NTNU

Norwegian University of
Science and Technology

Identification of Novel Genes associated with Rheumatoid Arthritis using Differential Gene Co-Expression Analysis

Marta Riise Moksnes

Chemical Engineering and Biotechnology
Submission date: June 2018
Supervisor: Eivind Almaas, IBT
Co-supervisor: André Voigt, IBT

Norwegian University of Science and Technology
Department of Biotechnology and Food Science

Summary

Large RNA-Seq and DNA microarray data sets have enabled the study of relationships between genes through co-expression analyses. Changes in gene co-expression patterns are often related to changes in biological function, and differential co-expression network analyses have become an important step in the comparison of co-expression profiles of different conditions. The CSD method for differential co-expression analysis is used to identify conserved, differentiated and specific correlation patterns between gene pairs over multiple conditions, analyzing pair-wise correlations in gene expression and the variability of these correlations within each data set. The variability in correlation is found from the analysis of independent sub-samples from the full data set. The minimum sub-sample size and required number of sub-samples in total gives a lower bound for the data set size. An alternative sub-sampling approach allowing dependence between the sub-samples was investigated in this study. The aim was to find out how the estimated variability was affected by sub-sample dependence compared to a low number of sub-samples, and to investigate whether or not sub-sample overlap could be justified as a means to allow for smaller data sets in a CSD analysis. The results of the study were clear: Sub-sample dependence had a much smaller negative impact on the estimated variability than having too few sub-samples. Fewer than 150 sub-samples should therefore be avoided, and for data sets smaller than 60 data points per gene, sub-sample dependence was suggested as a valid alternative to the original sub-sampling approach.

Rheumatoid Arthritis (RA) is an autoimmune disease affecting about 1% of the world-wide population. The disease involves a chronic inflammation of the joints, followed by progressive articular remodelling and damage, as well as many common comorbidities. The CSD method was applied on gene expression data from the synovial fluid of RA patients and healthy controls, with the aim of identifying novel genes that could be central in RA development. The alternative sub-sampling algorithm was implemented on the small control data set. The CSD analysis resulted in a differential co-expression network clearly enriched in genes with functions related to the disease. Eleven network hubs were identified: Three genes, *PDCD1*, *CTLA4* and *PRDM1*, had previously been associated with RA, while eight genes, *ZNF205-AS1*, *GPR18*, *LINC00426*, *SEPT1*, *ASAP2*, *ENPP1*, *IL2RG* and *GBP5* were new in this context. Like a high number of genes in the network, most of them were related to immune function or growth and tissue/organ development, and due to their high network connectivity, all of them were considered to be possible candidates for further research of RA. The three most highly connected genes, *PDCD1*, *ZNF205-AS1* and *GPR18*, contributed to a highly disassortative sub-network and were hypothesized to have a coordinating role in RA. *PDCD1* also was a strong connector between parts of the network with different correlation relationships. This is consistent with previous reports of RA association for this gene. The eight other genes were found in a sub-network dominated by differentiated co-expression and particularly enriched with genes related to cell differentiation and tissue and organ development and morphology. These were hypothesized to be involved in later stages of rheumatoid arthritis.

Samandrag

Store datasett frå RNA-Seq- og DNA mikromatrisestudier har gjort det muleg å studere samanhengane mellom gen via samuttrykksanalysar. Endringar i samuttrykksmønster mellom gen er ofte relaterte til endringar i biologisk funksjon, og analysar av differensielle samuttrykksnettverk har blitt viktige når ein skal samanlikne samuttrykksprofilar mellom ulike biologiske vilkår. CSD-metoden for analyse av differensielt samuttrykk vert brukt til å identifisere konserverte, differensierte og spesifikke korrelasjonsmønster mellom ulike biologiske vilkår, ved å analysere parvise korrelasjonar i genuttrykk, samt intern variasjon i desse korrelasjonane i kvart datasett. Variasjonen i korrelasjon vert rekna ut frå korrelasjonar i uavhengige utval av heile datasettet. Det naudsynte talet på utval, kombinert med den minste storleiken dei kan ha, resulterer i ei nedre grense for storleiken på datasetta. Ein alternativ framgongsmåte for å finne utval, der avhengigheit mellom utvala er tillate, vart utforska i denne studien. Målet var å finne ut korleis den estimerte variasjonen i korrelasjon vart påverka av avhengigheit mellom utvala samanlikna med det å ha for få utval. Motivasjonen for dette var å undersøke om overlapp mellom utvala kunne forsvarast slik at mindre datasett kunne bli brukt i ei CSD-analyse. Resultata av studien var tydelege: Avhengigheit mellom utvala hadde mykje mindre negativ påverknad på den estimerte variasjonen i korrelasjon enn det å bruke for få utval hadde. Færre enn 150 utval vart difor fråråda, og for datasett mindre enn 60 datapunkt per gen vart avhengigheit mellom utvala føreslått som eit gyldig alternativ til den originale tilnærminga for å finne utval.

Leddgikt er ein autoimmun sjukdom som rammar omlag 1% av befolkninga i verda. Sjukdomen fører med seg kronisk leddbetennelse, etterfulgt av gradvis omforming og skade på ledda, samt ei rekke vanlege følgesjukdomar. CSD-metoden vart nytta på genuttrykksdata frå leddvæske hos leddgiktpasientar og friske kontrollpersonar. Den alternative metoden for å finne utval til utrekning av variasjon i korrelasjon vart brukt på eit lite kontrolldatasett. Målet med analysa var å identifisere nye gen som kunne vere sentrale i utviklinga av leddgikt. CSD-analysa resulterte i eit differensielt samuttrykksnettverk der gen med funksjonar relaterte til sjukdomen var tydeleg overrepresenterte. Elleve nettverks-nav vart identifiserte: Tre gen, *PDCD1*, *CTLA4* og *PRDM1*, hadde tidlegare blitt assosiert med leddgikt, medan åtte gen, *ZNF205-AS1*, *GPR18*, *LINC00426*, *SEPT1*, *ASAP2*, *ENPP1*, *IL2RG* og *GBP5*, var nye i denne samanhengen. Dei fleste var relaterte til immunforsvaret eller til vekst, vevs- og organutvikling, og på grunn av knutepunktposisjonane dei hadde i nettverket vart alle elleve vurdert som mulege kandidatar for vidare forskning på leddgikt. Dei tre gena som var kopla til flest andre gen i nettverket, *PDCD1*, *ZNF205-AS1* og *GPR18*, bidrog sterkt til disassortativitet i ein del av nettverket, og ei hypotese var at desse hadde ei koordinerande rolle i leddgikt. *PDCD1* knytta også saman to delar av nettverket som var dominerte av ulike typar korrelasjonssamanhengar. Dette gir godt samsvar med tidlegare funn av leddgiktassosiasjon for dette genet. Dei resterande åtte gena var plassert i ein del av nettverket som var dominert av differensierte samuttrykksrelasjonar, og der gen relatert til vekst, samt vevs- og organutvikling og morfologi var særleg overrepresenterte. Ei muleg hypotese var difor at desse kunne vere involverte i seinare stadier av leddgikt.

Preface

This thesis is submitted to the Department of Biotechnology and Food Science at the Norwegian University for Science and Technology (NTNU), and concludes my M.Sc degree in Chemical Engineering and Biotechnology where I specialized in Biotechnology/Systems Biology.

I would like to express my gratitude to my supervisor, professor Eivind Almaas, for his valuable help in the form of shared insights, feed-back, comments, and not least his optimism and enthusiasm. With his support I got the courage to embrace new and challenging tasks in topics that were previously unfamiliar to me, without which I would never have learned so much and had so much fun. My co-supervisor, Ph.D. candidate André Voigt, also deserves thanks for giving me helpful advice on the way, on everything from neat little computer tricks to his thoughts and perspectives on the methodology and results of the project.

I would also like to acknowledge the help and encouragement that I got from several other Ph.D. candidates in Eivind's research group: Emil Karlsen, Martina Hall, Christian Schultz and Pål Røynestad all gave me useful ideas and valuable help with computer programming and statistics. Last, but not least, I would like to thank my friends and family for their patience, love and support.

Marta Riise Moksnes
Trondheim, June 2018

Table of Contents

List of Tables

List of Figures

Symbols and Abbreviations

C,S,D	=	Conserved, Specific, Differentiated
H	=	Node homogeneity
i,j	=	Gene pair
k	=	Condition or Node degree
k_p^{CSD}	=	Threshold value for C,S and D scores
L	=	Sample size in k_p^{CSD} determination
M	=	Number of gene pairs
n	=	Sub-sample size for σ determination
N	=	Data set size: Number of data points per gene
p	=	Importance level
RA	=	Rheumatoid Arthritis
S	=	Number of drawn sub-samples
o	=	Maximum permitted data point overlap between sub-samples
ρ	=	Spearman correlation coefficient
σ	=	Standard deviation of the mean
σ^2	=	Variance

Chapter 1

Introduction

"I think the next century will be the century of complexity"
- Stephen Hawking (San Jose Mercury News, January 2000)

1.1 Complex Biological Systems

A system is a set of interconnected components and the interactions between them. Our surroundings are always changing because they are composed of all sorts of interactions going back and forth between different actors. These interactions often act in messy patterns that are difficult to predict intuitively: We live in a world of *complex systems*. Countries, cities, airports and train stations interact by transport of people or cargo between them, building up networks with roads, flights and railroads. The people in these countries and cities interact with each other through a variety of channels; they talk to each other, work together, become friends with each other and interact on social media. They exchange money and products, e-mails and text messages, physical and sexual contact, and diseases. For e-mails to be transferred, the electronic components in our computers also interact with each other by sending electrical signals between them, in order to enable the computers to communicate over the Internet. These are just some examples; almost anywhere you look, a complex system can be found.

The desire to understand complex systems and to make sense of their seemingly unpredictable behaviour has lead to the field of *network science*. Network science is a relatively new field of science, and has experienced a rapid development during the last two decades. It builds upon the mathematical field of graph theory, which dates back to 1735, when the Swiss mathematician Leonard Euler gave a mathematical solution to the famous puzzle of the seven bridges in Königsberg. Network science uses both mathematical descriptions and computational modeling of networks in order to gain a deeper understanding of complex systems and their properties and behaviours. The objective is ultimately to try to predict and control their future behaviour. All systems can be represented as networks of components and interactions, and understanding the overall behaviour of different networks can lead to improvements in anything from logistics and infrastructure to medicine

and disease control.

Just like the world around you is complex, so is your body. A human body is its very own world of big and small complex systems, and we are highly dependent on the relationships between these systems on both a macroscopic and microscopic level. Analysis of biological networks on the molecular level is the objective of *systems biology*, a sub-field of network science. Molecular biological networks are found in humans, but also in other animals, plants and microorganisms. Two types of cellular networks commonly studied in systems biology are protein interaction networks and gene co-expression networks. In protein interaction networks, the nodes represent proteins and the links represent physical interactions between them, while in gene co-expression networks, the nodes represent individual genes, and the link between two nodes reflects a coordination in the timing and/or extent to which the two genes are transcribed into their corresponding messenger RNA, the primary function of which is to direct the synthesis of a given protein. It is also possible to study for instance the networks of biochemical reaction pathways or the interactions between metabolites. All of these systems can be studied together in order to understand the human body (or any other organism) as an integrated whole, or as a huge, complex machinery built up of numerous smaller components and systems. In the pursuit of integrating genomic, biochemical, physiological, behavioural and environmental information about the human body, we enter the field of *systems medicine*.

Like in any machinery, there are many things that can go wrong in the cellular systems in our bodies. Most of the time the consequences are small, but sometimes changes in our biological systems can have a very negative impact on our health and life. In order to find medical solutions to counter a given disease, it is essential to have knowledge about the underlying mechanisms behind its occurrence and sustainment: where in the machinery do things start to go wrong? Are there any loose bolts somewhere? Or maybe all components are working well on their own, while some biological cogwheels have teeth that cannot reach its counterpart anymore?

In a normal cellular machinery there are of course no metal bolts or cogwheels, so a more reasonable place to start the investigations is with genes or proteins. Living systems might be as different as a giant sequoia tree and a tiny sea slug, but they all have one thing in common: their cells contain DNA, on which there are genes; sometimes tens of thousands in number. All cells in a human being contain identical DNA molecules with the same genes in them, yet there is a huge variety of human cells with very different functions and properties. The genes act as templates for messenger RNAs, which in turn function as recipes for proteins to be produced. The regulation of what genes are used for RNA synthesis or not will determine what proteins can be produced in the given cell at that moment, and thereby possibly decide its fate. A mistake in the regulation of *gene expression*, the extent to which RNA is produced from the individual genes, may therefore have severe consequences for a cell. By using techniques such as RNA microarrays or RNA-Seq it is possible to monitor the gene expression of all the genes in our genome at a given time.

Any cell's environment is constantly monitored and used as input for gene regulation, so that the cell will be able to handle any situation as well as possible. The environmental input that a cell receives will guide its production of specific proteins. These proteins will bind to the DNA and thereby regulate the RNA producing machinery's access to different

genes. Gene expression can in this way be activated or repressed according to the situation that the cell faces [3].

As previously mentioned, there might still occur mistakes and disturbances in these highly regulated systems of genes, RNAs and proteins. Mutations in even one single gene can lead to severe diseases [4]. However, most of the time there is not just one or even a few genes that cause a disease. An alternative cause of disease is when the *interactions* between multiple genes cause trouble for the entire system. Instead of looking for *one* culprit among the genes, the researchers now have to look for the central network of criminals.

One disease with which a high number of genes are associated, and where the underlying mechanisms are by no means identified yet, is rheumatoid arthritis (RA), an autoimmune inflammatory disease that affects around 1% of the worldwide population [5]. The disease starts as a chronic inflammation of the synovial fluid of the small joints in the hands and feet, leading to highly painful, stiff and swollen joints. The disease can further result in tissue remodelling and damage, as well as spreading throughout the body, causing inflammation and malfunction in other organs. Patients with RA might also experience more general symptoms of disease, such as fever, fatigue, appetite loss and depression, and the disease is itself a risk factor for other potentially fatal conditions, such as liver failure and heart attack. Although there exists medical treatment for patients suffering from RA, there is no final cure. The heritability of the disease is estimated to be between 40% and 60% [6], so it is important to get as much knowledge as possible about the underlying genetics of the disease. One method to study the genetics of RA and other genetic diseases is through the above mentioned gene co-expression analysis.

Gene co-expression analysis methods are used to investigate the relationships between genes by studying their *co-expression*: the extent to which their expression is correlated. In a *differential* gene co-expression analysis the gene co-expression patterns from two or more conditions are compared to each other. Such an analysis is widely applicable, and can be used to compare for instance tissue types, related species, treatment effects to placebo effects, or healthy tissue to diseased tissue. Gene co-expression patterns often correlate well with biological functions [7, 8, 9, 10], and are therefore an essential step in trying to identify the molecular processes behind for instance evolution, cell differentiation, disease development or cell responses to new drugs. Today there are several online databases where big data sets of gene expression measurements on diseased and healthy tissue from many organisms are freely available [11, 12]. Mapping and analyzing co-expression patterns can potentially lead to the discovery of important genes or gene clusters associated with a given phenotype, leading us one step closer to understanding the mechanisms behind cell function and dysfunction [7, 8].

One method for differential co-expression analysis is the CSD (Conserved - Specific - Differentiated) method [2]. This method identifies change, loss or conservation of strong gene pair correlations across two conditions. The method strictly requires that the data sets from both conditions have a minimum number of data points per gene. Researchers are sometimes dependent on data they did not collect themselves, and it can therefore be difficult to control the size of the data sets. As the size requirement of the CSD method is equally strict on both data sets, it can be especially challenging to find large enough control data sets to use as a reference for disease data, for example. The existing data bases with general gene expression data from healthy individuals, for instance the GTEx

project database [12], are not exhaustive, and some tissue types are more difficult to find enough data on than others. The size requirements thus puts a restriction on what data sets can be used for a CSD analysis.

The aim of this study was two-fold. The first part focused on the development of the CSD method for differential gene co-expression analysis. The objective of this work was to study the possibility of developing the method to accommodate for smaller data sets than it was originally designed to handle. This would allow more data sets to be analyzed by the method, making it more broadly applicable. I wrote a Python script where the method modifications were implemented, and they were tested on a gene expression data set of healthy whole blood samples. The results were compared to the results from using the original method on the same data set. In the second part of the study, the CSD method was used to compare a data set with gene expression measurements obtained from the joint synovial fluid of patients diagnosed with rheumatoid arthritis, to a data set of measurements obtained from the joint synovial fluid of healthy individuals. The modifications in the method was applied to the reference data set, which was too small to be used otherwise. The objective of this analysis was to identify novel genes that could be essential in the development and/or sustainment of rheumatoid arthritis.

The thesis is organized to reflect the structure of the project. Chapter 2 provides background information and theory meant to help readers unfamiliar with the topics presented in the remaining chapters. This includes information about RA, RNA sequencing and differential gene co-expression analysis, including a description of the original CSD method, as well as topics from network theory and statistics that are relevant for the project. Because of the two-fold objective in this study I could have structured the remaining chapters in different ways. For easy navigation, the reader should be aware that I have chosen to structure all chapters so that the materials related to the method development are given in the first main section, and the materials related to the CSD analysis of rheumatoid arthritis are given in the second main section of each respective chapter: Chapter 3 first introduces the modifications of the CSD method and the steps for analyzing the results, as well as an overview of the structure of the script where the modifications were implemented. The second part goes through the different parts of the CSD analysis of RA. Chapter 4 first presents the results of the method study and development, and then the results of the CSD analysis. The results are also briefly analyzed in this chapter. Chapter 5 is likewise divided into a discussion of the method study results and a discussion of the results from the RA study. Chapter 6 finally gives a conclusion and outlook for both parts of the project.

Chapter 2

Background

This chapter will introduce the main topics underlying the method, discussion and results presented in this thesis. Most of the theory and background information presented will therefore be of direct relevance for the remaining chapters, while some parts are also included to give a more complete overview of the topics that are discussed. The main parts of the network theory are obtained from *Network Science* by Albert-László Barabási [1]. The reader is referred to this book for more extensive information about this topic. The section about the CSD method for differential gene co-expression analysis is based on the article by Voigt, Nowick and Almaas where the method was first described [2]. Additional sources are cited in the text. Some parts of this chapter are taken from the final report that I wrote for the course TBT4500 Specialization Project at NTNU in the fall 2017 as a preparation for this thesis.

2.1 Rheumatoid Arthritis

Rheumatoid arthritis (RA) is a chronic autoimmune inflammatory disease affecting about 1% of the worldwide population. The disease can occur at any age, and has a strong female predominance [5, 6]. The disease begins with immune cells attacking the synovial membrane of the joints, causing a chronic inflammation and swelling (hyperplasia) that is most prominent in the small joints of the hands and feet. RA is characterized by progressive articular remodelling and damage. The disease can further develop to cause inflammation in other organs, and is thus associated with a long list of comorbidities, for instance pulmonary, cardiovascular, liver, skeletal and psychological disorders, as well as early death [13].

The results from familial studies have shown that RA arises from a combination of genetic and environmental factors [14]. Twin studies have suggested the heritability of RA to be between 40 and 60 $\%$. Most of these results are from European populations, but a study of Japanese twins also gave comparable results [6]. The blood of most RA patients contains the auto-antibodies *rheumatoid factor* (RF) or *anti-citrullinated protein antibodies* (ACPA). RF is also present in other autoimmune diseases, while ACPA seems to

be more specific to RA. Nevertheless there are RA patients without any of these antibodies, showing that RA can be heterogeneously expressed in different patients [14]. This is also evident by the fact that not all RA patients responds to the same treatment methods, and patients who do not respond to one type of biologic therapy may respond well to another [14].

Human leukocyte antigens (HLAs), and polymorphisms in the corresponding genes, have been at the center of the RA research for many decades. The gene most highly associated with RA pathogenesis in previous research is the *HLA-DRB1* gene found in the major histocompability complex (MHC) region on Chromosome 6, a gene dense region that has been consistently shown to be associated with RA. Proteins from multiple of these genes build up the MHC protein complex that is involved in presenting proteins to developing T cells, making them able to recognize foreign molecules [15]. Many of the about 220 genes in the MHC region of Chromosome 6 have immunoregulatory functions [16]. *HLA* genes have been estimated to contribute between 11 and 37% to the RA heritability; more in ACPA-positive than in ACPA-negative patients. The moderate association between the *HLA* genes and RA heritability illustrates the combined effects of multiple genetic and non-genetic factors that are in play during RA pathogenesis. Some of the most relevant non-*HLA* genes associated with RA include *PADI4*, *PTPN22*, *TNFAIP3*, *STAT4* and *CCR6* [14, 17]. A list of 54 genes associated with RA is given in Appendix A1.

Environmental factors associated with RA include lung exposure to smoking, noxious solvents and silica dust, as well as alterations of the gastrointestinal microbiome, and bacterial infections, in particular periodontitis [5, 18]. These non-genetic factors have in common that they involve foreign particles that come in contact with one of the mucous layers of the body, either at respiratory, oral and/or intestinal sites. The mucosal sites normally function as barriers towards external environmental insults, such as pollutants, toxins and pathogens, and are tightly connected with the immune system [18]. Obesity and vitamin D deficiency have also been linked to RA [5].

2.2 RNA Sequencing

Active genes in a given cell are transcribed into their corresponding mRNAs. These are further processed and translated into protein, the final gene product. The *transcriptome* of a cell at a given physiological condition or developmental stage etc., is the full set of mRNA (and other RNA) transcripts present in the cell in that condition, as well as the quantity of these transcripts [19].

RNA sequencing (RNA-Seq) is a commonly used method for transcriptome profiling; the identification and quantification of RNA transcripts present in a biological sample at a given time. The RNA transcripts are converted to complementary pieces of DNA (cDNAs), before high-throughput DNA sequencing methods are used to determine the nucleotide sequence and relative abundance of each transcript [19].

Today the whole human genome is sequenced, and the genes are mapped, giving the sequence of more or less every human gene. The sequenced RNA transcripts from human tissue samples can therefore reveal the activity of each individual gene in each sample. The relative amount of RNA from each gene is known as the *expression profile* of the given tissue sample. Gene expression profiles from a high number of subjects can be

monitored over time, throughout a developmental process or in response to changes in the cell environment, for example a disease. RNA-Seq is therefore a useful tool in the study and analysis of differential gene expression. A wide range of data sets with RNA-Seq data from different experiments have been made publicly available online in databases such as GEO (Gene Expression Omnibus, [11]) and GTEx (The Genotype-Tissue Expression Project, [12]).

2.3 Differential Gene Expression

Although the same genetic material is found inside the nucleus of all the cells in your body, a blood cell differs greatly from a skin cell or a neuron. During development, cells differentiate into different cell types under control of various developmental signal molecules (growth factors) that control the expression of the different genes [20]. The differential expression of genes in response to various signals continues after the cells have settled into their specific cell types. The signals can now be any information from the cell environment, such as available nutrients, signal molecules released from neighboring cells, molecules released from infectious bacteria or damage to the cell. These signals lead to up- or down-regulation of genes coding for proteins called transcription factors (TFs). TF proteins can thus be viewed as an internal representation of a cell's environment [3]. Their function is to further regulate gene expression, making the cell as capable as possible of handling and responding to its current environment.

The TFs perform their regulation by binding physically to the DNA. The TF-DNA interaction will then either aid or prevent RNA Polymerase from accessing certain parts of the DNA, thereby activating or repressing transcription of genes in that area of the DNA double helix into RNA. The proteins of these genes might be new transcription factors or they can have other functions in the cell.

Differential gene expression studies are performed to analyze the change in gene expression profiles in response to some factor of interest. RNA-Seq or similar gene expression profiling methods are performed on tissue samples taken from two or more conditions of interest, returning specific expression profiles for each condition. One can then compare the cells' response to the chosen conditions by performing a statistical comparison between the expression profiles (see Section 2.5). Differentially expressed genes that are detected by this approach will be a natural choice for further investigation of the underlying mechanisms of the cells' response to the given conditions, for instance the mechanisms underlying a disease. In the best case, the results of such studies might even suggest new treatment methods, such as a point of attack for new drugs.

Proteins do not perform isolated tasks. They can not only regulate the production of themselves and each other, they also interact physically, and multiple proteins might affect the same parts of the cell and the same areas of the DNA, competing or cooperating, thereby modifying the effects and functions of each other. To a large degree, the function and behaviour of a cell results from the complex, simultaneous and combined effort of its proteins. The expression of one gene is therefore not independent of the expression of other genes. The fact that proteins are not working in isolation has led researchers to not only study the differential expression of single genes, but to look at pairs of genes and study the pair-wise relationships between their expression profiles. The synchronized

expression of gene pairs gives rise to co-expression networks (see Sections 2.6 and 2.7), which is a way to map genes and the relationships between them.

2.4 Network Theory

A network is defined as a set of components, referred to as *nodes* or *vertices*, and the interactions between them, called *links* or *edges* [1]. Networks can be represented visually, for instance as circles linked together by lines, as illustrated in Fig. 2.1, depicting the family member/relationship network of a fictional boy named Thomas.

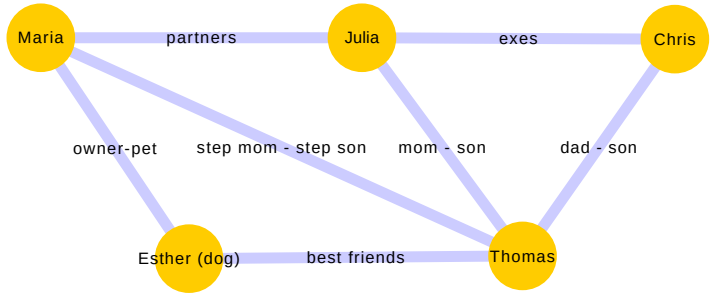


Figure 2.1: Visualization of an example network: A fictional family member/relationship network. The family members are represented with orange nodes, and their relationship is represented by blue links.

Any system where components interact or have some connection or relationship can thus be represented as networks. The examples in Chapter 1 illustrated the fact that network structures can be found in highly different contexts, in everything from trading and logistics to molecular biology. Network nodes can be as different as cities, persons and molecules, and network links can be as diverse as physical cargo transport, human relationships and statistical correlations of the translation processes of genes inside cells.

There are many types of complex systems that can be analyzed as networks just within the field of molecular biology. Because biological molecules are always connected to other molecules in some way or another; by chemical bonds, interactions, molecular conversions, or by affecting each others' activity, state or dynamics (for instance as enzymes), all of them can be viewed as components in a network [21]. Some examples of molecular networks in biology include gene regulatory networks, protein-protein interaction networks, metabolic networks, cell signaling networks and the previously mentioned gene co-expression networks that will be described in more detail in Chapter 2.6. The main questions in the analysis of biological networks are related to how the networks are connected and what impact this connectivity has on the functionality of the biological systems. Because networks are representations of real systems with distinct properties, there are alternative ways of constructing networks in order to capture the features of the system at hand. Some of the most common and basic network properties are presented in the following parts of this section, but it is not by any means an attempt to cover all possible

properties or characteristics of networks. Most of the topics will be of direct relevance for the following chapters, while some are included to make the sections more complete.

