Tel-Aviv University
Raymond and Beverly Sackler
Faculty of Exact Sciences
The Blavatnik School of Computer Science

Using differential co-expression for dissecting biological processes and revealing disease specific gene regulation

Thesis submitted in partial fulfillment of the requirements for
M.Sc. degree in the School of Computer Science, Tel-Aviv University

By

David Amar

The research work for this thesis has been carried out at Tel-Aviv University
under the supervision of

Prof. Ron Shamir

April 2012

Acknowledgements

First and foremost, I would to thank my advisor Prof. Ron Shamir for his help and guidance in every step of the way during my M.Sc studies. Most importantly, I wish to express my thanks for teaching me how to conduct research with integrity and thoroughness.

During my studies I had the privilege of collaborating with gifted researchers that became great friends. I would like to thank Oren Tsfadia, Louis Bradbury and Prof. Eleanore Wurtzel (CUNY) for working with me on the MORPH project and introducing me to plant biology. I would like to thank Ofer Lavi and Hershel Safer (TAU) for working with on the GENEPARK project and teaching me a lot in machine learning and the importance of organization. I would also like to thank Prof. Nir Osherov from the medical school at TAU for giving me the opportunity to work on Aspergillus Fumigatus data in two different projects. I would like to thank him and his students Haim Sharon and Shelly Hagag for working with me and for their unlimited patience while teaching me biology. I would like to thank Adi Maron-Katz for working with me on biclustering problems in fMRI data, and Ron Zeira and Yaron Orenstein for working with me on classification problems in flow-cytometry data. I wish to thank my lab mates Roye Rozov, Renana Meller, Guy Harari, Guy Karlebach, Dvir Netanely, Mukul Bansal, Arnon Paz, Eyal David, and Annelyse Thevenin, for helpful discussions, endless brain storming sessions and being great friends. I owe special thanks to Gilit Zohar-Oren for her administrative and moral help. I would also like to thank Igor Ulitsky for being available for every question I had and for his instructive comments.

Last but not least, I thank my wife Anat for her support and love.

Finally, I deeply thank the Edmond J. Safra Foundation for their financial support over the past two years. In addition, my research was partly supported by the GENEPARK project which is funded by the European Commission within its FP6 Programme (contract number EU-LSHB CT- 2006-037544), and by the Israel Science Foundation (grant no. 802/08)

"The whole of science is nothing more than a refinement of everyday thinking."

Albert Einstein

Abstract

Gene expression measurements can help in understanding diseases, by identifying differences between the tissues of sick and healthy individuals. Differential expression analysis is a well-established way to find gene sets whose expression is altered in the disease. Recent approaches to gene expression analysis seek differential co-expression patterns, wherein the level of co-expression of a particular set of genes differs markedly between disease and control samples. Such patterns can arise from a disease-related change in the regulatory mechanism governing that set of genes, and pinpoint dysfunctional regulatory networks.

Here we present DICER, a new method for detecting differentially co-expressed gene sets. The method utilizes a novel probabilistic score for differential correlation. In addition, DICER detects meta-modules: pairs of modules, where each module is correlated across all samples but the amount of correlation between the two modules is different in the disease and in the normal samples. We show that our method outperforms the state of the art in terms of significance and interpretability of the detected modules. Moreover, the gene sets discovered by DICER manifest regulation by disease-specific microRNA families. In a case study on Alzheimer's disease, DICER dissected biological processes and protein complexes into functional sub-units that are differentially co-expressed, thereby revealing inner structures in disease regulatory networks.

Contents

Acknowledgements 3
Abstract 7
Contents 9
1. Introduction and Summary 11
2. Background 13
2.1 Biological background 13
2.1.1 Basic biology 13
2.1.1.1 Historical background 13
2.1.1.2 What are cells? What are mitochondria? 14
2.1.1.3 What is a gene? How is the genetic information used for producing real biological functionality? 15
2.1.1.4 Gene regulation 16
2.1.2 Measuring gene expression 18
2.1.3 Neurodegenerative disorders 19
2.2 Computational background 20
2.2.1 Data representation 20
2.2.2 Co-expression analysis 20
2.2.3 Gene Clustering 22
2.2.4 Differential expression analysis 23
2.2.5 Differential co-expression analysis 24
2.2.5.1 CoXpress 26
2.2.5.2 DiffCoEx 26
2.2.6 Large scale biological networks 28
2.2.7 Enrichment analysis 28
2.2.7.1 The hypergeometric test 29
2.2.7.2 The FAME algorithm for miRNA enrichment analysis 30
2.2.8 Complexity theory background 31
2.2.8.1 Optimization and gap problems 31
2.2.8.2 Hardness of approximation 33
3. A novel measure of class specific differential correlation 35

3.1 A normalized score of differential correlation 35
3.2 Experimental tests 36
4. The DICER Algorithm 40
4.1 Algorithm overview 40
4.2 The probabilistic framework 44
4.3 Class-specific differential correlation analysis 46
4.4 The consistent correlation graph 47
4.5 Finding meta-modules 47
4.5.1 Hardness of approximation 47
4.5.2 Heuristic 48
5. Experimental results 51
5.1 Comparison with other differential correlation gene module discovery methods 51
5.2 Discovered gene sets are highly enriched with miRNA targets 55
5.3 Case study: Alzheimer's disease 57
5.4 Technical notes 62
5.4.1 Finding differentially correlated gene modules using DiffCoEx 62
5.4.2 Enrichment analysis of pathways and protein complexes 62
5.4.3 Enrichment analysis of miRNA families 63
5.4.4 Enrichment analysis of known disease and miRNA associations 63
Supplementary Tables 64
References 69

1. Introduction and Summary

High throughput measurement technologies, e.g., microarrays, mass spectrometry and next generation sequencing, are often used to compare different classes of individuals. By utilizing data produced by these techniques, one can try to discriminate different groups of patients and to decipher biological mechanisms that underlie a specific phenotype. Because these technologies produce many thousands of numeric attributes for each sample, their analysis raises formidable computational challenges.

Systems biology aims to dissect biological phenomena by integration of high-throughput data. Often, established biological knowledge is integrated in the analysis together with large-scale genomic measurements. For example, in discovery of pathways whose genes expression profiles are altered in a disease, the same data can be used to construct disease biomarkers [1-6]. Other methods integrate networks (e.g. protein-protein interaction network) with microarray experiments for finding differential genes that form dense sub-networks and utilize these data for improving classification [7-12]. Another key challenge in molecular biology is to understand the regulatory program that controls mRNA levels. The key components in this program are transcription factors (TFs) and microRNAs (miRNAs). TFs are proteins that activate or repress transcription by binding to short DNA sequences that typically reside in a gene's promoter. miRNAs are a class of endogenous non-protein-coding small RNAs, which repress gene expression at the posttranscriptional level by annealing to the 3'UTR of the mRNA. Several attempts in computational reverse engineering of regulatory mechanisms were performed by combining gene set detection methods with sequence based analysis for motif finding [13-17].

Complex supervised analysis of gene expression data has gone beyond identification of differential genes or pathways, to identify differential co-expression patterns. Differential co-expression is a situation in which the co-expression of genes changes among different phenotypes. Using the premise that co-expressed genes are more likely to be co-regulated, major changes in co-expression patterns may indicate changes in regulation. Several studies identified differentially co-expressed transcriptional factors known to be involved in cancer whereas their mean expression levels had hardly changed [18-20]. Another main motivation for performing differential co-expression analysis emerged from the need to find disease specific alterations in regulatory systems [21], and several studies found specific evidence for

differential co-expression patterns [18-20, 22-28] (see [21] for review). For example, Mentzen et al.[27] identified gene modules that are enriched with cell adhesion and growth factor related genes, and that manifest a significant decrease in co-expression in mammary gland tumors compared to wild type.

Here we describe DICER (Differential Correlation in Expression for meta-module Recovery), a new method for differential co-expression (DC) analysis. We developed a novel statistical score for DC, and show that it has significantly higher values in real data sets compared to randomized data sets. Given a set of gene expression profiles partitioned into different classes, DICER aims to detect gene sets that manifest correlation changes that are specific to a particular class of interest. DICER addresses two scenarios of differential correlation: (1) a group of genes that are differentially correlated in the tested class; we call such a group a *differentially correlated cluster*; and (2) a pair of gene sets where each set contains genes that are co-expressed in all classes and there is marked change in the correlation between the two sets in the tested class; we call such a pair a *meta-module,* and each of the sets is called a *module*. A meta-module can represent two (or more) sub-units of the same biological process where each of them is co-expressed throughout the tested phenotypes, but their interrelation is differs in a specific phenotype. This can happen, for example, if the regulation of one of the subunits is altered in the disease condition. Because meta-module detection is NP-hard to approximate within any constant factor, DICER uses heuristics to find meta-modules.

We tested the ability of DICER and other methods to find differentially correlated gene modules on five disease-related gene expression data sets. We discovered that DICER can detect more significant pathway enrichments, and that the modules discovered by DICER manifest higher correlation changes patterns. In addition, DICER modules are highly enriched with gene targets of miRNA families. These enrichments identify known miRNA-disease associations and suggest new miRNA candidates that affect the tested disease. In a case study on Alzheimer's disease data set we demonstrated how DICER can dissect known biological functions into biologically meaningful sub-units, which cannot be detected by standard differential expression analysis. We also show that such analysis can explain changes of gene activity that are not detected by analyzing changes at the mRNA level.

2. Background

This chapter lays out the background and terminology required for this thesis. In section 1 we shall introduce basic biological definitions and recent findings. We shall also discuss current available high throughput data that were used in this thesis, and give a brief introduction to neurodegenerative diseases. In section 2 we shall discuss and give formal definitions of the computational problems addressed. This section includes problems of: coexpression analysis, gene clustering, differential expression, large scale genetic networks, gene sets enrichment analysis, and differential coexpression. In the end of section 2 we shall give a background on computational complexity, the APX complexity class and discuss hardness of approximation schemes.

2.1 Biological background

In this section we introduce biological terms and definitions that are necessary for understanding the motivation of this thesis, and the computational problems that we deal with.

2.1.1 Basic biology

Here we present, briefly, basic biological terms. For more information on basic biology see [29, 30]. For additional information on gene regulation see [31, 32].

2.1.1.1 Historical background

Modern genetic research originated in the 19'th century, when genetics was first considered as a set of principles, combined with analytical procedures, and the notion of a gene as the biological entity responsible for defining traits was first introduced. In 1859 Charles Robert Darwin published his book *"On the Origin of Species"* introducing the evolution theory with compelling evidence. In 1866, the Augustinian monk Gregory Mendel performed a set of experiments that revealed the basic inheritance mathematics. Remarkably, until 1944 it was

believed that proteins carry the genetic information, and that the information in deoxy-ribonucleic acid (DNA) plays only a secondary role. Thanks to the outstanding experiments performed by Oswald Avery, Colin MacLeod and Maclyn McCarty this belief was shattered, and it was shown that the DNA is the major carrier of genetic material in living organisms. In 1953, based on research done by Rosalind Franklin, James Watson and Francis Crick deduced the three dimensional double-helix structure of the DNA and inferred its method of replication [30]. Later on, in 1975, Frederick Sanger and Alan Coulson published the first DNA sequencing procedure. In February 2001, the first draft of the human genome was completed [33], marking the beginning of the informatics revolution of biology.

2.1.1.2 What are cells? What are mitochondria?

The cell is the basic structural and functional unit of all living organisms. It is the smallest unit of life classified as a living thing. The cell is separated from its surroundings by at least one membrane. Specialized sub-units of the cell that have a specific function, and are also surrounded by a membrane, are called *organelles*. There are two types of cells: prokaryotic, and eukaryotic. In eukaryotic cells the genetic information stored in the DNA molecule is surrounded by a membrane in an organelle called the *nucleus*, whereas in prokaryotic cells the DNA is not surrounded by a membrane. All multi-cellular organisms are comprised of eukaryotic cells.

The *mitochondrion* (plural: mitochondria) is an organelle present only in eukaryotic cells. It is responsible for generating the cell supply of adenosine triphosphate (ATP), the main source of chemical energy in living cells. In addition, because of their main role in metabolism, the mitochondria are involved in a wide range of biological processes, such as signaling, cellular differentiation, programmed cell death (also called apoptosis), cell cycle, and cell growth [34]. Many, sometimes very different, diseases involve dysfunction of the mitochondria, of which many are brain disorders [35].

2.1.1.3 What is a gene? How is the genetic information used for producing real biological functionality?

The DNA molecule is comprised of four building blocks called *nucleotides*. A *gene* is a DNA segment that codes for a specific biological functionality. Upon demand, parts of the DNA chain are exposed and copied into single stranded ribonucleic acid (RNA) chain in a process called *transcription*. Most of these RNA chains are used for synthesis of proteins in a process called *translation*. Nonetheless, some genes are *non-coding* and contain functional information that is not carried by proteins. The translation process is accomplished by cell components called *ribosomes*. The RNA template molecule for protein synthesis is called the *messenger RNA* (mRNA). *Gene expression* is the process by which information from a gene is used in the synthesis of a functional gene product, or transcript.

Most of the cell functions are done by proteins, the main players of the machinery within the living cell, and are also used as communication means between and within cells. Proteins take care of maintaining cell structure and activity, allowing the cell to cope with external environment. Proteins can also be used for cell-to-cell signaling, secreted outside and used within the cell upon demand. There are many types of proteins, so cells of different tissues, while having the same genetic information (i.e. the same DNA), may have different mixtures of proteins. Even within the same cell, the protein types and amounts (or concentrations) may change with time, depending on the cells internal state and on outside conditions (e.g. stresses). In many cases proteins cannot function alone, as they are parts of a protein complex in which partners must physically connect, and even include non-protein cofactors such as RNA molecules, to produce biological functionality. These complexes can be very large and some of them contain dozens of different proteins. One of the main complexes within a cell is the ribosome, which is responsible for synthesis of new proteins when those are required.

A biological pathway is the set of molecular entities involved in a given biological process and the interrelations among those entities. While every protein in the pathway can be active on its own, the activity of most members of the pathway is needed for the pathway to carry out its biological process. Biological pathways can represent the flow of metabolic particles in a biological process and the genes responsible to carry out the chemical interactions. These pathways are called *metabolic pathways*. Biological pathways are also used to represent sequences of signaling interactions among different cellular entities in the cell. These pathways are called *signaling pathways*. Pathways are to some extent the biologist's

simplification or abstraction. Pathway boundaries are inherently fuzzy and somewhat subjectively defined, but they are valuable for understanding biology and organizing biological knowledge (e.g. a metabolic or signaling pathway). Although current knowledge about some biological pathways may be substantial and useful for systems-level analyses, not all genes that participate in and/or affect function of these pathways are known. For other pathways, only rudimentary information is known.

2.1.1.4 Gene regulation

The emergence of complex, multi-cellular life forms was accompanied and probably facilitated by increased complexity of gene regulation. Gene expression is tightly regulated to guarantee that the proper amount of gene product is present in the right cell at the correct time.

At the transcriptional level, the key regulatory players are proteins that bind the DNA sequences and control the flow of genetic information from DNA to RNA. These factors are called transcription factors (TFs) and can either increase or decrease gene expression of specific genes. TFs bind to the DNA and can turn on or off the transcription machinery of a specific gene. Moreover, transcription will not be carried out without dedicated TFs.

At the post-transcriptional level, numerous regulatory mechanisms were discovered in the past decades. These mechanisms can be very different, but most of them can be grouped under the umbrella of RNA regulation. RNA regulation processes exploit the chemical characteristics of the RNA molecule and the processes that synthesize mRNA. For example, raw mRNA molecules created in the transcription processes contain segments that are removed in a process called *splicing*. The retained segments are called *exons*, whereas the removed segments are called *introns*. A post-transcriptional regulatory process called *alternative splicing* can create genetic variation by using different splicing patterns for the same gene. Another most important attribute of the mRNA molecule is a tail of adenine molecules that is added after the transcription process. This tail can be added in different sites along the RNA, thus regulation of this process can create different mRNA molecules [36]. Last but not least, mRNA regulation can be carried out by short RNA molecules, with no more than two dozens of nucleotides, that bind to a sequence in the mRNA molecule and negatively regulate it (i.e. by promoting the degradation of the mRNA molecule). These short

RNA molecules are called micro-RNAs (miRNAs). They are abundant in many human cell types, and probably target most mammalian genes [37-39].

A summary of regulatory mechanisms is presented in Figure B1. TFs and miRNAs are similar in that both bind nucleotide sequences and have the ability to change the activity level of a specific gene. In both cases, multiple miRNAs and TFs may target the same genes that can be regulators themselves. Moreover, regulation patterns can change extremely upon different situations in which the cell needs to adapt. Thus, elucidating the regulators that are active in a specific phenotype and the interactions among them is a challenging problem of great importance.

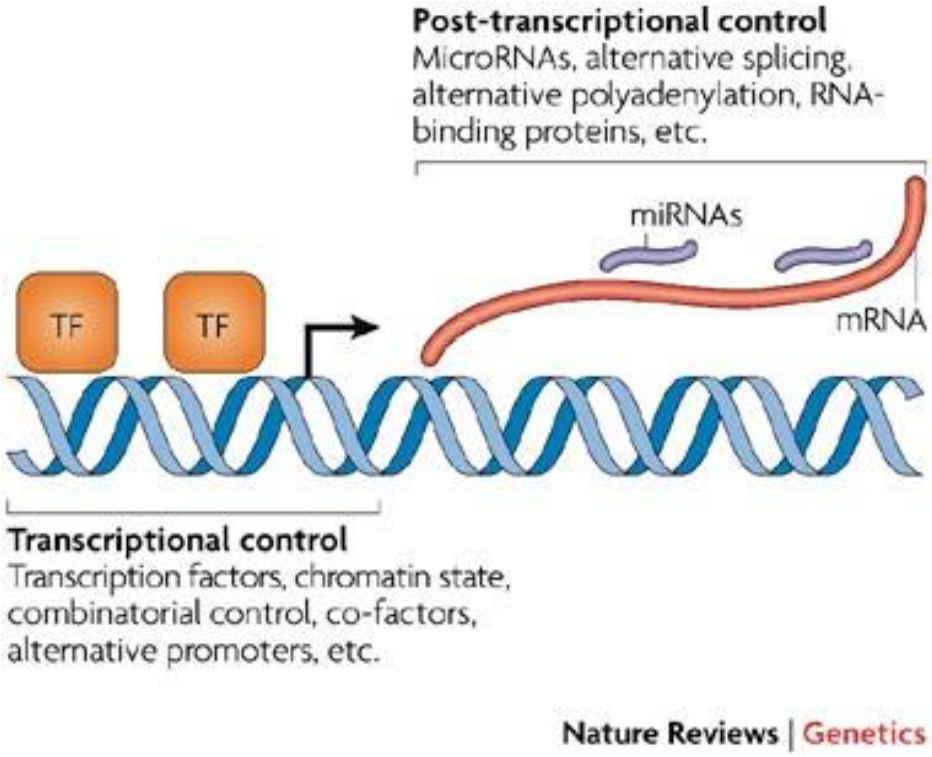


Figure B1 A summary of gene regulation mechanisms. The transcription factors bind the DNA molecule, typically before the start of the gene coding sequence (marked by an arrow). Many different mechanisms can regulate genes post-transcriptionally by controlling the mRNA molecule. Source: [32].