2.4.1 Network Connectivity and The Adjacency Matrix

The number of nodes in a network is denoted its *size*. Any node i has a given number of other nodes to which it is connected. These other nodes are node i's *nearest neighbors*. The *degree*, k_i, of a node is equal to its total number of nearest neighbors.

If the interactions in a system have directions (for instance that A affects B, but B does not affect A), the corresponding network should be a *directed* network, where each link has a designated direction. In an *undirected* network, the links do not have a direction because the relationship between two connected nodes is always equal for both nodes. An example of an undirected network is a correlation network, where it is impossible that one node correlates with another without the other also correlating with the first one.

A *connected component* of an undirected network is a set of nodes that are connected with each other in a way that makes it possible to start from any node and follow the links of the network to reach any other node in the component. If any connected component is much bigger than any other component of the network, it is called the *giant component* [1].

Some systems have heterogeneous connections between their components [22], for instance there might be a difference in the passenger capacity of a flight, the strength of a correlation or the intensity of a signal. Heterogeneous connections or interactions, can be represented as weighted links. The weight of a link quantifies its relative strength/capacity/intensity compared to the other links in the network. Weighted networks can be converted to unweighted networks with homogeneous links by choosing a threshold value for the weight of the links, and keeping all links with a weight above the threshold.

A simple way of representing the full connectivity structure of a network is its *adjacency matrix*, $\mathbf{A}$ [21]. $\mathbf{A}$ is a square matrix whose elements a_{ij} represent the presence or absence, and possibly the direction and/or strength, of a link between nodes i and j. In the case of an unweighted, undirected network, the adjacency matrix $\mathbf{A}$ takes a binary form: $a_{ij} = 1$ indicates the presence of a link between nodes i and j, whereas $a_{ij} = 0$ indicate the absence of a link between them. An example of the adjacency matrix of an unweighted, undirected network can be found in Fig. 2.2, which also illustrates the concept of cliques, which will be presented later in this Chapter. If the network is weighted, the entries of the adjacency matrix takes a continuous range of numbers reflecting this property, and if the network is directed, the entries of the adjacency matrix might also take both positive and negative values.

All connections of a component i is represented by the i'th row and the i'th column in the adjacency matrix. In the adjacency matrix of an undirected network, the values and order of the elements in the i'th row is equal to those of the i'th column, making the matrix symmetric around its diagonal. In the case of a directed network, the i'th row and the i'th column may differ, because each of them represents one direction of interaction. The degree of a node i in an undirected network can thus be found by counting the non-zero entries of either the i'th row or the i'th column of its adjacency matrix, while in the case of directed networks, the *total degree* of a node i, which is the sum of its *in-degree*, k_{in}, and *out-degree*, k_{out}, can be found by counting the non-zero entries of both the i'th row *and* column. The in- and out-degree gives the number of incoming and outgoing links,

respectively. If the nodes do not interact with themselves to form self-links, the entries of the diagonal of the adjacency matrix are all zero.

2.4.2 The Degree Distribution and Scale-Free Networks

The *degree distribution* of a network refers to the proportion of nodes within the network that has the degree k. In a random network, such as an Erdős–Rényi-network, links are randomly associated with nodes based on a uniform probability. The degree distribution of such an artificial, random network, is a binomial distribution, resembling a Gauss curve with smaller variance [21]. This means that most nodes have a degree close to the average degree, $< k >$.

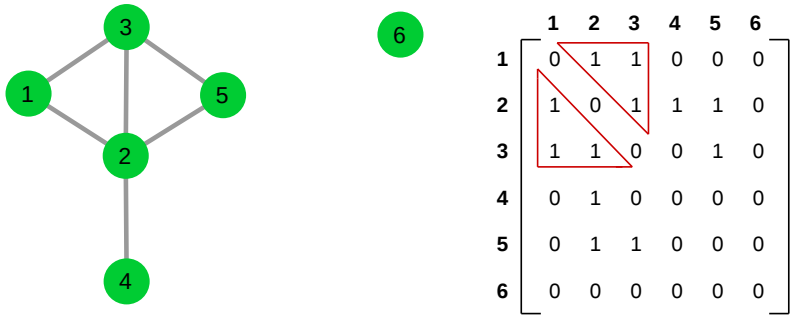
Most biological and other real-world networks are characterized by being so-called *scale-free*, meaning that their degree distribution follows a power-law distribution [1]. This can be formulated by Eq. (2.1), where p_k denotes the probability that a node will have the degree k, and γ is the degree exponent:

$$p_k \sim k^{-\gamma} \tag{2.1}$$

Scale-free networks are characterized by having many low-degree nodes and a few highly connected nodes [1]. Nodes with many more neighbors than the average degree in the network are called *hubs*. The scale-free topology makes them robust against random attacks, for instance randomly removing nodes or changing links in the network. The average *distance* in these networks are much shorter than in random networks. A network distance is defined as the shortest path length between two nodes, in other words the minimum number of links (or link weight in the case of weighted networks) needed to go from one node to another. Because of the hubs there are many alternative routes between two nodes if a non-hub node between them disappears. While robust against random attacks, scale-free networks are all the more vulnerable against targeted attacks: Selectively removing hubs will quickly cause the network to break down, partially or fully depending on how many hubs are removed [1]. Imagine the chaos that would arise if multiple hubs in the European airport network, such as London Airport Heathrow or Amsterdam Airport Schiphol, were immediately closed down for traffic!

2.4.3 Network Modules

Even if the full network is scale-free, local sub-networks can be organized differently and have other properties. A collection of nodes that forms a sub-network is called a *network module* [23]. The term does not have a unique definition, and can refer to any defined substructure in a given network. An example of a network module is the *clique*, in which all nodes are connected to all the others [1]. Finding cliques is no straight-forward task, but there are several computer algorithms for detecting different types of cliques in big networks. For a sufficiently small network, cliques can easily be identified just by looking at the network itself, or its adjacency matrix. A clique would be visible in the adjacency matrix as a group of nodes with no zero-entries between any of the nodes in the group. An example of the identification of one (of two) 3 node cliques in a network consisting of 6 nodes, is given in Fig. 2.2.



(a) **(b)**

Figure 2.2: Conceptual visualization of a clique (nodes 1, 2 and 3) in a network of six nodes, numbered 1 to 6 (a). The clique is also marked in the corresponding adjacency matrix (b).

2.4.4 Assortative and Disassortative Networks

Assortativity denotes the tendency of nodes of similar degree to connect to each other, while *disassortativity* denotes the tendency of nodes of similar degree to avoid connecting to each other in the network: highly connected nodes will tend to connect to nodes with low degree and vice versa [1].

Such degree correlations can be detected by for instance studying the *neighborhood connectivity distribution* of a network, found by plotting the average connectivity (degree) of the neighbors of nodes with degree k as a function of the degree k itself. By approximating a line through the resulting points, the network's degree correlation function, $k_{nn}(k)$, is found, expressed in Eq. (2.2):

$$k_{nn}(k) = ak^{\mu} \tag{2.2}$$

where k is the degree, μ is the correlation exponent, and a is just a regression constant. The type of degree correlation will be dependent on the correlation exponent, μ:

- Assortative networks: $\mu > 0$
- Neutral networks: $\mu = 0$
- Disassortative networks: $\mu < 0$

Fig. 2.3 illustrates the degree distributions of three real networks: one assortative, one neutral and one disassortative.

2.4.5 Node Parameters

The previously discussed node degree, k, is perhaps the most important parameter of nodes in a scale-free network, since it gives information about how connected the node is to the

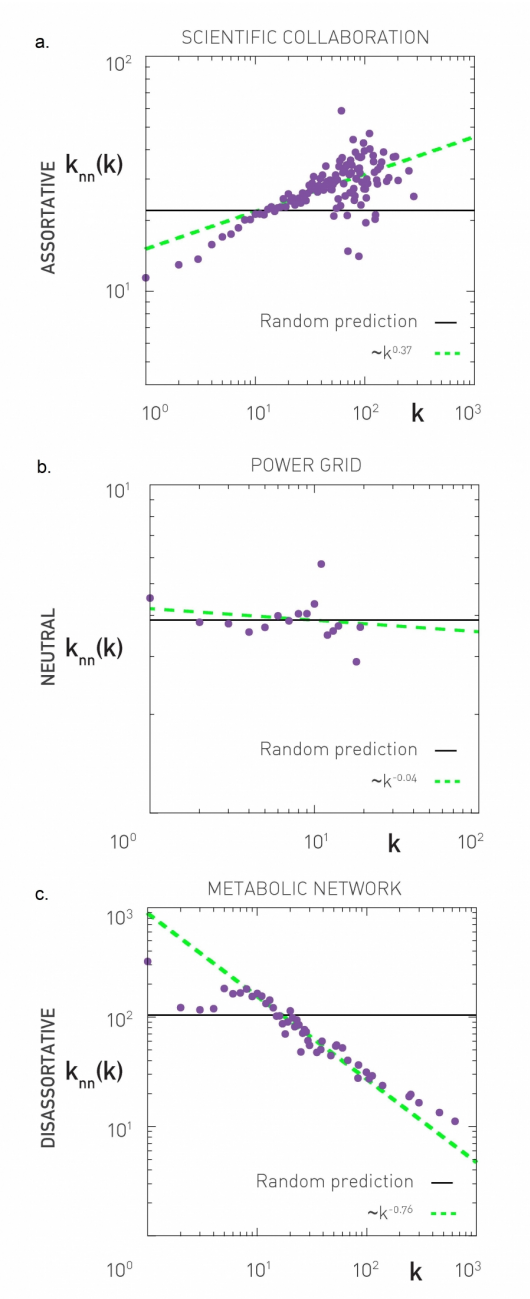


Figure 2.3: The degree correlation function $k_{nn}(k)$ for three real networks: An assortative collaboration network ($\mu = 0.37$), a neutral power grid network ($\mu = 0.04$) and a disassortative metabolic network ($\mu = 0.76$). The green, dotted line gives the regression line through the points, and the black line illustrates what would be expected if the degree correlation was completely random. Source: [1]

rest of the network. Other important parameters include various centrality measures. The *eccentricity* of a node *i* is defined as the maximum finite distance between node *i* and any other node in the network. The eccentricity of an isolated node is defined to be zero. The *betweenness centrality* of a node is the number of shortest paths in the network that go through that node. The *closeness centrality* of a node measures the sum of the shortest paths from the node to all other nodes in a connected component.

2.4.6 Network Parameters

Networks can be characterized by various parameters that reflect different network properties. The previously described degree-distribution and degree correlation are two such parameters. The *diameter* of a network is defined as the largest eccentricity in a connected component, while the *radius* is defined as the smallest of the non-zero eccentricities in the network. The *characteristic path length* is the average distance between nodes in the network.

2.5 Statistics

2.5.1 Correlation

When constructing a network, one needs a systematic way of deciding whether or not there will be a link between two nodes. The decision to keep or reject a possible link should be based on the type of connection the link reflects. Although some relations between components are binary of nature, this is most often not the case. In many networks, the ”connection” between two nodes is a continuous measure, for instance a correlation value between some property of the nodes. This is the case for gene co-expression networks, where any links represent a correlation of the levels of RNA from each gene. Most of the time the correlation coefficient of the expression of two genes is not strictly zero, but it isn't necessarily very high either. In order to create a meaningful gene co-expression network, the question to ask is not just whether there is *any* relationship between the expression levels of two genes, but how strong and *systematic* the relationship between them is. A strong correlation between two genes might indicate a biologically relevant relationship between them, so it is important to separate correlations that could just as easily have happened by coincidence, and those that it would be unlikely to find without there being a systematic mechanism behind them.

A correlation coefficient, typically denoted ρ, takes any value between -1 and 1. The closer ρ is to -1 or 1, the stronger is the negative or positive correlation, respectively. A strong positive correlation coefficient reflects two data series where the corresponding values increase or decrease together, while a strong negative correlation coefficient reflects the opposite: an increase in one of the data series is closely associated with a decrease in the other, or vice versa.

The closer ρ is to zero, the less do the two data series follow a given relationship [24]. Values sufficiently close to -1 or 1 are considered significant.

Pearson Correlation

The Pearson correlation coefficient can be used to calculate the linear relation between two raw data measurement series, for instance the normalized gene expression levels obtained in an RNA-Seq experiment. The Pearson correlation coefficient is given by Eq. (2.3) [25]:

$$\rho_{ij} = \frac{\text{cov}(i,j)}{\sigma_i \sigma_j}, \tag{2.3}$$

where cov(i,j) denotes the covariance between measurement series *i* and *j*, and σ_i and σ_j denote the standard deviations of the measurement series *i* and *j*, respectively.

Spearman Correlation

The equation for the Spearman correlation coefficient has the same form as the Pearson correlation, but instead of measuring the correlation between the raw data measurements, it is used to find the correlation of the *ranks* of the measurements.

The ranks are the numbers that would indicate the place of the measurements in a list where they were sorted by size. The rank of the numbers 4.5, 2.3, 7.3 and 1.0 would for example be 3, 2, 4 and 1, respectively. The rank of a gene's expression level is thus an alternative way of giving its relative expression level compared to the other measured genes from the same sample. An advantage of the rank correlation over the linear correlation is that the measurements do not have to be normalized in order for two correlation coefficients from two different data sets to be compared.

It follows that before calculating the Spearman correlation of the expression levels of two genes, the *N* measurements of each gene have to be converted to their corresponding rank in each sample. For two genes *i* and *j* with *N* data points each, the Spearman correlation coefficient is given by Eq. (2.4):

$$\rho_{\text{rg}(i)\text{rg}(j)} = \frac{\text{cov}(\text{rg}(i), \text{rg}(j))}{\sigma_{\text{rg}(i)} \sigma_{\text{rg}(j)}} \tag{2.4}$$

where rg(i) and rg(j) denote the ranks of genes i and j, respectively, cov(rg(i),rg(j)) denotes the covariance between rg(i) and rg(j), and $\sigma_{\text{rg}(i)}$ and $\sigma_{\text{rg}(j)}$ give the standard deviations of rg(i) and rg(j), respectively.

2.5.2 Computing Running Variance

The variance of a sample of N data points, $x_1, ..., x_N$, can be computed using Eq. (2.5) [25]:

$$\sigma^2 = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \bar{x})^2 = \sum_{i=1}^{N} x_i^2 - (N-1) \cdot \bar{x}^2 \tag{2.5}$$

where $\bar{x}$ is the mean of the data points. If new values, x_i, are added one at a time, and the final *N* is large, the computational task will be so big that a computer should calculate the variance. If using the left hand side version of the variance formula, the computer has to go through the data two times for each new x_i value: once for calculating the mean and once for calculating the differences.

The two-pass approach will be more time consuming, so the most intuitive solution might then be to make the computer accumulate the two sums of x_i^2 and x_i (to find $\bar{x}$) separately, and then find the difference according to the right hand side of Eq. (2.5). There is, however, a possible pitfall to this approach: If each x_i is very large or has many decimals, then the sum of squared x_i's will get even bigger, possibly losing precision because the size of the value exceeds the number of digits the computer has available for storing float numbers. If the difference between the two sums is small, this might lead to a serious loss of precision. In the worst case scenario, one might even get a *negative* number, even though a variance must always be positive. Not only will the results be far off, but it would also make it impossible to take the square root of the variances in order to get the standard deviations.

A solution to this problem was proposed by B.P. Welford in 1962 [26]. His numerically stable algorithm for computing running variance is given as the following recurrence formulas initialized with $M_1 = x_1$ and $S_1 = 0$:

$$M_k = M_{k+1} + \frac{x_k - M_{k-1}}{k} \tag{2.6}$$

$$S_k = S_{k-1} + (x_k - M_{k-1}) \cdot (x_k - M_k) \tag{2.7}$$

where the k^{th} estimate of the variance is found from Eq. (2.8) as long as $2 \leq k \leq N$:

$$\sigma^2 = \frac{S_k}{k-1} \tag{2.8}$$

2.5.3 Confounding

Confounding occurs when there seems to be a direct relationship between two variables, while in fact the apparent relationship is caused by a hidden factor not accounted for, a *confounding factor*. Confounding factors can easily lead to false conclusions. An example could be the following fictional study: You have bone samples from a group of elderly people and a group of young people. Almost all the elderly people have osteoporosis, and none of the young people do. From that you conclude that osteoporosis comes from old age. If the real reason for the high occurrence of osteoporosis in the group of elderly test subjects is that without your knowledge, they all live in the same nursing home where they get too little sun light and their diet does not contain vitamin D, this would be a confounding factor possibly leading you to make a false conclusion about the origin of osteoporosis.

Even if you take care to avoid such obvious sources of errors as in the previous example, it will be difficult to remove *all* possible confounding factors in a data set, for instance a gene expression data set. If a pair of genes, denoted *i* and *j*, shows an overall significant co-expression across the *N* samples in a condition *k*, and the co-expression is similarly big in all the *N* individuals, it is likely that a strong co-expression between these genes is typical for condition *k*. If, however, the gene pair shows a very strong correlation of the data for one or more sub-groups of the *N* samples, and a non-significant co-expression for the rest of the samples, the overall correlation coefficient could still end up being significant,

but there is a risk that this significance is due to confounding factor(s): some unknown condition or conditions that the individuals with strong co-expression have in common (age, unreported diseases, life-style etc.), instead of the condition *k* itself. Confounding factors would therefore typically lead to a high variance in the correlation calculated from many different subsets of the two full data series.

2.5.4 Boxplots

A boxplot is used to graphically visualize the distribution of numerical data [27]. An example boxplot is given in Fig. 2.4. Any set of numbers can be ordered, allowing us to find its different quartiles: The second quartile, Q2, is the middle value (median) of the data, while the first and third quartiles, Q1 and Q3, are defined to be the middle values between Q2 and the smallest and biggest values in the data set, respectively.

In a traditional boxplot there is a rectangle, "the box", where the first and third quartiles can be found as the upper and lower sides of the box. The height of the box (= Q3-Q1), called the interquartile range (IQR) thus gives the range within which 50% of the data points are found. The second quartile, Q2, representing the median of all the data points, are found as a horizontal line (the red line in Fig. 2.4) inside the box. Furthermore there are two *whiskers* extending above and below the box. The range of the whiskers are typically set between $W_{lower} = Q1 - C \cdot IQR$ and $W_{upper} = Q3 + C \cdot IQR$ where W_{upper} and W_{lower} represent the range of each whisker, and *C* is a constant that has to be chosen. If no data points are as high or low as these values, the whiskers will instead end at the highest and lowest data point. Any data point above W_{upper} or below W_{lower} are represented as outliers. In Fig. 2.4, each outlier is represented by "+".

2.5.5 Hypothesis Testing

In hypothesis testing, one compares a statistical hypothesis H1 to the null hypothesis H0. Typically the H1 hypothesis can be something like "there is a linear relationship between these two data sets", or "there is a significant difference between these two measurement series", while the respective null hypotheses H0 would be "there is no linear relationship between the data sets, they are completely uncorrelated", or "there is no significant difference between these measurement series".

To decide if there is a significant correlation between the two, one calculates the correlation coefficient ρ for the two data sets, and then compares it to the correlation coefficient distribution from completely uncorrelated data sets of the same size. If the calculated ρ could just as likely have been from two uncorrelated data sets, one keeps the null hypothesis, whereas if it is highly unlikely that the calculated ρ value would come from the comparison of two uncorrelated data sets, one rejects the null hypothesis in favor of the alternative hypothesis H1.

To decide whether to reject the null hypothesis or not, it is necessary to quantify "highly unlikely"; one needs to set a threshold for $|\rho|$, above which ρ values are considered significant. Tables with the probabilities (p-values) of given ρ values to occur from uncorrelated data sets of given sizes, are widely available online or in statistics literature. By convention, absolute p-values of 0.05 or below are considered significant, and 0.01 or below is considered highly significant. If you have calculated a correlation coefficient ρ

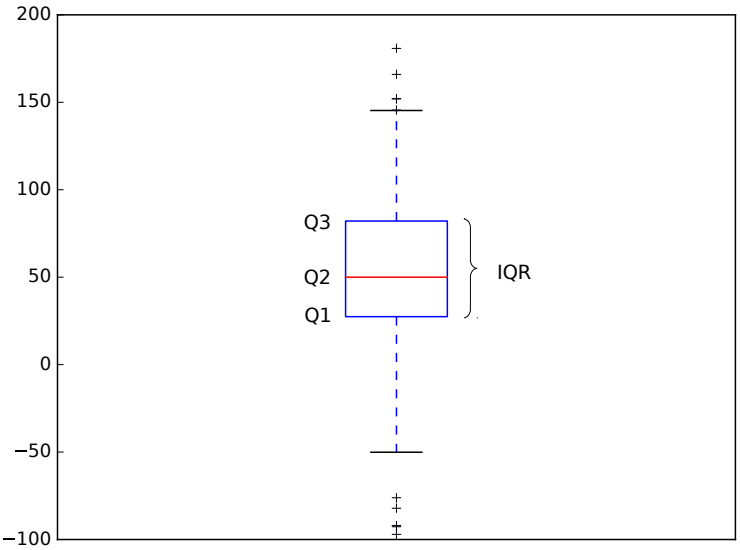


Figure 2.4: An example boxplot where the first, second and third quartiles, Q1, Q2 and Q3, as well as the interquartile region, IQR, are marked.

from two data sets, and find a corresponding p-value of 0.01, it means that you would only expect a ρ value this extreme or more (this close or closer to 1 or -1) to happen 1% of the times *if the null hypothesis H0 is true*. In other words, in 99% of the cases, a ρ value this good would come from a true systematic relationship between the data sets [25, 28].

For large data sets, the Fisher's z transformation can be used to find p-values from correlation coefficient values [23]. New z values are computed from the given ρ values using Eq. (2.9):

$$z = \frac{1}{2} \ln \frac{(1+\rho)}{(1-\rho)} \tag{2.9}$$

The obtained z values can be assumed to be normally distributed, and corresponding p-values can be found in any p-value table for normal distributions. Alternatively, one can use q-values, which are p-values adjusted for the false discovery rate discussed Section 2.5.8.

Following these norms there is still a chance of either rejecting the null hypothesis when the data sets are really unrelated, or keeping the null hypothesis even if the data sets are actually related. The first is called a *Type I error*, or *false positive result*, while the latter would be a *Type II error*, or *false negative result* [28]. In the specific case of constructing gene co-expression networks, a Type I error would be to put a link between two genes that have no biological relation in their expression levels, whereas a Type II error would be not putting a link between two genes that in reality do have a biological relation of their expression.

One type of statistical hypothesis testing is the *t-test*. The three most common varia-

tions of the *t*-test are the one-sample, two-sample and paired *t*-tests. Only the two-sample *t*-test will be explained in this section, as it is the one that will be relevant in later chapters.

As the name implies, there are two samples, or measurement series, in a two-sample *t*-test. The test is used to determine if two data sets are significantly different from each other or not. More precisely, it tests the null hypothesis that the two measurement series have the same average, meaning that they can be assumed to come from the same underlying distribution.

Let $y_1, ..., y_n$ be the n measurements in sample 1 and $x_1, ..., x_m$ be the m measurements in the sample 2. The average of the two samples are denoted $\bar{y}$ and $\bar{x}$, and the corresponding sample variances are denoted σ_x^2 and σ_y^2. The degrees of freedom, df, are found as $df = n + m - 2$. From these parameters, the *t*-statistic can be calculated using Eq. (2.10) [28]:

$$t = \frac{\bar{y_1} - \bar{y_2}}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \tag{2.10}$$

From df and the calculated *t*-statistic, one can find the associated p-value from a *t*-table. If the p-value is smaller than 0.05, the null hypothesis of equal mean values can be rejected, and the two samples can be assumed to come from different underlying distributions.

2.5.6 The Multiple Comparison Problem

The multiple comparison problem arises when several individual hypothesis tests are considered at the same time [29]. The significance values of the individual comparisons no longer reflect the true significance value of the combined set of multiple hypothesis tests, where the same data are used in many different comparisons. The more comparisons that are made simultaneously, the more false positives are expected [24].

This would to a great extent be relevant for the construction of large gene co-expression networks. There are between $20\,000$ and $30\,000$ genes in the human genome, and comparing all these genes pair-wise would result in between $1.9 \cdot 10^8$ and $4.5 \cdot 10^8$ simultaneous hypothesis tests. Even if one does not study the correlation of all the genes in the whole genome, but restricts the analysis to for instance 10^3 genes, this would result in almost $5 \cdot 10^5$ comparisons. A p-value of 0.05, normally considered significant, would allow about $25\,000$ false positive results. If the goal of a differential co-expression analysis is for example to identify genes (or relations between genes) involved in a disease (using the "guilt-by-association" principle), one could end up with a high number of "miscarriages of justice". Your network could easily become rather useless as "evidence". The Bonferroni correction and the false discovery rate are two possible ways of handling the multiple comparison problem.

2.5.7 The Bonferroni Correction

There are many different procedures that can be used to protect us from calling differences significant when they in fact are not. To avoid getting too many false positives, the significance level has to be adjusted for the number of simultaneous comparisons. An easy way to do this is to use the Bonferroni correction [24].

The Bonferroni correction simply divides the individual significance level (0.05) by the number of tests being conducted (e.g. corresponding to number of gene pairs), and uses this as the new significance level for all the individual tests [29]. This works well with relatively few variables, but it is not difficult to see that with for instance 10^3 genes, this requires an individual significance level of $5 \cdot 10^{-5}$, which is very low. The more comparisons you make, the lower the significance level gets for each case, with the result that it becomes more and more difficult to detect even highly significant correlations. As the overall Type I error rate goes down, the Type II error rate would therefore increase. The risk of getting many false discoveries decreases dramatically, but you might not get too many valuable discoveries either. It is therefore often advisable to use other correction methods when making a *very* high number of hypothesis tests [29]. However, if there are enough samples (enough simultaneous measurements of the genes at study), the more reliable each correlation value would get, and the easier it is to still get results below the Bonferroni corrected significance level even with a high number of tests.

2.5.8 The False Discovery Rate

A different way of performing multiple comparison correction is by using the *false discovery rate* (FDR). The false discovery rate quantifies the expected number of false positive results. The false positive rate of the Bonferroni correction gives the percentage of all results that will be truly negative, while the FDR gives the percentage of the *declared* positive results that will be truly negative. The FDR has a greater power, but is less conservative, so the number of Type I errors will be larger than if using the Bonferroni correction [30].