2.1.2 Measuring gene expression

In biological research, it is often required to acquire a global molecular "snapshot" of the cell at different situations. This snapshot can be later used for a comparative analysis to derive the molecular changes between different cells. If technology would allow it, the best option for such a snapshot would provide a quantification of the concentration of all proteins and metabolic particles in the cell. However, techniques for recording these quantities en masse are still under development. Alternatively, it is possible to measure the concentration of all RNA molecules in the cell using DNA microarrays or next generation sequencing (NGS) technologies, and use these measurements as approximation for the current activity of all genes (and indirectly their protein products).

A DNA microarray is a small solid surface consisting of thousands to millions of microscopic spots of DNA oligonucleotides called probes. Each probe can hybridize (bind) to specific RNA fragments, depending on its nucleotide content. In a specific design, the oligonucleotides of each probe are chosen so that it can hybridize to a single gene's mRNA. Extracting RNA material from the cell, fragmenting it and creating physical contact between this material and the surface of a microarray allows for the probes to hybridize to the RNA that was present in the cell at the time of extraction. The larger the number of fragments present for a gene, the higher the hybridization intensity. Later on, the microarray is scanned, and the amount of RNA bound to each probe is measured. This allows for measuring the transcription levels of all genes within a living cell or tissue. A gene expression profile is the set of all levels associated with all genes.

Recently, a new generation of DNA sequencing technologies called Next Generation Sequencing (NGS) or Deep Sequencing have being developed. These methods can read millions of short DNA or RNA chains. These chains can be compared to a known sequence of a reference genome. For a short review of these methods and additional references see [40]. Recently, these methods were applied also to RNA sequencing (in a technique called RNA-Seq), allowing for high throughput and accurate measurement of gene expression profiles. Producing such profiles is becoming cheaper and more accurate, but the challenges of analyzing these profiles are the same as in microarrays. All methods and gene expression analysis algorithms covered throughout this work are also applicable to expression profiles obtained by next generation sequencing, after an appropriate normalization of the mRNA reads and their summarization in expression profiles.

2.1.3 Neurodegenerative disorders

Neurodegenerative disorders are a class of diseases in which progressive loss of neurons occur. Excitotoxicity and apoptosis are two main causes of neuronal death [41] , and related pathways ,e.g. oxidative stress and mitochondria impairment, were shown to play a key role in multiple neurodegenerative brain disorders [42]. In particular, many apoptotic signals emerge from the mitochondria [43]. For most of these disorders early steps in the disease cascade are still unknown [44]. For example, the role of the vascular system in the progression of the neurodegenerative process is still unclear. The traditional view explains the vascular lesion as an epiphenomenon, while other views relate this phenomenon as a possible cause of neurodegeneration [45, 46]. Several studies by Zlokovic and colleagues suggested that vascular damage in Alzheimer's disease (AD) occurs initially and leads to neurodegenerative changes [47].

Another interesting aspect of neurodegenerative diseases is the relation between neuron loss and the immune system. Traditionally, presence of immune cells in the central nervous system (CNS) was thought to be detrimental for the organism [48], as shown in multiple sclerosis [49], stroke [50] and depression [51]. A large variety of neurodegenerative diseases, including AD and PD, were shown to be associated with chronic neuroinflammation [52, 53]. However, other studies suggested a physiological role of immune cells such as monocytes and T cells in brain repair [54-56]. In addition, immune-deficient mice were shown to suffer from cognitive impairment [57]. The findings in this growing field have initiated several clinical trials with the aim of boosting immune responses in the CNS of individuals with spinal cord injury, multiple sclerosis and AD [58].

AD is the most common progressive neurodegenerative brain disorder in human. AD is a complex progressive condition that involves sequentially interacting pathological cascades, including the interaction of amyloid-β (Aβ) aggregation with plaque development, and the hyperphosphorylation and aggregation of tau protein with formation of tangles. Together with associated processes, such as inflammation and oxidative stress, these pathological cascades contribute to loss of synaptic integrity and progressive neurodegeneration [59].

2.2 Computational background

In this section we lay out the computational background of this work. Each sub-section deals with different type of computational problems. Within each sub-section we give references for additional information that is not covered here.

2.2.1 Data representation

Gene expression data can be represented as a real matrix $D \in \mathbb{R}^{n\ x\ m}$, where n is the number of genes in the data and m is the number of samples. Each row in the matrix contains the expression pattern of a specific gene, and each column represents the expression profile of a sample. Thus, columns can represent different experiments, conditions, cells, or individuals. Each entry $D_{i,j}$ can represent ratios or absolute values. In our settings we will consider only the latter. The pattern of gene i is the i^{th} row in D. The expression profile of a sample j is the j^{th} column of D. In many cases we are also given a mapping of each sample (i.e. each column in D) to a label that represents a disease. That is, each sample j is given a label $l \in l_1, l_2, \dots, l_K$, where K is the number of labels in the data. The simplest case is where there are two possible labels: case (diseased) and control (normal).

2.2.2 Co-expression analysis

Gene coexpression analysis aims to detect gene pairs that are coordinated in their expression profiles. Co-expression is usually measured by similarity (or distance) between the vectors that represent the gene profiles. The biological rationale for this analysis is that genes that have similar expression patterns are more likely to be part of the same biological process than randomly selected genes. This paradigm is called "guilt by association". Another premise is that coexpressed genes are more likely to be co-regulated. Based on these assumptions, gene coexpression analysis has been widely used for gene function prediction [60-63].

The Pearson correlation coefficient is a co-expression measure that quantifies the linear correlation of two vectors that represent expression profiles. Let $\bar{u}$ denote the average of a

vector of real numbers u. The Pearson correlation r of two gene profiles x and y is defined as:

$$r = \frac{\sum_{i=1}^{m}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{m}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{m}(y_i - \bar{y})^2}}.$$

This score is between -1 and 1 inclusive. In the extreme case where the values of x and y satisfy $y = ax + b$ and $a > 0$ then $r = 1$. If $a < 0$ then $r = -1$. If x and y are independent then $r = 0$. Pearson correlation equalizes the expression patterns by subtracting the mean and dividing by the standard deviation. Hence, the correlation reflects relative trends (common deviations) and not absolute values. A drawback of the Pearson correlation coefficient is its sensitivity to outliers. Two tested genes might have high correlation simply because of single "bad" condition that posses extremely high expression values [60].

A general probabilistic framework for coexpression analysis was presented in [64, 65]. Although the original goal of the algorithm was to find gene clusters that are highly co-expressed, the statistical framework is generic and can be used for co-expression analysis. In this framework co- expression values are assumed to follow a bimodal distribution, i.e. values can belong to one of two distributions. The first distribution is of gene pairs form different clusters, denoted as *non-mates*, whereas the second distribution is of gene pairs from the same clusters, denoted as *mates*. For co-expression analysis, we assume instead that mates are gene pairs with similar expression patterns, whereas non-mates are gene pairs with dissimilar expression patterns. Denote the distribution density functions as f_{mates} and $f_{non-mates}$. Assume that the probability that a gene pair belongs to the 'mates' distribution is p, independently for each pair. We define the log-likelihood ratio of a correlation value r as:

$$LLR(r) = log\ \frac{p\ f_{mates}\ (r)}{(1-p)f_{non-mates}(r)}$$

When using the Pearson correlation coefficient, the assumption that f_1 and f_0 are normal distributions (with possibly different means and standard deviations) has theoretical and practical justifications [65]. Under this assumption, we can derive a LLR score:

$$LLR_{1,0}(r) = \log\frac{p\ f\ (r|\mu_{mates}, \sigma_{mates}\ )}{(1-p)f\ (r|\mu_{non-mates}, \sigma_{non-mates})}$$

$$= log\frac{p\sigma_{non-mates}}{(1-p)\sigma_{mates}} + \frac{(r - \mu_{non-mates})^2}{2\sigma_{non-mates}^{\ 2}} - \frac{(r - \mu_{mates})^2}{2\sigma_{mates}^{\ 2}}$$

The LLR score can take positive and negative values. A positive value will be observed when the correlation between two genes is more likely to present mates than non-mates. All model parameters can be estimated using expectation maximization (EM) algorithm [64].

2.2.3 Gene Clustering

Given a gene similarity matrix, the goal of gene clustering is to assign genes into groups called clusters, such that genes that are assigned to the same cluster are similar, whereas genes assigned to different clusters are non-similar. There are many formulations for the clustering problem, and most of them are NP-hard. Therefore, approximations and heuristics are used. For example, in correlation clustering [66], the input is a complete graph in which nodes represent genes and edges can be labeled as similar or dissimilar. Similarity edges are labeled '+' and dissimilarity edges are labeled '-'. The number of agreements in a clustering solution is defined as the number of '+' edges within clusters plus the number of '-' edges between different clusters. The number of disagreements is the number of '-' edges inside clusters plus the number of '+' edges between clusters. Maximizing agreements (and the equivalent problem of minimizing disagreements) is NP-hard, but Bansal et al. [66] give a constant approximation algorithm for minimizing disagreements, and a PTAS for maximizing agreements.

Clustering methods have been used in vast number of fields. One of the most basic and intuitive approaches to clustering are hierarchical methods [13, 67-69]. These methods construct a tree-like structure to explore the relations among entities. For example, average linkage hierarchical clustering builds the tree structure by iteratively selecting the pair of entities with maximum similarity and uniting them to form a new cluster. Then, the similarity of this cluster with every other entity (which can be a gene or another cluster) is defined as the average similarity of its components to the components of the other object. This approach can be easily applied to distances instead of similarities by selection of the pair of entities with the minimal distances, while the averaging process remains the same.

In many applications it is not necessary to form a partition of the genes. Several methods were developed to find homogeneous gene sets that not necessarily cover all genes [64, 70]. These methods hold the potential to remove outlier genes, and therefore are more robust. Because genes can be a part of different biological processes or participate in multiple protein

complexes, several methods allow assigning genes to multiple clusters [71-76]. The solution provided by these methods is called fuzzy clustering.

2.2.4 Differential expression analysis

Gene expression studies of disease typically include a comparison of gene expression profiles among different classes. The simplest case is binary classification, where samples can take one of two possible labels, typically, diseased and healthy tissues. Expression profile comparisons are usually done using a statistical test for significance of the difference in mean expression level of a single gene between the classes [77]. One of the simplest tests available is Student's t-test. It is a statistical hypothesis test where the null hypothesis is that the means of the gene expression values in the two classes are equal. It is based on the assumption that the expression values within each class follow a normal distribution with equal variance in both classes, and that the samples are independent from each other. For a tested gene, we have two vectors of expression values x and y. The t-statistic is defined as:

$$\frac{\bar{x} - \bar{y}}{S_{x,y}\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

Where: $$S_{x,y} = \sqrt{\frac{(n_1-1)S_x^2 + (n_2-1)S_y^2}{n_1+n_2-2}}$$

Here, n_1 and n_2 are the sizes of x and y respectively, and S_x^2 and S_y^2 are the estimated standard deviations of x and y respectively. Under the null hypothesis, this statistic follows the Student's t-distribution with $N - 2$ degrees of freedom, where $N = n_1 + n_2$. Thus, statistical significance (p-value) can be calculated exactly.

When performing many statistical tests the threshold for significance should be determined so that the number of false positives would remain low. The standard approach is called FDR (False Discovery Rate), and it ensures that the expected fraction of false positives, out of all accepted tests, would remain low [78]. All statistical tests performed in this thesis were set to 0.05 FDR.

Several methods were developed to calculate the significance of the differential expression for predefined sets of genes [3, 79-82].Usually these methods partition the genes within the

sets to differential and not differential, or weight the genes by their importance. In most cases, these methods identify cases of *up-regulation*, where the expression level in the cases is larger than in the control, and *down-regulation*, where the expression level in the cases is lower than in the control. For example, in GSEA (Gene Sets Enrichment Analysis) [3] all genes are ranked according to their differential expression measurement (where the first rank is the most up-regulated gene and the last rank is the most down-regulated gene). Given a set of genes, the genes tendency to appear concentrated in ranking is tested using the Kolmogorov-Smirnov (KS) test. Thus, if the genes in the set are up-regulated then they would have high ranks, their rank distribution would tend to get lower rank-scores than the distribution of all other genes, and the KS p-value would be significant. Lee et al. [2] used a very simple algorithm for selection of the most differentially expressed genes within pathways, where the goal is to find a subset of pathway genes such that if the selected gene patterns are averaged the resulting expression profile manifests the best differential expression score . The discriminative score of a pathway is defined as the t-test score of the expression pattern resulting from averaging the expression patterns of the selected genes. The selection of genes in the pathway is done in a greedy fashion, starting with the most differentially expressed gene and iteratively adding genes. In each iteration, addition of the gene with the next best t-test score is considered, until no addition increases the discriminative score. As a result of the algorithm, pathways can be ranked according to their discriminative score. To determine the number of pathways to use, the top discriminative pathways are chosen using a cross-validation process such that the number of selected pathways optimizes the area under the receiver operating characteristic (ROC score), of the classification quality.

2.2.5 Differential co-expression analysis

Complex supervised analysis of gene expression data has gone beyond identification of differential genes or pathways, to identify differential co-expression patterns. Differential co-expression is a situation in which the level of co-expression of genes changes among different phenotypes. Figure B2 demonstrates a case in which co-expression in healthy individuals is markedly higher than in sick individuals. Using the premise that co-expressed genes are more likely to be co-regulated, major changes in co-expression patterns may indicate changes in regulatory factors.

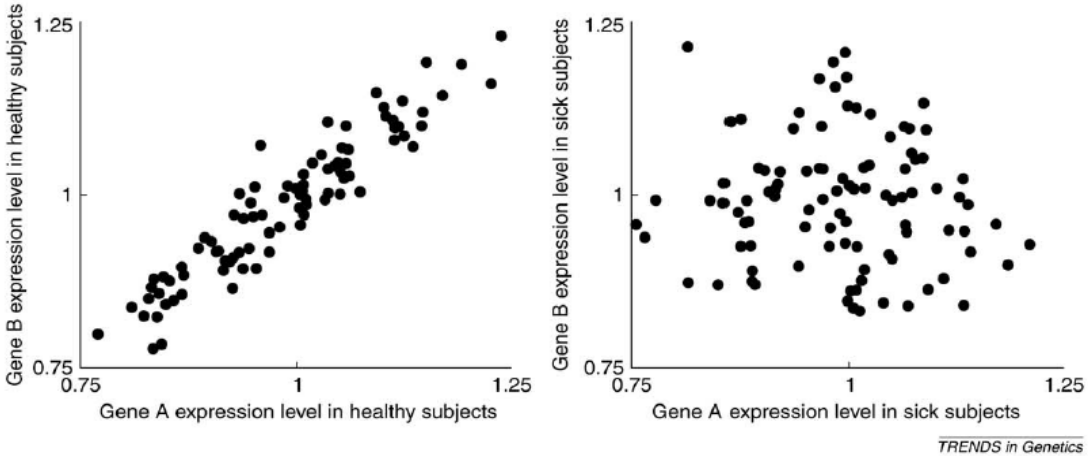


Figure B2 Example of differentially co-expressed gene pair. The x and y coordinates show the expression levels of two genes. Each dot shows the expression levels of the two genes in one individual. Left: healthy subjects. Right: sick subjects. The genes are highly correlated in healthy individuals and are un-correlated in sick patients. Source: [21].

Several studies of cases patients and healthy controls identified differentially co-expressed transcription factor pairs known to be involved in cancer whereas their mean expression levels had hardly changed [18-20]. This demonstrates that differential co-expression provides a new type of discriminative information that is beyond what is merely obtained by looking at changes in expression intensities. This motivation for performing differential co-expression analysis emerged from the need to find disease-specific alterations in regulatory systems [21]. Several studies found specific evidence for differential co-expression patterns that indicate changes in regulatory systems, or are characteristic of the cell state at specific phenotypes [18-20, 22-28] (see [21] for review). For example, Mentzen et al.[27] identified gene modules that are enriched with cell adhesion and growth factor related genes, and that manifest a significant decrease in co-expression in mammary gland tumors compared to wild type.

Several computational approaches were developed for performing differential co-expression analysis, including detection of differentially correlated gene modules or gene specific analysis [83-86]. Lai et al. [18] developed a statistical framework for analysis of a single gene of interest, and showed that genes associated with cancer may manifest differential co-expression with many other genes. Gene Set Coexpression Analysis (GSCA) [87] was proposed to test for differential co-expression of known pathways. For each pathway, GSCA summarizes the change in co-expression over all gene pairs in the pathway and estimates significance using permutation tests. Below we describe in more detail two methods for finding differential co-expression on the gene set level.

2.2.5.1 CoXpress

CoXpress [85] finds coexpressed gene modules using one of the classes in the data, and then tests if these modules show a different co-expression pattern in the other classes. First, hierarchical clustering is used to cluster the genes according to their similarity in the tested class. This is done using distance-based average linkage hierarchical clustering in which the distance between two genes is $1-r$, where r is the Pearson correlation between the expression profiles of two genes, restricted to the tested class. Gene clusters are determined by cutting the tree structure at a specific height. The next step assesses the significance of the clusters. Each cluster with more than three genes is tested for significance of differential coexpression. Significance is estimated by sampling random groups of the same size, in each class, and creating an empirical distribution of correlation within groups in each class. Then, a gene group is called differentially co-expressed if it manifests significantly high coexpression in the tested class, while its correlation is not significant in the other classes.

CoXpress was extended to handle known gene sets (e.g. GO terms). It was successfully applied together with differential expression analysis, and complex discovery in protein-protein interaction data, in a comparative study of mammary gland tumors development in mice [27] . While there were clusters that were up-correlated and up-regulated (i.e. gene sets that manifest higher expression values and higher co-expression in a specific class), some of the discovered gene modules were up-regulated in tumors, but had a decreased co-expression pattern. This demonstrates the complex relations between differential expression and differential co-expression.

2.2.5.2 DiffCoEx

A recently proposed sophisticated method called DiffCoEx [86] looks for differentially co-expressed gene modules. DiffCoEx has five steps. For simplicity we discuss here binary class data sets. First, the gene pair-wise correlation matrix is calculated for each class k:

$$C[k]: c_{i,j}^{[k]} = corr\ (gene_i, gene_j)$$

Where the index k indicates that the correlation is calculated on samples in class k only. Second, the adjacency difference matrix is defined as:

$$D\colon d_{i.j} = \ (\ 0.5 * |sign\ (c_{i,j}^{1}) * (c_{i,j}^{1})^{2} - sign\ (c_{i,j}^{2}) * (c_{i,j}^{2})^{2}\ |\ )^{\beta/2}$$

In this matrix, high values of $d_{i.j}$ indicate that the coexpression status of gene i and gene j changes significantly between the two conditions. This adjacency matrix is defined such that it only takes values between 0 and 1. Note that this score does not discriminate between up-correlation and down-correlation when analyzing a specific class of interest, as it merely quantifies the level of absolute difference of correlation. The soft threshold parameter β is taken as a positive integer and is used to transform the correlation values so that the weight of large correlation differences is emphasized compared to lower, less meaningful, differences. In the DiffCoEx implementation β is set to 6.

Third, derive a dissimilarity matrix from D:

$$T\colon t_{i,j} = 1 - \frac{\sum_k d_{i,k} d_{k,j} + \ d_{i,j}}{\min\left(\sum_k d_{i,k}\ , \sum_k d_{j,k}\ \right) + \ 1 - \ d_{i,j}}$$

The score $t_{i,j}$ is between 0 and 1, and is called the topological overlap dissimilarity score [88]. Intuitively, a low value of $t_{i,j}$ denotes high similarity of i and j in the differential coexpression network. Moreover, it means that gene i and gene j have significant correlation changes with the same large group of genes. This property of the dissimilarity score allows DiffCoEx to detect both gene modules that manifest a marked change in the correlation and module-to-module changes.