2.6 Gene Co-Expression Networks

Gene expression data can be obtained by performing an RNA microarray analysis (see Section 2.2). Before constructing a gene co-expression network, all the genes need to be measured simultaneously in a sufficient number of independent samples. In the complete data set, the expression levels of each gene are represented by a vector with as many elements as there are samples analyzed. Each vector element thus contains one measurement of the abundance of mRNA from that gene in a tissue sample from one person. The expression levels of all the genes measured in the same tissue sample will have the same placement in their respective gene's expression vector.

As previously mentioned, a *gene co-expression network* is a network where the nodes represent genes, and the links indicate a synchronized gene expression pattern between the connected genes. Two genes can have positively correlated expression (the up- and downregulation patterns follow each other, simultaneously or with some delay), negatively correlated expression (upregulation of one gene is associated with downregulation of the other and vice versa) or no correlation of expression. The connection between two genes in a co-expression network is the correlation of their respective expression vectors. In unweighted gene co-expression networks, any correlation above a given threshold is considered significant, and should be represented as a link in the network. Genes connected by a link are said to be *co-expressed*. In weighted gene co-expression networks, the strength of each link will reflect the strength of the correlation.

The motivation behind constructing these networks is that gene co-expression patterns are found to often correlate with a given phenotype or biological function [7, 8, 9, 10]. Constructing a network from pairwise gene co-expressions makes it possible to find network characteristics and structures that would otherwise be difficult to detect if you were to study the genes or even pairs of genes separately.

Expression data always comes from experimental measurements of RNA levels, and will therefore always be noisy due to measurement uncertainty. It is important to be careful when analyzing the data, making sure to be aware of possibly false conclusions. When performing an unweighted network analysis it is important to set a correlation threshold for co-expression links so that they are likely to reflect significant biological, real-life relations between the genes as well as possible. Without such a correlation threshold, one could easily end up with a complete (fully connected) network, where all the nodes are connected to all the others, and from which no meaningful information can be drawn.

2.7 Differential Gene Co-Expression Analysis

Differential gene co-expression analysis is the analysis of how the pairwise correlations in gene expression differ between two or more conditions. The analysis is performed in order to explain for instance a biological system's level of dysfunction, or the cell response to given conditions. "Conditions" can in this case refer to the presence or absence of a variety of categories and factors, such as disease or illness, substance abuse, exposure to pollutants, medical treatment, lifestyle factors, tissue type, species and so on. Any condition can be studied by use of the same methods.

There are several approaches to the analysis of the change in gene co-expression, and the choice of method will usually reflect the type of information one is interested in. The methods differ both in the way they identify differential gene co-expression, in their main focus of interest and in their output. One approach is to create one co-expression network for each condition to be studied, and then simply see what links or nodes have appeared or disappeared when going from one condition to another [31, 32]. The alternative approach is to construct *one* differential co-expression network, where a link between two genes represents the *change* in co-expression for that gene pair between two conditions [2, 33, 34]. The change detected with either approach, might either only be the occurrence/disappearance of a link between the conditions, or additionally the change of link type, from positive to negative correlation, can be included [2]. Some approaches identify differentially expressed genes instead of focusing on the links [32, 33, 35] and some focus on detecting different network modules [35, 36, 37].

2.8 The CSD Method

2.8.1 CSD networks

The CSD method for differential co-expression analysis [2] is used to create a single network that describes three different types of co-expression changes for pairs of genes between two given conditions. A CSD network consists of nodes, representing genes, and

links, representing a *condition-wise change or conservation* of the co-expression relationship between two genes. A network link reveals, in other words, whether or not the correlation between two genes has changed significantly between the two conditions.

The pair-wise gene co-expression is first calculated for each condition separately. Gene co-expression is represented by the Spearman correlation coefficient, $\rho_{ij,k}$, for a given gene pair (*i,j*) over all the *N* gene expression data points representing condition *k*. A *co-expression relationship* in a given condition can either be a strong positive correlation ($\rho_{ij,k}$ close or equal to 1), a strong negative correlation ($\rho_{ij,k}$ close or equal to -1), or a weak/no correlation ($\rho_{ij,k}$ close or equal to 0). From the co-expression relationships, three types of *differential* co-expression relationships are recognized by the CSD method: *conserved* (denoted C), *specific* (denoted S) and *differentiated* (denoted D) co-expression:

- A conserved (C) link in the CSD network represents a situation where a significant co-expression relationship between two genes has not changed between the two conditions: The correlation is strong, and with the same sign, in both conditions.

- A specific (S) link represents situations where there is a strong correlation of any sign between the genes in one condition and a weak or no correlation in the other.

- A differentiated (D) link represents the situation where a strong gene pair correlation changes sign when going from one condition to the other.

The three possible differential co-expression relationships between two genes are visualized as a gene co-expression score surface plot in Fig. 2.5. The Spearman correlation coefficients ρ_1 and ρ_2 for a single gene pair in conditions 1 and 2 are represented on one axis each, giving a plot with eight co-expression areas of interest. Two blue areas represent C relationships, with strong, same sign correlation coefficients in both conditions, two red areas represent D relationships with strong, oppositely signed correlation coefficients in the two conditions, and four green areas represent S relationships, with a strong correlation of either sign in one condition and a close to zero correlation coefficient in the other condition. The white area represents combinations of correlation coefficients that will not result in a link in the CSD network. This includes situations where one or both correlations are neither particularly strong or particularly weak, and situations where both correlations are weak.

The C, S or D relationship between a pair of genes *i* and *j* is determined by calculating gene relationship scores, C_{ij}, S_{ij} and D_{ij}, given in Eqs. (2.11), (2.12) and (2.13) respectively [2]:

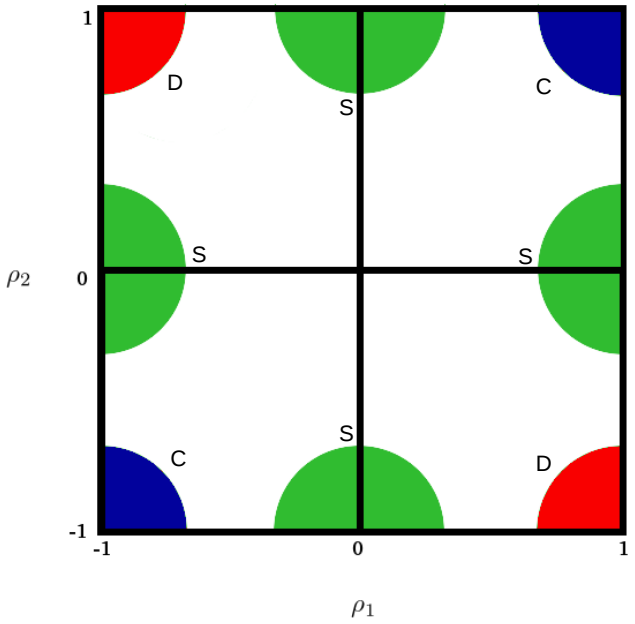


Figure 2.5: The gene co-expression score surface gives a general representation of the combinations of correlation coefficients from two conditions that will correspond to the three different types of co-expression relationships; C, S and D. The variables ρ_1 and ρ_2 represents the Spearman correlations of a given gene pair in conditions 1 and 2 respectively. The green areas corresponds to specific (S) differential co-expression, the blue areas correspond to conserved (C) co-expression, and the red area corresponds to differentiated (D) co-expression. The co-expression relationship types are also indicated by their respective letter, C, S or D, next to their corresponding colored area. The white area represents combinations of correlations that will not result in a network link. Source: [2]

$$C_{ij} = \frac{|\rho_{ij,1} + \rho_{ij,2}|}{\sqrt{\sigma_{ij,1}^2 + \sigma_{ij,2}^2}} \tag{2.11}$$

$$S_{ij} = \frac{||\rho_{ij,1}| - |\rho_{ij,2}||}{\sqrt{\sigma_{ij,1}^2 + \sigma_{ij,2}^2}} \tag{2.12}$$

$$D_{ij} = \frac{|\rho_{ij,1}| + |\rho_{ij,2}| - |\rho_{ij,1} + \rho_{ij,2}|}{\sqrt{\sigma_{ij,1}^2 + \sigma_{ij,2}^2}} \tag{2.13}$$

2.8.2 Internal Co-Expression Variations in Each Condition

The C_{ij}, S_{ij} and D_{ij} scores are corrected for possible confounding factors in the data set by including the variable $\sigma_{ij,k}^2$ in the equations. This is an estimate of the variability in the Spearman correlation coefficient for each gene pair *i,j* within the data set representing condition k. A high variability in correlation between two genes reflects that groups of the data points for those genes are very differently correlated than the other data points, which could be caused by some unknown factor affecting only a sub-group of the subjects from which the tissue samples are taken. If the scores were not corrected for high variabilities within the data set, there would be a risk of creating a false impression that an observed co-expression between two genes is related to the studied condition, when in fact it is related to some unknown factor not accounted for. A high $\sigma_{ij,k}^2$ will therefore reduce the value of the co-expression scores, C_{ij}, S_{ij} and D_{ij}.

An estimate for the internal variance in co-expression for each gene pair within each condition is computed as the standard deviation of the mean of the Spearman correlation coefficients calculated for each independent sub-sample of size n that can be drawn from the N data points. The sub-sampling is done using the following sub-sampling algorithm [2]:

1. The total of N data points per gene are sequentially numbered.

2. The N data points are divided into non-overlapping sub-samples of size n. With $n = 7$ and $N = 49$ this would result in 7 initial sub-samples.

3. The first data point is used as the initiating data point, n^*. A new sub-sample is built by starting with data point n^*. While sequentially iterating through the data points, new data points are added by the following criterion: if the current data point has never previously co-occurred in a sub-sample with any of the data points already in the current sub-sample, it is added to it.

4. When the size of the current sub-sample reaches n, a new sub-sample is initiated with the same initiating data point, n^*, and step 3 is repeated.

5. When no valid sub-sample of size n can be drawn using n^* as the initiating data point, step 3 is repeated using $n^* = n^* + 1$ as the next initiating data point.

6. The procedure is complete when no more allowed sub-samples of size n can be constructed.

This algorithm ensures that the highest possible number of independent sub-samples of a fixed size n is drawn from the full data set. The independence is ensured by the condition that the intersection between any two sub-samples cannot be more than one data point.

The sub-sampling strategy of the CSD method accounts for the fact that confounding factors could change the "real" correlation within the sub-populations of samples that they affect. The rationale for using only independent sub-samples, where no *pairs* or bigger sets of data points are part of multiple sub-samples, is therefore that calculating correlations over the same data points multiple times could possibly lead to an underestimation of the real variability within the data.

In order to detect confounding factors as well as possible, and also get a high number of sub-samples from which each $\sigma_{ij,k}$ can be calculated, the sub-sample size n should be small, while still allowing reasonable calculations of correlation coefficients. Voigt, Nowick and Almaas [2] found a sub-sample size of $n = 7$ to be the minimum requirement to calculate a meaningful Spearman correlation, and that in order to get enough data points for the estimation of $\sigma_{ij,k}$, the data set should have a total of at least $N = n^2$ data points per gene. This restricts the method to data sets with a minimum of 49 data points per gene per condition.

2.8.3 Thresholds for the C_{ij}, S_{ij} and D_{ij} scores

Looking into Eqs. (2.11), (2.12) and (2.13), note that taking the absolute values ensures positive numerators in each fraction, while the root sum square of the standard deviations of the means, $\sigma_{ij,k}$, in the denominator can be any positive number, with no limitation on how close they are to 0. The scores can therefore take values from 0 to infinity. The C_{ij}, S_{ij} and D_{ij} scores quantify the extent of each C, S and D co-expression relationship for a given gene pair, but the values of the three scores are *not* directly comparable, as the scores follow different distributions.

In order to obtain one single, meaningful network in which all links are comparable to each other, the three scores have to be mapped to a common scale. Three threshold values, k_p^C, k_p^S and k_p^D, are determined so that each of them corresponds to an *importance level*, p. The importance level is not the same as the traditional p-value common in standard hypothesis testing. In contrast to a p-value, the importance level in the CSD method is not based on an hypothesis testing situation, but on the probability of obtaining a given score from the distributions of the calculated scores. The threshold values,$k_p^{C,S,D}$ and associated importance levels $p^{C,S,D}$ are determined by the following method (The C score with its associated threshold value and importance level is used as an example):

1. Calculate the C score for all M gene pairs, resulting in a total set of M different C_{ij} scores.

2. Draw m samples, s_i, from the total set of C scores, with each sample having a size $L << M$.

3. The threshold value k_p^C is determined as the average of the maximal values per sample: $k_p^C = \frac{1}{m} \sum_{i=1}^{m} \max_{\{s_i\}} C$

4. The associated importance level p is set as $p = 1/L$.

The same procedure is used for the threshold values and importance levels of the S and D scores. By choosing a common importance level p for the three scores, thereby getting an appropriate sample size L, it is possible to get comparable threshold values for the three types of scores.

Each pair of genes has a calculated Spearman correlation coefficient, $\rho_{ij,k}$, for each of the two conditions k, resulting in a pair of correlation coefficients, ($\rho_{ij,1}$, $\rho_{ij,2}$). As explained previously, the C_{ij} score is designed to increase the closer a pair of Spearman correlations are to the values $(-1,-1)$ or $(1,1)$, and to get small otherwise. Similarly, the S_{ij} score is designed to increase the closer the pair of Spearman rank correlations get to $(1,0)$, $(0,1)$, $(-1,0)$ or $(0,-1)$, and to become low otherwise, and the D_{ij} score is designed to increse the closer a pair of Spearman rank correlation gets to $(-1,1)$ or $(1,-1)$, and to become low otherwise. Eqs. (2.11), (2.12) and (2.13) thus ensure that for any combinations ($\rho_{ij,1}$, $\rho_{ij,2}$), at most *one* of the C_{ij}, S_{ij} and D_{ij} scores will be above their respective threshold k_p, returning only one type of link between genes *i* and *j*. Adjusting the importance value corresponds to increasing or decreasing the areas in Fig. 2.5, considered to reflect a C, S or D relationship, and it has to be set to a value that does not make the areas overlap. The importance value can be adjusted to a level that ensures a network size and link density that is suitable for further analysis.

2.8.4 Node Homogeneity

A node in the CSD network, representing a given gene i, can be characterized by its distribution of link types, t (C, S or D). This characteristic is called node homogeneity, H_i, and is calculated as shown in Eq. (2.14) [2]:

$$H_i = \sum_{t \in \{C,S,D\}} \left(\frac{k_{t,i}}{k_i} \right)^2 \tag{2.14}$$

where $k_{C,i}$, $k_{S,i}$ and $k_{D,i}$ denote the number of C, S and D-type links, respectively, and k_i is the degree of node i.

Chapter 3

Materials and Methods

This chapter is divided into two main sections. Section 3.1 gives an overview of the method study and development, while Section 3.2 describes the method and parameters used in the CSD analysis of rheumatoid arthritis. The CSD method was chosen for this analysis because it provides a more extensive and complete description of the co-expression changes between two conditions than other comparable methods: It introduces a new type of co-expression change (the condition-wise sign change of a strong correlation of the gene expression of two genes), as well as including both loss and conservation of strong correlations of gene pairs between the two conditions at study. The reader is referred to Section 2.8 for a complete description of the original method.

3.1 Method Study and Development

To respond to the issue of data set size limitations in the original CSD method [2], its sub-sampling strategy was modified so that instead of drawing only independent sub-samples, a higher number of less independent (more overlapping) sub-samples were drawn at random, until a fixed percentage of sub-samples had been drawn or discarded. The new, alternative sub-sampling algorithm was tested on smaller and smaller sub-sets of a bigger data set, and the results obtained were analyzed and compared to the results using the original sub-sampling algorithm on the same data. The objective of the method development study was to test if and to what extent allowing a higher dependence between the sub-samples would lead to a significant over- or underestimations of the gene pair correlation variabilities, $\sigma_{ij,k}$, in each data set *k*, thus uncovering whether or not this would allow data sets with fewer than 49 data points per gene to be used to create an informative CSD network.

3.1.1 Parameters Affecting σ_{ij}

It was essential to get an overview of the parameters that could affect the computed σ_{ij} values prior to the construction of an alternative sub-sampling algorithm . In order for the reader to better follow the different steps in the method study, the four most important

parameters affecting the calculated σ_{ij} are presented in this section; the *size of the data set*, N, the *number of sub-samples*, S, the *sub-sample size*, n, and the *dependence between the sub-samples*, given by the maximum number of data points that any two sub-samples are allowed to overlap, o.

A large enough S will ensure that σ_{ij} reaches a final, stable value, which will not be significantly changed by adding more sub-samples. A high number of sub-samples will also increase the probability of detecting confounding factors, which is the whole point of both sub-sampling procedures. If sub-samples are allowed to overlap, a higher S will lead to an increase in the average dependence between the sub-samples that are drawn.

The number of sub-samples that is possible to find is dependent on the combination of the other three parameters. If possible, increasing N by making more measurements per gene would be the easiest way of getting more independent sub-samples. Lowering the sub-sample size, n, will also allow more sub-samples to be drawn, but there is a limit to how small n can be without reducing the validity of the Spearman correlation: If there are too few data points per sub-sample, the Spearman correlation coefficient will not be meaningful anymore. The choice of sub-sample size, n, will obviously affect how many sub-samples of a given dependence that can be drawn in total. Increasing the dependence between sub-samples by allowing a higher o is a third way of increasing S.

Originally *no* dependence between sub-samples was allowed (see Section 2.8): No *pair* of data points was allowed in any two sub-samples. Allowing pairs of data points to appear multiple times, thus allowing some dependence between sub-samples, would allow more sub-samples to be drawn, but it could also shift the resulting σ_{ij} to become a worse estimate of correlation variability. With random drawing of sub-samples, this effect would be more apparent for smaller data sets, as fewer data points to draw from would increase the chance of drawing overlapping sub-samples. If allowing a given overlap between the sub-samples only resulted in a reasonably small change in $\sigma_{ij,k}$, one could still benefit from increasing the allowed sub-sample overlap, possibly getting good enough estimates for σ_{ij} even with quite small data sets.

As an illustration of the substantial change in the number of sub-samples that can be found with just a small increase in the permitted overlap, imagine a data set with $N = 9$ data points per gene and a sub-sample size $n = 3$. Using the original requirement of maximum one data point overlap, this will result in 12 sub-samples by using the original sub-sampling algorithm described in Section 2.8. Allowing a sub-sample overlap of 2 in stead of 1, will in this example increase the number of possible sub-samples with 84.

3.1.2 Data Collection

The gene expression data set used to test the new sub-sampling algorithm came from whole-blood samples of healthy individuals and was published as part of the GTEx project [38, 39]. The data set was found in their online database [12], and contained 338 data points per gene.

To investigate the consequences of varying the allowed sub-sample overlap on differently sized data sets, all 338 measurements of the first 100 genes of this data set was extracted from the full data set and used for computation of reference values. Smaller data sets were sliced out from this set in order to produce smaller data sets. The reason for only using the first 100 genes was to reduce the time consumption, but as this resulted in

4950 gene pairs for all the different data sets, it was considered to be enough genes for the purpose of studying the results of the method modifications.

The smaller data sets were created by extracting the first 300, 200, 100, 60, 50, 40, 30, 25 and 20 data points per gene, respectively. The data set sizes were chosen to be closer to each other when approaching and going below the lower size limit of the original CSD method, $N = 49$ data points per gene. This was done because any effects of higher maximum permitted overlap between sub-samples would be bigger with smaller data sets as a result of there being fewer valid sub-samples to draw.

3.1.3 The Alternative Sub-Sampling Algorithm

The original, deterministic sub-sampling algorithm was not easily changed to include a higher dependence between the drawn sub-samples. The alternative algorithm that was proposed and implemented in a script was therefore based on random drawing of sub-samples that were accepted or discarded based on how much overlap they had with previously drawn sub-samples. The alternative algorithm is presented in the following paragraphs.

A gene expression data set consists of *N* RNA measurements per gene. All the gene expression measurements of a given gene can be thought of as a vector, and elements in the same place in two genes' expression vectors come from the same tissue sample and may be compared. For each gene pair the new algorithm drew multiple sets of seven random expression vector entry numbers and put them together as new sub-samples. Each new sub-sample was tested for overlap with previously drawn sub-samples and accepted or discarded based on the chosen maximum overlap.

The "maximum sub-sample overlap", *o*, was defined as the maximum number of data points that a given sub-sample had in common with any other sub-sample. For example the set $\{1, 2, 3, 4, 5, 6, 7\}$ has $o = 1$ data point overlap with the set $\{7, 8, 9, 10, 11, 12, 13\}$, and the basis for the correlation calculations of these two sets would have been independent. Furthermore, the set $\{1, 2, 8, 9, 14, 15, 16\}$ has no more than *two* data points overlap with any of the other two sets, and would be allowed together with the previous two sub-samples if the maximum sub-sample overlap was set to $o = 2$. Put in another way: It has no subsets of *three* data points in common with any of the other sub-samples. Allowing a sub-sample overlap of $o = 3$, ensures that no subset of *four* data points will be found together in two sub-samples, and so on. Choosing $o = 6$ gives completely random dependence for sub-samples of size $n = 7$, as the only restriction is that no complete sub-sample is used twice.

The following algorithm was implemented for the calculation of σ_{ij} *for each gene pair i,j* in a data set with N data points per gene expression vector, a sub-sample size n, a maximum number of valid sub-samples per gene, S_{max}, and a maximum overlap of o data points per sub-sample:

1. Assign integers in increasing order from 1 to N to each measurement column in the data set.

2. Draw n random, unique integers in the interval [1, N] as the "current sub-sample". If more than 2 data points in the corresponding expression vector entries of genes

i or *j* has the value zero; discard the sub-sample, and repeat step 2 until a valid sub-sample is found or a given discard percentage is reached.

3. Compute the Spearman correlation coefficient for genes *i* and *j* from the expression vector elements given by the current sub-sample.
4. Repeat step 2, and proceed to step 3 if the current sub-sample has no more than o data points overlap with any previously drawn sub-sample. Discard the sub-sample if this requirement is not met, and repeat 2.
5. Repeat steps 1 to 4 until a given percentage of the sub-samples are discarded or an upper limit, S_{max}, for the number of valid sub-samples is reached.
6. Calculate the mean and the associated standard deviation of the mean, $\sigma_{ij,k}$, of the Spearman correlation coefficients from all valid sub-samples.

The decision not to calculate Spearman correlations of sub-samples with more than two zero entries per gene was to avoid problems with computing Spearman correlation values with too many tied ranks. Other tied values were not expected because the measurements had a high number of decimals. Gene pairs where the computed $\sigma_{ij,k}$ was based on too few accepted sub-samples (below a limit S_{min}) were filtered out from the data set before further analysis was performed.

3.1.4 Choice of Method Parameters

Discard Percentage

The discard percentage of the sub-sampling algorithm was set to 99%. This parameter was set after testing a few different values: Using $95\,\%$ resulted in significantly fewer valid sub-samples, while increasing the percentage to 99.9999% did not result in many more valid sub-samples, but increased the computation time significantly.

Sub-Sample Overlap, *o*

The test data sets described in Section 3.1.2 were tested with maximum permitted sub-sample overlaps of $o = 1, 2, 3$ and 6. Permitted sub-sample overlap of $o = 1$ corresponded to independent sub-samples like in the original method, only with a different way of finding them. Adjusting o to 2 or 3 allowed a small to medium sub-sample dependence, while $o = 6$ represented the highest possible permitted overlap between non-identical sub-samples.

Sub-Sample Size, *n*

Different sub-sample sizes were evaluated by Voigt, Nowick and Almaas [2], who found that the sub-sample size could not be reduced below $n = 7$. Based on this, the method study in this project was restricted to sub-samples of $n = 7$ data points, because the new algorithm was to be tested on very small data sets, and it was assumed to already be difficult to draw a sufficient number of sub-samples of $n = 7$ when testing with the smallest sub-sample overlaps.

The Number of Sub-Samples, *S*

When using the new sub-sampling strategy with various numbers of allowed sub-sample overlap, each σ_{ij} would be calculated from the given number of sub-samples, S, that had met the criteria of allowed overlap and number of zeros. With a low sub-sample overlap and/or few data points, relatively few sub-samples would be drawn before the discard percentage was reached. With too few sub-samples, the σ_{ij} that was calculated would be too dependent on the initial drawing pattern and not reliable as an estimate of correlation variability. When increasing the data set size or the allowed sub-sample overlap, the maximum number of sub-samples would get higher and higher, possibly leading to very time-consuming calculations.

Taking these factors into consideration, it was necessary to set both the upper limit, S_{max}, for the number of sub-samples that the algorithm was allowed to draw per gene pair before it was reasonable to stop, and the lower limit of sub-samples, S_{min}, where a σ_{ij} based on fewer sub-samples than this limit was not used in the further tests of the method.

Since the value of σ_{ij} was computed continuously for each new sub-sample (see Section 3.1.6), one could think of σ_{ij} as a signal that would vary according to the number of sub-samples, before it stabilized after a given number of sub-samples had been drawn. The lower limit for the number of sub-samples, S_{min}, was set at the lowest number of sub-samples that would produce a σ_{ij} output that was relatively stable compared to the values obtained with a lower *S*. The upper limit for the number of sub-samples, S_{max}, was set at a level where the σ_{ij} signal had been stable for a significant number of sub-samples already, ensuring as reliable values as possible within a reasonable computation time.

The σ_{ij} "signal" was measured continuously for 28 gene pairs, increasing S_{max} gradually from 1 to 200. The parameters S_{min} and S_{max} were set to 50 and 200, respectively, based on the results from this pre-analysis. An evaluation of the limits is given in Chapter 4. The 28 gene pairs were all pair-wise combinations of eight selected genes. The genes were chosen carefully so that gene pairs with both positive and negative, as well as strong, medium and weak correlations were present in the selection. The identity of the eight genes used are given in Appendix A2.

3.1.5 Evaluation of the Alternative Algorithm

The value σ_{ij} was computed using the new sub-sampling algorithm on the 4950 gene pairs on each of the ten data sets of different size N, with four different values of the maximum permitted sub-sample overlaps o. This gave one measurement series per combination of N and o. These were denoted σ_o^N, and compared to a common series of "true" reference data. The reference data series σ_1^{338} (using the full data set of 338 samples per gene, and only independent sub-samples, $o = 1$) was computed independently from the first measurements, and defined as the "true standard deviations of the means", denoted σ^T.

Stability of the Computed σ Values

The stability of the results from the new sub-sampling algorithm were investigated by computing σ_1^{338} and σ_6^{338} 100 separate times for the same 28 gene pairs used for deciding the S limits in Section 3.1.4. The distributions of σ_o^{338} values for each gene pair were

visualized as boxplots in order to see the spread of the values. The relative variation in σ for these gene pairs was found as the coefficient of variation, CoV, which measured the variance of each distribution divided by the mean value of σ for each gene pair.

Evaluation of the Effect of Sub-Sample Dependence

The comparison between the σ_o^N series and the reference series was done by dividing $\sigma_{o,ij}^N$ for each gene pair *i,j* by its corresponding reference value σ_{ij}^T. The distribution of each σ_o^N/σ^T series was plotted as boxplots with whiskers set to span the following ranges: $W_{upper} = Q3 + 1.5 \cdot IQR$ and $W_{lower} = Q1 - 1.5 \cdot IQR$. The different boxplots were analyzed in order to investigate how the overall σ distributions were affected by the combination of the various parameters, o, N and S.

The distributions of the σ_o^N values were also compared to the distributions of σ^T using a two-sample *t*-test. This was done to check at which combinations of o and n the σ_o^N series were so different that they could be assumed to come from a different underlying distribution. For each o, a *t*-test was performed on the σ_o^N series with decreasing values of N until the distributions were found to be significanly different, using a significance level of 0.05.

Investigation of Outliers

To investigate if there was any system to what gene pairs had the biggest variation in their computed σ values, and therefore ended up as outliers in the boxplots, some of the σ_o^N/σ^T distributions and associated outliers were investigated further. As stated previously, the boxplot whiskers were set to have a length of $1.5 \cdot IQR$, so that any σ/σ^T value above the value $Q3 + 1.5 \cdot IQR$ or below $Q1 - 1.5 \cdot IQR$ were defined as outliers.

The total number of outliers, as well as the associated percentage of gene pairs ending up as outliers, were found for the σ_o^N/σ^T distributions with the following (N, o) combinations: (338,1), (300,1), (100,1), (338,6), (100,6), (50,6) and (30, 6). Furthermore, a comparison was done to find the number and percentage of common outlier gene pairs between two and two of the chosen σ_o^N/σ^T distributions. The distributions were chosen so that the combinations of distributions with high/high, high/low and low/low N values in combination with high/high, high/low and low/low o values were compared. This was done to ensure that outliers from different types of distributions were investigated without having to investigate all the 29 possible distributions. The latter would have been too time consuming.

The most extreme outlier gene pairs were also studied in more detail. Four of the σ_o^N/σ^T boxplots were chosen for this final outlier analysis: σ_1^{338}/σ^T, σ_1^{100}/σ^T, σ_6^{338}/σ^T and σ_6^{30}/σ^T. These represented big and small data sets, and big and small o values. The five most extreme outliers were identified, as well as the five gene pairs lying closest to the medians of each distribution. The overall correlation coefficient ρ and the associated σ^T for these gene pairs were found from the σ^T series and compared to each other.

Evaluation of the Reference Data

To test the validity of the σ^T series as a good representation of the true correlation variability within the data set, the σ^T values for all the 4950 gene pairs were compared to the corresponding series of values obtained by using the original sub-sampling algorithm on the same gene pairs and 338 data points. The data series found using the original algorithm was denoted $\sigma_1^{338}(orig)$. The distribution of the pair-wise ratios $\sigma^T/\sigma_1^{338}(orig)$ was visualized as a boxplot equivalent to the ones described in previous paragraphs. In order to evaluate if the two algorithms gave equally large variations in the resulting σ_{ij} values that they produced, a similar boxplot was also made for the distribution of $\sigma_1^{338}(orig1)/\sigma_1^{338}(orig2)$ where $\sigma_1^{338}(orig1)$ and $\sigma_1^{338}(orig2)$ were the results obtained from two different runs of the CSD-CS software on the same data set. Because the CSD-CS software is deterministic in the sense that it draws the exact same sub-samples each time if all the parameters are the same, the different runs were performed setting the software parameter *randomSeed* to 3 and 12 respectively. This parameter decided a starting point for the sub-sampling process, ensuring that the exact same sub-samples were not redrawn on the second run.

3.1.6 Software Implementation of the Alternative Algorithm

In order to perform the calculations of Spearman correlation coefficients and standard deviations of the means of sub-sample correlations for a given data set with different maximum sub-sample overlaps o, I wrote a python script. The script imported a gene expression data set, created all possible pairs of genes from the set, and computed the Spearman correlation coefficients for each gene pair. The script further implemented the new sub-sampling algorithm (see Section 3.1.3) to find standard deviations of the means of a given number of sub-sample correlations. Two versions of the script was made: the first version computed σ_{ij} with maximum sub-sample overlaps of $o = 1, 2$ and 3, while the second version used sub-samples with maximum sub-sample overlap $o = 6$. Both versions of the script are found in Appendix A4.

For each value of o, the standard deviations of the means of up to S_{max} valid sub-samples were calculated on the run for each new sub-sample, either until S_{max} valid sub-samples had been found or a given percentage of all drawn sub-samples had been discarded. Welford's algorithm for the calculation of running variance [26] was implemented for the variance calculations because it ensured a high numeric stability and precision of the results. All results for one gene pair were calculated before proceeding to the next gene pair.

In the first version of the script, all possible combinations of either pairs, triplets or fours (for permitted overlaps of $o = 1$, $o = 2$ and $o = 3$, respectively) of data points from the currently drawn sub-sample were created for each drawn sub-sample. If none of them had been used before, the sub-sample was used for correlation and variance calculations, and all possible orders of the new pairs, triplets or fours from this sub-sample were added to a dictionary structure of used combinations. This dictionary was used as a reference for the next sub-sample and so on. If any of the pairs/triplets/fours in a drawn sub-sample were found in the dictionary already, the sub-sample was discarded.

The second version of the script was simplified for the calculation of ρ_{ij} and σ_{ij} when the sub-sample overlap was set to $o = 6$, since storing all combinations of 6 numbers in

a dictionary structure was a cumbersome procedure. Instead each sub-sample was transformed to a *set* structure (unordered collections of objects), and all used sub-samples were stored in a set of sets. If any new sub-sample was not in the set of previously used sub-samples, the new sub-sample was accepted.

To reduce the overall computation time, the possibility of running parallel computations was included in both versions of the script, now distributing the gene pair computations on the available amount of CPUs on the computer.

While the first version of the script was only used for small data sets in the method development part, the second version of the script was later used for a full size gene expression data set. The memory demands of the second version of the script were therefore reduced, the run time was further improved and a progress bar was added so that one could more easily keep track of the process. The script was furthermore made more robust against errors, both by including some exception handling that would allow the program to continue if something went wrong during one of the calculations, just writing an error message in the result file, and by saving the results at user specified intervals, so that all computations did not have to be rerun if the script was to stop.

As the parallelization function that was used did not have a built-in way of saving results on the go, the second version of the script was written so that it divided the list of gene pairs into a given number of chunks, computed the data for each chunk at a time, and wrote the results to file before the next chunk was fed to the parallelization function. This reduced the RAM demands and made the script more robust to failures, by reducing the potential amount of lost results. The script was set to keep track of all successfully processed chunks of gene pairs, and return the identity of any non-successful chunks. At a rerun the script would access this list and only redo the computations on the chunks of gene pairs that were not successfully processed on previous runs.

In addition to optimizing the existing code, the computation time was sped up by using multiple computers. The second version of the script was therefore made so that the user could fill in the number of machines and the number of cores per machine, in order for the script to divide the gene pairs into groups of corresponding size. For each run on a new machine, the user then had to write the correct machine number into the script.

The final output of both scripts was a text-file containing a table of gene pairs and their associated Spearman correlation coefficients, ρ_{ij}, the associated σ_o^N or $(\sigma_o^N)^2$ (by choice) for either $o = 1$, $o = 2$ and $o = 3$ *or* $o = 6$ (depending on the version), as well as the number of sub-samples, S, from which each σ_o^N was calculated.

The first version of the script was run on the test data set of 100 genes and $N = 338$ measurements per gene. With S_{max} set to 150, the computations took less than 1.5 hours running on 24 parallel $3.47\,\mathrm{GHz}$ Intel Xeon X5690 cores with $126\,\mathrm{GB}$ RAM on Ubuntu 16.04. The second version of the script was run with a data set of roughly $25\,000$ genes and 152 measurements per gene as input. With S_{max} set to 150, the computations took about 7 days running on a system consisting of 160 parallel $2.4\,\mathrm{GHz}$ Intel Xeon E25-2680 cores distributed on 7 virtual machines ranging from $62.9\,\mathrm{GB}$ to $189\,\mathrm{GB}$ RAM, running Ubuntu 16.04.

3.1.7 CSD-CS Software for Original Sub-Sampling Algorithm

The CSD-CS software developed by M.O. Helland in his master project at NTNU, 2017 [40] was used for the calculation of σ_{ij} values using the original sub-sampling algorithm with only independent sub-samples. The CSD-CS program has implemented the original sub-sampling approach of the CSD method described in section 2.8, and can produce a table of gene pairs and their associated ρ_{ij}, and σ_{ij}^2. Worth noting is that this implementation of the CSD method does use *all* valid sub-samples that can be found with the original sub-sampling algorithm, but limits the number of sub-samples to a given number equivalent to the parameter S_{max} described in the previous sections. The software is available on GitHub (https://github.com/magnusolavhelland/CSD-Software).

3.2 CSD Analysis of Rheumatoid Arthritis

3.2.1 Data Collection

The data used in the analysis of rheumatoid arthritis was published as a part of a study by Guo, Walsh, Fearon, Smith, Wechalekar, Yin et al. in 2017 [41], and accessed through the Gene Expression Omnibus (GEO) database (accession number GSE89408). The data set consisted of total RNA sequencing measurements (25049 genes) of joint synovial biopsy samples of 152 subjects with rheumatoid arthritis and 28 healthy control subjects. One tissue sample per subject had been analyzed without replicates. The data set also contained measurements from 38 individuals with other types of arthritis, all of which were excluded from the CSD analysis.

3.2.2 Network Construction

Gene expression data from the synovial fluid of patients with RA and healthy control subjects were used to generate a CSD network based on the method described in Section 2.8. The RA data set was big enough for the original sub-sampling algorithm of the CSD method, while the reference data set was too small: Using the original sub-sampling strategy described by Voigt, Nowick and Almaas [2], 28 data points per gene would not lead to a large enough number of sub-samples for calculating the internal variability in the data set.

The original sub-sampling strategy was therefore used only for the RA data set, while the modified sub-sampling strategy described in Section 3.1.3 was used for the reference data set. When calculating the $\sigma_{ij,ref}$ of each gene pair in the reference data set, sub-sample overlaps of up to 6 data points were permitted instead of just 1, resulting in a higher number of sub-samples of random dependence instead of a too low number of independent sub-samples. These sub-samples were used for the calculation of the variability within the reference data set.

The results of the method study, which will be presented and analysed in Chapter 4, indicated that with the available methods, this approach would give the best representation of the internal correlation variance in a data set as small as $N = 28$. Other alternatives would have been to ignore the internal variance in the reference data set, or to not perform the analysis at all. One should however be aware that a substantial number of the estimated

$\sigma_{ij,ref}$ values will be less precise representations of correlation variability in the data set than what could have been found with more data. Although not optimal, the noise that was expected in the data as a result of this approach was accepted as the best option available.

In accordance with the results of the method development study the maximum number of sub-samples were set to $S_{max} = 150$, while the lower limit was set to $S_{min} = 100$. Gene pairs that did not meet this criterion were filtered out from the results after the correlation and variance calculations. The large majority of the gene pairs had reached S_{max}, so the gene pairs that were discarded due to a low number of valid sub-samples typically had reached S values much lower than S_{min}, due to a high occurrence of zeros in the gene expression data for one or both genes. Gene pairs with sub-sampling or correlation computation errors (for instance by too many tied ranks in the full data set) were also filtered out from the data set. RNA coding genes were kept in the network even if they do not code for protein products as they might have regulatory functions.

The two correlation/variance data sets were finally used to create a CSD network as described in Section 2.8.

3.2.3 Biological Process Enrichment Analysis

In order to establish if the network was enriched with genes related to specific biological processes, a GO (gene ontology) enrichment analysis was performed using the GO Enrichment Analysis tool of the Gene Ontology Consortium [42, 43, 44] available online (http://www.geneontology.org/page/go-enrichment-analysis). The category "Biological Process" was selected for the enrichment search. The GO Enrichment Analysis tool identifies a biological process as *enriched* in the group of input genes if genes related to the given process is under- or over-represented in the network compared to what would have been expected by chance (with a correction for multiple comparisons, including only results with a false discovery rate lower than 0.05). The results also return the number of expected genes in the relevant categories, as well as an enrichment factor, or *fold enrichment*, indicating how big the over- or under-representation is.

The biological process enrichment analysis was performed on the full network obtained with an importance level of 10^{-5}, as well as the three groups of all nodes containing C-, S- or D-type links respectively.

3.2.4 Software for CSD Analysis of RA

The Spearman correlation coefficients and variances for all gene pairs in the RA data set were found using the CSD-CS software described in Section 3.1.7, developed by M.O. Helland in his master project at NTNU, 2017 [40]. The software was given the following parameters: sub-sample size: 7, maximum number of sub-samples: 200, randomSeed: 4. Because the RA data set was bigger, an upper limit for the number of sub-samples was increased to 200.

The corresponding values for the reference data set was found using the Python script (second version) described in Section 3.1.6 and given in Appendix A4. The filtration of the correlation/variance-data obtained by the Python script, was done using a short, self-written Python script given in Appendix A5, and both the RA and reference correlation/-variance files were sorted using the *sort* command of the Linux terminal.

From the correlation/variance-data, the C-, S- and D-scores were found, and a CSD network was constructed using the CSD software by Voigt, Nowick and Almaas [2], available on GitHub (https://github.com/andre-voigt/CSD). The sample size L was set to 10^4, 10^5 and 10^6 on three separate runs of the software. This corresponds to importance values p of 10^{-4}, 10^{-5} and 10^{-6}, respectively.

The CSD network of gene pairs and associated CSD link types was visualized and analyzed using Cytoscape (v.3.5.0) [45]. Cytoscape is a freely available, open source software for network data integration, analysis and visualization (www.cytoscape.org).

The ratio of C-, S- and D-links on each node, as well as the node homogeneity, H, were found using another self-written Python script found in Appendix A6. These values were imported into Cytoscape together with the network.

Chapter 4

Results and Analysis

The results in this chapter are divided in two parts according to the two different parts of the study. Section 4.1 contains the results of the method development study and Section 4.2 presents the results from the CSD analysis of rheumatoid arthritis. The results are also analyzed in each respective section.

4.1 Results: Development of the CSD Method

4.1.1 Impact of the Number of Sub-Samples, *S*

The main difference between how the new and the original sub-sampling algorithms were built was that the new algorithm drew random data points for each sub-sample and accepted or discarded the sub-samples based on the overlap with previously drawn sub-samples. In contrast, the original algorithm found new valid sub-samples in a systematic approach, not wasting time on drawing sub-samples that would later be discarded. The consequence of drawing the sub-samples in a random fashion was that for each new valid sub-sample it became increasingly more difficult to draw more valid sub-samples. An important question was therefore how many valid sub-samples that could be drawn within a reasonable amount of time when using the new sub-sampling approach.

Table 4.1 shows the average number of valid sub-samples (each of 7 data points), S_{new}, that the new script was able to identify from the data sets of different sizes, N, with maximum sub-sample overlaps of $o = 1, 2, 3$ and 6. Table 4.1 also has a column called S_{orig}, showing the maximum possible number of independent sub-samples found using the original, systematic sub-sampling approach. I got the S_{orig} data from by co-supervisor, André Voigt, who was also the main author of the article on the original CSD method (Voigt, A. 2018, written communication, May 10th). The reason why the number of sub-samples seems to stop at 200 is that $S_{max} = 200$ was set as an upper limit for the number of sub-samples to be drawn (see next paragraph). This was also set as a cap on the S_{orig} data to avoid confusion. The reader should therefore be aware that for the data set sizes $N = 200, 300$ and 338, Table 4.1 does not display the true differences in drawing

capacity for the different maximum overlap values, as 200 sub-samples was reached for all the different o values, and the computations were stopped when this limit for S was reached. As shown in Table 4.1, the new algorithm could find less than half the number of independent sub-samples that could be obtained with the original, deterministic method in most of the cases. At $N = 60$, the original method can only find 28 sub-samples, which at first might look like a mistake, since it is lower than the number of sub-samples at $N = 50$. This is not a mistake, but a numerical coincidence caused by the combination of $n = 7$, $N = 60$ and the steps of the sub-sampling algorithm. At $N = 59$, the original sub-sampling algorithm will find 46 sub-samples, and at $N = 61$ it will find 45, showing that this is a very local phenomenon. For N smaller than 50 data points per gene, the data sets were too small for the original sub-sampling algorithm to have a big effect, and the two algorithms therefore have equal performance when it comes to finding independent sub-samples.

Table 4.1: The average number, S_{new}, of valid sub-samples (7 data points) that the new sub-sampling algorithm was able to find from data sets of size N, with maximum sub-sample overlaps of $o = 1, 2, 3$ and 6. The numbers are rounded to integers. The number of independent sub-samples found using the original, deterministic sub-sampling algorithm, S_{orig}, is given as a reference. NB! *The highest number of sub-samples in the table is 200 because S was limited to $S_{max} = 200$. **The S_{orig} value for $N = 60$ is correctly lower than at $N = 50$ due to a numerical coincidence. See explanation in the text.

N	S_{new}, $o = 1$	S_{new}, $o = 2$	S_{new}, $o = 3$	S_{new}, $o = 6$	S_{orig}, $o = 1$
338	200*	200*	200*	200*	200*
300	200*	200*	200*	200*	200*
200	200*	200*	200*	200*	200*
100	62	200*	200*	200*	138
60	22	162	200*	200*	28**
50	15	93	200*	200*	34
40	9	47	200*	200*	8
30	5	19	119	200*	5
25	4	11	54	200*	4
20	3	5	20	200*	3

A related question was how many sub-samples were *necessary* in order to obtain a stable σ value as an estimate of the correlation variability in the data set. Fig. 4.1 shows the σ_1^{338} signal for 28 gene pairs as a function, *g(S)*, of the number of sub-samples, S. As described in Chapter 3, the gene pairs were selected so that both positive and negative, as well as strong, medium and weak correlations were represented. For S between 1 and 50, there is much fluctuation in each *g(S)* "signal": for one of the gene pairs the signal reaches values almost ten times higher than its final value. At $S = 50$, the signal plots start to descend slowly towards a plateau where the values are not significantly changed as S increases. Between $S = 150$ and $S = 200$ the plots are almost horizontal, meaning that additional sub-samples do not change the σ values very much in this S range. $S = 50$ and $S = 200$ were therefore chosen as the upper and lower limits of S, S_{min} and S_{max}, in the further method study. An equivalent plot for σ_3^{338} as a function of S is found in Appendix

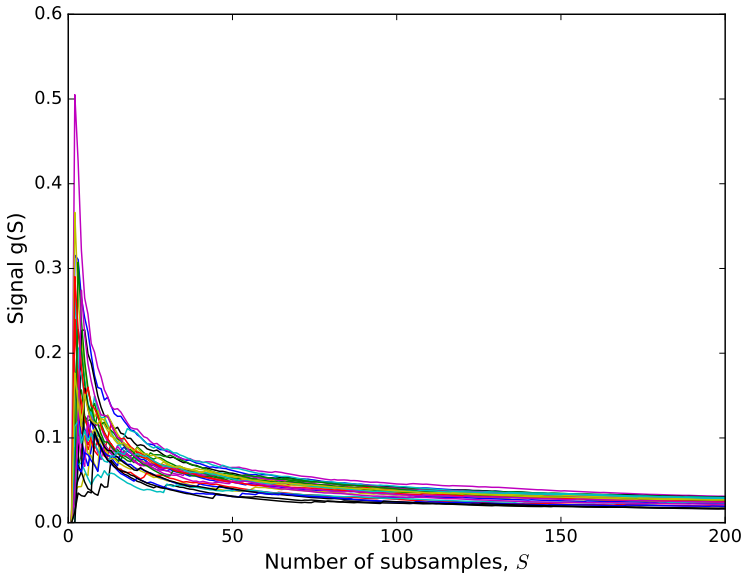


Figure 4.1: The function *g(S)* represents the σ_1^{338} signal for 28 gene pairs (with high, medium and low, as well as both positive and negative correlations) as a function of the number of sub-samples from which they are calculated, S. Each graph represents the signal of one gene pair.

A7. This plot is nearly identical to Fig. 4.1, which justifies using the same limits with increased overlap between sub-samples.

Because S_{max} was set to 200, the end values of the signals in Fig. 4.1 correspond to the "true" standard deviations used as reference in the further calculations, σ^T, for these gene pairs. It is however difficult to distinguish the different values from each other in the plot. In order to get a better overview of the final σ^T values in the data set, the distribution of σ^T for all 4950 gene pairs in the data set is given as a histogram in Fig. 4.2. The range of all the σ^T values is divided into 50 bins of equal size, and the frequency of gene pairs with σ^T values within each of the bins determines the height of the corresponding columns.

The signals in Fig. 4.1, as well as the σ^T values visualized in Fig. 4.2, are all outputs from a single run of the random sub-sampling script, and even if each signal ends up at a stable value, it would not necessarily end up at the *same* value on a second run with new sub-samples. Fig. 4.3a shows boxplots of the σ_{ij} values ($N = 338$ and $o = 1$) at $S = 200$ obtained from 100 independent runs of the random sub-sampling script on the same 28 gene pairs that were used for Fig. 4.1. The figure shows that there are indeed variations in the final σ_{ij} values, but for the majority of the runs, these variations are reasonably small. There are of course differences between the gene pairs, but in general the boxes have interquartile ranges of roughly 0.002, meaning that the distance from the median to the first and third quartile (the upper or lower side of the box) will be more or less close to 0.001 (depending on where in the box the median is found). The whiskers span a range of approximately 0.005-0.006, meaning that a point lying at the outer end of a whisker could deviate approximately 0.003 from a given median. Fig. 4.2 shows that the majority

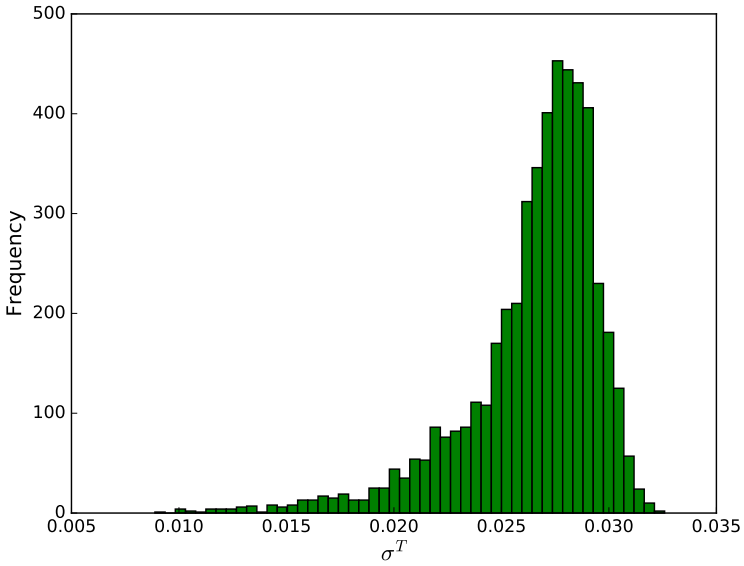
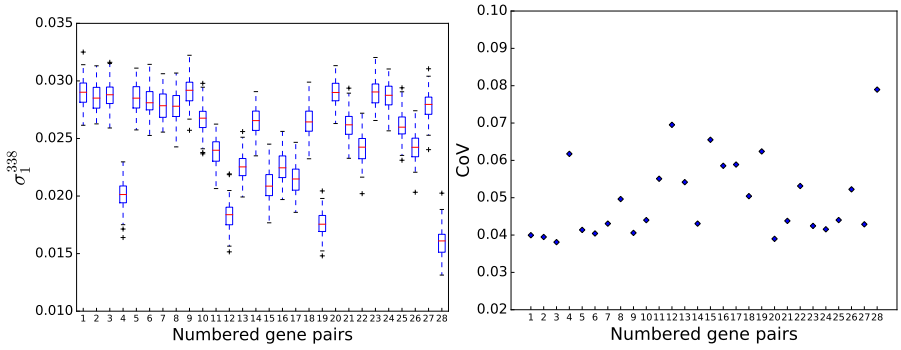


Figure 4.2: The distribution of the σ^T values for all 4950 gene pairs in the data set is visualized as a histogram with 50 bins representing value intervals for σ^T. The height of each column represents the number of gene pairs with a σ^T in the corresponding interval.

of the σ^T values lie in the interval $[0.025, 0.030]$. With a median value of $\sigma_{ij} = 0.025$, a deviation of 0.001 from itself equals a 4% change, while a deviation of 0.003 equals a 12% change.

There are, however, a significant number of gene pairs with σ values that are much lower than 0.025. For these gene pairs, an absolute deviation of 0.001 or 0.003 will represent a much higher percentage deviation than for the majority of the gene pairs. Fig. 4.3b shows the coefficient of variation, CoV, for the same 28 gene pairs. The CoV represents the relative variation in each of the 28 σ_1^{338} series, given as the standard deviation of the 100 σ values divided by the mean of the same values. Comparing Fig. 4.3a and Fig. 4.3b, one can clearly see that gene pairs with a low σ^T have higher CoVs and vice versa. This means that even two very uniformly correlated gene expression series are still so noisy that the final σ values from multiple computations have an absolute variation comparable to the more inconsistently correlated gene expression series. A plot equivalent to Fig. 4.3a, only with $o = 6$ as the only change, is given in Appendix A3. The two plots have no major differences; they have comparable interquartile ranges and whisker ranges for all the gene pairs. This indicates that for the given data set size and number of sub-samples, the values found with $o = 6$ are equally stable as with $o = 1$. This will be investigated further later in this chapter.



(a) **(b)**

Figure 4.3: Stability of the computed σ^T values: Fig. (a) shows boxplots of the σ_{ij} values ($N = 338$ and $o = 1$) at $S_{max} = 200$ obtained from 100 independent runs of the random sub-sampling script on 28 gene pairs, *ij*, with various correlation coefficients. Fig. (b) shows the coefficient of variation (CoV) for the same 28 gene pairs. The gene pairs are numbered for easy comparison between (a) and (b).

4.1.2 The Combined Effects of *N*, *S* and *o*

Figs. 4.4 - 4.7 show boxplots of multiple σ_o^N/σ^T series with maximum overlaps of $o = 1, 2, 3$ and 6, respectively. A maximum overlap $o = 1$ represents independent sub-samples, while a maximum overlap $o = 6$ represents the maximum dependence between sub-samples. Each series represents running the new sub-sampling algorithm on all 4950 gene pairs in data sets of different sizes N (338, 300, 200, 100, 60, 50, 40, 30, 25, 20) with $S_{max} = 200$. Data set sizes not represented in the different plots means that the minimum number of sub-samples, $S_{min} = 50$, was not obtained for any of the gene pairs for that combination of N and o.