Fourth, the dissimilarity matrix is used as input to a hierarchical clustering algorithm, and modules are detected using a dynamic tree cut procedure [89]. This procedure starts by cutting the tree at the top (producing a small number of large clusters) and then explores the hierarchy structure. Through an adaptive process it splits large clusters that are likely to contain distinct sub-clusters, and merges close clusters that are neighbors in the hierarchy. This process terminates when the clusters become stable.

Finally, the resulting gene modules are assessed for significance using permutation tests. Here the modules detected by DiffCoEx are compared to random gene sets of the same size similarly to the process described for CoXpress.

In a separate post-processing optional step, the significance of the module-to-module relation can be compared to random gene set pairs of the same size. Since this process is not part of the module detection algorithm, DiffCoEx may produce gene modules even on

randomly created data sets. Moreover, DiffCoEx might produce both gene sets and gene set pairs that are not significantly differentially correlated.

2.2.6 Large scale biological networks

The holy grail of biology is to model the complex cellular systems to reveal fundamental biological processes. One step toward this goal is to obtain a system level overview. One such overview that is common today is to represent the set of molecular components as a biological network. In these networks, nodes represent cellular entities such as genes, proteins, or other molecules. Edges represent interactions between entities that are dependent or collectively carry out a biological function. For example, protein interaction networks describe physical relations between proteins. This information can help in modeling signaling pathways and in elucidating de novo protein complexes [90-92]. Another example is co-citation networks, in which an edge is added between two proteins that are mentioned together in a scientific text. In functional similarity networks, an edge is added between two proteins that are predicted to have the same function [93].

Today's models of biological networks are noisy and incomplete. They may contain many false positive edges due to inaccurate high-throughput methods and may miss many real edges due to partial knowledge. Nevertheless, computational methods were successfully applied on such networks and led to prediction of gene functions [94]. In this work we used such a tool called GENEMANIA [61, 95, 96]. This tool integrates many networks including protein interactions, co-localization, functional similarity, and predicted functional similarity. This tool is mainly applicable to small scale analysis in which we want to learn what is known about a single gene, or to summarize the known interactions among a small group of genes.

2.2.7 Enrichment analysis

Biological data available today can be helpful in determining the functionality of a particular gene. The data include metabolic and signaling pathways, biological processes, molecular function, protein complexes, and cellular localization. Moreover, we have today association of regulatory factors to their target genes. For example, several tools and databases provide predicted and validated miRNA targets [37, 97, 98]. Gene enrichment analysis uses biological

knowledge to assign biological meaning to some group of genes. In this process we ask if a group of genes is likely to be related to some biological process or is it under common regulation. Usually, we start with a group of genes that were detected by differential expression analysis, or with gene sets that were coexpressed in some experiment. In this section we shall discuss two enrichment analysis methods. The first is the hypergeometric test that is very general, and can be used as enrichment score in every application. The second is the FAME algorithm that was developed specifically for miRNA enrichment analysis.

2.2.7.1 The hypergeometric test

This is the most broadly used enrichment analysis. Formally, given an underlying set G of all genes, a gene set A and an a priori defined group of genes B we test the significance of the intersection of A and B. B can denote a group of genes in a biological process, a pathway, a protein complex, or the targets of some regulatory factor. The null hypothesis of the test is that the genes in A are randomly selected from the group G. Thus under the null hypothesis the size of the intersection between A and B follows a hypergeometric distribution, where $|A|$ is the number of sampling trials, $|B|$ is the number of successes and $|G|$ is the population size. The probability to for $|A \cap B| = x$, where $x \leq \min(|A|, |B|)$ is:

$$P_{hg}(x) = \frac{\binom{|B|}{x}\binom{|G| - |B|}{|A| - x}}{\binom{|G|}{|A|}}$$

Thus, the p-value is:

$$\sum_{x \geq |A \cap B|}^{\min(|A|,|B|)} P_{hg}(x)$$

Although this test is straightforward and intuitive, when testing different defined groups of genes B_i with the same group A, it does not take into account dependencies among different tested sets. In addition, if N gene sets are analyzed vis-à-vis M predefined gene groups, the number of statistical tests amounts to N*M. Thus, multiple test correction (e.g. FDR) is mandatory in order to control the number of false positives.

2.2.7.2 The FAME algorithm for miRNA enrichment analysis

FAME (Functional Assignment of MicroRNAs via Enrichment) [99] is a permutation-based statistical method that tests for over- and under- representation of miRNA targets in a set of target genes. Unlike standard enrichment analysis tools, this method addresses specific characteristics of miRNA regulation process during the analysis, and therefore has superior statistical power. In addition, unlike most enrichment analysis procedures, FAME integrates the analysis of many gene groups (i.e. the group of targets of every miRNA) and corrects for the multiple testing. FAME uses both computationally predicted and biologically validated gene targets as the set of genes targeted by each miRNA. For robustness, only conserved miRNA target sites are taken into account when testing for over representation.

FAME accounts for the predicted strength of each miRNA-target pair, and for the number of miRNAs regulating each gene. First a weighted bipartite graph $G = (M, T, E, W)$ is constructed in which miRNAs families (constituting the set M) are connected to their predicted target genes (which constitute the set T). For every hit of the miRNA sequence in the 3'UTR of a gene, an edge is added to E, weighted by the strength of the prediction (an integer value, available in TargetScan [37]). Thus, parallel edges are allowed. Following the construction of the graph, $N = 2000$ random graphs are created using degree preserving permutations: For each possible edge weight w, a long sequence of independent edge shuffling steps is performed, by replacing a pair of edges $(a, b), (c, d)$ of weight w by the pair $(a, d), (c, b)$ of the same weight. This step preserves the number of edges with weight w of each node in the graph. The number of random edge shuffles for each random graph is $5|E|$. The randomized graphs are used to evaluate the significance of the overlap between a miRNA M' and a set of genes T'. For each such pair let $W(M', T')$ be the sum of weights between M' and T' in G. This score is compared to the same scores induced by M' and T' in the random graphs and an empirical p-value is calculated. Finally, all p-values are corrected using the FDR procedure.

2.2.8 Complexity theory background

In this section we provide a background on computational complexity theory and focus on hardness of approximation. We start with some basic definitions of optimization problems and then discuss gap problems and APX-hard problems. For more details on basic concepts in complexity theory see [100, 101]. For more information on hardness of approximation see [102].

The main effort in complexity theory is to classify problems according to the amount of resources they require. The central resource is the amount of time needed, as a function of the length of the input. Many classical problems in computation theory are formulated as decision problems. Given an alphabet Σ, a set $L \subseteq \Sigma^*$ is called a *language* over Σ. A string x in Σ^* is called a *positive input* if $x \in L$ and a *negative input* if $x \notin L$. A polynomial time mapping reduction, also denoted as Karp reduction, f from a decision problem A to a decision problem B, is a polynomial time procedure that maps each positive input of A to a positive input of B, and each negative input of A to a negative input of B. We mark $A \leq_p B$ to denote the case where a Karp reduction from A to B exists.

The most fundamental time complexity classification is into problems that can be solved efficiently and those that not. The class P represents the set of problems that can be solved in polynomial time, and is considered roughly to be the set of efficiently solvable problems. Proving that a problem cannot be solved efficiently is occasionally achievable, but in most cases it is not known how to prove that. Many problems are not known to be in P, nor proven to be outside P, and are part of the NP class, the class of all problems for which there is a polynomial time verifier. The most fundamental open question in computer science is whether $P = NP$. While this problem is still open, we are able to classify problems as NP-hard. A computational problem L is NP-hard is every other problem in NP is reducible to L by a polynomial time procedure.

2.2.8.1 Optimization and gap problems

In many cases we are interested in optimization problems. Here, given an input x for problem A, x has a set of possible (feasible) solutions, and each solution w is scored according to an objective function $o_A(x, w)$. An optimization algorithm should return a best

solution. For example, we are given as input a graph and the optimization problem is to find an independent set (IS) – a group of nodes that induces an edge-free graph – of maximum size. For simplicity, from now on we discuss only maximization problems, and assume that feasible solutions have positive values.

For an optimization problem A and an input x, denote $opt(x)$ as the value of an optimal solution for an input x. M_A is called an algorithm for A if for every input x it produces a feasible solution w to A (not necessarily one with an optimal value) . M_A is called a *constant factor approximation algorithm* if its solution deviates from the optimal solution by at most a factor of c for all inputs. For maximization problem A, the outcome of M_A is a solution w for x satisfying:

$$\frac{o_A\ (x, w)}{opt(x)} \geq c$$

Note that here c is between 0 and 1. For a minimization problem we can replace the left hand side by its inverse. The *APX* class of optimization problems is the set of problems for which there exists a polynomial time constant factor approximation algorithm [103, 104].The class PTAS (Polynomial Time Approximation Scheme) is the set of optimization problems for which for every $0 < \varepsilon < 1$ there is a polynomial time algorithm that guarantees an approximation factor of $1 - \varepsilon$. The degree of the polynomial is dependent on ε, and can even be an exponential function of $\frac{1}{1-\varepsilon}$. By definition, every problem that has a PTAS is also in APX.

Another type of decision problems that are closely related to optimization problems are gap problems. For maximization problem A, a function $g(x)$ that quantifies the size of the input x, and real values $a, b, a < b$ define the gap problem $gap - A[a, b]$ that assigns a possible input x to one of the three options:

- If $opt(x) \geq\ b * g(x)$ then x is a positive input.
- If $opt(x) \leq a * g(x)$ then x is a negative input.
- Else ($a * g(x) < opt(x) <\ b * g(x)$) x is called a "don't-care" input.

An algorithm for this problem must accept all positive inputs, reject all negative inputs and may either accept or reject any other input. For example, for the maximum IS $g(x)$ is the number of nodes in the graph and $0 < a, b < 1$.

Lemma: Given a polynomial time approximation algorithm M_A for A that guarantees an approximation factor $\frac{a}{b}$, it can be used to solve the decision problem $gap - A[a, b]$

Proof: We use the following algorithm: Fix the size function g(x). Run M_A on x to get a feasible solution w. If $o_A(x,w) < a * g(x)$ then declare x as a negative input. Otherwise declare x as positive.

If x is a positive instance then $opt(x) \geq b * g(x)$ and the solution w obtained by M_A satisfies $o_A(x,w) \geq \frac{a}{b} opt(x) \geq a * g(x)$ and the algorithms accepts. If x is a negative instance then $opt(x) \leq a * g(x)$. Since $o_A(x,w) \leq opt(x)$ the algorithm rejects. ■

Corollary: If $gap - A[a,b]$ is NP-hard then so is the $\frac{a}{b}$ approximation of A.

A *gap-preserving* reduction f is a polynomial time Karp reduction from a gap problem $gap - A[a,b]$ to another gap problem $gap - B[c,d]$ such that for every positive input x for $gap - A[a,b]$, $f(x)$ is a positive input to $gap - B[c,d]$, and for every negative input x for $gap - A[a,b]$, $f(x)$ is a negative input to $gap - B[c,d]$. The don't-care inputs can be mapped to good arbitrarily. We denote the reduction by $\leq_p$.

2.2.8.2 Hardness of approximation

Problems that cannot be approximated within a constant factor unless $P = NP$ are problems whose approximation within any constant factor c is NP-hard. In most practical cases only heuristics can be used to solve them. Clearly, such problems are APX-hard.

As a result of the corollary in the previous section, a way to prove that a problem A has no constant factor approximation algorithm (unless P=NP) is to show that for every constant c there exist an NP-hard gap problem $gap - A[a,b]$, such that $\frac{a}{b} = c$. A problem shown to be NP-hard to approximate within any constant factor problem is the maximum clique problem [105, 106]. In this problem we are given an un-weighted, undirected graph G and the optimization objective is to find a maximum cardinality clique. Thus, NP-hardness of approximation of a maximization problem A can be proved if for every constant factor c there exists a gap-preserving reduction:

$gap - MaxClique[a,b] \leq_p gap - A[ka,kb]$

Where $\frac{a}{b} = c$, and $k > 0$.

We note that a strong non-approximability result was later proved by Hastad for the maximum clique problem, under stronger complexity assumptions [107].

3. A novel measure of class specific differential correlation

Given two genes and two classes, a naïve approach to measure the differential correlation between the genes would be to calculate the absolute difference in correlation between the classes. For multi-class data sets, different pairs of classes may manifest different distributions in their differential correlation and normalization is required. We assume that gene similarities within each class are distributed normally, such that each class has different mean and standard deviation. We estimate the distribution parameters of each class directly from the gene expression data. When analyzing two specific classes, we calculate the differential correlation scores of all gene pairs and standardize them. We use the class distributions to estimate the expected distribution of differences in co-expression, and score a pair by subtracting from its co-expression difference the mean difference and dividing by the standard deviation. The result is called the T-score of the pair. In multi-class data sets, this class pair-wise analysis produces several T-scores for the same pair of genes. Because we are interested in T-scores that quantify the differential correlation with respect to a specific class, we calculate the T-score of the specific class of interest with each of the other classes and integrate the different scores by taking the minimum (in absolute value) if the sign of all T-scores are consistent, and assign a zero score otherwise.

3.1 A normalized score of differential correlation

For genes u,v and a class D of profiles, define $R_D^{u,v}$ to be the Pearson correlation between u and v in that class. Given two classes D_i and D_j and a pair of genes u and v we assume that the correlations within each class are normally distributed with class-specific parameters:

$$R_{D_i}^{u,v} \sim N(\mu_i, \sigma_i)$$

$$R_{D_j}^{u,v} \sim N(\mu_j, \sigma_j)$$

We also assume that the correlations are independent. Hence the expected distribution of the difference satisfies:

$$R_{D_i}^{u,v} - R_{D_j}^{u,v} \sim N(\mu_i - \mu_j, \sqrt{\sigma_i^2 + \sigma_j^2})$$

All class-specific parameters ($\mu_i, \sigma_i, \mu_j, \sigma_j$) are directly evaluated from the input data. We can then calculate the normalized score, which we call the T-score (or the pairwise differential correlation score) of u and v:

$$T_{D_i,D_j}^{u,v} = \frac{\left(R_{D_i}^{u,v} - R_{D_j}^{u,v}\right) - (\mu_i - \mu_j)}{\sqrt{\sigma_i^2 + \sigma_j^2}}$$

In multi-class data sets, for two genes u and v, multiple T-scores are calculated. When analyzing the differential correlation with respect to a specific class D_k, we perform a "one vs. all" analysis. For each pair of genes u and v we check if all T-scores, calculated between D_k and all other classes, have the same sign (i.e. for the class D_k and each other class j, we check the sign of $T_{D_i,D_j}^{u,v}$). If the sign is consistent for all classes $i \neq k$ then we define the aggregated T-score as:

$$T_{OVA}(u,v) = \begin{cases} min_{i \neq k}\left(T_{D_i,D_k}^{u,v}\right) & if\ the\ sign\ is\ positive \\ max_{i \neq k}\left(T_{D_i,D_k}^{u,v}\right) & if\ the\ sign\ is\ negative \end{cases}$$

Otherwise we set $T_{OVA}(u,v) = 0$. Under this definition, positive aggregated scores mark the cases in which the correlation under D_k is higher than under all other classes. We call this situation 'up-correlation'. Negative aggregated scores indicate lower correlation of the pair under D_k. We call this situation 'down-correlation'. Zero score is obtained when the differential correlation of u and v is not consistent when D_k is compared to the other classes.

3.2 Experimental tests

In order to test if differential correlation is a wide phenomenon in real biological data we analyzed five categorical disease gene expression data sets (**Table 1**). For every data set we performed random permutations of the sample labels and calculated the integrated T-scores. If differential correlation is prevalent we expect that the real data sets will have higher T-

scores (in absolute values) than the randomized data sets. We compared the distributions of the real scores and the random scores on each data set. **Figure 1a** shows the T-score distributions for each data set. In the two-class data sets (AD, NDD and Lung cancer) there is a clear separation between the real and random distributions: both distributions are centered on zero T-score, whereas the variance in the real data sets is larger. In the multiclass case (IBD, SLE), because we assign non-zero T-scores only to the cases in which the sign of the T-score is consistent, at least 40% of the scores are zero both for real and random data sets. However, when we focus on the upper tail of the distributions (T-score > 0.5) we observe that the distribution of the real data set has a heavier upper tail than the distribution of the permuted data sets. **Figure 1b** shows a comparison of the standard deviations of each distribution in each data set. Since there is no large difference in the mean of the distributions (the difference was below 0.05 in all data sets) and there was a large difference in the variance, we conclude that high (absolute) T-scores are more likely on real data sets than on the permuted data sets.

Data set description	Data set name	classes	Class of interest	samples	GEO ID (ref)
Brain samples of patients with Alzheimer's disease and controls	AD	2	Alzheimer's disease (AD)	363	GSE15222 [38]
Brain samples from 6 different neurodegenerative diseases and control	NDD	2	Neuro-degenerative disorder (NDD)	118	GSE26927 [n/a]
Blood samples from healthy controls and Bowel diseases: Crohn, ulcerative colitis.	IBD	3	Crohn's disease	128	GSE3365 [39]
Healthy controls vs. lung cancer	Lung cancer	2	Cancer	187	GSE4115 [40]
Inflammatory and infectious diseases and healthy individuals	SLE	6	Systemic lupus erythematosus (SLE)	270	GSE22098 [98]

Table 1 The data sets used in this study. Gene expression profiles from five studies were obtained from GEO[108] using the series matrix of each data set. To reduce noise and focus on genes that vary across the study, in each data set we used the 3000 probes showing maximum variation and then merged probes by the probe to Entrez ID mapping.