Fig. 4.4 shows boxplots of σ_1^N/σ^T for $N = 338, 300, 200$ and 100. The pair-wise ratios of σ_1^{338} against σ^T (σ_1^{338}) are based on two different runs of the new sub-sampling algorithm. This gives a boxplot with a median of 1, meaning that the median of the 4950 σ_1^{338}/σ^T ratios represents a gene pair with equal σ values at both runs. The first, and third quartiles, Q1 and Q3, only span approximately 1.0 ± 0.05, and combined with the fact that the median is exactly 1, this shows that 50% of the gene pairs had σ values that were no more than 5% different from the σ values from the indepentent, second computation. Apart from some outliers, most of the remaining σ_1^{338} values deviated roughly between 5% and 20% from the reference value with equal parameters. The most extreme outlier gene pair deviated almost 40% from itself on the two different runs. This first boxplot illustrates the variation in σ solely due to different choice of sub-samples. Given that the σ^T values found with my script are equivalent to the values obtained using the original algorithm (which will be discussed later), any combinations of N and o resulting in σ_o^N/σ^T boxplots similar to the first boxplot from the left in Fig. 4.4 would therefore indicate that the parameters are justifiable to use in a CSD analysis.

The boxplots of σ_1^{300}/σ^T and σ_1^{200}/σ^T in Fig. 4.4 are more or less equal to the one

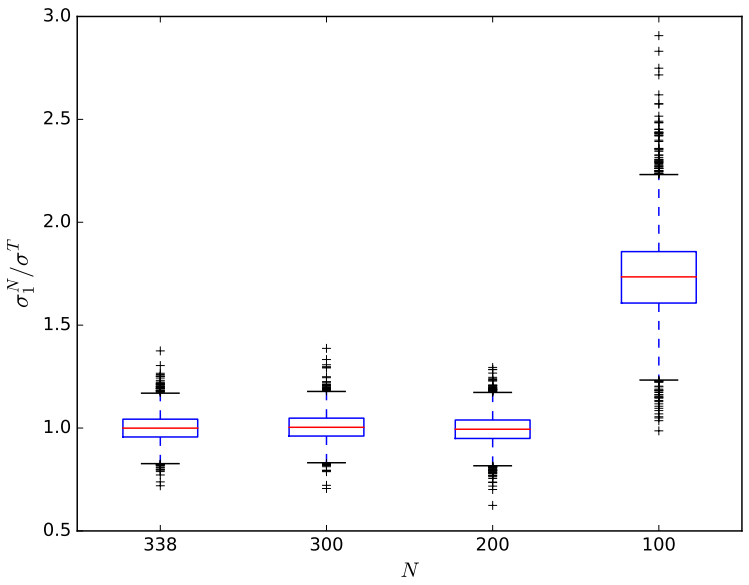


Figure 4.4: Boxplots of four distributions of the relative standard deviations of the means, σ_o^N/σ^T, coming from calculations of σ_o^N for 4950 gene pairs, using the random sub-sampling approach with the parameters $o = 1$, $N = 338, 300, 200$ and 100, and $S_{max} = 200$. The corresponding σ^T values come from an independent run of the random sub-sampling script with $o = 1$, $N = 338$ and $S_{max} = 200$, and are used as "true" reference values for each gene pair.

for σ_1^{338}/σ^T, but the σ_1 values show a collective, large positive deviation from their corresponding σ^T values when $N = 100$. The median of the ratios is around 1.75, meaning that the σ middle value increased with 75% of its true value. Apart from the lowest outliers, none of the σ_1^{100}/σ^T ratios are close to 1. The box itself is also much wider, and the whiskers range from around 1.25 to around 2.25. The upper outliers almost reach 3.0, meaning that a few gene pairs had σ_1^{100} values almost 300% as big as σ^T.

What could be the cause of the huge change in the results when going from $N = 200$ to $N = 100$ in Fig. 4.4? Table 4.1 shows that $N = 100$ is the highest data set size where the strategy of randomly drawing sub-samples *fails* to draw 200 independent ($o = 1$) sub-samples. At this data set size, only 62 sub-samples were drawn on average.

Fig. 4.5 shows boxplots of σ_2^N/σ^T for $N = 338, 300, 200, 100, 60$ and 50, and Fig. 4.6 shows boxplots of σ_3^N/σ^T for $N = 338, 300, 200, 100, 60, 50, 40, 30$ and 25. The higher sub-sample overlap, the smaller data sets can be used while still obtaining $S > S_{min}$, and the more data sets also reaches $S = S_{max}$. With this in mind, one can see that Figs. 4.5 and 4.6 show trends similar to Fig. 4.4. For data set sizes where S_{max} was reached (down to and including $N = 100$ and $N = 40$ in Figs. 4.5 and 4.6 respectively), the medians lie close to 1.0, but when S approaches S_{min} the medians of the plots start to increase significantly, and the boxes, whiskers and outliers span much larger ranges.

Looking more closely into Figs. 4.5 and 4.6, it is possible to see a small change in the boxplots *before* the decrease in S towards S_{min}; a much smaller increase of the whiskers

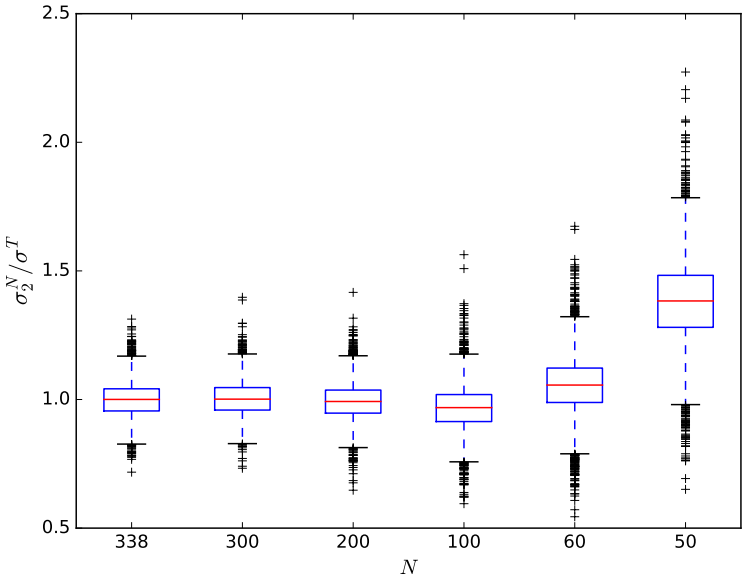


Figure 4.5: Boxplots of six distributions of the relative standard deviations of the means, σ_o^N/σ^T, coming from calculations of σ_o^N for 4950 gene pairs, using the random sub-sampling approach with the parameters $o = 2$, $N = 338, 300, 200, 100, 60$ and 50, and $S_{max} = 200$. The σ^T values come from an independent run of the random sub-sampling script with $o = 1$, $N = 338$ and $S_{max} = 200$, and are used as true reference values for each gene pair.

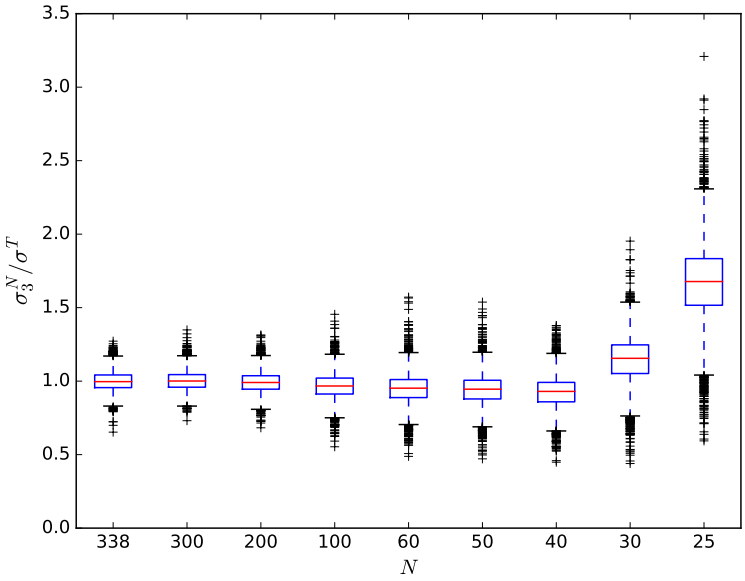


Figure 4.6: Boxplots of nine distributions of the relative standard deviations of the means, σ_o^N/σ^T, coming from calculations of σ_o^N for 4950 gene pairs, using the random sub-sampling approach with the parameters $o = 3$, $N = 338, 300, 200, 100, 60, 50, 40, 30$ and 25, and $S_{max} = 200$. The σ^T values come from an independent run of the random sub-sampling script with $o = 1$, $N = 338$ and $S_{max} = 200$, and are used as true reference values for each gene pair.

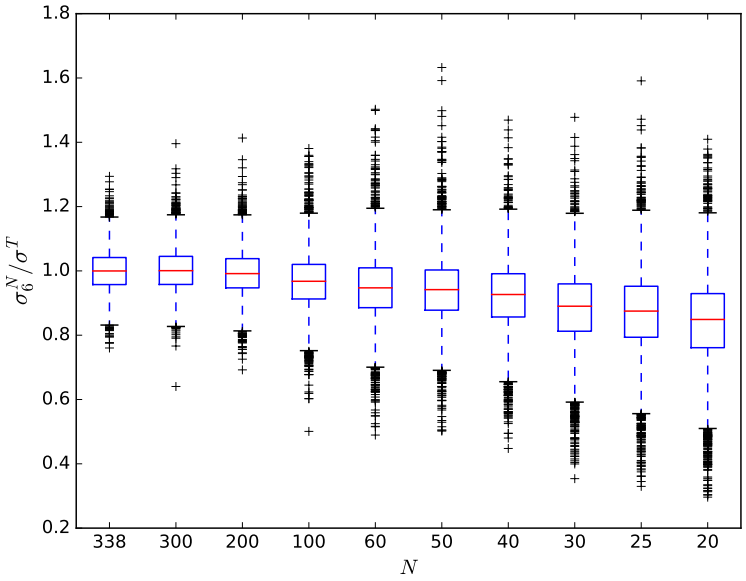


Figure 4.7: Boxplots of ten distributions of the relative standard deviations of the means, σ_o^N/σ^T, coming from calculations of σ_o^N for 4950 gene pairs, using the random sub-sampling approach with the parameters $o = 6$, $N = 338, 300, 200, 100, 60, 50, 40, 30, 25$ and 20, and $S_{max} = 200$. The σ^T values come from an independent run of the random sub-sampling script with $o = 1$, $N = 338$ and $S_{max} = 200$, and are used as true reference values for each gene pair.

and interquartile ranges of the boxes when going from left to right in both figures. These increases are accompanied by medians that are *lowered* slightly. These changes are unrelated to S being below S_{max}, and are not present in Fig. 4.4, where all sub-samples are independent. This trend is seen much more clearly in Fig. 4.7, showing boxplots of σ_6^N/σ^T for N as low as 20. Here $S = S_{max}$ for all data set sizes, and the sub-samples are allowed to have a maximum dependence. For N down to 100, the boxplots do not deviate much from the reference plot σ_1^{338}/σ^T in Fig. 4.4, but the smaller the data sets get, the more the medians are lowered, and the bigger ranges do the boxes, whiskers and outliers span.

Looking at Figs. 4.4 - 4.7 together, one can see a general trend where lower N and S values (fewer data points and fewer valid sub-samples) lead to a higher and higher overestimation of σ regardless of o, while higher o values (increased dependence between sub-samples) lead to an underestimation of σ when S is not a factor. Furthermore, the overestimation of σ when lowering S towards S_{min} seems to be much stronger than the underestimation of σ when increasing o to its extreme value of 6. To further test the interaction between S and o, S_{max} was decreased to 150, and σ_6^N/σ^T was re-calculated for $N = 100, 60, 50, 40, 30, 25$ and 20, which were the data set sizes where the effect of $o = 6$ was most significant with $S_{max} = 200$. The results are found in Fig. 4.8. One can see that in contrast to Fig. 4.7, the medians of the box-plots are *above* 1 for N as low as 25, while the median of the boxplot is actually very close to 1 at $N = 20$, and the boxplot itself is comparable to the boxplot of σ_2^{60}/σ^T in Fig. 4.5.

While Figs. 4.4 - 4.7 show clear trends for σ when varying N and o, the boxplots do not explicitly show what the σ distributions look like or how similar they are. Table 4.2 shows the *t*- and p-values resulting from a two-sample *t*-test of several σ_o^N series against the σ^T series. The zero hypothesis of each *t*-test was that the two series came from the same underlying distribution, and the alternative hypothesis was that the two series came from two different distributions. Any p-value below 0.05 would indicate that the zero hypothesis should be rejected. The results from the *t*-tests show that just by decreasing N by a bit more than one third, from 338 to 200, any σ_o^N series becomes so different from σ^T that they appear to come from two different underlying distributions. This result is independent of o, and indicates that for instance the distribution of σ_6^{300} is more similar to the σ^T distribution than the σ_1^{200} distribution is. This illustrates the impact of the available data points on the results.

4.1.3 Evaluation of σ^T as Reference Value

In order for Figs. 4.4 - 4.7 to be meaningful representations of the new sub-sampling algorithm, it was essential to test if σ^T was a good representation of the "true" internal variability in the data set, and thus could be used as reference values for other σ_o^N's. The σ^T (equal to σ_1^{338}) computed using my own implementation of the new sub-sampling algorithm, was therefore plotted against σ_1^{338} obtained using the CSD-CS software (see Section 3.1.7), which had implemented the original sub-sampling algorithm, on the same data set.

A boxplot of the pair-wise ratios $\sigma_1^{338}(new)/\sigma_1^{338}(orig.)$ from the calculations of σ_1^{338} using the new and original method respectively, is given in Fig. 4.9a. The plot looks very similar to that of σ_1^{338}/σ^T in Fig. 4.4, which means that the results from the CSD-CS

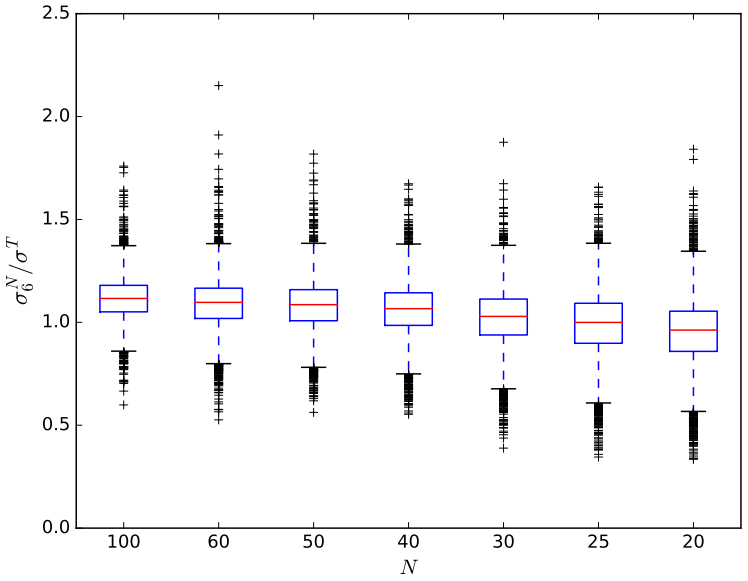


Figure 4.8: Boxplots of seven distributions of the relative standard deviations of the means, σ_o^N/σ^T, coming from calculations of σ_o^N for 4950 gene pairs, using the random sub-sampling approach with the parameters $o = 6$ and $N = 100, 60, 50, 40, 30, 25$ and 20, and $S_{max} = 150$. The σ^T values come from an independent run of the random sub-sampling script with $o = 1$, $N = 338$ and $S_{max} = 200$, and are used as true reference values for each gene pair.

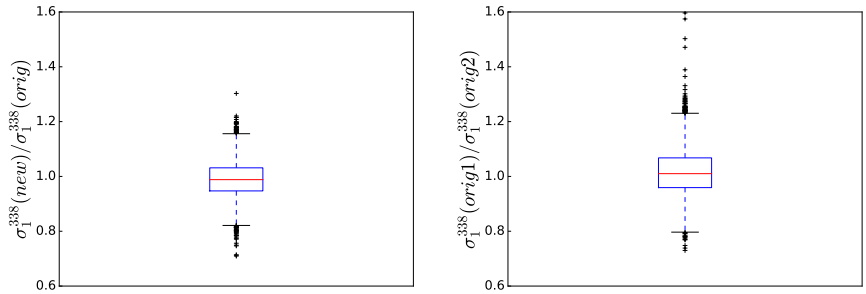
Table 4.2: *t*- and p-values from a two-sample *t*-test of 12 σ_o^N series against the common reference σ^T series. Any p-value below 0.05 indicates that the two series are drawn from different underlying distributions.

N*, *o	**t-value**	**p-value**
338, 1	-0.42	0.672
300, 1	1.20	0.229
200, 1	-2.66	0.008
338, 2	-0.69	0.489
300, 2	0.67	0.500
200, 2	-3.48	0.001
338, 3	-0.94	0.346
300, 3	0.20	0.845
200, 3	-3.79	0.000
338, 6	-0.62	0.540
300, 6	0.51	0.607
200, 6	-3.57	0.000

software do not differ more from the results obtained with my script, than the results from two runs of my script differ from each other. Fig. 4.3a shows some variation in the σ values on many runs with my script, but in order to find out if the results from the CSD-CS software and my random sub-sampling script creates an equal amount of variation with the same parameters S and o, it would be of further interest to check how much variation there is in the results from several runs of the CSD-CS software as well. This could indicate if the variation seen in Fig. 4.9a is mainly caused by the variation in my script or if both my script and the CSD-CS software seem to be contributing equally to the variation. Fig. 4.9b shows a boxplot of the pair-wise ratios of σ_1^{338} from two runs of the CSD-CS software on the same data as before, limiting the software to draw $S = 200$ sub-samples. The only difference between the two runs was varying the "randomSeed" parameter of the program, to avoid that it produced the exact same sub-samples both times. Figs. 4.9b and 4.9a look very similar to each other, and also to the σ_1^{338}/σ^T plot in Fig. 4.4. The results from these two runs of the CSD-CS software were actually slightly more different than the results from the two alternative softwares were from each other. With additional tests the exact differenes might change. Figs. 4.9b and 4.9a illustrate that both methods contribute with a similar amount of variation in the results. The σ^T values used as a reference could therefore be assumed to be equally good as the σ_1^{338} values found using the CSD-CS software with the original sub-sampling method.

4.1.4 Investigation of the σ_o^N/σ^T Boxplots' Outliers

The outliers in the σ_o^N/σ^T boxplots in Figs. 4.4-4.7 were investigated in more detail. Table 4.3 gives the number of outliers for the boxplots that were compared, while Table 4.4 shows the number of outliers that the compared boxplots had in common, as well as the percentage of outliers that this number represented for both groups respectively. The overall Spearman correlation and the associated σ^T for the five most extreme outlier gene



(a) **(b)**

Figure 4.9: Fig. (a) displays the boxplot of the distribution of the pair-wise ratios $\sigma_1^{338}(new)/\sigma_1^{338}(orig)$ for 4950 gene pairs, used as a comparison of σ_1^{338} computed using the new sub-sampling approach, denoted (new), and the original, deterministic method for finding independent sub-samples, denoted (orig). Fig. (b) shows the boxplot of the pair-wise ratios of $\sigma_1^{338}(orig1)$ $/\sigma_1^{338}(orig2)$ for 4950 gene pairs, used as a comparison of σ_1^{338} computed with two runs of the original, deterministic method for finding independent sub-samples. The sub-sampling is initiated at different places in the data set for the two runs, which are denoted (orig1) and (orig2), respectively.

pairs, as well as the five gene pairs closest to the median, were found for the boxplots of σ_1^{338}/σ^T, σ_1^{100}/σ^T, σ_6^{338}/σ^T and σ_6^{30}/σ^T, and given in Tables 4.5-4.6. These plots were chosen so that both big and small data sets, and high and low o values were represented.

Table 4.3: Number of outlier genes in chosen σ_o^N/σ^T boxplots, and the percentage of all 4950 gene pairs that this number represents. N denotes the data set size, and o gives the maximum permitted overlap of sub-samples in the calculations of σ_o^N.

N, o	**Number of outliers**	**Percentage outliers**
338, 1	53	1.07%
300, 1	44	0.88%
100, 1	84	1.70%
338, 6	60	1.21%
100, 6	116	2.34%
50, 6	146	2.95%
30, 6	141	2.85%

Tables 4.5-4.6 show that, with a few exceptions, the outliers were strongly correlated gene pairs with low σ^T values, while the gene pairs close to the boxplot medians were weakly correlated with high σ^T (see Fig. 4.2 for reference). This is consistent with the results found in Fig. 4.3, which demonstrated that for a few example gene pairs, the variability in the σ values found using the random sub-sampling approach were of the same magnitudes, independent of the magnitudes of the σ values. The investigation of outliers further confirmed that the variation in the final σ values from calculations with $S_{max} = 200$

Table 4.4: Number of outliers in common between chosen σ_o^N/σ^T boxplots, as well as the percentage of each plot's total number of outliers that this represents. N denotes data set size, and o gives the maximum permitted overlap of sub-samples in the calculations of σ_o^N.

N1, o1	***N2, o2***	**Number of outliers in common**	**Percentage of outliers in common, plot1, plot2**
338, 1	300, 1	7	13%, 16%
338, 1	100, 1	6	11%, 7%
338, 1	338, 6	9	23%, 19%
100, 1	100, 6	35	42%, 30%
50, 6	100, 6	51	35%, 44%
30, 6	100, 6	32	23%, 22%

was not much lower for highly correlated gene pairs with low correlation variability. Fig. 4.2 shows that the number of gene pairs with a relatively low correlation variability ($\sigma < 0.025$) accounts for a much higher amount of the total number of gene pairs than the highest number of outliers that was detected ($o = 6$ and $N = 100, 60$ and 30). The numbers of outliers are still quite high, especially compared to the number of gene pairs with *very* low correlation variability ($\sigma < 0.020$). However, this does not mean that the majority of highly correlated gene pairs are outliers. One should also note that even a substantial overestimation of a σ that should have been low, would still normally be below the highest σ values. These over- or underestimations would therefore not be able to change the characteristics of an entire CSD network.

Table 4.5: Stats for boxplot outlier gene pairs: The Spearman correlation, ρ_{ij}, and associated σ_{ij}^T, for the five most extreme outlier gene pairs in the boxplots of σ_1^{338}/σ^T and σ_1^{100}/σ^T (Fig. 4.4) and σ_r^{338}/σ^T and σ_3^{30}/σ^T (Fig. 4.7). N denotes the data set size, and o gives the maximum permitted sub-sample overlap in the calculations of σ_o^N.

N, o	**Gene pair**	**Spearman's** ρ	σ^T
338, 1	ENSG00000240618.1, ENSG00000228327.2	0.879	0.014
	ENSG00000188290.6, ENSG00000160072.15	0.720	0.021
	ENSG00000237683.5, ENSG00000239906.1	0.729	0.014
	ENSG00000188290.6, ENSG00000176022.3	−0.495	0.019
	ENSG00000197785.9, ENSG00000189409.8	0.731	0.015
100, 1	ENSG00000131591.13, ENSG00000186827.6	0.770	0.018
	ENSG00000162572.15, ENSG00000162576.12	0.649	0.021
	ENSG00000176022.3, ENSG00000175756.9	0.782	0.013
	ENSG00000160087.16, ENSG00000272455.1	0.730	0.014
	ENSG00000160072.15, ENSG00000197785.9	0.853	0.011
338, 6	ENSG00000239906.1, ENSG00000241860.2	0.773	0.014
	ENSG00000224956.5, ENSG00000269227.1	−0.073	0.024
	ENSG00000160072.15, ENSG00000197785.9	0.853	0.011
	ENSG00000127054.14, ENSG00000160072.15	0.852	0.016
	ENSG00000131591.13, ENSG00000160072.15	0.857	0.012
30, 6	ENSG00000227232.4, ENSG00000240618.1	−0.328	0.027
	ENSG00000238009.2, ENSG00000230021.3	0.646	0.019
	ENSG00000233750.3, ENSG00000188976.6	−0.718	0.015
	ENSG00000237683.5, ENSG00000239906.1	0.729	0.014
	ENSG00000228463.4, ENSG00000186891.9	−0.594	0.020

Table 4.6: Stats of gene pairs close to the boxplot medians: The Spearman correlation, ρ, and the associated σ^T, for five gene pairs lying on or close to the medians in the boxplots of σ_1^{338}/σ^T and σ_1^{100}/σ^T (Fig. 4.4) and σ_r^{338}/σ^T and σ_3^{30}/σ^T (Fig. 4.7). N denotes the data set size, and o gives the maximum permitted sub-sample overlap in the calculations of σ_o^N.

N, o	**Gene pair**	**Spearman's** ρ	σ^T
338, 1	ENSG00000198744.5, ENSG00000224870	0.223	0.028
	ENSG00000198744.5, ENSG00000187730.6	0.123	0.028
	ENSG00000198744.5, ENSG00000182873.4	0.152	0.027
	ENSG00000229376.3, ENSG00000235098.4	−0.085	0.030
	ENSG00000229376.3, ENSG00000269227.1	0.016	0.030
100, 1	ENSG00000235373.1, ENSG00000228594.1	−0.070	0.029
	ENSG00000235373.1, ENSG00000248333.3	−0.276	0.027
	ENSG00000240618.1, ENSG00000260179.1	0.072	0.028
	ENSG00000240618.1, ENSG00000227775.3	0.257	0.030
	ENSG00000228327.2, ENSG00000187583.6	−0.197	0.029
338, 6	ENSG00000177757.1, ENSG00000272512.1	−0.061	0.028
	ENSG00000177757.1, ENSG00000197530.8	−0.096	0.029
	ENSG00000188976.6, ENSG00000179403.10	0.189	0.027
	ENSG00000187583.6, ENSG00000160075.10	−0.052	0.030
	ENSG00000272141.1, ENSG00000179403.10	0.119	0.028
30, 6	ENSG00000250575.1, ENSG00000187608.5	−0.245	0.026
	ENSG00000225972.1, ENSG00000224956.5	0.081	0.032
	ENSG00000224956.5, ENSG00000197530.8	0.204	0.027
	ENSG00000188290.6, NSG00000127054.14	0.613	0.018
	ENSG00000188290.6, ENSG00000078369.13	−0.094	0.028

4.2 Results: Application to Rheumatoid Arthritis Gene Expression Data

4.2.1 CSD Network

The gene expression data from 25049 genes measured from the synovial fluid of RA patients and healthy controls resulted in a CSD network of 2287 nodes (genes) and 3610 links (Fig. 4.10), using an importance level of $p = 10^{-5}$. The CSD network has approximately the same amount of the three link types: 1148 C-type links, 1285 S-type links and 1177 D-type links. The majority of the nodes and links are interconnected in a giant component of 1681 (73.5%) nodes and 3191 (88.4%) links. In addition to the giant component, there are four connected components with more than 10 nodes, the biggest having 23 nodes and 35 links. The remaining nodes form 211 connected components with fewer than 10 nodes each. The waste majority of these small components only consists of two nodes. With very few exceptions, all links outside the giant component are C- and S-type links. Other importance levels ($p = 10^{-4}$ and $p = 10^{-6}$) were also investigated, but were found to be too dense or too sparse. Network visualizations and size parameters are given in Appendix A8.