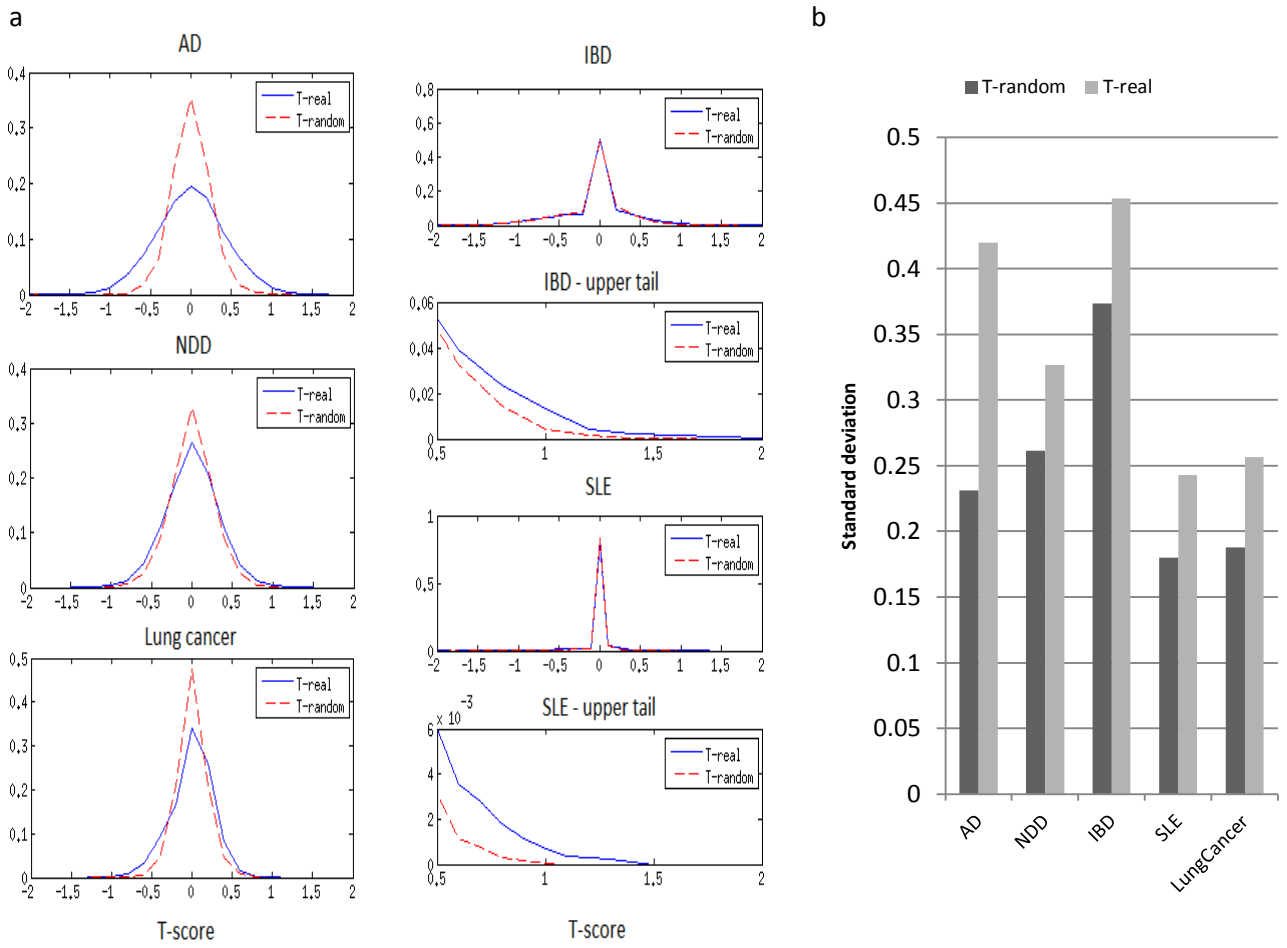


Figure 1 T-score distributions in real and randomized data sets. a) The distributions of the T-scores in the real and randomized data sets. The variance of the distributions is larger for the real T-scores, whereas the average might not change. Since in the IBD and SLE data sets most T-scores are around zero, we also show the upper tails of the distributions. b) The standard deviation of the T-scores in the real and randomized data sets. The standard deviation is larger in all tested data sets, indicating that high T-scores (in absolute values) are more probable in the real data sets. Randomized data sets were generated by shuffling sample labels. Results are the average of 50 randomized datasets.

4. The DICER Algorithm

4.1 Algorithm overview

We developed a novel algorithm, called DICER (Differential Correlation in Experssion for meta module Recovery) for extracting gene modules that manifest differential correlation, with respect to a specific phenotype. **Figure 2** shows an overview of the algorithm. Generally, our algorithm has four phases: (1) normalizing the T-scores, (2) finding class-specific up or down correlated gene pairs and clusters, (3) consistent similarity analysis, and (4) integration to find meta-modules. Our method detects two different types of gene sets: differentially correlated gene sets (clusters) and meta-modules. A class-specific differentially correlated gene cluster is a group of genes that are significantly more correlated (or significantly less correlated) in the tested class. We denote up-correlated gene cluster as UCC and down-correlated gene cluster as DCC. A meta-module is a pair of gene modules, in which each module is a set of genes that are correlated across all phenotypes, whereas two genes that belong to different modules are differentially correlated. We denote an up-correlated meta-module as UCMM and a down-correlated meta-module as DCMM.

Figure 3 shows two examples of gene sets detected by DICER in the Alzheimer's disease (AD) and Lung cancer data sets (ref). **Figure 3a** shows an up-correlated gene cluster of 242 genes that was discovered in the AD data set. The average correlation of these genes in the controls class is 0.437 whereas the average correlation in the AD class is 0.715. This cluster is significantly enriched with many functional terms ($p< 0.05$ after FDR correction). It contains 80 genes related to cerebellum activity (p=3.7E-10), 13 Spliceosome genes (p=1.29E-6) and genes that belong to protein complexes related to miRNA processing: large Drosha complex (6 genes, p=7.53E-6) and DGCR8 multiprotein complex (5 genes, p=3.1E-5). Hence, AD patients show higher levels of correlation of these processes. **Figure 3b** gives an example of down-correlated meta-module in the Lung cancer data set. This meta-module contains two modules of sizes 39 and 77, the average correlation between the modules is -0.43 in the controls class and -0.86 in the Lung cancer class. The average correlation within the modules is above 0.75 in the two classes. The large module is significantly enriched with C complex Spliceosome (4 genes p=8.4E-3) and protein complexes related to miRNA processing: large Drosha complex (3 genes, p=4.4E-3) and DGCR8 multiprotein complex (3 genes, p=9.7E-4).

Unlike the AD case, the miRNA related complexes in this data are correlated in all classes. Most genes in the small module are poorly functionally annotated as most genes (>95%) are not assigned to GO biological processes and Kegg pathways. Thus, the meta-module presented in Figure 3b can provide new candidate genes that are related to lung cancer phenotype. Figure 3c shows the expression values of two genes of the meta-module, ALPK1 and RAD23B. They are negatively correlated in the lung cancer class samples (r=-0.76) but are un-correlated in the controls (r=-0.12).

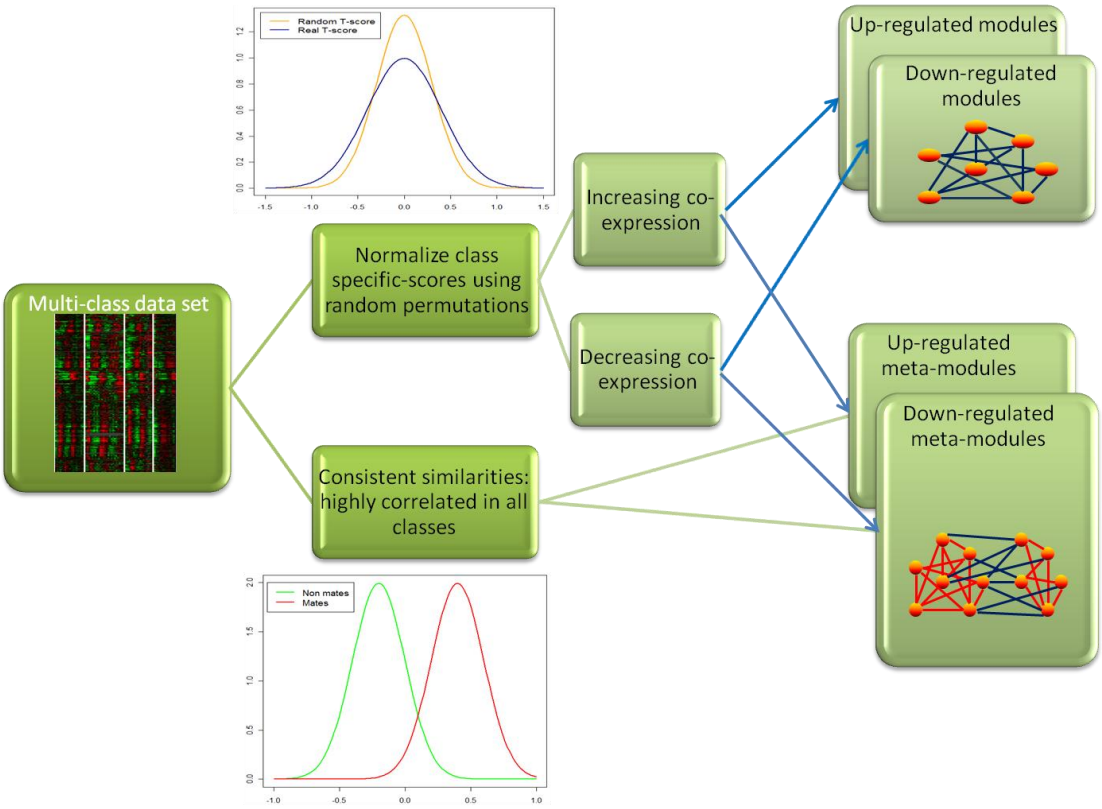


Figure 2 Overview of the class specific differential correlation analysis. The input is a set of expression profiles from different classes of samples. T-scores are computed for the class of interest and are normalized using the T-scores calculated on random data sets, created by shuffling the sample labels. The normalized scores are then used to find gene clusters that manifest differential correlation in the tested class compared to all other classes (up/down-correlated modules; blue edges indicate class-specific differential co-expression). A second similarity analysis is performed in order to detect gene pairs that are co-expressed in all classes. In each class an EM algorithm is used to divide the correlations to high ('denoted as 'mates', red distribution) and low (denoted as 'non-mates', green distribution) and consistent similarities are defined as cases in which gene pairs are mates in all classes. The normalized T-scores and the consistent correlations are used to find pairs of gene modules in which each module is a group of consistently correlated genes (red edges), whereas the correlation between the modules is differential (blue edges), these module pairs are denoted as meta-modules.

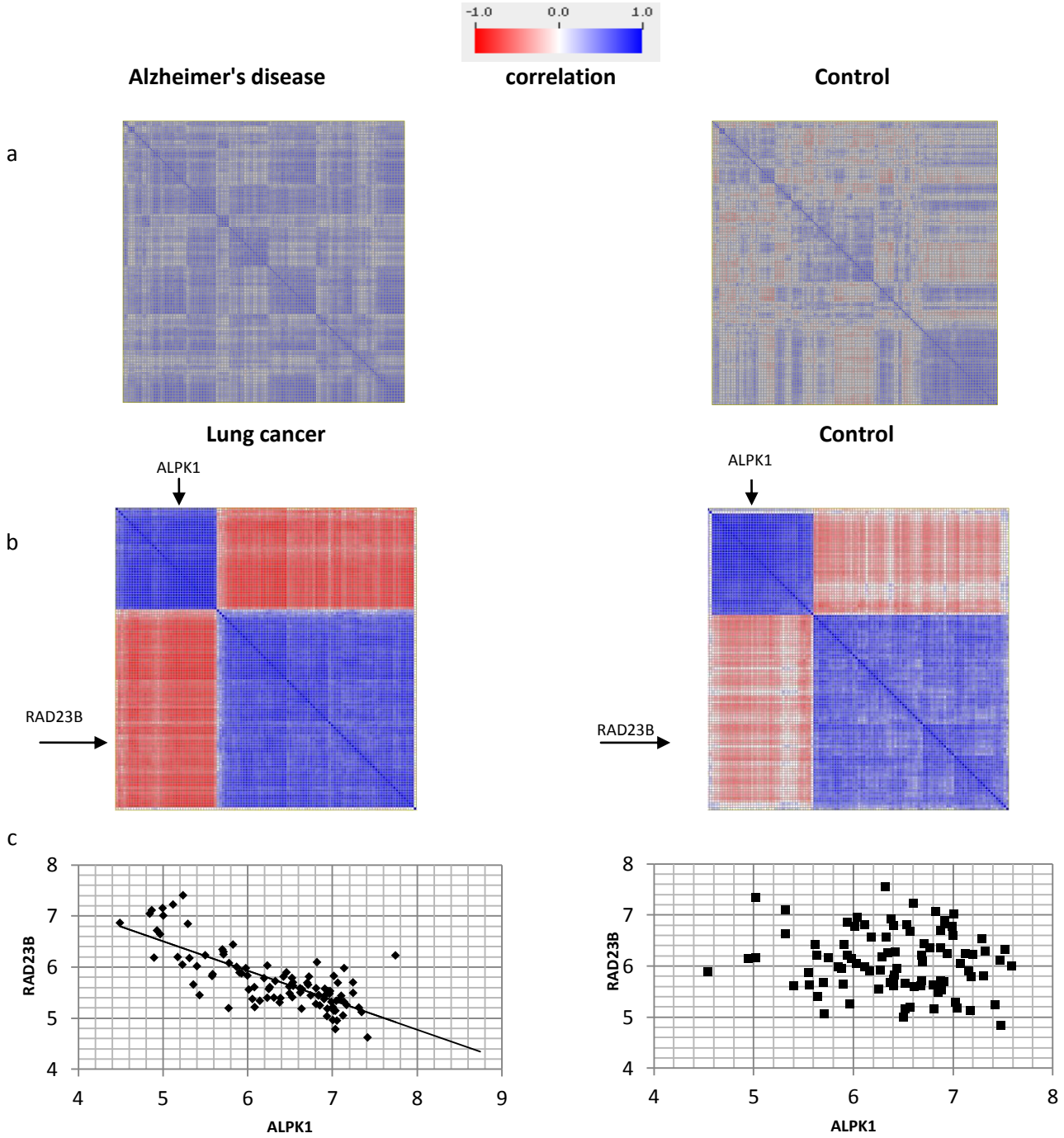


Figure 3 Examples of differential correlation patterns. a) An up-correlated 242-gene cluster discovered in the AD data set. The correlation matrices of the cluster genes in the AD and control classes are shown. The average correlation is 0.72 and 0.44 in the AD and the control classes, respectively. b) A down-correlated meta-module discovered in the lung cancer data. It contains two gene modules of sizes 39 and 77. The correlation matrices of the meta-module genes are shown for the lung cancer and the control classes. The correlation between the two modules is -0.43 in the control class, whereas the correlation in the lung cancer class drops to -0.86. Each module is a group of genes that are highly correlated in both classes: the average correlation within each module is > 0.75. c) The correlation between genes RAD23B and ALPK1. The two genes are marked by arrows in b. Each dot corresponds to a patient and the axes mark the logarithm of expression values of the two genes in that patient. The genes are negatively correlated in the lung cancer class (r=-0.76) but are un-correlated in the controls (r=-0.12).

In the first analysis of DICER, we calculate the T-scores with respect to a specific class of interest. We then create synthetic data sets by randomly shuffling the patients' labels, and compare the original T-scores to the ones calculated on the randomized data sets. By calculating the log likelihood ratios (LLR) between the two samples, positive scores are assigned to gene pairs whose T-score is not likely to occur by chance. We use the LLR scores as edge weights between genes to construct two graphs: up correlations graph (denoted as G_{up}) and down correlations graph (denoted as G_{down}). We then use hierarchical clustering [67, 68] on both graphs to find UCCs and DCCs (**Figure 2**, top right). Clusters are determined by going up the hierarchy as long as the sum of LLR scores within a group is positive. This statistical model for scoring a cluster is similar to that of [64] , and ensures that the accepted gene clusters are likely to represent a significant phenomenon. Only clusters that contain at least 15 genes are accepted.

In addition, we perform statistical analysis to detect gene pairs that are consistently correlated in all classes. Our analysis follows the model presented in [64]. The goal is to find gene pairs that are co-expressed in all classes. For each class we compute the gene correlations and divide them, using an Expectation Maximization (EM) algorithm, into pairs showing high correlation ('mates') and low correlation ('non-mates'). We then calculate, for each correlation score, the LLR score between the mates probability and the non-mates probability. Gene pairs that are co-expressed in all classes will induce a positive LLR score in each class; therefore, we assign for each gene pair the minimal LLR score. We denote the resulting graph as the Consistent Correlation graph (CG).

The final step of the algorithm finds meta-modules (**Figure 2**, bottom right). The algorithm receives as input two weighted graphs: the CG graph and a class-specific differential correlation (DC) graph (i.e. either G_{up} or G_{down}). A simple greedy procedure iteratively finds a meta-module and removes its genes from the graphs. The procedure takes an edge with high DC weight and examines the local neighborhoods of its endpoints. We remove genes that do not manifest high scores in CG in their own neighborhood and genes that do not manifest high scores with the other neighborhood in the DC graph. **Figure 4a** shows a simple example of a module pair discovery. We improve the initial solution by merging meta-modules (**Figure 4b**) and adding single genes to a meta-module (**Figure 4c**). **Figure 4d** shows the final meta-module.

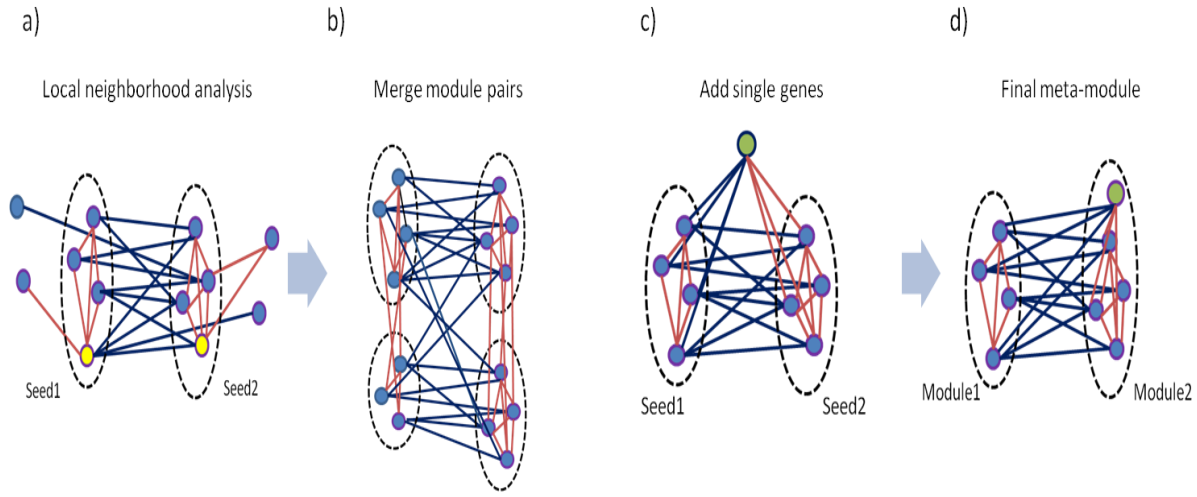


Figure 4 Overview of the steps of the meta-module discovery algorithm. The seeds that will form the basis for modules of a meta-module are encircled with dashed lines in a-c. Blue edges correspond to differentially correlated gene pairs. Red edges correspond to consistently correlated gene pairs. a) The construction starts from the edge between the yellow nodes. A seed is formed around each of them, containing a set of consistently correlated genes, whereas edges between the two seeds correspond to differentially correlated genes. Genes that are consistently correlated with one of the seeds but are not differentially correlated with the other are excluded. Genes that are differentially correlated with one seed but are not consistently correlated with the other are removed as well. b) Merging two meta-modules. The resulting meta-module has high differential correlation between the two sides and high consistent correlation within each side. c) Addition of a single gene to a meta-module. The gene colored green is added to seed2 because is differentially correlated with seed1 and consistently correlated with seed2. d) The final meta-module. The two sub-groups of the meta-module are denoted as modules.

We discovered that in many cases our algorithm produces many meta-modules, therefore we scored each meta-module by the sum of LLR scores between the modules and use the top ten UCMMs and top ten DCMMs as output. As we shall show, the large number of modules that we produce compared to other method does not come at the expense of their quality.