Fig. 4.10 shows that although the giant component is interconnected by all three link types, links of the same type tend to group together. This tendency is strongest for D-type links and weakest for S-type links. In Fig. 4.11, visualizing the giant component of the network, the nodes are colored according to which link type they are dominated by. A node is defined to be dominated by a link type if more than 2/3 of its links are of a given type. Blue, green and red nodes thus have at least 2/3 of C-, S- and D-type links, respectively. Yellow nodes are not dominated by one link type only. The majority of the nodes in Fig. 4.11 are blue, green or red, further confirming the tendency of link types to group together.

4.2.2 Degree Distribution

The degree distribution of the CSD network was found to follow a heavy-tail power-law distribution with a degree exponent $\gamma = 1.7$. This means that the network topology is far from random, leading us to take a closer look at the network hubs in the next step of the analysis.

Extracting the networks of nodes connected only by C-, S- or D-type nodes and analyzing them separately revealed that these sub-networks also had degree distributions that could be approximated with power-law functions with degree exponents $\gamma_C = 2.0$, $\gamma_S = 1.5$ and $\gamma_D = 1.3$. As already seen, the C-network does not have any nodes with as high degrees as those found in the S- and D-networks, but it nevertheless shows a topology of hundreds of low-degree nodes and a few much more connected nodes. Visualizations of the separate C-, S- and D-networks are found in the Appendix A9.

4.2.3 Hubs and Assortativity

Network hubs were defined as nodes with more than 40 neighbors. This choice was the same as was used in a comparable study [2], and resulted in a reasonably low number of

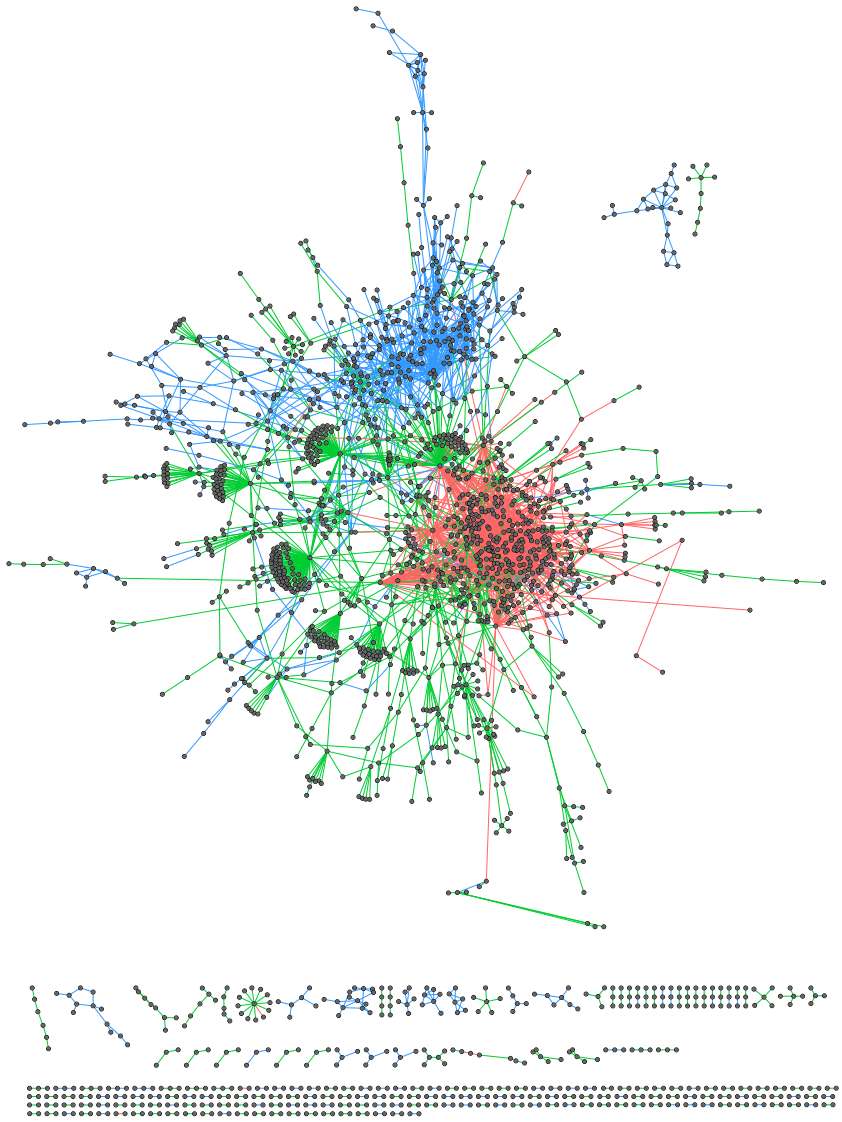


Figure 4.10: Visualization of the CSD network generated using an importance value of $p = 10^{-5}$. The links are colored according to type: C-type links are blue, S-type links are green and D-type links are red.

Table 4.7: The hubs of the CSD network with their total degree, k, the number of connections of each type, k_C, k_S, k_D, and the node homogeneity, H, of each hub.

Hub genes	k	k_C	k_S	k_D	H
PDCD1	98	0	56	42	0.51
ZNF205-AS1	90	0	90	0	1.00
GPR18	76	0	75	1	0.97
LINC00426	71	0	0	71	1.00
SEPT1	64	0	5	59	0.86
CTLA4	53	0	0	53	1.00
ASAP2	44	0	6	38	0.76
ENPP1	44	0	4	40	0.83
IL2RG	44	1	0	43	0.96
PRDM1	43	0	0	43	1.00
GBP5	41	0	0	41	1.00

hubs: Eleven genes were identified as network hubs in the giant component. These are listed in Table 4.7 and marked as enlarged, black nodes in Fig. 4.11. In order to see the extent to which the hubs are dominated by one type of link, the number of C-, S- and D-type connections, k_C, k_S, k_D, as well as the node homogeneity, H, is given for each hub in the table. The biggest hub, *PDCD1*, had a comparable number of S- and D-links, while the second and third biggest hubs, *ZNF205-AS1* and *GPR18*, were completely dominated by S-links. The remaining hubs were highly dominated by D-type links. No hubs were dominated by C-links; in fact there was only *one* C-link connected to any of the hubs. The three highest connected nodes that were dominated by C-links had degrees of $k = 29$, $k = 28$ and $k = 24$, respectively, and all three had node homogeneities of $H = 1.00$, meaning that the connections with all neighbors were of the conserved type. Six of the eleven hubs were among the genes with highest betweenness centrality: *PDCD1*, *GPR18*, *SEPT1* and *ZNF205-AS1* were the four genes with the highest betweenness centrality (in that order), while *ASAP1* and *LINC00426* genes were on the 6th and 7th place, respectively. *PDCD1* and *SEPT1* were also the two genes having the highest closeness centralities in the network.

Fig 4.11 reveals that the topology seems to differ according to link type distribution. The sub-network of only S-links seems to be much more disassortative than the sub-networks of C- or D-type links. Highly connected nodes with a high proportion of S-links, which includes the three biggest hubs in the CSD network, are connected to a high proportion of neighbors with low degree. This tendency can be seen as green "bouquets" in Fig. 4.11. Although close to each other in the network, the three biggest hubs in the network are not directly linked, but are separated by low degree nodes connected to each of them. Because the D- and C-type links are so tightly interconnected, it is difficult to see the topology of these networks by visual inspection of the figure. The neighborhood connectivity distributions of the separate C-, S- and D-networks (Figs. 4.12, 4.13 and 4.14) show that the C-network is assortative, while both the S- and D-networks are disassortative; however, the S-network is much more strongly disassortative than the D-network. The C- and D-networks show about equally strong assortativity and disassortativity re-

spectively. The topologies of the C-, S- and D-type networks differ somewhat from what has previously been found using the CSD method: In the CSD network from two different brain regions, Voigt, Nowick and Almaas found a highly assortative C-network with a densely connected core, and a stronger disassortativity for the D-network than for the S-network [2]. They hypothesized that the tightly co-regulated cluster of C-dominated genes could reflect more robust regulatory mechanisms, which are less likely to change.

Known information about biological functions and/or disease associations of the identified hub genes is summarized in Table 4.8 and elaborated in the following paragraphs. Several of the hub genes are linked to the immune system, either by being directly involved in the function, differentiation or development of B or T cells, by being connected to other parts of the immune system, or by being associated with RA or other autoimmune or immune system related diseases. Many of their gene products have GTPase activity or a broader ability to cleave nucleoside-3-phosphates, and several are associated with cancer. Two of the hub genes are RNA genes, which do not code for protein products, but which can possibly have regulatory functions. The link between an autoimmune disease and the immune system is obvious, but these other functions are also interesting in the context of RA. GTPases are important in signaling pathways related to the cell cycle [46], and cancer is characterized by abnormal growth. As RA involves severe tissue deformities, genes associated with cell differentiation or abnormal growth are also considered to be of relevance.

The *PDCD1* gene codes for a membrane protein are expressed in pro-B-cells, and is thought to play a role in their differentiation [47]. The protein has also been suggested to be important in T cell function, and for prevention against autoimmune diseases [48]. It has previously been associated with RA [49], and also with other inflammatory diseases, such as Systemic Lupus Erythematosus [50]. The gene *GPR18*, which codes for a G-protein coupled receptor, has been suggested to be involved in immune system regulation [51], for example in the directed migration of immune cells to sites of interest, mediated by a lipid-based signaling mechanism in the central nervous system [52]. The *CTLA4* gene is a member of the immunoglobulin family, and codes for a protein that contributes to T cell inhibition [53]. The gene has been suggested to play a key role in thyroid autoimmunity, and polymorphisms in the gene has been associated with RA, type I Diabetes Mellitus and other autoimmune diseases [54]. *IL2RG* codes for a protein found in many interleukin receptors, and is important for T cell development [55] and other parts of the immune system. Mutations in the gene is associated with severe immunodeficiency disorders [56]. The *PRDM1* gene codes for a protein repression of β-interferon gene expression [57]. Induced by B lymphocytes, *PRDM1* is a master regulator of plasma cell differentiation [58], and has previously been associated with RA [59] and T cell lymphomas (cancer) [60]. The *ASAP2* gene is involved in regulation of vesicular transport, cellular migration (via GTPase activation) and autophagy [61]. Immune cells show a high degree of migration throughout the body, but more important: autophagy can affect both innate and adapted immunity [62].

The gene *ENPP1* codes for a protein that is able to cleave all nucleoside-3-phosphates, and it is also found to be important for the survival of long-lived antibody-secreting plasma cells (derived from B cells) [63]. The gene is associated with a long list of diseases and symptoms, among them osteoarthritis [64], type II Diabetes [65], decreased kidney func-

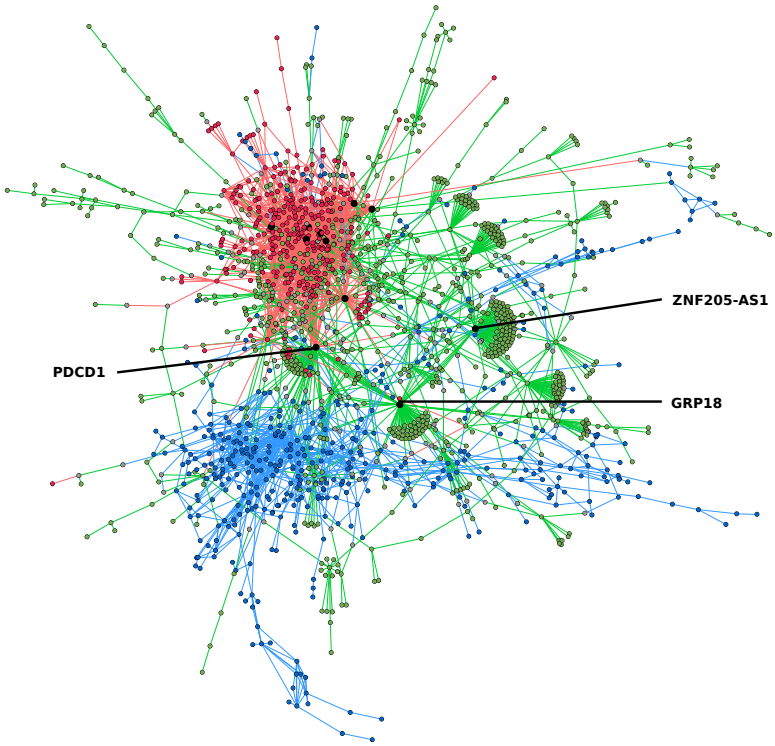


Figure 4.11: Visualization of the giant component of the CSD network generated using an importance value of $p = 10^{-5}$. The links are colored by link type (C-types are blue, S-types are green and D-types are red), and the nodes are colored according to link type dominance: Any node where more than 2/3 of the links connected to it are of a given type is colored with the same color as the dominating link type. Yellow nodes are not dominated by any link type. Hubs ($k > 40$) are coloured black and enlarged for emphasis. The topology of the three largest hubs, *PDCD1*, *ZNF205-AS1* and *GRP18*, indicates a strong disassortativity in the part of the network dominated by S-links (see the text), and are therefore marked explicitly.

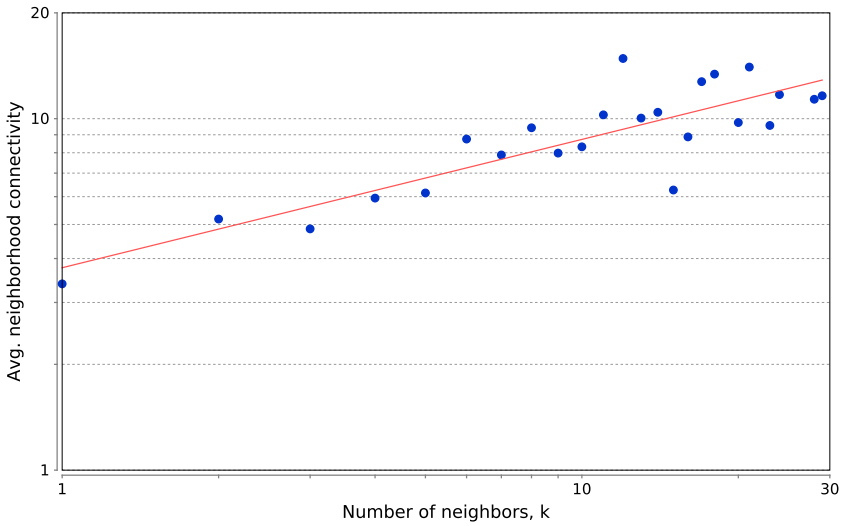


Figure 4.12: Neighborhood connectivity distribution of the C-network: The loglog plot of node degree, k, versus average connectivity of all neighbors of degree k-nodes reveals a weak assortativity: The nodes have a tendency to connect to other nodes of similar degree. The regression line through the data points gives a positive correlation exponent $\mu = 0.37$, and the correlation between the data points and the line is 0.8.

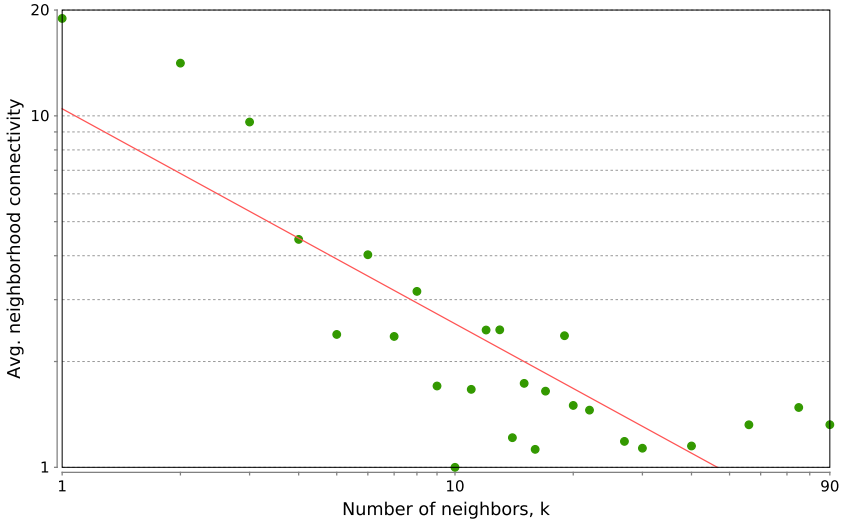


Figure 4.13: Neighborhood connectivity distribution of the S-network: The loglog plot of node degree, k, versus average connectivity of all neighbors of degree k-nodes reveals a strong disassortativity: Highly connected nodes have a tendency to link to nodes with low degree and vice versa. The regression line through the plot gives a negative correlation exponent $\mu = -0.61$, and the correlation between the points and the line is 0.9.

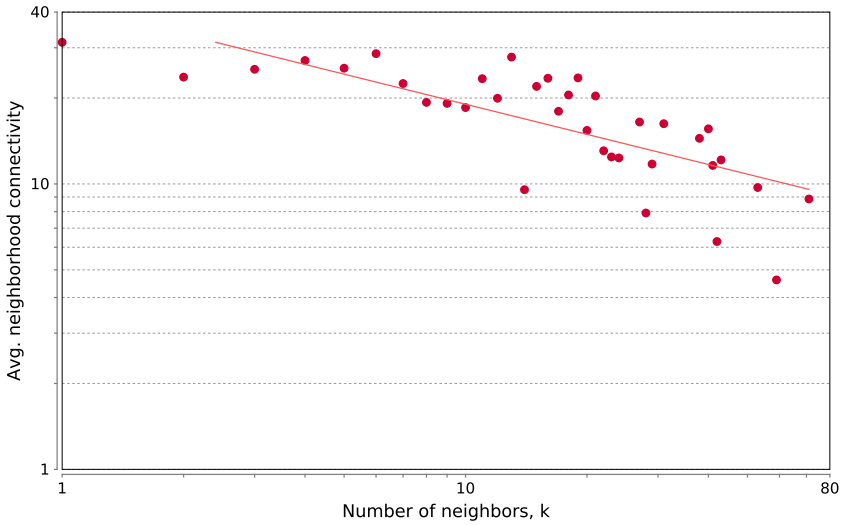


Figure 4.14: Neighborhood connectivity distribution of the D-network: The loglog plot of node degree, k, versus average connectivity of all neighbors of degree k-nodes reveals a weak disassortativity: Highly connected nodes have a tendency to link to nodes of low degree and vice versa. The regression line through the plot gives a correlation component $\mu = -0.35$, and the correlation between the points and the line is 0.8 .

tion [66] and arterial calcification [67]. The nucleoside-3-phosphate cleaving ability of the *ENPP1* gene is shared by *SEPT1* and *GBP5*, which both code for two different GTPases. The protein product of *SEPT1* is a member of the septin family of GTPases, showing high expression in lymphoid tissue [68]. The gene has been found to be involved in cytokinesis [69], and has also been associated with oral cancer [70]. The *GBP5* gene codes for a GTPase that can activate NLRP3 inflammasome assembly, and thus plays a role in immunity and inflammation [71]. Just like *SEPT1*, this gene has also been suggested to have a cancer related function, and has been shown to be highly expressed in T cell tumor tissue [72]. Overexpression of G-protein coupled receptors (GPRs or GPCRs) has also been linked to tumor progression [73], possibly also linking the previously mentioned *GPR18* to cancer development.

The gene *ZNF205-AS1* does not code for a protein, but for a single stranded RNA oligonucleotide with an *antisense* function, meaning that it can bind to the mRNA of the *ZNF205* gene and block its translation into protein. The *ZNF205* gene has been identified as a transcription repressor [74]. *LINC00426* is also an RNA gene and does not code for a protein product. To my knowledge, there is no known function or disease associated with this gene.

Note that some of the genes are referred to by alternative names in the cited literature: *PD1* is an alias for *PDCD1*, *ZNF205* is also referred to as *ZNF210* or *Rhit*, and aliases of *PRDM1* include *PRDI-BF1* and *Blimp1*.

Table 4.8: The eleven most highly connected hub genes in the network are given with known associated biological functions or processes, as well as previous disease and symptom associations, if any. The order of the genes corresponds to their degree in the network, starting with the most highly connected node.

Hub gene	Biological Function or Process	Disease Association
PDCD1	Differentiation of pro-B-cells [47] Regulation of T cell function[48] Prevention of autoimmune diseases [48]	RA [49] Systemic Lupus Erythematosus [50]
ZNF205-AS1	Non-coding RNA that blocks translation of transcription repressor [74]	
GPR18	G-protein coupled receptor activity [51] Possible immune system regulation [51]	Cancer [73]
LINC00426	Non-coding RNA	
SEPT1	Cytokinesis [69]	Cancer [70]
CTLA4	T cell inhibition [53]	RA [54] Type I Diabetes Mellitus [54] Other autoimmune diseases [54]
ASAP2	Vesicular transport [61] Autophagy [61] Cell migration [61]	
ENPP1	Nucleic acid cleavage [63] Survival of long-lived plasma cells [63]	Type II diabetes [65] Decreased kidney function [66] Arterial calcification [67] Osteoarthritis [64]
IL2RG	T-cell development [55] Interleukin receptors [55]	Immunodeficiency disorders [56]
PRDM1	Repression of β-interferon gene expression [57] Plasma cell differentiation [58]	RA [59] Lymphocyte cancer [60]
GBP5	Immunity [71] Inflammation [71] GTPase activity [71]	Cancer [72]

4.2.4 GO Functional Enrichment

The full network, as well as the separate C-, S- and D-networks, showed high enrichment of genes related to multiple biological processes. The general GO categories, which included a high number of genes, were moderately enriched in the CSD network. Most of the general terms enriched in the full network were related to regulation of various other processes. The majority of the more specific processes associated with a lower number of genes were very highly enriched. Most of them fell into one of three groups: processes that could be directly or indirectly linked to the immune system, processes linked to inflammatory activity and processes associated with cell differentiation.

Table 4.9 gives an overview of the main general biological terms that were enriched in the full CSD network, and their fold enrichment (FE). The most relevant and specific sub-processes enriched in the full network, as well as the extracted C-, S- and D-networks, are described in more detail in the following paragraphs and summarized in Table 4.10, together with their fold enrichment. Table 4.9 is included to provide a context for the specific processes in Table 4.10. The complete lists of biological processes where associated genes were significantly enriched in the four networks, as well as the fold enrichment and the false discovery rates for each process, is too long to be written out in its entirety in the Appendix, but a URL to the four lists are given in Appendix A10. Enrichment factors larger than 1.0 represents an overrepresentation of genes related to the given process, while factors smaller than 1.0 represents an under-representation. All results are given with a correction for multiple comparisons, only including results with a false discovery rate less than 0.05.

The by far most highly enriched processes in the full CSD network were related to the MHC protein complex (10-fold enrichment), known to be strongly related to RA pathogenesis [16]. The most significantly enriched MHC related processes included MHC protein complex assembly and peptide antigen (immune response trigger) processing and binding to the MHC. There were however just five genes related to the MHC, and although all five were represented in the network, resulting in a high enrichment factor, they did not make up a large proportion of the network. Other enriched processes directly involved with the immune system included the following categories: immune system processes, regulation of phagocytosis, the interleukin-1-mediated signaling pathway, the T cell receptor signaling pathway and regulation of B cell differentiation. The most significant other biological processes that were highly enriched in the full CSD network were related to cell motility, protein localization to chromosomes/telomeres, regulation of catabolic processes and of protein localization to Cajal bodies. Immune cells need to have a directed cell motility, since they migrate to wounded or inflammed sites in the body. Furthermore, premature aging of immune system T-cells, with accelerated loss of telomeres, have been associated with RA, and it has been suggested that this is due to excessive proliferative pressure or inadequate maintenance of the telomeres [75]. Cajal bodies have been found to participate in the delivery of telomerase (telomere extendig enzyme) to the telomeres, and a high activity of Cajal bodies allow highly proliferative cells, for instance in tumors, to grow ”indefinitely” [76].

The C-network showed a strong enrichment in processes related to cellular component organization and assembly. The process with the strongest enrichment (almost 27-fold) in this category was related to the assembly of antigens with the already mentioned MHC

Table 4.9: The main GO biological process categories that had an enrichment of associated genes in the full CSD-network are given with their respective fold enrichment (FE). The categories are ordered according to their fold enrichment.

Main GO category	**FE**
Regulation of stem cell differentiation	2.66
Pos. regulation of autophagy	2.33
Macroautophagy	2.24
Regulation of intracellular transport	1.93
Protein folding	1.90
DNA replication	1.89
Cell division	1.86
Regulation of cellular localization	1.78
Neg. regulation of immune system process	1.70
Immune system development	1.61
Regulation. of cell activation	1.57
Cell proliferation	1.52
Cellular component organization or biogenesis	1.43
Immune syst. process	1.38
Pos. regulation. of cellular process	1.32
Regulation. of cellular metabolic process	1.31
Regulation. of response to stimulus	1.30
Regulation. of cell death	1.29
Response to stress	1.23
Regulation. of signalling	1.21

Table 4.10: The GO biological processes that had the strongest enrichment of associated genes in the full CSD-network and the separate C-, S- and D-networks. The fold enrichment is given for each process.

Network	**Enriched Process**	**FE**
CSD Network	Peptide antigen assemblywith MHC protein complex	10.18
	Actin polymerization-dependent cell motility	8.48
	Pos. regulation of establishment of protein localizationto telomere	7.12
	Regulation of receptor catabolic process	6.78
	Pos. regulation of protein localization to Cajal body	6.78
C-network	Peptide antigen assembly with MHC class II protein complex	26.91
	Beta selection	26.91
	Positive regulation of protein lipidation	20.18
	Positive regulation of ER tubular network organization	20.18
	Actin polymerization-dependent cell motility	17.94
	Pos. regulation of protein localization to Cajal body	17.94
	Pos. regulation of establishment of protein localization to telomere	16.14
	Neg. thymic T cell selection	13.45
	Positive regulation of RNA export from nucleus	13.45
	Protein heterotrimerization	12.23
S-network	ER-signalling pathway	4.54
	Fatty acid β-oxidation	3.89
	Ubiquitin-dependent protein catabolic process	3.53
	Anaphase-promoting complex-dependent catabolic process	3.49
	Regulation of telomere maintenance via telomere lengthening	3.48
D-network	Pharyngeal arch artery morphogenesis	24.98
	T-helper 17 cell differentiation	21.86
	Pos. regulation of vascular endothelial cell proliferation	20.82
	Regulation of cell fate specification	16.65
	Paraxial mesoderm development	12.27
	Pos. regulation of cardiac muscle cell proliferation	12.14
	Outflow tract septum morphogenesis	11.66
	Positive regulation of interleukin-2 production	9.71
	T cell costimulation	8.55
	Cellular defense response	7.03
	Negative regulation of T cell proliferation	7.03

complex. The C-network was also enriched in processes related to immune system development (among them T cell differentiation, activation and selection), in which beta selection processes were the most highly enriched. Genes related to negative thymic T cell selection were also enriched by a factor of 13 in the C-network. Like the beta selection, negative thymic selection also contributes to the maturation of T cells. RA patients have been shown to have a lower diversity of T cells, which is also typical for the immune dysfunction in elderly people, partly attributed to the loss of thymic function [77]. The C-network also showed a high enrichment of genes related to the regulation of biological processes, regulation of cellular component organization, localization and its regulation, negative T cell selection and regulation of transport. The most highly enriched categories in these groups include the positive regulation of protein lipidation, endoplasmic reticulum (ER) organization, RNA export from nucleus, ATP synthesis, protein oligomerization and skin morphogenesis, as well as the previously mentioned protein localization to telomeres and Cajal bodies. Protein lipidation is involved in the regulation of numerous biological pathways. It can affect protein binding to cell membranes, but also the regulation of signaling processes, and can be very important for the control of protein localization and function [78]. The endoplasmic reticulum has many functions, including folding and modification of proteins and the synthesis of phospholipids and steroids [79]. Of particular relevance to RA is the fact that steroids are commonly used for their anti-inflammatory effects, for instance in the treatment of RA [80]. RNA export and protein oligomerization is also obviously linked to protein production and folding. All the mentioned specific processes were enriched by more than a 6-fold in the C-network, most of them by more than a 10-fold. It must however be noted that there were not always a high number of genes related to each specific process, but since all of the genes related to for instance antigen assembly with the MHC complex were present, which was much more than what would have been expected by chance, this resulted in a high fold enrichment.

The S-network was most enriched in categories related to regulation of metabolic and signalling processes. It is worth mentioning that these processes showed a lower enrichment than the most specific processes in the C-network (between 2.3-fold and 4.5-fold enrichment, compared to up to a 27-fold enrichment), but the processes that were enriched in the S-network were broader and included a higher number of genes. Prominent biological processes enriched in the S-network included processes related to the ER-nucleus signalling pathways, biosynthesis of sphingolipids (also involved in signalling pathways), fatty acid β-oxidation, ubiquitin-dependent protein catabolism and protein polyubiquitination, telomere maintenance, stem cell differentiation and cell division, RNA processing and stability, and DNA repair of double DNA strand breaks. Deficiency of the previously mentioned telomerase is associated with premature immune aging, which is again associated with loss of critical immune function [81]. Patients with RA have furthermore been shown to have defects in the DNA repair mechanisms related to DNA double strand breaks [81]. Ubiquitination of proteins normally marks them for degradation by proteasomes, and ubiquitination of the MHC is also involved in cellular modulation of immune recognition. The ubiquitin-proteasome system is an important part of the protein metabolism and quality control in eukaryotes, and is involved in processes such as oxidative stress and inflammation [82]. The E3 ubiquitin ligase has been shown to have an anti-arthritic effect [83].

The D-network showed a high enrichment of genes related to developmental processes such as tissue and organ development, respiratory system development, circulatory system development, immune system development, connective tissue development, immune cell activation, cytoskeleton development, and the regulation of developmental processes. The most highly enriched processes in these groups were pharyngeal arch artery morpogenesis, T-helper 17 cell differentiation (in particular T-helper 17 type immune response), regulation of vascular endothelial cell proliferation and blood vessel endothelial cell migration, regulation of cell fate specification, regulation of interleukin-2 production and T cell activation, differentiation and proliferation and also cellular defense responses. Genes related to these processes were all enriched by more than a 6-fold and up to a 25-fold in the case of pharyngeal arch artery morphogenesis. The high enrichment of genes related to processes that actively regulate the development of a high number of tissues and organs was not seen in the C- or S-networks.

4.2.5 Genes Previously Associated with RA

Appendix A1 gives a list of 54 genes that have previously been associated with RA. Out of these there are 13 genes in the list that can be identified in the CSD network: *HLA-DRB1*, *PDCD1*, *CTLA4*, *PRDM1*, *FCRL*, *RASGRP1*, *IKZF3*, *IL2RA*, *IL2RB*, *CD2*, *CD40*, *CD28* and *DDX6*.

The genes *PDCD1*, *CTLA4* and *PRDM1* are already identified as network hubs in previous sections. The genes *FCRL* and *IKZF3* were linked to another hub gene, *ASAP2*, and *RASGRP1* was connected to two hubs: *ASAP2* and *ENPP1*.

The gene *HLA-DRB1* is the single gene that has shown the strongest association with RA in previous research, but being connected to only *one* other gene, making up a 2-node component, it had a very modest role in the CSD network. Other *HLA*-genes were found in more prominent positions in the network: the gene *HLA-DOB* had an intermediate degree of $k = 11$, and was found in the D-network of the giant component. The genes *HLA-DMA*, *HLA-DMB*, *HLA-DPA1*, *HLA-DPB1*, *HLA-DQA1* and *HLA-DRA* were linked together in the second largest component of 23 nodes connected exclusively by C-links. *HLA-DRA*, *HLA-DMA*, *HLA-DRB1* and *HLA-DMB* were in fact three of the four genes making antigen assembly with the MHC complex so highly enriched in the C-network.

Chapter 5

Discussion

5.1 Method Study

The new sub-sampling algorithm has both advantages and disadvantages. One possible advantage of the new algorithm compared to the implementation of the original algorithm that was used, is that all parts of the data set are sampled with equal probability in the implementation of the new sub-sampling. Although the original algorithm could in theory find *all* valid sub-samples, the available CSD-CS software implementing the original algorithm (see Section 3.1.6) would stop after a given number of sub-samples were drawn. Because of the nature of the algorithm, this meant that data points positioned in the beginning of the data set would be sampled more often than data points in the end of the data set. Any confounding factors expressed towards the end of the data set would therefore be less likely to be detected. It is not given how big an advantage this means for the new algorithm, as it has its own problem with finding valid sub-samples.

The latter point leads us to the most obvious disadvantage with the new sub-sampling algorithm: the random drawing of sub-samples. It quickly becomes challenging to draw more valid sub-samples if the sub-samples are forced to be independent or nearly independent. If fully independent sub-samples were found to be necessary, it would clearly be advisable to use the original sub-sampling algorithm. With only *one* extra point overlap between sub-samples, the new algorithm could produce between two and three times as many *nearly* independent sub-samples as the number of completely independent sub-samples found with the original approach. When the sub-sample overlap is allowed to be fully random, with overlap of up to 6 data points between 7 data point sub-samples, the random sub-sampling algorithm was able to reach the chosen maximum number of 200 sub-samples even for data sets as small as 20 data points per gene. With high sub-sample dependence this might even be too much for the smallest data sets, as will be discussed later. As the new algorithm was not meant to replace the original algorithm for drawing *independent* sub-samples, the limited number of independent and nearly independent sub-samples ($o = 1$ or 2) was not considered to be a big problem for the method study itself.

The number of valid sub-samples that can be found for a given gene will be highly dependent on the actual values in the gene expression data set. Gene pairs where one or both of the genes have many zeros in any of the data sets will in practice be filtered out in the final network due to S being too low. The reason for this is that the Spearman rank correlation is dependent on ranking the data by magnitude, and it can therefore not be calculated if there are too many identical values in a sub-sample. The data set that was used for testing happened to have very few zero entries, which meant that few sub-samples were rejected as a result of a high content of zeros. Using the same approach on a different data set might therefore lead to a lower average number of valid sub-samples than what was found in this study. The results should therefore always be filtered based on the actual number of sub-samples that was drawn for a given gene pair.

The filtering of gene pairs where one or both genes have many zero-entries in the data set can in turn lead to loss of significant links in the CSD network. If one gene is expressed at various levels while the other gene is almost completely silent (many zero-entries) in one of the data sets, this gene pair should be registered as ”uncorrelated”, but will instead be filtered out from the network. If the same genes are expressed and strongly correlated in the other data set, they should have had an S-link between them that will instead be lost. Since any gene with many zero entries in the data set will be paired with all the other genes in the data set, it is possible that quite a substantial amount of significant S-links are lost due to this issue. The smaller the data set, the more will any fraction of zeros affect the overall results, as there will be a lower number of non-zero data points left to draw from. This could therefore also affect C- and D-links. An example would be if we have a small data set, for instance $N = 30$: If half of the data points for one gene is zero, and the other half has high values, and the sample-wise opposite is true for another gene, then these genes should be registered as ”anti-correlated”. Instead they might also be filtered out from the network because there are only 15 non-zero data points left, from which it could be difficult to draw enough valid sub-samples with a given permitted overlap.

The problem with a high content of zeros in the data set is however not a problem that is exclusive for the new sub-sampling approach, since the original sub-sampling algorithm also finds the Spearman correlation of its sub-samples. The problem might still be smaller with the original systematic approach, as all possible sub-samples could be checked before the gene pair is filtered out. With random drawing, on the other hand, it is possible that the same data points with zeros in them are drawn over and over again by chance, causing a fast increase in the discard rate.

The upper and lower limits for the number of sub-samples, S, were based on the plot of 28 gene pairs' σ signals as a function of S (Fig. 4.1). The early high fluctuations seen in the signal plot, and which were later dampened with higher S values, make sense mathematically: With a large number of sub-samples, any single value will influence the mean value much less than with a low number of sub-samples. Because gene expression data is normally quite noisy, the early fluctuations were expected. Because the ”signal” represented the standard deviation of a mean, it also makes sense that the fluctuations did not go *below* the final values at any point, since this would mean that the random sampling by coincidence happened to first draw a collection of very similar sub-samples from a very noisy data set. This would be unlikely, especially given the low number of gene pairs that were tested, and the relatively low number of data points in total. Because the 28 gene

pairs were chosen so that gene pairs with both positive and negative, as well as highly, medium and poor correlations were present, this plot should be quite representative for the behaviour of σ for differently correlated gene pairs. The signals could have been plotted separately for high, medium or low correlations to see the difference in how fast the signals became stable, but as S_{min} would have to be based on the signals stabilizing the slowest, this was not considered important.

Even though the biggest fluctuations were gone at $S = 50$, the values had not stabilized fully, so using this as the lower limit for S can be questioned. Even though the 28 signals were not very close to their final values at $S = 50$, they seemed to have reached their correct *order*, meaning that it was still possible to distinguish high and low σ gene pairs to some degree, even if the magnitude of the σ values would be too high. It was therefore more or less reasonable to use this lower limit for comparison purposes, sacrificing precise results for the sake of having data on low o and low N situations. In Figs. 4.4-4.7 one can clearly see the effect of this choice, namely the fact that the majority of the σ values deviated highly from their reference values even before the average S became as low as 50. This confirmed that this lower limit was indeed too low to get very precise results, and that this low number of sub-samples should be avoided in an actual CSD analysis. Setting the lower limit, S_{min}, to for instance 100 would be more appropriate, and an even higher number would be preferable. To see why correctly ordered, but overestimated σ values would be so problematic, imagine that one of the two data sets are much bigger than the other, and therefore able to produce more precise (lower) σ values, while the smaller data set gets highly over-estimated σ values. The over-estimated σ values will then dominate over the correctly estimated σ values from the bigger data set, and potentially shift the C-, S- and D-scores massively. In the worst case scenario, constantly over-estimating the σ values in only one data set could give a false impression of much confounding in one data set, and at the same hide the effects of real confounding factors in the other data set.

At $S = 200$, the signals had not changed much compared to lower S values, and this was set to be the upper limit for S in the further calculations. The 28 σ_{ij} signals (Fig. 4.1) could still change slightly more with higher S values, but there would also be a trade-off between time consumption and precision: Based on the difference between $S = 150$ and $S = 200$, there seems to be very small changes to be gained by increasing S above 200, while increasing the number of sub-samples for each gene pair by for instance 50 or 100 will significantly slow down the computation time, especially if S_{max} is close to the upper S limit for that data set size. One should also keep in mind that σ is dependent on what sub-samples were drawn in the given sub-sampling session, so that one could easily obtain a different σ when rerunning the sub-sampling procedure. It is impossible to say which of them would be the most correct one, since σ would never be more than an *estimate* of the internal variability in the correlation. It is therefore no guarantee that increasing S more and more will always result in a better σ. For these reasons, $S_{max} = 200$ was considered to be a good upper limit for S.

If the computation time had not been an issue, a better way of getting a more precise σ than increasing S could be to run the sub-sampling multiple times and use the average of the results. This would however slow down the computation time even more than increasing S, and is unfortunately not seen as a realistic option for the current implementation of the sub-sampling algorithm. One reason why this is unfortunate, is that averaging over

multiple runs of sub-sampling would not just have given more precise σ values in general, but could also have solved another major issue with the sub-sampling procedure, namely the fact that the results vary equally much for high and low σ values (Fig. 4.3). One could have expected that a low internal variation in correlation would also give a lower variation in low σ values, but based on σ calculated for 28 gene pairs 100 times this did not seem to be the case. The investigation of outliers from the full data set confirmed that highly correlated genes with low σ were highly overrepresented as extreme outliers, both with high and low o and N. This is most likely because gene expression measurements are not very precise, so gene expression data seemed to be very noisy, even for highly correlated gene pairs with low σ. This means that the most interesting gene pairs, namely those with a low internal variation in correlation over the data set, have a lower precision in σ relative to their absolute magnitude than what gene pairs with high internal variation have. This is a significant weakness of the sub-sampling approach that will cause some noise in the CSD network. It is likely that this is also an issue with the original sub-sampling approach.

For the σ_o^N series that were investigated, there were never more than 3% of the 4950 gene pairs that ended up as outliers (Table. 4.3). How many percentages of the strongly correlated gene pairs that this accounted for was not found explicitly, but since the compared σ_o^N series had between 7% and 44% of the outlier gene pairs in common, which gene pairs that ended up as outliers could not be random. The σ^T distribution (Fig. 4.2) also showed that the numbers of outliers made up a significant proportion of the number of genes with low σ, especially for the combinations of low o and N. All the σ_o^N series were not investigated due to time constraints, but both high and low o and N values were represented, and the trend was the same for all of them, indicating that the ”choice” of outlier gene pairs had more to do with their internal correlation as such, than the data set size or maximum overlap between sub-samples. The same trend was seen from the coefficient of variation for the 28 gene pairs where the σ was computed 100 independent times (Fig. 4.3b). The fact that highly and most uniformly correlated gene pairs had a bigger relative variability is a significant disadvantage which can be assumed to apply to both sub-sampling algorithms.

While the variations in σ caused by random sub-sampling are bound create some noise, it would most likely not be devastating for the CSD network. The variations in σ seen for the 28 gene pairs with various correlations tested 100 times each (Figs. 4.3a,A1) were not so big that the most extreme results of the highest and lowest σ values were overlapping, so a gene pair with very low internal correlation variation would never be mistaken for having a very high internal variation and vice versa due to this factor. Single links (gene pairs) are normally less interesting than network structures involving several gene pairs, such as hubs or clusters. Unless σ is maximally overestimated for the majority of the genes connected to a big network hub, or for the majority of the gene pairs making up a cluster, these network structures would not disappear as a consequence of slightly bad variation estimates. Because the sub-sampling is completely random, it is also highly unlikely that this would happen by chance.

How big deviation from the true value each outlier represented varied greatly, as the boundaries of the whiskers in each boxplot were determined by the interquartile ranges of the given box. The most extreme outlier of a narrow boxplot could therefore be a much better estimate of σ than a different σ lying within the box in a broad plot with a big

interquartile range. As the variation in σ due to different choices of sub-samples alone was shown to not be that big (Fig. 4.4, first boxplot), most of the deviations from the true σ values used as reference were attributed to the variations in N, S and o, which were also the parameters that were initially regarded to be of main interest. Some of the variation was also assumed to come from the fact that the correlation variability might truly change when a high number of data points are removed.

Each of the above mentioned parameters, N, S and o, had a clear impact on σ relative to the reference values, σ^T. Lowering N would at some point lead to a forced lowering of the number of sub-samples, S, that was possible to draw (Table 4.1). The lowering of S caused a significant overestimation of σ (Figs. 4.4-4.6), visible even with $S_{avg} = 162$, which was long before S_{avg} was close to $S_{min} = 50$. On the other hand, increasing the allowed overlap between sub-samples, o, caused a significant underestimation of σ relative to σ^T (Fig. 4.7). The overestimating effect of lowering S was clearly stronger than the underestimating effect of increasing o, as the upward shift and broadening of the boxplots with a lower S were much steeper and bigger than the downward shift and broadening of boxplots that followed an increase in o. Even with the parameters $S_{max} = 150$ (which would not be assumed to be a *big* decrease in S_{max}) and $o = 6$ (which was the maximum possible value of o for this sub-sample size), the boxplots were shifted upwards even with data sets as small as $N = 30$ (see Fig. 4.8), which was small enough that the effect of increasing o would clearly be relevant. The sub-sample size, n, was a fourth factor which could be analyzed in further studies of the method. In this study the sub-sample size was fixed at $n = 7$, assuming that for the sake of getting more sub-samples, it was best to have as small sub-samples as possible within the limits of the Spearman correlation calculations. Because a given o, for instance $o = 3$, would represent a different relative overlap with differently sized sub-samples, it is possible that one could benefit from increasing n to for instance 8 or 9, reducing the sub-sample dependence from a given o at the possible cost of poorer confounding identification. This is something that could be tested in the future.

The reason why increasing o had a bigger effect on smaller data sets was that a given o did not imply that the overlap was *fixed* at a given number of data points each time, but that any randomly drawn sub-sample would be discarded *if it happened to have* an overlap of more than o data points with any previously drawn sub-samples. This means that with a big enough N, setting $o = 6$ will not be likely to cause so many (if any) highly independent sub-samples, while the chance for randomly drawing overlapping sub-samples will increase with a low N, due to fewer data points to draw from. The average overlap between sub-samples is not known in any of the sub-sampling procedures, so it is not possible to know from the results of this study exactly *how* dependent the sub-samples were in each case, and how a known average overlap would affect the results. It was, however, possible to see at which N/o combinations the results started to deviate much from the reference. A consequence of the heightened effect of $o = 6$ on small data sets was also that a higher S_{max} might actually *worsen* the results, because a higher number of sub-samples will force more sub-samples to a higher dependence. For data sets with $N = 30$ or below, it might therefore be better to decrease S_{max} from 200 to for instance 150, as shown in Fig. 4.8.

Lowering N leads to a higher dependence between sub-samples, increasing the underestimating effect that a high o has on σ, while at the same time forcing S to be lower,

increasing the overestimating effect on σ. In both the overestimating case of low S and low o, and the underestimating case of high S and high o, it is therefore not possible to use the boxplots to separate the effect of either low S or high o from the effect of low N itself. The lowering of N not only meant lowering the number of sub-samples possible to draw, or heightening the average overlap between sub-samples if this was permitted, but it also meant a loss of data points. This could only be seen by the *t*-tests performed on the different σ_o^N series relative to σ^T (Table 4.2). The *t*-tests showed that the σ_6^{300} distribution was on average closer to the σ^T distribution than σ_1^{200} was, even though σ^T and σ_1^{200} had the same number of completely independent sub-samples, while σ_6^{300} possibly had some sub-sample dependence. This could not be seen from the boxplots, but meant that the loss of 100 data points was itself changing the σ distribution more than a high o did on a high N data set.

It is possible that, by chance, the data points that were removed in order to get smaller data sets had a significant difference in correlation compared to the remaining data points, either because of confounding factors or because of inaccurate measurements, meaning that not sub-sampling over this data would change the σ values enough for it to be detected by the *t*-test. However, this effect was not necessarily very big, because with many σ values in each group, a *t*-test would be able to detect very small changes as significant changes of the distributions. In this case, each group had almost 5000 σ_{ij} values in it. The fact that for instance σ_1^{200} seems to have a different distribution than σ^T does therefore not automatically mean that σ_1^{200} is a bad estimate of the internal correlation variability within the data set. Increasing the data set with 100 more measurements could just as easily have made a different distribution than σ^T, not meaning that 338 data points is too low. The *t*-test still shows the obvious: that either random noise or confounding can indeed play a role when calculating correlations, and no matter how the data is measured or how many times, one cannot get data from data points that are not there. One should therefore obviously try to get as many measurements as possible. The fact that the removed data points contribute to the overall σ when they are included could also make us question the choice of σ^T as reference for σ found in the smaller data sets, since these data sets might rightfully have different correlation variabilities. Since the problem with the small data sets is that you cannot find enough valid independent sub-samples, σ^T was considered to be the best option to use for reference purposes. However, it seems very unlikely that the difference in data points could produce the clear over- and underestimating trends in Figs. 4.4-4.7.

The consequences of over- and underestimating σ are similar, but not identical. Overestimation of σ would lead to a false impression that the gene pairs in the given data set have a higher internal variability in correlations than is actually the case. This will lower the associated C-, S- or D-scores unnecessarily. If the scores are lowered enough, the overestimation will lead to more false negative results, as it will result in the loss of significant links that should have been present in the CSD network. As previously discussed, an overall overestimation in just one of the two data sets can hide real confounding in the other data set, thereby also leading to false positives. If σ is underestimated in one data set, on the other hand, there might be an increase in false positives, but not in false negatives: One might get links in the network that should not have been present due to confounding factors that were not detected in the same network, but confounding factors can still be detected in the other data set. Overestimation of σ is therefore likely to have even more

severe consequences for the CSD network than underestimation.

The overestimating power of low S was also *stronger* than the underestimating power of high o, strengthening the conclusion that low S values are much worse for the final network than high o values. If you have to choose, it will be preferable to choose a high number of more dependent sub-samples rather than a low number of fully independent sub-samples. In this context it is worth noting that the original sub-sampling algorithm of the CSD method found fewer than 50 sub-samples for data sets of size $N = 60$ and $N = 50$. Seeing the huge deviations from σ^T that was found using on average 62 independent sub-samples at $N = 100$ (Table 4.1 and Fig. 4.4), the 34 independent sub-samples that the original sub-sampling method was able to identify from $N = 50$ data points seems to be far too low. The results of this study therefore indicate that the estimated lower limit of $N = 50$ for the original sub-sampling method may have been too optimistic.

Knowing that random drawing fails to find anything close to the maximum number of valid sub-samples no matter the sizes of N, S and o, it would have been very beneficial to find a systematic algorithm that could identify a higher number of sub-samples with up to a given number of overlapping data points, thereby increasing S while keeping o as low as possible. In addition to the increased possibility of drawing invalid sub-samples in itself, the initial random sub-sampling pattern can also possibly reduce the maximum number of valid sub-samples to be drawn later, due to the combination of sub-samples that are drawn in the beginning. Using a systematic algorithm, it could have been possible to come close to an exhaustive selection of sub-samples. This would allow even very small data sets to be used with a low dependence rather than a high dependence, which would most likely produce better results for these data sets than the random sub-sampling algorithm that was proposed in this thesis would. This was however not within the scope of this thesis, and is suggested for further research.

Another aspect of the method was the run-time of the script that was written and used for correlation and variance computation. As the two versions of the script were mainly written for use in this project, the requirement of finishing the computations within ”a reasonable amount of time” was set in order for the results to be finished within the time limits of the thesis. While this aim was met, it is obvious that it is costly, both in terms of time and energy consumption, to run 160 parallel processes for about one week, which was roughly the time spent on a full gene expression data set ($N = 28$ and $S_{max} = 150$) in the second part of the study. Although not within the scope of this project, I would therefore, for any future use, suggest implementing the new sub-sampling algorithm using for instance C++, which is much more efficient than Python. With faster computations, one could for instance also run the calculations multiple times and use the average of the obtained σ values, reducing the number of outliers in the results. Prefiltration of genes with a very high occurrence of zeros (instead of just filtering out gene pairs with $S < S_{min}$) could also have lowered the computation time, as each gene is possibly part of tens of thousands of gene pairs.

5.2 CSD Analysis of Rheumatoid Arthritis

Rheumatoid arthritis (RA) is a disease that is highly heterogeneous with a multitude of genetic and environmental factors affecting the pathogenesis and disease development. I

therefore expected it to be challenging to find clear disease related patterns in the CSD network. The network was however enriched with genes related to a high number of regulatory processes (Tables 4.9 and 4.10), and the strongest process enrichments were related to immune function, inflammation, growth, cell differentiation and morphology. Since RA is an autoimmune inflammatory disease associated with severe tissue remodelling and deformation, which in turn demands a high proliferating activity in the affected tissues, this suggests that the network represented real, meaningful relationships between the genes that were highlighted. The network followed a power-law degree distribution, showing that the network topology was far from random. Assuming that this non-randomness reflects some sort of biological function, it is likely that the network hubs play a special role in RA. It is, however, important to remember that the C-, S- and D- links do not correspond to gene regulatory networks or protein-protein interaction networks. The CSD network can therefore not be used alone to draw any conclusions about the exact regulatory mechanisms underlying the development or progression of RA.

The main topological differences between the C-, S- and D-type networks (see Fig. 4.11 and Figs. A5-A7) indicate a possible difference between the regulatory mechanisms leading to these link types. The C-network is weakly assortative (see Fig. 4.12), has a more narrow degree distribution and is somewhat densely connected. This is quite similar to what was seen in Voigt, Nowick and Almaas' CSD analysis of different brain regions [2], although their C-network was more strongly assortative and dense. Both the D- and S-networks are disassortative (see Figs. 4.13-4.14), but the more widespread S-network is much more disassortative than the very densely connected D-network. It is possible that the high disassortativity is due to the connectivity distribution alone (since the biggest hubs are S-dominated). This pattern is nevertheless not the same as what was found in [2], where the D-network was the most disassortative. The difference between the C-, S- and D-type networks are also reflected in the fact that they are enriched with genes connected to different groups of biological processes.

The C-network was highly enriched in processes related to T cell maturation and selection, in addition to various processes that are important for general cell function and maintenance, for instance ATP synthesis, RNA export from the nucleus and processes related to cell signaling and pathway regulation in general. A conserved correlation between genes related to general cell function and maintenance processes (maybe also to especially important regulatory pathways) can be interpreted as a particular robustness in the related mechanisms, beneficial for cell survival. The enrichment of genes related to T cell maturation in the C-network was more difficult to explain: T cells are essential for the immune system, and the MHC complex has consistently been shown to be associated with RA. The MHC related genes were, however, not found to have very prominent positions in the CSD network. It is not obvious why these genes appeared in a small C-component in the network, but considering that the MHC gene products build up a protein complex together, a tight co-regulation is not very strange. It is worth noting that this smaller component was connected to the giant component when the importance level for the network was increased (see Appendix A8). Although having conserved co-expression between them, this group of genes could nonetheless be regulated by other genes that has a changed regulation pattern in RA patients.