4.2 The probabilistic framework

We adopted the framework of [64]. In the following sections we will use this framework to compare the $T_{OVA}(u,v)$ on real and random data sets, and to compare high and low correlation values within each class. Therefore, we first describe here this framework in a generic manner. We assume that scores belong to one of two distributions with density

functions f_1 and f_0, where the probability of belonging to the first distribution is p, and define the log-likelihood ratio score of a score s as:

$$LLR(s) = log\ \frac{p\ f_1\ (s)}{(1-p)f_0(s)}$$

In our analyses we assume that f_1 and f_0 are normal distributions with different means μ_1, μ_0 and standard deviations σ_1, σ_0. p_1 is the prior probability that a score is sampled from f_1. Hence we transform a score x into a LLR score:

$$LLR_{1,0}(x) = log\ \frac{p_1\ f\ (x|\mu_1, \sigma_1\ )}{(1-p_1)f\ (x|\mu_0, \sigma_0)} = log\frac{p_1\sigma_0}{(1-p_1)\sigma_1} + \frac{(x-\ \mu_0)^2}{2{\sigma_0}^2} - \frac{(x-\ \mu_1)^2}{2{\sigma_1}^2}$$

Define $G = (V, E)$ to be a weighted undirected graph, in which the nodes correspond to genes and edge weights correspond to LLR scores. Given a set of edge scores C we would like to test the following two hypotheses:

$$H_0^C: C\ is\ a\ group\ of\ scores\ sampled\ from\ f_0$$

$$H_1^C: C\ is\ a\ group\ of\ scores\ sampled\ from\ f_1$$

Let $P\left(H_i^C\ \middle|C\right)$ be the posterior probability of H_i^C for $i = 0,1$. Under a simplifying assumption that the set of random variables within C are independent, define:

$$P\left(H_i^C\middle|C\right) = \prod\nolimits_{c\in C} p_1^i(1-p_1)^{1-i}\ f_i(c|\mu_i, \sigma_i)$$

Thus:

$$\log\frac{P\left(H_1^C\ \middle|C\right)}{P\left(H_0^C\ \middle|C\right)} = \sum_{c\in C} LLR_{1,0}(c)$$

For a set of scores C we accept H_1^C if and only if $P\left(H_1^C\ \middle|C\right) > P\left(H_0^C\ \middle|C\right)$. Thus we accept H_1^C if and only if $\sum_{c\in C} LLR_{1,0}(c) > 0$.

4.3 Class-specific differential correlation analysis

We first perform an analysis of a specific class D_k against all other classes by calculating the T_{OVA} score for each gene pair. We denote the resulting distribution as T_{OVA}^{real}. Next, we create random data sets by permuting class labels and calculate the T_{OVA} scores. The process is repeated 20 times and we denote the resulting distribution T_{OVA}^{random}. We then transform the T_{OVA}^{real} scores to LLR scores as described above using the distributions of T_{OVA}^{real} and T_{OVA}^{random}. An important parameter in this process is the prior p assigned for T_{OVA}^{real}. By decreasing this parameter we can control the number of T-scores that receive a positive score. Setting low prior means that less T-scores will get a positive LLR score. Because this parameter depends on the tested data set we set it to:

$$Pr_{T_{OVA}^{real}}(x \geq \mu_{T_{OVA}^{random}} + 2\sigma_{T_{OVA}^{random}})$$

We note that if T_{OVA}^{real} and T_{OVA}^{random} are equal then by using this prior almost all LLR scores would be negative.

Next, we define two weighted, undirected graphs in which we assign a node for each gene and an edge between each gene pair. The first graph is denoted G_{up} and its edge weights are defined as:

$$w(u,v)_{up} = \begin{cases} -LLR_{T_{OVA}^{real},T_{OVA}^{random}}\left(T_{OVA}(u,v)\right) & if\ T_{OVA}(u,v) < 0 \\ LLR_{T_{OVA}^{real},T_{OVA}^{random}}\left(T_{OVA}(u,v)\right) & otherwise \end{cases}$$

The rationale here is that only gene pairs that were up-correlated in D_k will be assigned positive scores. Similarly, we define the second graph, denoted as G_{down}:

$$w(u,v)_{down} = \begin{cases} -LLR_{T_{OVA}^{real},T_{OVA}^{random}}\left(T_{OVA}(u,v)\right) & if\ T_{OVA}(u,v) > 0 \\ LLR_{T_{OVA}^{real},T_{OVA}^{random}}\left(T_{OVA}(u,v)\right) & otherwise \end{cases}$$

We then use average linkage hierarchical clustering [67, 68] to find sub-graphs in G_{down} and G_{up} . Going from the leaves up, we merge two gene sets as long as the sum of edge scores between the merged sets is positive. Sets of size > 15 are defined as clusters. The rationale is that these sets correspond to gene modules that are differentially correlated, as they are more likely to represent a real correlation change than expected by chance.

4.4 The consistent correlation graph

In this analysis we calculate all gene-pair correlations in every class. We then divide the correlation scores to high correlations (denoted as 'mates') and low correlations (denoted as 'non-mates') within each class. We assume that the mates and non-mates distributions are normal, and use expectation maximization (EM) algorithm to represent the data as a mixture of two Gaussians [109]. The EM evaluates their parameters and the prior probability that two randomly chosen elements are mates. Because the distributions may vary among different classes, we perform the EM step in each class separately. We use these parameters to calculate the LLR scores as in the previous section. Gene pairs that are co-expressed in all classes will induce a positive LLR score in each class; therefore, to ensure that only gene pairs that are consistently co-expressed in all classes would be assigned with a positive score, we assign for each gene pair the minimal LLR. We use these scores as weights of edges in the undirected graph of consistent correlations, denoted as CG.

4.5 Finding meta-modules

4.5.1 Hardness of approximation

We show here the hardness of the problem of finding a maximal meta-module.

The formal statement of the problem is as follows: we are given two edge-weighted graphs G=(V,E) and G'=(V,E') with the same vertex set V. A meta-module is two disjoint vertex sets S, T in V such that the sum of edge weights of S and T in G is positive and the sum of weights between S and T is positive in G'. The goal is to find a meta-module of maximum cardinality $|S \cup T|$.

Theorem: Finding a maximum size meta-module is NP-hard to approximate within any constant factor.

Proof: We show that the problem is NP-hard to approximate within a constant factor via a gap preserving reduction from the max-clique problem . Given the input graph G'' = (V'',E'') with *n* nodes for the maximum clique problem, we define the node set for G and G' as $V = V_1 \cup V_2$ where V_1 and V_2 are copies of V''. For every edge (u, v) in E'', let u_1, u_2 and v_1, v_2 be the copies of u and v respectively. In G we set $w(u_1, v_1) = 1$ and $w(u_2, v_2) = 1$. In

G' we set $w(u_2, v_1) = 1$, $w(u_1, v_2) = 1$, $w(u_1, u_2) = 1$, and $w(v_2, v_1) = 1$. All other edges are scored -$4n^2$. The reduction is clearly polynomial.

Note first that any module in a meta-module cannot contain a negative edge, since the sum of the weights in such a module would be negative. Hence, every module must correspond to a clique in G''. If there is a clique C with at least b nodes in G'', then it will induce a meta-module with at least 2b nodes in G and G', by taking the two copies of C in G and G' as the modules. If a meta-module with at least 2k nodes exists in G and G', then one of its modules has at least k nodes, and such a module corresponds to a clique in G''. In other words, if there is no clique of size a in G'' then there is no meta-module of size 2a in G and G'. Thus, we have shown a gap preserving reduction:

$$Max\ clique[a, b] \leq_p\ Max\ meta - module\ [2a, 2b]$$

Since the max-clique problem is NP-hard to approximate within a constant factor [105-107], we conclude that the maximal meta-module detection is also NP-hard to approximate within a constant factor.■

4.5.2 Heuristic

In the final analysis we use the graphs CG and either G_{down} or G_{up} to find meta modules. We describe the analysis using CG and G_{up}; the analysis of CG and G_{down} is analogous. We define two disjoint gene sets U and V as $friends$ if the sum of edge weights between U and V in G_{up} is positive. Using the probabilistic framework, two modules that are friends are more likely to represent a real correlation change than expected by chance. We define an up-correlated meta-module MM as a pair of non-overlapping gene modules (M_i, M_j) that satisfy: (1) each module M_i is a sub-graph of CG with a positive sum of edge weights, and (2) M_i and M_j are $friends$.

We now present our algorithm for meta-module detection. We developed a three-phase heuristic for meta-modules detection: (1) initial module pairs detection, (2) a greedy merge of pairs, and (3) addition of single genes to pairs. In the next section we shall describe each phase of the algorithm.

Initial module-pairs detection

We use a simple greedy local heuristic to find pairs of initial modules, which we call seeds. Using the definition of G_{up}, we assume that only a small fraction of the edges would be assigned with positive scores. Let E' be the set of positive edges in G_{up}, and let $G' = (V, E')$ be the un-weighted graph induced by E'. We iteratively select the edge (u, v) in G' such that u and v have together a maximal number of neighbors. Let C_1 be the set of nodes that are neighbors of u and not neighbors of v, and let C_2 be the set of nodes that are neighbors of v and not neighbors of u. We repeatedly remove nodes from C_i whose sum of edge weights with C_i in CG is non positive, or have a non positive sum of edge weights $C_j, j \neq i$ in G_{up}.

To determine the order of node removal we score each node by the sum of scores in CG with its set plus the sum of scores with the other set in G_{up}. When inspecting which node to remove we consider three possible candidates: (1) the node that has the minimal score, (2) the node that has the minimal score with the other set in G_{up} , and (3) the set that has the minimal score with its own set in CG. If all three candidates have positive scores we stop and accept the meta-module. We found out that in many cases the first candidate manifests negative scores both within its group (in CG) and with the other group (in G_{up}). Therefore, in these cases we remove this gene. However we observed cases in which node (1) manifests positive score within its group (in CG) or with the other group (in G_{up}), while having a negative score. In these cases we use node (1) as a "guide" for the removal stage: if it has a negative score in G_{up} (with the other set) then the edges between the two seeds in G_{up} are not heavy enough, and we remove node (2). Otherwise we remove node (3). Once a pair is detected, its genes are removed and the process repeats.

Greedy merge of module pairs

In the second phase we improve the solution by merging meta-modules. Let $MM_1, MM_2, \ldots, MM_M$ be the current set of meta-modules discovered in phase 1, where each meta-module MM is a pair of two non-overlapping gene modules. A pair of meta-modules $MM_i = \{M_i^1, M_i^2\}$ and $MM_j = \{M_j^1, M_j^2\}$ can be merged if one of the merging options $\{M_i^1 \cup M_j^1, M_i^2 \cup M_j^2\}$ or $\{M_i^1 \cup M_j^2, M_i^2 \cup M_j^1\}$ leads to a gain in the scores both within the

modules and between them. We iteratively merge the best gain meta-module pair until the gain is negative.

Adding single genes to meta-modules

In the third phase we improve the solution by adding single genes that do not belong to any module to meta-modules. A gene g can be added to a meta-module $MM_i = \{M_i^1, M_i^2\}$ if one of the merging options $\{M_i^1 \cup g, M_i^2\}$ or $\{M_i^1, M_i^2 \cup g\}$ leads to a gain in the scores within the modules and between them. We iteratively look for the best gene and meta-module gain, and add the gene to the appropriate module. We stop this process when the best gain is negative.

5. Experimental results

5.1 Comparison with other differential correlation gene module discovery methods

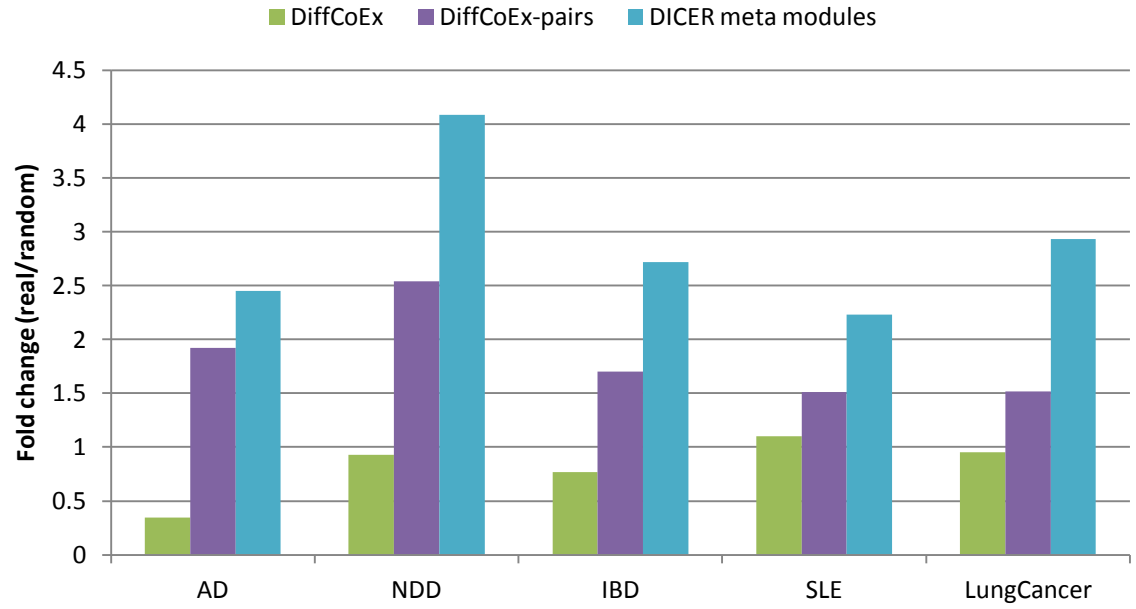
Extant differential correlation-based methods look for gene modules with altered correlation patterns among different classes. For example, the CoXpress method [85] uses hierarchical clustering to find gene modules, and tests their significance using random sampling. A recent method called DiffCoEx [86] transforms the correlation differences to a distance matrix in which two genes are close if both have significant correlation changes with the same group of genes. Unlike DICER, DiffCoEx does not output module pairs, and even if two modules are differentially correlated, the correlation within each module can be differential as well. Another difference is that the DICER algorithm uses statistical normalization of the DC scores to ensure that the accepted modules are significant.

We compared DICER with DiffCoEx and CoXpress on the five data sets described in **Table 1**. In all cases CoXpress did not find significant clusters that contain at least 15 genes. DiffCoEx and DICER detected modules in all data sets. On average DiffCoEx solution covers 29% of the genes (in 4.2 modules on average) whereas our approach covers 34% of the genes (in 23 modules on average, see **Supplementary Table 1** for module statistics). DICER detected differentially correlated gene clusters only in the AD and SLE data sets.

We first compared DICER and DiffCoEx in terms of the significance of differential correlation change. For every method and each pair of modules we created 200 random module pairs of the same sizes, calculated the average absolute change of correlation between the modules, and measured the fold change between the real module pair and the average of the random module pairs. In addition, for each DiffCoEx module we created 200 random modules of the same size, calculated the average absolute change in correlation within each module, and measured the fold change between the real module and the average of the random modules. Because many of DiffCoEx modules and module pairs did not manifest a fold change above 1, we compared the average fold-change of all DICER meta-modules to those of the top two DiffCoEx modules. We also included in the comparison out of all possible DiffCoEx module pairs, the ones with fold change above 1.1. The results are shown in **Figure 5a**. In five data sets the fold change between DiffCoEx modules was higher than within modules, and the fold

change of DICER meta-modules was higher than both. In addition, the fold change within DiffCoEx modules is close or below one. Thus, in most cases DiffCoEx can be used to discover differentially co-expressed module pairs. We also inspected the within- and between- module absolute correlation change. Results on the AD and Lung cancer data sets are shown in **Figure 5b**. Both methods achieve a marked separation between the within-module and between-modules correlation changes. The between-module correlation change distribution of DICER is significantly shifted towards higher absolute correlation changes compared to the between-module distribution of DiffCoEx (Kolmogorov-Smirnov $p < 1E\text{-}20$).

a



b

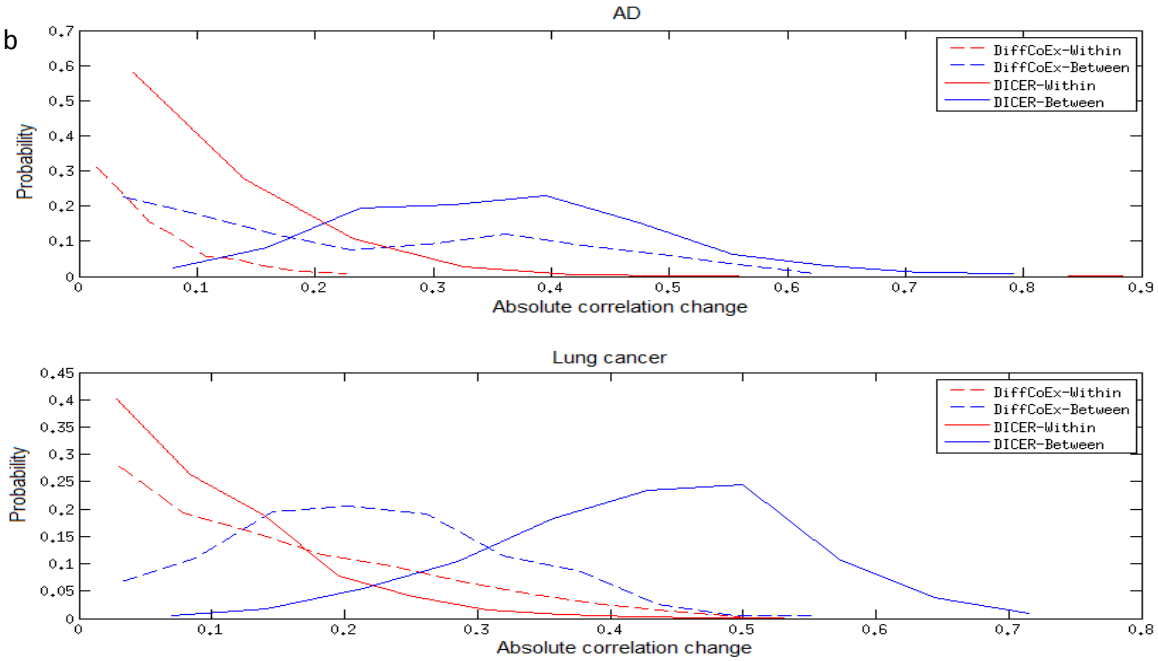


Figure 5 The distribution of within and between-module scores for DICER and DiffCoEx. a) The discovered differential co-expression pattern compared to patterns derived from random gene sets. For each discovered module and module pair we created 200 random gene sets of the same size and calculated their absolute differential co-expression. We then calculated the fold change between the real modules and the average of the random gene sets. Similarly, for every module pair we generated 200 random module pairs of the same size and calculated the fold change of absolute correlation change between the real module-pairs and the average of the random module-pairs. The green bars show the average of the top two DiffCoEx modules in each dataset. For testing DiffCoEx module pairs (purple bars) we took into account only module pairs with fold change above 1.1. For DICER (blue bars), the top ten up-correlated and the top ten down-correlated module pairs were taken into account. b) The distribution of within- and between-module absolute correlation for DICER and DiffCoEx, in the AD and lung cancer data sets.

We also compared the functional enrichment of DICER and DiffCoEx modules, by performing KEGG pathways enrichment analysis (FDR threshold of 0.05 followed by redundancy filtering, see section 5.4). The number of significantly enriched pathways by each method is shown in **Figure 6a**. Both methods did not achieve significant enrichment on the IBD data set. DiffCoEx did not achieve significant enrichment in both the AD and Lung cancer data sets either, while DICER results had 14 and 24 enriched pathways respectively. In the NDD and SLE data sets the two methods reported similar numbers of pathways. In addition we calculated the average enrichment factor of the accepted terms. Enrichment factor is defined as the ratio between the fraction of pathway genes in the tested module and the fraction of the pathway genes among all expressed genes in the data set. Therefore high enrichment factor scores indicate specificity of the gene set to the tested pathway. The results are shown in **Figure 6b**. In all cases DICER had an advantage of 50% or more in terms of average enrichment factor.

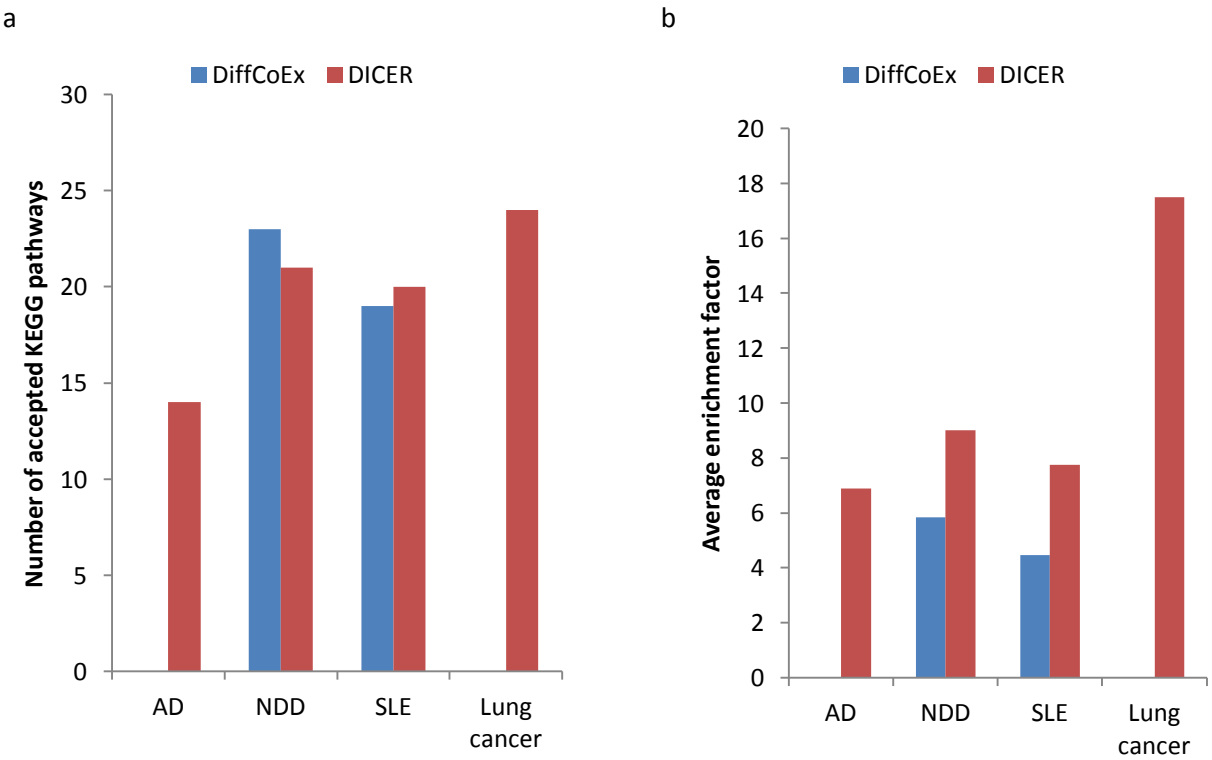


Figure 6 KEGG pathway enrichment analysis. The modules of DiffCoEx and DICER were tested for KEGG pathway enrichment using the hyper-geometric test with 0.05 FDR for multiple tests correction. Both methods did not report significant enrichment on the IBD data set. a) The number of enriched pathways. b) Average enrichment factors of the enriched sets. The enrichment factor is defined as the ratio between the fraction of the pathway genes in the tested set and the fraction of the pathway genes in the data set.

5.2 Discovered gene sets are highly enriched with miRNA targets

Under the assumption that genes that are co-expressed are more likely to be co-regulated, changes in co-expression may be due to changes in regulatory patterns. Therefore, we tested if differentially correlated gene modules are significantly enriched with genes that are targets of specific miRNA families, and if there are cases in which the discovered gene targets emerge from different modules within the same meta-module. We used the FAME algorithm [99] for miRNA binding site enrichment analysis. Acceptance threshold was set to 0.05 after FDR correction. We tested for miRNA enrichment in the three types of gene sets that were discovered by DICER: (1) consistently co-expressed modules, (2) meta-modules, and (3) differentially correlated clusters. The results are shown in **Figure 7**. Except for the modules in the lung cancer data, in every data set and gene set type combination, DICER revealed significant enrichments (see **Supplementary Table 2** for full list of families in each case). Notably, in some cases the same miRNA was found in different gene sets. For example, miRNA family mir-124/506 was detected in two meta-modules in the AD data set. However, in all cases some miRNA families were detected only in the meta-modules or only in the gene clusters. In contrast, DiffCoEx obtained very few miRNA enrichment results: no miRNA families were detected in the IBD, SLE and lung cancer data sets, one in AD and one in the NDD data set.

To test if the discovered miRNA families are known to be associated with the tested disease, we used the mir2disease database [110] and tested if the overlap between the detected miRNA families and the annotations in mir2disease is significant (see section 5.4). Mir2disease contained miRNA-disease associations for three out of the five diseases in our experiments: AD, NDD, and lung cancer (marked results in **Supplementary Table 2**). In the AD case, significant overlaps were obtained in clusters (six known AD-related miRNAs, p=0.0023), and marginally significant overlaps were detected in modules (six miRNA families, p=0.068). In the NDD data set, significant overlap was obtained for modules (all five miRNAs detected in the modules are associated with NDD, p=0.0016). In the lung cancer, seven out of 12 enriched miRNA families detected in meta-modules are associated with the disease (p=0.037). In contrast, the DiffCoEx algorithm obtained no significant enrichments. We conclude that disease-specific miRNAs can be detected by focusing on differential correlation patterns.

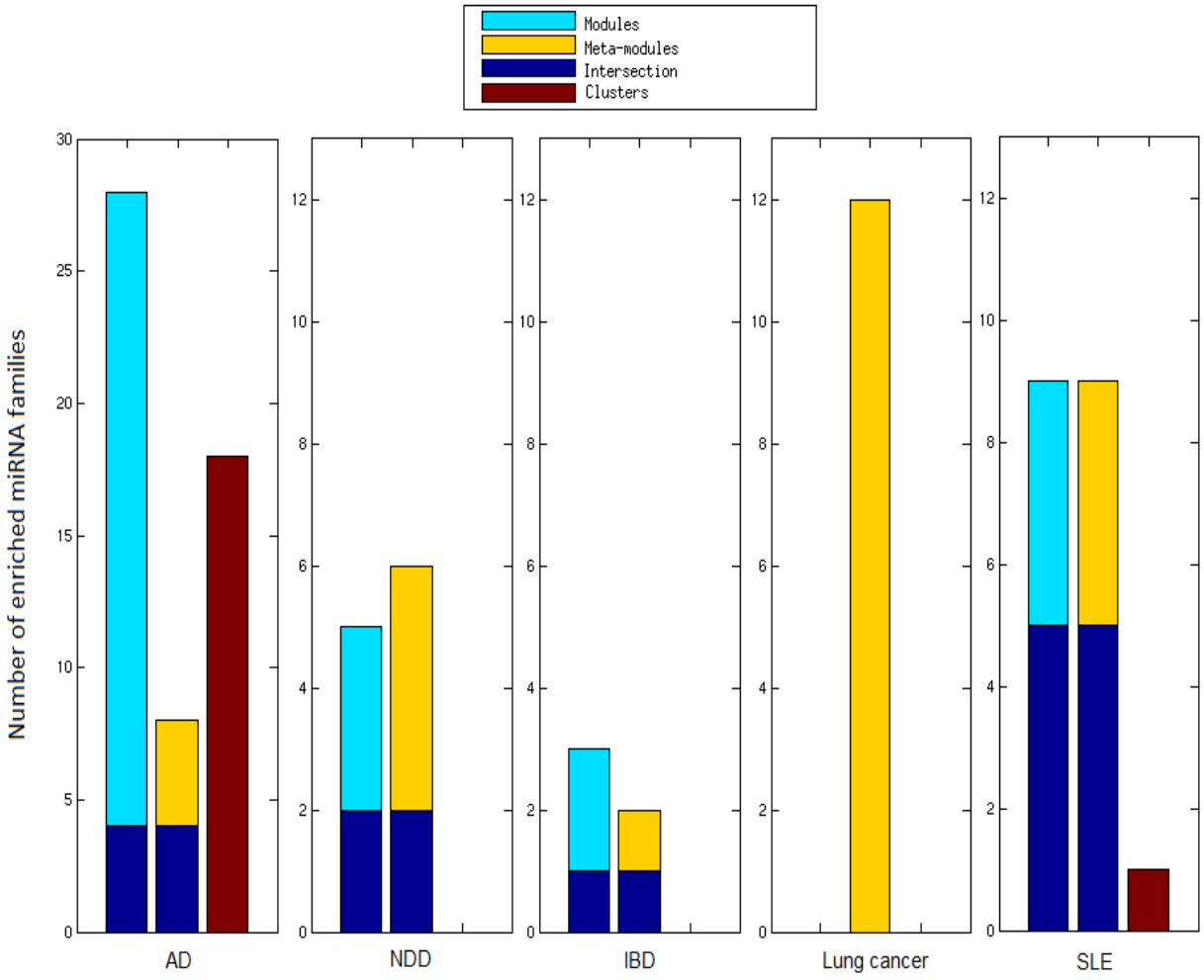


Figure 7 microRNA target enrichment in gene sets constructed by DICER. For every dataset we used the FAME algorithm to test for enrichment in targets of miRNA families in the gene sets generated by DICER. These included gene clusters, meta-modules and modules (the sub-groups of meta-modules). P-values were corrected for multiple testing (0.05 FDR). Because the modules are sub-groups of meta-modules we also calculated the intersection between enriched miRNA families in meta-modules and modules. Note that all data sets except the AD are on the same scale.

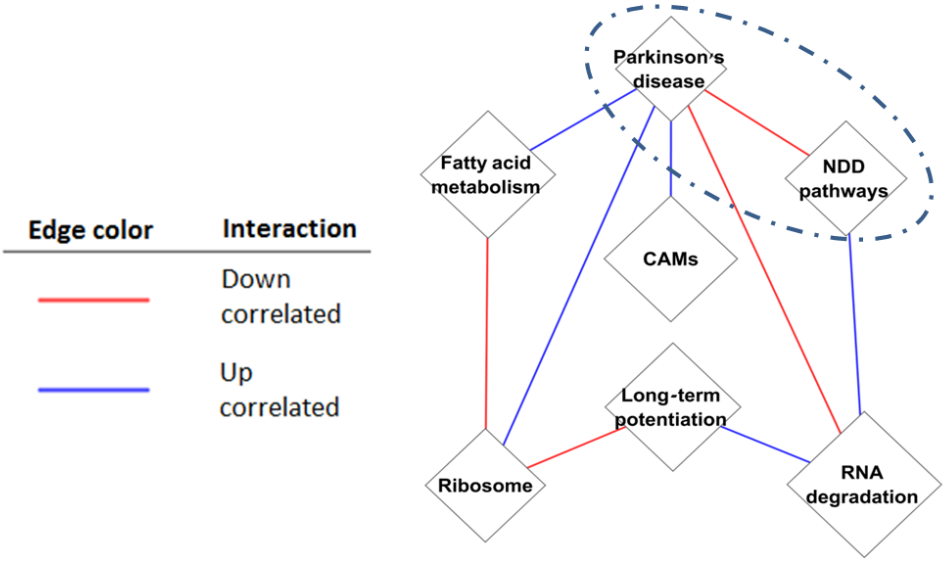
5.3 Case study: Alzheimer's disease

Alzheimer's disease (AD) is the most common progressive neurodegenerative brain disorder in human. AD is a complex progressive condition that involves sequentially interacting pathological cascades, including the interaction of amyloid-β (Aβ) aggregation with plaque development, and the hyperphosphorylation and aggregation of tau protein with formation of tangles. Together with associated processes, such as inflammation and oxidative stress, these pathological cascades contribute to loss of synaptic integrity and progressive neurodegeneration [59].

We compared the enriched miRNA families that were detected by DICER on differentially correlated clusters or meta-modules to the mir2disease database. These miRNA families cover 10 (24%) of the miRNAs that are associated with AD: mir-101, mir-106, mir-124a, mir-125, mir-26b, mir-29a, mir-29b-1,mir-363, mir-9, and mir-93. Furthermore, three of these miRNAs were annotated as causal for AD: mir-106, mir29a and mir-29b-1. The mir-17-5p/20/93.mr/106/519.d family was detected in an up-correlated gene cluster, and these miRNAs where shown to target the amyloid precursor gene (APP), and therefore have potential relevance to human neurodegenerative disorders [111]. Down regulation of these miRNAs may favor high APP levels in AD [112]. Mir-106, which is annotated as causal in mir2disease, was shown to interact with APP [113]. mir-101 was shown to be down-regulated in sporadic AD brains [113]. Mir-124, together with mir-125, was shown to be abundantly represented in AD hippocampus [114] . It was also shown that mir-124 is involved in nervous system development by regulation of neuron-specific alternative splicing [115]. Loss of mir-29a\b-1 in sporadic AD patients was shown to be correlated with increased BACE1/beta-secretase expression, where the cleavage of BACE1/beta-secretase is known to be the rate limiting stage of Aβ production [116]. mir-9, together with mir29-a and mir29-b1, was shown to regulate BACE1 expression in vitro [116].In addition to finding known AD-related miRNAs the FAME analysis of the DICER modules provides new candidate miRNAs with relevance to AD. For example, mir-216 was found enriched in a UCMM (p=0.001), and was predicted by FAME to target the solute-carrier family member gene SLC1A2, which is important in excitatory glutamate clearance in the central nervous system. This miRNA was shown to be expressed in glioblastoma and astroblastoma cell lines [117]. Together with mir-203, which was enriched in a UCM (p=5E-4), this miRNA was validated to target GABA receptor α1 subunit [118]. GABA receptors are known as the inhibitory receptors in the central nervous system [119]. Taken together, this shows that DICER can detect well established disease-

related regulatory factors, and can also point out new candidates that may affect the tested disease.

a



b

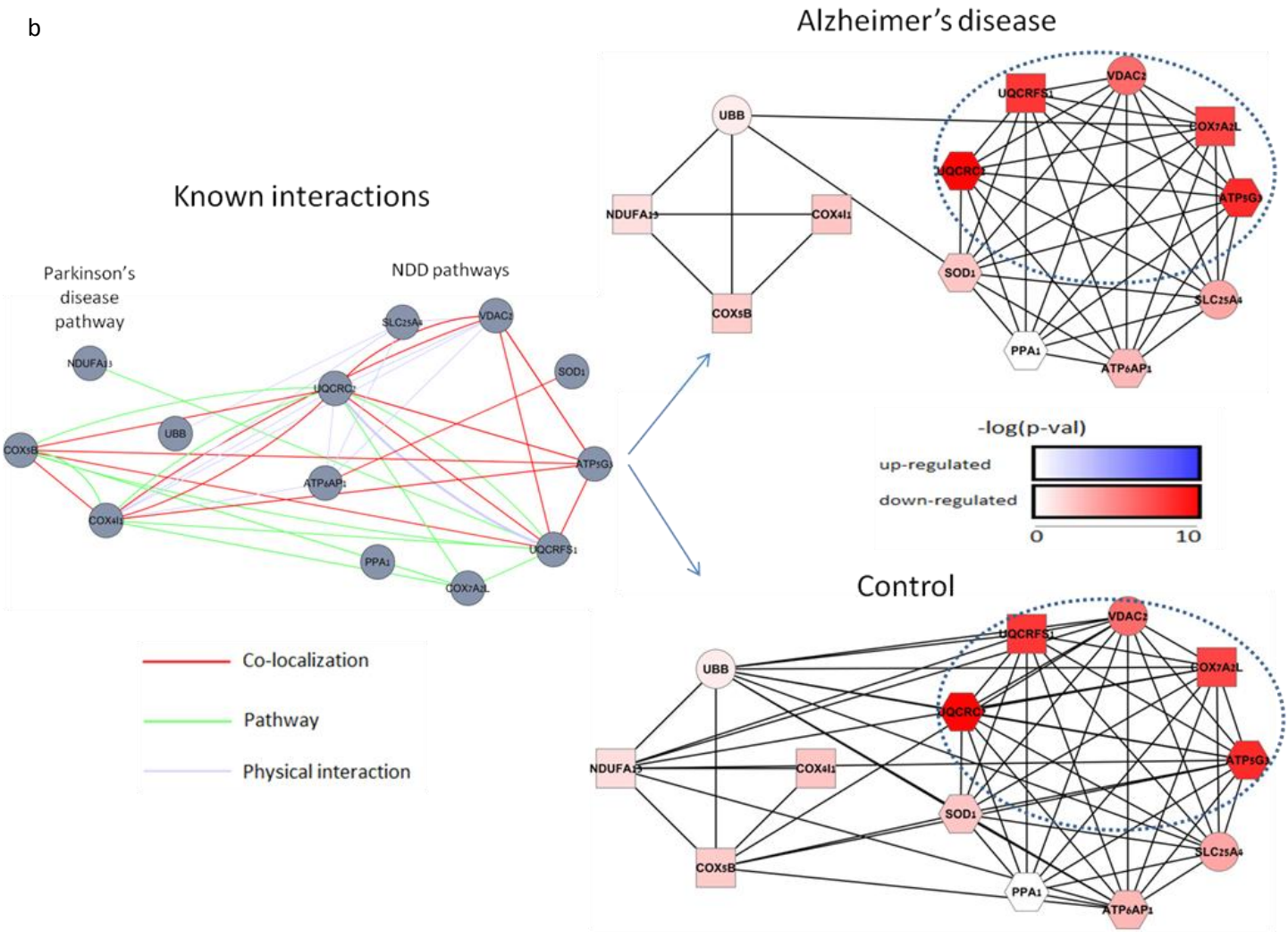


Figure 8 Differential correlation map of modules enriched with KEGG pathways discovered in the Alzheimer's disease (AD) data. a) An overview of the pathways differential correlation. Nodes represent gene modules, the node label reflects a pathway that was enriched in the gene module, and edges correspond to differential correlation (Blue edges mark increased correlation in AD. Red edges mark decreased correlation in AD). Node size is proportional to the size of the module. The enriched pathways are noted on the module. NDD pathways refer to Parkinson's disease (PD), Huntington's disease, Alzheimer's disease and oxidative phosphorylation. CAMs refer to cell adhesion molecules pathway; b) Analysis of the differential correlation between the PD and the NDD modules (the circled sub-graph in a). Left: the set of known interactions according to GENEMANIA. Most known interactions are between the modules. Right: co-expression networks of the same genes for AD patients and controls. Rectangular nodes are genes related to oxidoreductase activity, hexagons indicate genes related to phosphate metabolic process. An edge between two genes indicates correlation > 0.3 in the tested class. The average correlation between the modules was 0.3 in the controls and 0 in the AD class. Node colors indicate the differential expression between case and control, measured as log-p-value (t-test) of the tested gene. The genes encircled in the NDD pathway module are also part of the PD pathway. These genes are also down-correlated in AD, whereas all other genes show only mild differential expression.

Unlike GSCA [3], which searches for differential correlation patterns among known pathways, DICER does not use known pathway information. Therefore by analyzing the meta-modules found by DICER we can detect differential relations between genes of different biological processes. Moreover, we can dissect the genes of a biological process into sub-groups that are differentially correlated. We demonstrate these abilities of DICER for pathways and protein complexes related to AD.

We created the differential correlation map of modules enriched with KEGG pathways (0.05 FDR followed by redundancy filter, see section 5.4). A summary map is shown in **Figure 8a.** For example, a module enriched with cell adhesion molecules (CAMs) was found to be up-correlated with a module enriched with genes related to Parkinson's disease (PD). Two modules that were enriched with pathways that are directly related to NDD (oxidative phosphorylation, PD, AD, and HD) were down-correlated. **Figure 8b** shows these two modules in detail. GENEMANIA analysis [61, 95, 96] shows that known interactions are mainly between the modules (co-expression and predicted interactions were excluded). The DICER analysis suggests that although the modules share similar functionality, their genes can be partitioned into two so that each sub group is highly homogeneous (correlation above 0.8 both in cases and controls), whereas the correlation between sub-groups is lower in AD samples (class specific co-expression networks are shown). In addition, both groups contain genes related to PD (all genes in group 1, and the encircled genes in group 2) and oxidoreductase activity

(rectangular nodes; group 1 contains COX4I2, COX5B and NDUFA13, group 2 contains COX7A2L, SOD1 and UQCRFS1). In contrast, only group 2 contains genes that are related to phosphate metabolic process (hexagon nodes, genes UQCRC2, ATP6AP1, SOD1, ATP5G3, PPA1). This group is consistently correlated with PD genes that are strongly down-correlated (the encircled genes in group 2). Both groups contain cytochrome c oxidase (COX) genes. COX impairment is well established in AD [120, 121]. The first group contains the gene COX7A2L that is down-regulated in AD, whereas the second group contains the genes COX4I2 and COX5B that do not show a significant differential correlation. The differential co-expression between these subparts of COX can be explained by the association of APP with the mitochondria import channels, causing a blockage for incoming of the nuclear-encoded COX subunits 4 and 5B[121]. This can also explain why the genes COX4I2 and COX5B are not necessarily down-regulated at the mRNA level, while COX is impaired. This example demonstrates that disease-specific DC-based analysis can detect differential networking of functional sub-units within pathways without using any prior knowledge, and thus lead to new hypotheses regarding their roles.

Figure 9a shows the differential correlation map of modules enriched with protein complexes (0.05 FDR), in the AD data. Here again we detect differential correlation between different protein complexes, for example, decreased correlation between the spliceosome and ribosome genes. In **Figure 9a** two up-correlated gene modules that were enriched with ribosomal genes are marked. The first group was enriched with cytoplasmatic ribosomal genes (seven genes, p=0.002), whereas the second was enriched with both 60S ribosomal unit genes (seven genes, p=1.68E-7, enrichment factor 21.3) and in genes that belong to the Nop56p-associated pre-rRNA complex (six genes, p=5.09E-6, enrichment factor 12.2). We note that these two complexes highly overlap: only two out of six Nop56 genes are not annotated as members of the 60S complex. In addition, only the first group contains 40S ribosomal unit genes (four genes). **Figure 9b** focuses on these two modules. GENEMANIA analysis of the modules indicates that all ribosomal genes, from both modules, are highly connected in all three types of interactions: co-localization, physical interaction, and pathway (co-expression and predicted interactions were excluded). However, by comparing the co-expression in controls and AD (**Figure 9b**) we observe that the modules are highly correlated only on AD. In addition, 40S complex genes are up-regulated in AD, whereas 60S genes show only mild differential expression. Thus, we observe both increased coordination of the ribosome sub-complexes, and increased activity of 40S, indicating major transcriptomic changes in AD.

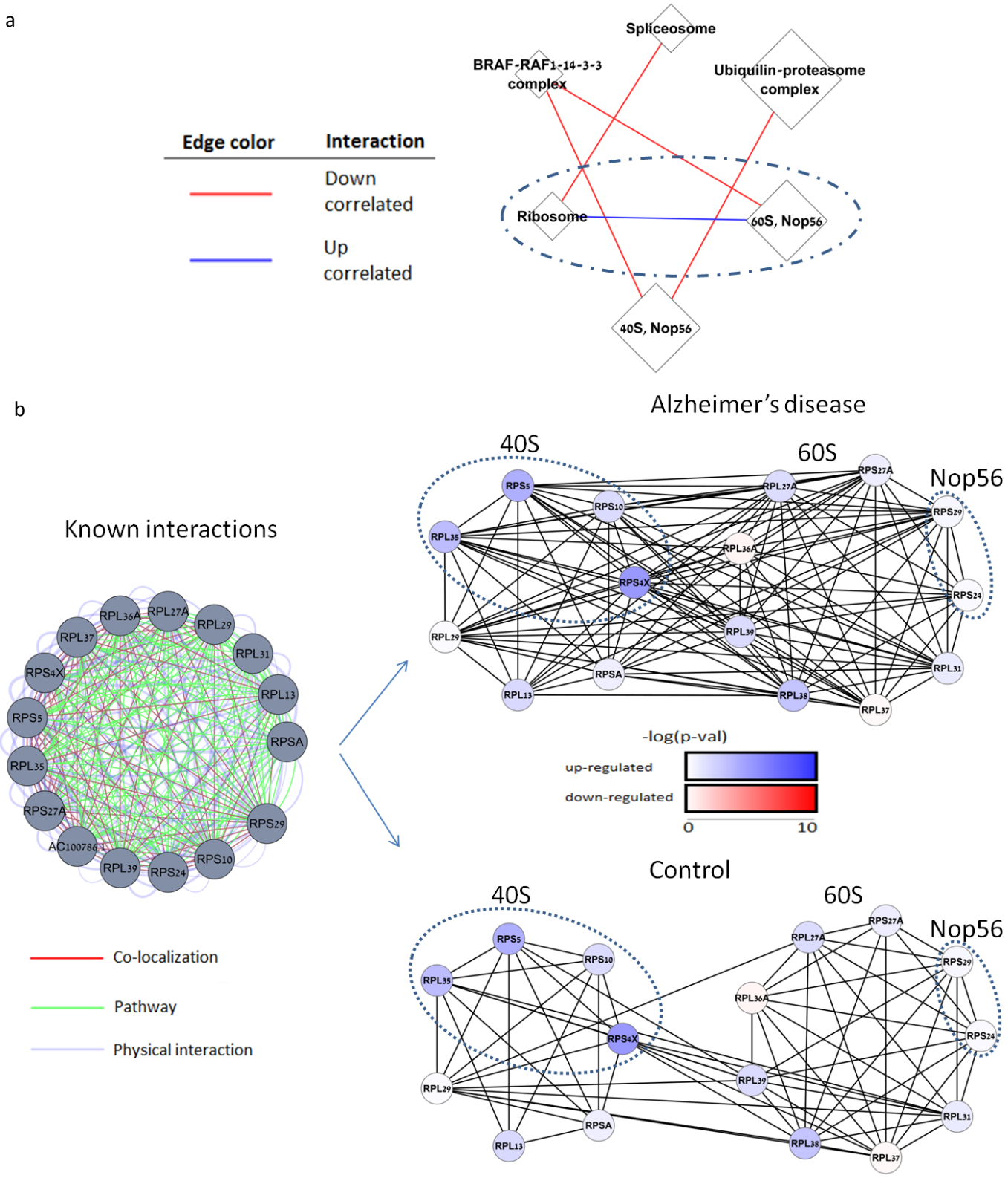


Figure 9 Ribosomal sub-complexes discovered in the Alzheimer's disease (AD) data. a) Overview of the differential correlation map of modules enriched with protein complexes. Node size is proportional to the size of the module. The enriched pathway names are noted

on the module. 40S: 40S cytoplasmatic Ribosome complex, 60S: 60S cytoplasmatic Ribosome complex, Nop56: Nop56p-associated pre-rRNA complex. Blue and red edges mark increased and decreased correlation in AD, respectively. ; b) Analysis of the differential correlation in the Ribosome and 60S, Nop56 meta-module encircled in a. Left: the known interactions between the meta-module genes according to GENEMANIA. Right: co-expression networks of the same genes for AD patients and controls. An edge between two genes indicates correlation > 0.5 in the tested class. The average correlation between modules was 0.4 and 0.75 in the controls and AD class, respectively. Node colors show differential expression between AD and control, measured by the log-p-value (t-test) of the tested gene. Encircled subgroups: proteins belonging to 40S cytoplasmatic Ribosome and Nop56 complex. 40S complex genes are up-regulated in AD, whereas 60S genes show only mild differential expression.

5.4 Technical notes

5.4.1 Finding differentially correlated gene modules using DiffCoEx

We used the R implementation of DiffCoEx with default parameters.

5.4.2 Enrichment analysis of pathways and protein complexes

We performed KEGG [122] and protein complex enrichment analysis of gene sets by calculating hyper-geometric p-values and false discovery rate correction for multiple testing [78]. The background set of the hyper-geometric score was set to be the (filtered) set of genes in the tested data set. Human protein complex annotations were extracted using BioMart [123, 124].

Because many KEGG pathways are highly overlapping, after performing the initial enrichment analysis, we used redundancy filtering of the KEGG pathway enrichment results. For every cluster, we checked every two enriched terms, and if the Jaccard coefficient between the gene sets of these terms in the tested cluster, was above 0.5, we kept the term with the lower p-value.

5.4.3 Enrichment analysis of miRNA families

We used the FAME algorithm [99] to test for enrichment of miRNA families in gene sets. We used 2000 sampling steps for evaluating enrichment p-values and corrected for multiple testing (FDR, 0.05).

5.4.4 Enrichment analysis of known disease and miRNA associations

Given a set of miRNA families detected in a data set of a specific disease we tested if the overlap with the associations in mir2disease is significant. To perform this test we converted the associations in mir2disease from MirBase ids [97] to miRNA families. We then calculated the hyper-geometric p-value for the overlap between the tested miRNAs set and the set of miRNA families associated with the tested disease. The background set for this test was all miRNA families that had at least one gene target in the tested data set.

Supplementary Tables

Data set	DiffCoEx meta-modules		DICER meta-modules	
	Number	average size	Number	average size
AD	3	48.66	50	60.1
NDD	5	80.2	16	37.15
IBD	2	36.5	3	35.66
Lung cancer	2	278.5	20	69.2
SLE	9	232	24	53.16

Supplementary Table 1: DiffCoEx and DICER meta-module statistics.

6

Data Set	Gene set	miRNA	Number of genes	P-value (FDR 0.05)	Enrichment factor
Meta modules					
AD	DCMM2	mir-124/506	28	5.00E-04	2.112
	DCMM5	mir-876-5p	4	5.00E-04	6.976
	DCMM5	mir-499/499-5p	7	5.00E-04	3.602
	DCMM7	mir-377	8	0.001	3.607
	DCMM8	mir-504	4	0.001	6.982
	DCMM9	**mir-125/351**	4	0.01	4.346
	UCMM6	mir-216/216b	4	0.001	5.413
	UCMM9	mir-124/506	10	5.00E-04	3.065
	UCMM9	mir-340/340-5p	7	0.005	2.972
NDD	DCMM1	mir-410	4	0.003	5.18
	DCMM1	mir-183	4	0.003	4.44
	DCMM8	mir-182	5	0.014	2.802
	DCMM8	mir-124/506	6	0.017	2.416
	UCMM5	**mir-30a/30a-5p/30b/30b-5p/30cde/384-5p**	4	0.028	2.975
	UCMM6	**mir-9**	4	0.004	3.747
Lung	DCMM1	mir-320/320abcd	29	5.00E-04	2.19
	DCMM4	mir-149	4	5.00E-04	8.762
	DCMM4	**mir-9**	6	0.005	3.058
	DCMM4	mir-300	5	0.026	2.807
	DCMM5	**mir-141/200a**	9	0.001	3.259
	DCMM6	mir-34a/34b-5p/34c/34c-5p/449/449abc/699	4	5.00E-04	6.909
	DCMM8	**mir-25/32/92/92ab/363/367**	6	0.004	2.814
	UCMM1	**mir-15/16/195/424/497**	4	0.028	2.498
	UCMM1	**mir-203**	4	0.028	2.874
	UCMM1	**mir-200bc/429**	4	0.036	2.456
	UCMM3	mir-590/590-3p	7	0.001	3.552
	UCMM5	**mir-130/301**	4	0.015	3.309
SLE	DCMM2	mir-590/590-3p	5	0.004	3.855
	DCMM2	mir-124/506	6	0.019	2.536
	DCMM2	mir-182	4	0.026	2.664
	DCMM3	mir-182	10	0.001	2.682
	DCMM6	mir-383	4	5.00E-04	7.853
	DCMM6	mir-205	5	0.002	3.942
	DCMM6	mir-216/216a	4	0.006	5.04

	DCMM6	mir-155	4	0.015	3.238
	DCMM9	mir-1/206	6	0.002	3.936
	UCMM2	mir-590/590-3p	4	0.008	3.749
Differential gene clusters					
AD	DCC11	mir-543	6	0.001	3.757
	DCC11	**mir-101**	6	0.011	2.998
	DCC12	mir-200bc/429	5	0.024	2.52
	DCC12	mir-590/590-3p	4	0.027	2.566
	DCC2	**mir-9**	4	0.028	2.922
	DCC5	**mir-9**	4	0.012	3.063
	DCC6	mir-140/140-5p/876-3p	4	0.003	4.759
	DCC9	**mir-29abc**	4	0.03	3.186
	UCC0	mir-203	4	5.00E-04	11.549
	UCC1	mir-410	4	0.004	4.328
	UCC1	mir-186	4	0.005	3.571
	UCC10	**mir-26ab/1297**	4	0.014	3.689
	UCC10	mir-124/506	6	0.035	2.509
	UCC3	mir-148/152	4	0.004	4.246
	UCC4	**mir-29abc**	4	0.037	2.905
	UCC4	**mir-17-5p/20/93.mr/106/519.d**	4	0.046	2.517
	UCC6	mir-25/32/92/92ab/363/367	4	0.013	3.588
	UCC8	mir-124/506	4	0.042	2.52
SLE	DCM0	mir-9	4	0.009	3.883
Modules					
AD	up_2	mir-199/199-5p	4	0.003	5.05
	up_4	mir-30a/30a-5p/30b/30b-5p/30cde/384-5p	7	5.00E-04	3.299
	up_4	mir-183	4	0.003	5.038
	up_4	mir-590/590-3p	6	0.005	3.123
	up_4	mir-133	5	0.029	2.897
	up_10	mir-300	5	0.005	3.581
	up_10	mir-181	4	0.034	2.381
	up_10	mir-27ab	4	0.046	2.233
	up_10	mir-590/590-3p	4	0.048	2.488
	up_13	mir-216/216b	4	5.00E-04	5.975
	up_14	mir-128	4	0.029	2.937
	up_14	**mir-17-5p/20/93.mr/106/519.d**	4	0.04	2.537
	up_16	mir-590/590-3p	7	0.002	3.442
	up_16	**mir-15/16/195/424/497**	6	0.026	2.271
	up_16	mir-320/320abcd	4	0.027	2.827
	up_17	mir-205	4	7.50E-04	6.441

	up_17	mir-204/211	5	0.002	4.504
	up_17	mir-125/351	4	0.006	4.202
	up_17	mir-153	4	0.013	3.625
	up_18	mir-124/506	10	5.00E-04	3.644
	up_18	mir-340/340-5p	6	0.004	3.25
	down_0	mir-200bc/429	20	5.00E-04	2.251
	down_1	mir-590/590-3p	17	5.00E-04	3.032
	down_4	mir-300	4	0.022	3.243
	down_4	mir-590/590-3p	4	0.035	2.791
	down_5	mir-124/506	28	5.00E-04	2.492
	down_7	mir-96/1271	5	0.002	3.695
	down_10	**mir-34a/34b-5p/34c/34c-5p/449/449abc/699**	4	0.014	3.1
	down_10	mir-448	4	0.016	3.156
	down_11	mir-124/506	15	0.001	2.254
	down_11	let-7/98	8	0.004	2.973
	down_13	mir-19	4	0.017	3.173
	down_13	mir-200bc/429	4	0.025	3.08
	down_14	mir-377	6	5.00E-04	4.314
	down_14	**mir-26ab/1297**	7	0.003	2.735
	down_15	**mir-15/16/195/424/497**	6	0.014	2.654
	down_15	mir-200bc/429	5	0.041	2.285
	down_16	**mir-17-5p/20/93.mr/106/519.d**	6	0.008	2.61
	down_17	mir-135	4	0.011	3.622
	down_19	**mir-125/351**	4	0.008	5.117
NDD	up_0	**mir-103/107**	4	0.001	7.761
	up_7	**mir-9**	7	0.015	2.601
	up_7	**mir-26ab/1297**	4	0.019	2.941
	up_11	**mir-30a/30a-5p/30b/30b-5p/30cde/384-5p**	4	0.028	2.913
	down_4	**mir-17-5p/20/93.mr/106/519.d**	4	0.02	3.012
SLE	up_5	mir-590/590-3p	4	0.001	6.266
	up_9	mir-124/506	4	0.035	2.385
	down_1	mir-15/16/195/424/497	6	5.00E-04	4.517
	down_1	mir-182	4	0.015	3.253
	down_4	mir-590/590-3p	5	5.00E-04	7.188
	down_4	mir-124/506	4	0.014	3.357
	down_6	mir-182	6	0.002	3.612
	down_6	mir-96/1271	5	0.016	3.104
	down_12	mir-383	4	5.00E-04	11.699
	down_12	mir-26ab/1297	4	0.029	2.742
	down_18	mir-1/206	4	0.012	3.746
	down_18	mir-30a/30a-5p/30b/30b-5p/30cde/384-5p	4	0.029	2.888

Supplementary Table 2: miRNA family enrichment analysis results. For each module, meta-module and cluster detected by DICER the enrichment was tested using the FAME algorithm. Names bolded indicate miRNA that have known relation with the disease according to the mir2disease database. Down and up-correlated meta-modules are indicated as DCMM and UCMM, respectively. Modules are indicated by up and down followed by their number. Differential gene clusters are indicated by DCC and UCC.

References

1. Glazko, G.V. and F. Emmert-Streib, *Unite and conquer: univariate and multivariate approaches for finding differentially expressed gene sets.* Bioinformatics, 2009. **25**(18): p. 2348-54.
2. Lee, E., et al., *Inferring pathway activity toward precise disease classification.* PLoS Comput Biol, 2008. **4**(11): p. e1000217.
3. Subramanian, A., et al., *Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles.* Proc Natl Acad Sci U S A, 2005. **102**(43): p. 15545-50.
4. Oron, A.P., Z. Jiang, and R. Gentleman, *Gene set enrichment analysis using linear models and diagnostics.* Bioinformatics, 2008. **24**(22): p. 2586-2591.
5. Kim, S.Y. and D.J. Volsky, *PAGE: parametric analysis of gene set enrichment.* BMC Bioinformatics, 2005. **6**: p. 144.
6. Ulitsky, I., et al., *DEGAS: de novo discovery of dysregulated pathways in human diseases.* PLoS One. **5**(10): p. e13367.
7. Ideker, T., et al., *Discovering regulatory and signalling circuits in molecular interaction networks.* Bioinformatics, 2002. **18 Suppl 1**: p. S233-40.
8. Rajagopalan, D. and P. Agarwal, *Inferring pathways from gene lists using a literature-derived network of biological relationships.* Bioinformatics, 2005. **21**(6): p. 788-793.
9. Cabusora, L., et al., *Differential network expression during drug and stress response.* Bioinformatics, 2005. **21**(12): p. 2898-2905.
10. Liu, M., et al., *Network-based analysis of affected biological processes in type 2 diabetes models.* Plos Genetics, 2007. **3**(6): p. 958-972.
11. Rapaport, F., et al., *Classification of microarray data using gene networks.* BMC Bioinformatics, 2007. **8**.
12. Ideker, T., et al., *Network-based classification of breast cancer metastasis.* Molecular Systems Biology, 2007. **3**.
13. Spellman, P.T., et al., *Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization.* Molecular Biology of the Cell, 1998. **9**(12): p. 3273-3297.
14. Tavazoie, S., et al., *Systematic determination of genetic network architecture.* Nature Genetics, 1999. **22**(3): p. 281-285.
15. Wyrick, J.J. and R.A. Young, *Deciphering gene expression regulatory networks.* Current Opinion in Genetics & Development, 2002. **12**(2): p. 130-136.
16. Segal, E., R. Yelensky, and D. Koller, *Genome-wide discovery of transcriptional modules from DNA sequence and gene expression.* Bioinformatics, 2003. **19**: p. i273-i282.
17. Shamir, R., et al., *Allegro: Analyzing expression and sequence in concert to discover regulatory programs.* Nucleic Acids Research, 2009. **37**(5): p. 1566-1579.
18. Lai, Y., et al., *A statistical method for identifying differential gene-gene co-expression patterns.* Bioinformatics, 2004. **20**(17): p. 3146-55.
19. Carter, S.L., et al., *Gene co-expression network topology provides a framework for molecular characterization of cellular state.* Bioinformatics, 2004. **20**(14): p. 2242-50.
20. Kostka, D. and R. Spang, *Finding disease specific alterations in the co-expression of genes.* Bioinformatics, 2004. **20 Suppl 1**: p. i194-9.
21. de la Fuente, A., *From 'differential expression' to 'differential networking' - identification of dysfunctional regulatory networks in diseases.* Trends in Genetics, 2010. **26**(7): p. 326-333.

22. Choi, J.K., et al., *Differential coexpression analysis using microarray data and its application to human cancer.* Bioinformatics, 2005. **21**(24): p. 4348-4355.
23. Reverter, A., et al., *Simultaneous identification of differential gene expression and connectivity in inflammation, adipogenesis and cancer.* Bioinformatics, 2006. **22**(19): p. 2396-404.
24. Elo, L.L., et al., *Systematic construction of gene coexpression networks with applications to human T helper cell differentiation process.* Bioinformatics, 2007. **23**(16): p. 2096-103.
25. Fuller, T.F., et al., *Weighted gene coexpression network analysis strategies applied to mouse weight.* Mamm Genome, 2007. **18**(6-7): p. 463-72.
26. Hudson, N.J., A. Reverter, and B.P. Dalrymple, *A differential wiring analysis of expression data correctly identifies the gene containing the causal mutation.* PLoS Comput Biol, 2009. **5**(5): p. e1000382.
27. Mentzen, W.I., M. Floris, and A. de la Fuente, *Dissecting the dynamics of dysregulation of cellular processes in mouse mammary gland tumor.* BMC Genomics, 2009. **10**: p. 601.
28. Southworth, L.K., A.B. Owen, and S.K. Kim, *Aging Mice Show a Decreasing Correlation of Gene Expression within Genetic Modules.* Plos Genetics, 2009. **5**(12).
29. Alberts, B., *Molecular biology of the cell*, 2008, Garland Science,: New York.
30. Lehninger, A.L., D.L. Nelson, and M.M. Cox, *Lehninger principles of biochemistry*. 3rd ed2000, New York: Worth Publishers.
31. He, L. and G.J. Hannon, *MicroRNAs: Small RNAs with a big role in gene regulation (vol 5, pg 522 2004).* Nature Reviews Genetics, 2004. **5**(8): p. 522-+.
32. Chen, K. and N. Rajewsky, *The evolution of gene regulation by transcription factors and microRNAs.* Nature Reviews Genetics, 2007. **8**(2): p. 93-103.
33. Venter, J.C., et al., *The sequence of the human genome.* Science, 2001. **291**(5507): p. 1304.
34. McBride, H.M., M. Neuspiel, and S. Wasiak, *Mitochondria: More than just a powerhouse.* Current Biology, 2006. **16**(14): p. R551-R560.
35. Zeviani, M. and S. Di Donato, *Mitochondrial disorders.* Brain, 2004. **127**: p. 2153-2172.
36. Proudfoot, N.J., A. Furger, and M.J. Dye, *Integrating mRNA processing with transcription.* Cell, 2002. **108**(4): p. 501-12.
37. Friedman, R.C., et al., *Most mammalian mRNAs are conserved targets of microRNAs.* Genome Research, 2009. **19**(1): p. 92-105.
38. Ambros, V., *The functions of animal microRNAs.* Nature, 2004. **431**(7006): p. 350-355.
39. Bartel, D.P., *MicroRNAs: Genomics, biogenesis, mechanism, and function.* Cell, 2004. **116**(2): p. 281-297.
40. Schuster, S.C., *Next-generation sequencing transforms today's biology.* Nat Methods, 2008. **5**(1): p. 16-8.
41. Emerit, J., A. Edeas, and F. Bricaire, *Neurodegenerative diseases and oxidative stress.* Biomedicine & Pharmacotherapy, 2004. **58**(1): p. 39-46.
42. Drachman, D.B., et al., *Cyclooxygenase 2 inhibition protects motor neurons and prolongs survival in a transgenic mouse model of ALS.* Annals of Neurology, 2002. **52**(6): p. 771-778.
43. Friedlander, R.M., *Mechanisms of disease: Apoptosis and caspases in neurodegenerative diseases.* New England Journal of Medicine, 2003. **348**(14): p. 1365-1375.
44. Haass, C., *Initiation and propagation of neurodegeneration.* Nat Med. **16**(11): p. 1201-4.
45. Zlokovic, B.V., *Neurodegeneration and the neurovascular unit.* Nature Medicine, 2010. **16**(12): p. 1370-1371.

46. Carmeliet, P., S. Zacchigna, and D. Lambrechts, *Neurovascular signalling defects in neurodegeneration.* Nature Reviews Neuroscience, 2008. **9**(3): p. 169-181.
47. Zlokovic, B.V., et al., *Pericytes Control Key Neurovascular Functions and Neuronal Phenotype in the Adult Brain and during Brain Aging.* Neuron, 2010. **68**(3): p. 409-427.
48. Kipnis, J. and M. Schwartz, *Controlled autoimmunity in CNS maintenance and repair - Naturally occurring CD4+CD25+ regulatory T-cells at the crossroads of health and disease.* Neuromolecular Medicine, 2005. **7**(3): p. 197-206.
49. O'Connor, K.C., A. Bar-Or, and D.A. Hafler, *The neuroimmunology of multiple sclerosis: Possible roles of T and B lymphocytes in immunopathogenesis.* Journal of Clinical Immunology, 2001. **21**(2): p. 81-92.
50. Becker, K.J., et al., *Immunologic tolerance to myelin basic protein decreases stroke size after transient focal cerebral ischemia.* Proceedings of the National Academy of Sciences of the United States of America, 1997. **94**(20): p. 10873-10878.
51. Dantzer, R., et al., *From inflammation to sickness and depression: when the immune system subjugates the brain.* Nature Reviews Neuroscience, 2008. **9**(1): p. 46-57.
52. Block, M.L. and J.S. Hong, *Microglia and inflammation-mediated neurodegeneration: Multiple triggers with a common mechanism.* Progress in Neurobiology, 2005. **76**(2): p. 77-98.
53. McGeer, E.G., A. Klegeris, and P.L. McGeer, *Inflammation, the complement system and the diseases of aging.* Neurobiology of Aging, 2005. **26**: p. S94-S97.
54. Ziv, Y., et al., *Immune cells contribute to the maintenance of neurogenesis and spatial learning abilities in adulthood.* Nature Neuroscience, 2006. **9**(2): p. 268-275.
55. Schwartz, M., R. Shechter, and Y. Ziv, *New GABAergic Interneurons supported by myelin-specific T cells are formed in intact adult spinal cord.* Stem Cells, 2007. **25**(9): p. 2277-2282.
56. Schwartz, M. and E. Hauben, *T cell-based therapeutic vaccination for spinal cord injury.* Spinal Cord Trauma: Regeneration, Neural Repair and Functional Recovery, 2002. **137**: p. 401-406.
57. Kipnis, J., et al., *Adaptive immunity affects learning behavior in mice.* Brain Behavior and Immunity, 2008. **22**(6): p. 861-869.
58. Popovich, P.G. and E.E. Longbrake, *Perspectives - Opinion - Can the immune system be harnessed to repair the CNS?* Nature Reviews Neuroscience, 2008. **9**(6): p. 481-493.
59. Blennow, K., M.J. de Leon, and H. Zetterberg, *Alzheimer's disease.* Lancet, 2006. **368**(9533): p. 387-403.
60. Usadel, B., et al., *Co-expression tools for plant biology: opportunities for hypothesis generation and caveats.* Plant Cell Environ, 2009. **32**(12): p. 1633-51.
61. Mostafavi, S., et al., *GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function.* Genome Biol, 2008. **9 Suppl 1**: p. S4.
62. Manfield, I.W., et al., *Arabidopsis Co-expression Tool (ACT): web server tools for microarray-based gene expression analysis.* Nucleic Acids Res, 2006. **34**(Web Server issue): p. W504-9.
63. Mutwil, M., et al., *GeneCAT--novel webtools that combine BLAST and co-expression analyses.* Nucleic Acids Res, 2008. **36**(Web Server issue): p. W320-6.
64. Sharan, R., A. Maron-Katz, and R. Shamir, *CLICK and EXPANDER: a system for clustering and visualizing gene expression data.* Bioinformatics, 2003. **19**(14): p. 1787-1799.
65. Sharan, R. and R. Shamir, *CLICK: a clustering algorithm with applications to gene expression analysis.* Proc Int Conf Intell Syst Mol Biol, 2000. **8**: p. 307-16.
66. Bansal, N., A. Blum, and S. Chawla, *Correlation clustering.* Machine Learning, 2004. **56**(1-3): p. 89-113.

67. Eisen, M.B., et al., *Cluster analysis and display of genome-wide expression patterns.* Proceedings of the National Academy of Sciences of the United States of America, 1998. **95**(25): p. 14863-14868.
68. Sokal, R.R. and C.D. Michener, *A statistical method for evaluating systematic relationships.* Univ. Kansas Sci., 1958(Bull., 38:): p. 1409-1438.
69. Saitou, N. and M. Nei, *The neighbor-joining method: a new method for reconstructing phylogenetic trees.* Mol Biol Evol, 1987. **4**(4): p. 406-25.
70. Van Dongen, S., *Graph Clustering by Flow Simulation.* In PhD Thesis. University of Utrecht., 2000.
71. Belacel, N., M. Cuperlovic-Culf, and R. Ouellette, *Fuzzy J-Means and VNS methods for clustering genes from microarray data.* Bioinformatics, 2004. **20**(11): p. 1690-1701.
72. Dembele, D. and P. Kastner, *Fuzzy C-means method for clustering microarray data.* Bioinformatics, 2003. **19**(8): p. 973-980.
73. Du, P., et al., *From disease ontology to disease-ontology lite: statistical methods to adapt a general-purpose ontology for the test of gene-ontology associations.* Bioinformatics, 2009. **25**(12): p. i63-8.
74. Fu, L.M. and E. Medico, *FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data.* BMC Bioinformatics, 2007. **8**.
75. Gasch, A.P. and M.B. Eisen, *Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering.* Genome Biology, 2002. **3**(11).
76. Wang, P.H., *Pattern-Recognition with Fuzzy Objective Function Algorithms - Bezdek,Jc.* Siam Review, 1983. **25**(3): p. 442-442.
77. Cui, X.Q. and G.A. Churchill, *Statistical tests for differential expression in cDNA microarray experiments.* Genome Biology, 2003. **4**(4).
78. Benjamini, Y. and Y. Hochberg, *Controlling the False Discovery Rate - a Practical and Powerful Approach to Multiple Testing.* Journal of the Royal Statistical Society Series B-Methodological, 1995. **57**(1): p. 289-300.
79. Lee, E., et al., *Inferring Pathway Activity toward Precise Disease Classification.* Plos Computational Biology, 2008. **4**(11).
80. Su, J.J., B.J. Yoon, and E.R. Dougherty, *Accurate and Reliable Cancer Classification Based on Probabilistic Inference of Pathway Activity.* PLoS One, 2009. **4**(12).
81. Tian, L., et al., *Discovering statistically significant pathways in expression profiling studies.* Proceedings of the National Academy of Sciences of the United States of America, 2005. **102**(38): p. 13544-13549.
82. Guo, Z., et al., *Towards precise classification of cancers based on robust gene functional expression profiles.* BMC Bioinformatics, 2005. **6**.
83. Cho, S.B., J. Kim, and J.H. Kim, *Identifying set-wise differential co-expression in gene expression microarray data.* BMC Bioinformatics, 2009. **10**.
84. Ihmels, J., et al., *Comparative gene expression analysis by a differential clustering approach: Application to the Candida albicans transcription program.* Plos Genetics, 2005. **1**(3): p. 380-393.
85. Watson, M., *CoXpress: differential co-expression in gene expression data.* BMC Bioinformatics, 2006. **7**.
86. Tesson, B.M., R. Breitling, and R.C. Jansen, *DiffCoEx: a simple and sensitive method to find differentially coexpressed gene modules.* BMC Bioinformatics, 2010. **11**.
87. Choi, Y. and C. Kendziorski, *Statistical methods for gene set co-expression analysis.* Bioinformatics, 2009. **25**(21): p. 2780-2786.
88. Ravasz, E., et al., *Hierarchical organization of modularity in metabolic networks.* Science, 2002. **297**(5586): p. 1551-1555.
89. Langfelder, P., B. Zhang, and S. Horvath, *Defining clusters from a hierarchical cluster tree: the Dynamic Tree Cut package for R.* Bioinformatics, 2008. **24**(5): p. 719-720.

90. Ito, T., et al., *A comprehensive two-hybrid analysis to explore the yeast protein interactome.* Proc Natl Acad Sci U S A, 2001. **98**(8): p. 4569-74.
91. Stelzl, U., et al., *A human protein-protein interaction network: a resource for annotating the proteome.* Cell, 2005. **122**(6): p. 957-68.
92. Kerrien, S., et al., *IntAct--open source resource for molecular interaction data.* Nucleic Acids Res, 2007. **35**(Database issue): p. D561-5.
93. Jensen, L.J., et al., *STRING 8-a global view on proteins and their functional interactions in 630 organisms.* Nucleic Acids Research, 2009. **37**: p. D412-D416.
94. Sharan, R., I. Ulitsky, and R. Shamir, *Network-based prediction of protein function.* Molecular Systems Biology, 2007. **3**.
95. Montojo, J., et al., *GeneMANIA Cytoscape plugin: fast gene function predictions on the desktop.* Bioinformatics, 2010. **26**(22): p. 2927-8.
96. Warde-Farley, D., et al., *The GeneMANIA prediction server: biological network integration for gene prioritization and predicting gene function.* Nucleic Acids Res, 2010. **38**(Web Server issue): p. W214-20.
97. Griffiths-Jones, S., et al., *miRBase: microRNA sequences, targets and gene nomenclature.* Nucleic Acids Research, 2006. **34**: p. D140-D144.
98. Kozomara, A. and S. Griffiths-Jones, *miRBase: integrating microRNA annotation and deep-sequencing data.* Nucleic Acids Research, 2011. **39**: p. D152-D157.
99. Ulitsky, I., L.C. Laurent, and R. Shamir, *Towards computational prediction of microRNA function and activity.* Nucleic Acids Res, 2010. **38**(15): p. e160.
100. Sipser, M., *Introduction to the theory of computation*. 2nd ed2006, Boston: Thomson Course Technology. xix, 431 p.
101. Arora, S. and B. Barak, *Computational complexity : a modern approach*2009, Cambridge ; New York: Cambridge University Press. xxiv, 579 p.
102. Arora, S. and C. Lund, *Hardness of Approximations.* PWS Publishing Company, 1996(Approximation Algorithms for NP-Hard Problems (D. Hochbaum, editor)): p. 399-446.
103. Khanna, S., et al., *On syntactic versus computational views of approximability.* Siam Journal on Computing, 1998. **28**(1): p. 164-191.
104. Ausiello, G., *Complexity and approximation : combinatorial optimization problems and their approximability properties*. 2nd corrected printing. ed2003, Berlin ; New York: Springer. xix, 524 p.
105. Arora, S. and S. Safra, *Probabilistic checking of proofs: A new characterization of NP.* Journal of the Acm, 1998. **45**(1): p. 70-122.
106. Arora, S., et al., *Proof verification and the hardness of approximation problems.* Journal of the Acm, 1998. **45**(3): p. 501-555.
107. Hastad, J., *Clique is hard to approximate within n(1-epsilon).* Acta Mathematica, 1999. **182**(1): p. 105-142.
108. Barrett, T., et al., *NCBI GEO: mining tens of millions of expression profiles - database and tools update.* Nucleic Acids Research, 2007. **35**: p. D760-D765.
109. RaywardSmith, V.J., *Mathematical classification and clustering - Mirkin,B.* Journal of the Operational Research Society, 1997. **48**(8): p. 852-853.
110. Jiang, Q., et al., *miR2Disease: a manually curated database for microRNA deregulation in human disease.* Nucleic Acids Res, 2009. **37**(Database issue): p. D98-104.
111. de Strooper, B. and P. Lau, *Dysregulated microRNAs in neurodegenerative disorders.* Seminars in Cell & Developmental Biology, 2010. **21**(7): p. 768-773.
112. Maes, O.C., et al., *MicroRNA: Implications for Alzheimer Disease and other Human CNS Disorders.* Curr Genomics, 2009. **10**(3): p. 154-68.

113. Hebert, S.S., et al., *MicroRNA regulation of Alzheimer's Amyloid precursor protein expression.* Neurobiol Dis, 2009. **33**(3): p. 422-8.
114. Lukiw, W.J., *Micro-RNA speciation in fetal, adult and Alzheimer's disease hippocampus.* Neuroreport, 2007. **18**(3): p. 297-300.
115. Makeyev, E.V., et al., *The MicroRNA miR-124 promotes neuronal differentiation by triggering brain-specific alternative pre-mRNA splicing.* Mol Cell, 2007. **27**(3): p. 435-48.
116. Hebert, S.S., et al., *Loss of microRNA cluster miR-29a/b-1 in sporadic Alzheimer's disease correlates with increased BACE1/beta-secretase expression.* Proc Natl Acad Sci U S A, 2008. **105**(17): p. 6415-20.
117. Landgraf, P., et al., *A mammalian microRNA expression atlas based on small RNA library sequencing.* Cell, 2007. **129**(7): p. 1401-14.
118. Zhao, C., et al., *Computational prediction of MicroRNAs targeting GABA receptors and experimental verification of miR-181, miR-216 and miR-203 targets in GABA-A receptor.* BMC Res Notes, 2012. **5**(1): p. 91.
119. Owens, D.F. and A.R. Kriegstein, *Is there more to GABA than synaptic inhibition?* Nature Reviews Neuroscience, 2002. **3**(9): p. 715-727.
120. Pickrell, A.M., H. Fukui, and C.T. Moraes, *The role of cytochrome c oxidase deficiency in ROS and amyloid plaque formation.* Journal of Bioenergetics and Biomembranes, 2009. **41**(5): p. 453-456.
121. Devi, L., et al., *Accumulation of amyloid precursor protein in the mitochondrial import channels of human Alzheimer's disease brain is associated with mitochondrial dysfunction.* Journal of Neuroscience, 2006. **26**(35): p. 9057-9068.
122. Kanehisa, M. and S. Goto, *KEGG: Kyoto Encyclopedia of Genes and Genomes.* Nucleic Acids Research, 2000. **28**(1): p. 27-30.
123. Smedley, D., et al., *BioMart - biological queries made easy.* Bmc Genomics, 2009. **10**.
124. Haider, S., et al., *BioMart Central Portal-unified access to biological data.* Nucleic Acids Research, 2009. **37**: p. W23-W27.