Another possible explanation for the enrichment of MHC related genes in the C-

network might be that the gene expression data were taken from joint synovial fluid samples, which is the site where the first inflammation occurs in RA patients. The maturation of T cells does not happen in the joints, but in the thymus, so when the T cells have been recruited to the joints, the T cell selection and maturation has already taken place. It could have been interesting to compare the CSD network from synovial gene expression with a similar RA CSD network based on, for instance, the blood samples or the thymus. It is also important to remember that RA does not involve a breakdown of the immune system as such. The problem is rather that the immune system attacks the patients' own cells in addition to foreign compounds. With this in mind, it does not seem unlikely that the relationship between some tightly connected immune system related genes could be conserved. The immune system is much more *active* in RA patients, but the CSD method does not detect changes in expression levels, so a conserved link does not mean that the *level* of gene activity is the same between the two conditions, only that the co-expression between these genes is conserved.

The S-network represented genes where a co-expression pattern has changed at one place or another: it might represent a loss of connection between two genes that are normally coordinated in healthy tissue, or the appearance of new connections that are normally not present. The S-network was enriched in processes related to different regulatory and signalling pathways, to various metabolic processes and energy production. Processes of particular interest included processes related to immune recognition and inflammation, ubiquitination (marking proteins for phagocytosis), cell division and stem cell differentiation, as well as DNA repair and telomere maintenance mechanisms, which should be highly active during cell proliferation. There were no very specific processes enriched in the S-network compared to the C- and D-networks, which was most likely the reason why the highest enrichment factors were generally lower in the S-network than in the other two. The S-dominated hubs had a high number of neighboring nodes that were not connected to any other nodes in the network. Seen together with the high betweenness centrality of the two S-dominated hubs and the high enrichment of regulatory processes in the S-network (as well as in the CSD network as a whole), this might indicate that these hubs play a role in coordinating a high number of other genes. Both *ZNF205-AS1* and *GPR18* has already been linked to regulatory functions. Not previously linked directly to RA, these seem like interesting genes to pay more attention to in further RA research. In order to get a deeper insight into the possible regulatory relationships and the loss or gain of biological functions, it would have been interesting to investigate which correlations were strong in the disease state and which were strong in the healthy tissue. The biggest hub in the network, *PDCD1*, had a high number of S-links to low degree neighbors, but it also had almost an equal number of D-links, and thus connected the S- and D-dominated parts of the CSD network by a substantial number of links. Not surprisingly, it also had the highest beweeenness and closeness centrality in the network. This gene has previously been linked to RA and, just like *ZNF205-AS1* and *GPR18*, this gene could also be hypothesized to have some coordinating role in RA.

The waste majority of hubs were however found in the D-network, and although the D-network was weakly disassortative, this was the link type that was most clearly limited to *one* area of the network. This could mean that the biological processes associated with the D-network were especially closely connected and possibly strongly regulated.

Although the D-network was also enriched in genes associated with the immune system, the feature that distinguished the D-network the most from the C- and S-networks was that it had a high occurrence of genes directly linked to specific tissue and organ development, as well as proliferation and growth of non-immune cells. The D-network could therefore possibly be assumed to be linked more tightly to the severe tissue remodelling and deformation processes that first affect the joints, and can later affect other tissues and organs in RA patients. Big, overreaching processes such as tissue and organ remodelling has to be highly organized and regulated, and this might be the reason why the D-type links are extra tightly grouped together. D-links did not reflect the loss of coordination, but an *altered* coordination between genes, possibly being related to regulatory malfunction rather than a complete halt of cellular processes. The enrichment of genes related to late stage symptoms of the disease imply a possible connection between the D-dominated hubs and the deformities seen in later stages of RA. Apart from one of them being directly associated with decreased kidney function, arterial calcification and osteoarthritis, the D-dominated hubs were nonetheless highly associated with the immune system, inflammations and autoimmune diseases, again emphasizing the interconnectedness of the different processes.

Several of the network hubs were also associated with cancer. Since cancer involves unimpeded growth, one could hypothesize that defects in these genes, or in the regulation of them may cause the abnormal growth and tissue remodelling seen in RA patients. One should, however, be careful about drawing such conclusions, as no gene is *only* an oncogene, but always has another function that could possibly be altered and lead to other cellular consequences than abnormal growth. Two of the network hubs were RNA genes, and although more difficult to find information about, they should not be discarded as irrelevant. They could be related to relevant regulatory mechanisms, and were therefore included in the analysis.

A strength of the method is that both the overall correlation and weighting of correlation variability is taken into account when calculating the C-, S- and D-scores. This ensures that only gene pairs with consistent co-expression across most of the measurements are highlighted in the final network. All genes, and especially those of the S- and D-networks, would therefore be of interest in further studies of the disease mechanisms of RA. It could also be of further interest to compare the CSD network to a protein interaction network, in order to see how the products of the different genes found in the CSD network actually interact. Although the genes in the network could be assumed to be of direct relevance to RA, there is still a chance that significant genes are not present in the network: S- and D-type links could be lost due to a high occurrence of zeros in the data set. As discussed previously, this is especially challenging with S-type links. An additional source of noise in this particular analysis was the new sub-sampling algorithm used on the reference data set. The fact that the resulting network did still show a strong connection to RA supports the assumption that the new sub-sampling approach did not have too grave consequences for the structure of the CSD network. It is still worth being aware that some links might have been included that should not have been, and vice versa.

The correction for correlation variability could itself have a downside as well: It is presupposed that a high correlation variability is caused by confounding factors, ignoring the fact that some diseases, like RA, are heterogeneously expressed in different patients. It is impossible to tell if a high correlation variability within a data set is caused by real

confounding factors or by heterogeneous expression of the given condition. Diseases are, in general, first characterized based on their phenotype outcome, and not on their underlying molecular mechanisms, since these are more difficult to identify. In some cases it might therefore be multiple molecular mechanisms leading to the same disease or syndrome, making it difficult to identify co-exression patterns that correspond to anything meaningful in a biological context. If multiple molecular mechanisms can lead to the same phenotype, gene correlations could become less uniform, and genes and correlations that are indeed related to the disease could end up being filtered out from the network. This would, however, be a challenge with any systems biology approach, and is not an exclusive disadvantage of the CSD method. On the upside, this also means that the CSD network could be assumed to reflect only the factors that different RA patients have in common, possibly capturing the most essential features of the disease. Since the network showed so clear associations with immune function and inflammation, it could be assumed to be of high relevance for understanding the genetic mechanisms underlying at least some aspects of RA.

Chapter 6

Conclusion and Outlook

The implementation of an alternative sub-sampling algorithm for the CSD method, allowing dependence between sub-samples, showed that a low number of sub-samples had a strong overestimating effect on the correlation variability, while sub-sample dependence had a weaker underestimating effect. A low number of sub-samples thus had a higher negative impact on the correlation variabilities than a higher sub-sample dependence did. In order to get as good estimates as possible of the correlation variabilities of small data sets, the number of sub-samples had to be weighed against the maximum permitted overlap.

Based on the results of this study, data sets smaller than 50 data points are recommended to have about 150 sub-samples if the alternative algorithm is used, especially if a high sub-sample dependence is allowed. Bigger data sets are recommended a higher number of sub-samples, for instance 200. The maximum sub-sample overlap should in general be kept as low as possible within these requirements. The smaller data sets are recommended to have fewer sub-samples to reduce the impact of the high dependence that is required to find a sufficient number of sub-samples. For data sets with 60 or fewer data points per gene allowing some sub-sample dependence is in general suggested in order to increase the number of sub-samples, independent of sub-sampling approach. For data sets smaller than the limitations set by the original CSD method, which was 49 data points per gene, it was not possible to find enough sub-samples with a low overlap using the new algorithm, but when allowing for highly dependent sub-samples, a sufficient number was easily found for data sets as small as 20 data points per gene. Allowing a high overlap between sub-samples gave significantly more noisy results, but could nevertheless be a justifiable option for small data sets.

Developing an algorithm for the deterministic identification of sub-samples with a low number of overlapping data points is suggested for further research, as this would give better results for small data sets and reduce noise in the CSD network. More research is also suggested in order to further investigate the trade-off between reduction of over-estimation by *increasing* the number of sub-samples, and reduction of under-estimation by *decreasing* the same number of sub-samples (in order to reduce the average sub-sample dependence). Finding a general rule for the combination of sub-sample number and overlap for a given

data set size would have made the new sub-sampling algorithm more easily applicable on new small data sets. The alternative sub-sampling algorithm should also preferably be implemented in a more efficient programming language, in order for the computations to be less time and energy consuming.

Two data sets of gene expression data from the synovial fluid of rheumatoid arthritis (RA) patients and healthy controls were analyzed using the CSD method with the alternative sub-sample algorithm, and resulted in a CSD network highly enriched with genes related to immune function, inflammation and differentiation, growth and morphology of cells, tissues and organs. Eleven hub genes were identified in the network: *PDCD1*, *ZNF205-AS1*, *GPR18*, *LINC00426*, *SEPT1*, *CTLA4*, *ASAP2*, *ENPP1*, *IL2RG*, *PRDM1* and *GBP5*. Eight of them, *ZNF205-AS1*, *GPR18*, *LINC00426*, *SEPT1*, *ASAP2*, *ENPP1*, *IL2RG* and *GBP5* have, to my knowledge, not previously been linked to RA. Two of them, *ASAP2* and *ENPP1*, were directly linked to at least one known RA associated gene each. Per the data presented, these eleven genes can be central in RA progression, and are thus possible candidates for further research of the disease. The three most highly connected hubs, *PDCD1*, *ZNF205-AS1* and *GPR18*, contributed strongly to a disassortative network pattern and were hypothesized to have a coordinating role between a high number of other genes. The *PDCD1* gene, which was the biggest network hub, was found to connect the parts of the network dominated by either specific or differentiated co-expression respectively, and had previously been associated with rheumatoid arthritis. The results of this analysis further supported the position of this gene in RA. The remaining eight genes, *LINC00426*, *SEPT1*, *CTLA4*, *ASAP2*, *ENPP1*, *IL2RG*, *PRDM1* and *GBP5*, were positioned in a sub-network dominated by differentiated co-expression between genes. The differentiated links were highly interconnected and genes connected to cell differentiation, growth, tissue and organ development and morphology were over-represented in this part of the CSD network. These genes are hypothesized to be involved in later stages of the disease development.

More research is clearly needed in order to understand the genetic mechanisms underlying the development of rheumatoid arthritis. As RA is a heterogeneous disease, a possible next step could be to divide the gene expression samples into different groups based on differences on phenotype, such as samples from ACPA positive and ACPA negative RA patients, and either compare them to each other directly with a differential co-expression analysis, or combine both groups with healthy control samples to make multiple CSD networks that could be compared to each other. In this way one could possibly get more uniform correlation patterns in each data set, and maybe find interesting gene relationships that were not visible in this study due to high correlation variabilities. Furthermore, one could have studied the S-links of the CSD network in more detail to find out which correlations were present in which condition. It could also have been interesting to compare multiple CSD networks based on different tissue types from RA patients and controls, for instance both synovial fluid samples and blood samples. This could possibly capture different stages or aspects of the disease. The genes identified in this study could also have been compared to other types of networks, such as gene regulatory networks or protein interaction networks in humans.

Reference list

[1] Barabási A. Network Science [Internet]. Cambridge University Press; 2016 [cited 2018 June 10]. Available from: http://networksciencebook.com/.

[2] Voigt A, Nowick K, Almaas E. A composite network of conserved and tissue specific gene interactions reveals possible genetic interactions in glioma. PLoS Comput Biol. 2017;13(9):e100573. Available from: https://doi.org/10.1371/journal.pcbi.1005739.

[3] Alon U. An introduction to systems biology. New York: Chapman & Hall/CRC; 2007.

[4] Kerem B, Rommens JM, Buchanan JA, Markiewicz D, Cox TK, Chakravarti A, et al. Identification of the cystic fibrosis gene: genetic analysis. Science. 1989;245(4922):1073–1080. Available from: http://dx.doi.org/10.1126/science.2570460.

[5] McInnes IB, Schett G. Pathogenetic insights from the treatment of rheumatoid arthritis. Lancet. 2017;389(10086):2328–37. Available from: https://doi.org/10.1016/S0140-6736(17)31472-1.

[6] Terao C, Raychaudhuri S, Gregersen PK. Recent Advances in Defining the Genetic Basis of Rheumatoid Arthritis. Annu Rev Genom Hum G. 2016;17(10086):273–301. Available from: https://doi.org/10.1146/annurev-genom-090314-045919.

[7] Aoki K, Ogata Y, Shibata D. Approaches for Extracting Practical Information from Gene Co-expression Networks in Plant Biology. Plant Cell Physiol. 2007;48(3):381–390. Available from: https://doi.org/10.1093/pcp/pcm013.

[8] Kharachenko P, Church GM, Vitkup D. Expression dynamics of a cellular metabolic network. Mol Syst Biol. 2005;1(1). Available from: https://doi.org/10.1038/msb4100023.

[9] Jansen R, Greenbaum D, Gerstein M. Relating Whole-Genome Expression Data with Protein-Protein Interactions. Genome Res. 2002;12(1):37–46. Available from: https://doi.org/10.1101/gr.205602.

[10] Yang Y, Han L, Yuan Y, Li J, Hei N, Liang H. Gene co-expression network analysis reveals common system-level properties of prognostic genes across cancer types. Nat Commun. 2014;5(3231). Available from: https://doi.org/10.1038/ncomms4231.

[11] NCBI. Gene Expression Omnibus[Internet]. Bethesda, MD: NCBI;. 2018 [cited: 23.01.2018]. Available from: https://www.ncbi.nlm.nih.gov/geo/.

[12] Institute B. GTEx Portal[Internet]. Boston, MA: The Broad Institute of MIT and Harvard;. 2018 [cited: 15.02.2018]. Available from: https://www.gtexportal.org/home/.

[13] McInnes IB, Schett G. The Pathogenesis of Rheumatoid Arthritis. N Engl J Med. 2011;365:2205–2219. Available from: https://dx.doi.org/10.1056/NEJMra1004965.

[14] Kochi Y, Suzuki A, Yamamoto K. Genetic basis of rheumatoid arthritis: A current review. Biochem Bioph Res Co. 2014;452(2):254–262. Available from: https://doi.org/10.1016/j.bbrc.2014.07.085.

[15] Janeway CA, Travers P, Walport M, Shlomchik MJ. Immunobiology. 5th ed. New York: Garland Science; 2001.

[16] Newton JL, Harney SMJ, Wordsworth BP, Brown MA. A review of the MHC genetics of rheumatoid arthritis. Genes and Immun. 2004;5:151–157. Available from: https://doi.org/10.1038/sj.gene.6364045.

[17] Kurkó J, Besenyei T, Laki J, Glant TT, Mikecz K, Szekanecz Z. Genetics of Rheumatoid Arthritis: A Comprehensive Review. Clin Rev Allergy Immunol. 2013;45(2):170–179. Available from: https://doi.org/10.1007/s12016-012-8346-7.

[18] Catrina AI, Joshua V, Klareskog L, Malmström V. Mechanisms involved in triggering rheumatoid arthritis. Immunol Rev. 2016;269:162–174. Available from: https://doi.org/10.1111/imr.12379.

[19] Wang Z, Gerstein M, Snyder M. RNA-Seq: a revolutionary tool for transcriptomics. Nat Rev Genet. 2009;10(1):57–63. Available from: https://dx.doi.org/10.1038%2Fnrg2484.

[20] Purves D, Augustin G, Fitzpatrick D, Hall WC, Lamantia A, White LE. Neuroscience. 5th ed. Sunderland, MA: Sinauer; 2012.

[21] Voit EO. A first course in systems biology. New York, NY: Garland Science, Taylor & Francis Group; 2013.

[22] Barrat A, Barthélemy M, Pastor-Sarrotas M, Vespignani A. The architecture of complex weighted networks. PNAS. 2004;101(11):67–92. Available from: https://doi.org/10.1142/9789812771681_0005.

[23] Horwath S. Weighted Network Analysis. New York, NY: Springer; 2011.

[24] Norušis MJ. IBM SPSS Statistics 19 Guide to Data Analysis. Upper Saddle River, NJ 07458: Prentice Hall Inc.; 2011.

[25] Taylor JR. An introduction to Error Analysis. The study of uncertainties in physical measurements. 2nd ed. University Science Book; 1997.

[26] Welford BP. Note on a method for calculating corrected sums of squares and products. Technometrics. 1962;4(3):419–420. Available from: https://doi.org/10.2307/1266577.

[27] McGill R, Tukey JW, Larsen WA. Variations of Box Plots. Am Stat. 1978;32(1):12–16. Available from: https://doi.org/10.2307/2683468.

[28] Larsen RJ, Marx ML. Introduction to Mathematical Statistics and Its Applications. 5th ed. Essex: Pearson Education Limited; 2014.

[29] Dubitzky W, Wolkenhauer O, Yokota H, Cho K, editors. Encyclopedia of Systems Biology. New York: Springer-Verlag; 2013.

[30] Benjamini Y, Hochberg Y. Controlling the False Discovery Rate: a Practical and Powerful Approach to Multiple Testing. JR Statist Soc B. 1995;57(1):289–300. Available from: http://dx.doi.org/10.2307/2346101.

[31] Choi JK, Yu U, Yoo OJ, Kim S. Differential coexpression analysis using microarray data and its application to human cancer. Bioinformatics. 2005;21(24):4348–4355. Available from: https://doi.org/10.1093/bioinformatics/bti722.

[32] Reverter A, Ingham A, Lehnert SA, Tan S, Wang Y, Ratnakumar A, et al. Simultaneous identification of differential gene expression and connectivity in inflammation, adipogenesis and cancer. Bioinformatics. 2006;22(19):2369–2404. Available from: https://doi.org/10.1093/bioinformatics/btl392.

[33] Liu B, Yu H, Tu K, Li C, Li Y. DCGL: an R package for identifying differentially coexpressed genes and links from gene expression microarray data. Bioinformatics. 2010;26(20):2637–2638. Available from: https://doi.org/10.1093/bioinformatics/btq471.

[34] Gao X, Arodz T. Detecting Differentially Co-expressed Genes for Drug Target Analysis. Procedia Comput Sci. 2013;18:1392–1401. Available from: https://doi.org/10.1016/j.procs.2013.05.306.

[35] Zheng C, Yuan L, Sha W, Sun Z. Gene differential coexpression analysis based on biweight correlation and maximum clique. BMC Bioinf. 2014;15(Suppl 15):S3. Available from: https://doi.org/10.1186/1471-2105-15-S15-S3.

[36] Amar D, Safer H, Shamir R. Dissection of Regulatory Networks that Are Altered in Disease via Differential Co-expression. PLoS Comput Biol. 2013;9(3):e1002955. Available from: https://doi.org/10.1371/journal.pcbi.1002955.

[37] Fukushima A. DiffCorr: An R package to analyze and visualize differential correlations in biological networks. Gene. 2013;518(1):209–214. Available from: https://doi.org/10.1016/j.gene.2012.11.028.

[38] Lonsdale J, Thomas J, Salvatore M, Phillips R, Lo E, Shad S, et al. The genotype-tissue expression (GTEx) project. Nature Genetics. 2013;45(6):580–585.

[39] Edgar R, Domrachev M, Lash AE. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. Nucleic Acids Res. 2002;30(1):207–210.

[40] Helland MO. Implementation and Application of Method for Differential Correlation Network Analysis [Master thesis]. Trondheim: NTNU; 2017. Available from: https://github.com/magnusolavhelland/CSD-Software.

[41] Guo Y, Walsh AM, Fearon U, Smith MD, Wechalekar MD, Yin X, et al. CD40L-Dependent Pathway Is Active at Various Stages of Rheumatoid Arthritis Disease Progression. J Immunol. 2017;198(11):4490–4501. Available from: https://doi.org/10.4049/jimmunol.1601988.

[42] Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, et al. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat Genet. 2000;25(1):25–29. Available from: http://dx.doi.org/10.1038/75556.

[43] S Carbon S, Mungall CJ, Munoz-Torres MC, Thomas PD, Courtot M, Roncaglia P, et al. Expansion of the Gene Ontology knowledgebase and resources. Nucleic Acids Res. 2017;45(D1):D331–D338. Available from: https://doi.org/10.1093/nar/gkw1108.

[44] Mi H, Huang X, Muruganujan A, Tang H, Mills C, Kang D, et al. PANTHER version 11: expanded annotation data from Gene Ontology and Reactome pathways, and data analysis tool enhancements. Nucleic Acids Res. 2017;45(D1):D183–D189. Available from: https://doi.org/10.1093/nar/gkw1138.

[45] Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, et al. Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks. Genome Res. 2003;13(11):2498–2504.

[46] Marshall C. How do small GTPase signal transduction pathways regulate cell cycle entry? Curr Opin Cell Biol. 1999;11(6):732–736. Available from: https://doi.org/10.1016/S0955-0674(99)00044-7.

[47] Fingera LR, Pua J, Wassermandb R, Vibhakara R, Louiec E, Hardyd RR, et al. The human PD-1 gene: complete cDNA, genomic organization, and developmentally regulated expression in B cell progenitors. Gene. 1997;197(1-2):177–187. Available from: https://doi.org/10.1016/S0378-1119(97)00260-6.

[48] Fife BT, Pauken KE. The role of the PD-1 pathway in autoimmunity and peripheral tolerance. Ann N Y Acad Sci. 2011;1217:45–59. Available from: https://doi.org/10.1111/j.1749-6632.2010.05919.x.

[49] Prokunina L, Padyukov L, Bennet A, de Faire U, Wiman B, Prince J, et al. Association of the PD-1.3A allele of the PDCD1 gene in patients with rheumatoid arthritis negative for rheumatoid factor and the shared epitope. Arthritis Rheum. 2004;50(6):1770–1773. Available from: https://doi.org/10.1002/art.20280.

[50] Prokunina L, Castillejo-López C, Oberg F, Gunnarsson I, Berg L, Magnusson V, et al. A regulatory polymorphism in PDCD1 is associated with susceptibility to systemic lupus erythematosus in humans. Nat Genet. 2002;32(4):666–669. Available from: https://doi.org/10.1038/ng1020.

[51] Kohno M, Hasegawa H, Inoue A, Muraoka M, Miyazaki T, Oka K, et al. Identification of N-arachidonylglycine as the endogenous ligand for orphan G-protein-coupled receptor GPR18. Biochem Bioph Res Co. 2006;347(3):827–832. Available from: https://doi.org/10.1016/j.bbrc.2006.06.175.

[52] McHugh D, Hu SSJ, Rimmerman N, Juknat A, Vogel Z, Walker JM, et al. *N*-arachidonoyl glycine, an abundant endogenous lipid, potently drives directed cellular migration through GPR18, the putative abnormal cannabidiol receptor. BMC Neuroscience. 2010;11(44). Available from: https://doi.org/10.1186/1471-2202-11-44.

[53] Pérez BF, Codner DE, Angel BB, Balic NI, Carrasco PE. +49 A/G genetic polymorphism of cytotoxic T lymphocyte associated antigen 4 (CTLA-4) in type 7 diabetes: association with autoantibodies and cytokines. Rev Med Chil. 2009;137(3):321–328. Available from: https://doi.org/10.3892/ijo.31.5.1021.

[54] Jacobson EM, Tomer Y. The CD40, CTLA-4, thyroglobulin, TSH receptor, and PTPN22 gene quintet and its contribution to thyroid autoimmunity: Back to the future. J Autoimmun. 2007;28(2–3):85–98. Available from: https://doi.org/10.1016/j.jaut.2007.02.006.

[55] Kalman L, Lindegren ML, Kobrynski L, Vogt R, Hannon H, Howard JT, et al. Mutations in genes required for T-cell development: IL7R, CD45, IL2RG, JAK3, RAG1, RAG2, ARTEMIS, and ADA and severe combined immunodeficiency: HuGE review. Genet Med. 2004;6(1):16–26. Available from: https://doi.org/10.1097/01.GIM.0000105752.80592.A3.

[56] Puck JM, Pepper AE, Bédard PM, Laframboise R. Female germ line mosaicism as the origin of a unique IL-2 receptor gamma-chain mutation causing X-linked severe combined immunodeficiency. J Clin Invest. 1995;95(2):895–899. Available from: https://doi.org/10.1172/JCI117740.

[57] Keller AD, Maniatis T. Identification and characterization of a novel repressor of beta-interferon gene expression. Genes Dev. 1991;5(5):868–87. Available from: https://doi.org/10.1101/gad.5.5.868.

[58] Ying HY, Su ST, Hsu PH, Chang CC, Lin IY, Tseng YH, et al. SUMOylation of Blimp-1 is critical for plasma cell differentiation. EMBO Rep. 2012;13(7):631–637. Available from: https://doi.org/10.1038/embor.2012.60.

[59] Eyre S, Bowes J, Diogo D, Lee A, Barton A, Martin P, et al. High-density genetic mapping identifies new susceptibility loci for rheumatoid arthritis. Nat Genet. 2012;44:1336–1340. Available from: https://doi.org/10.1038/ng.2462.

[60] Krishnan C, Warnke RA, Arber DA, Natkunam Y. PD-1 expression in T-cell lymphomas and reactive lymphoid entities: potential overlap in staining patterns between lymphoma and viral lymphadenitis. Am J Surg Pathol. 2010;34(2):178–189. Available from: https://doi.org/10.1097/PAS.0b013e3181cc7e79.

[61] Seuter S, Ryynänen J, Carlberg C. The ASAP2 gene is a primary target of 1,25-dihydroxyvitamin D3 in human monocytes and macrophages. J Steroid Biochem Mol Biol. 2014;144(A):12–18. Available from: https://doi.org/10.1016/j.jsbmb.2013.08.014.

[62] Heath RJ, Xavier RJ. Autophagy, Immunity and Human Disease. Curr Opin Gastroenterol. 2009;25(6):512–520. Available from: https://doi.org/10.1097/MOG.0b013e32833104f1.

[63] Wang H, Gonzalez-Garcia I, Traba J, Jain S, Conteh S, Shin DM, et al. ATP-degrading ENPP1 is required for survival (or persistence) of long-lived plasma cells. Sci-Rep UK. 2017;7(17867):2045–2322. Available from: https://doi.org/10.1038/s41598-017-18028-z.

[64] Suk EK, Malkin I, Dahm S, Kalichman L, Ruf N, Kobyliansky E, et al. Association of ENPP1 gene polymorphisms with hand osteoarthritis in a Chuvasha population. Arthritis Res Ther. 2005;7(5):R1082–1090. Available from: https://doi.org/10.1186/ar1786.

[65] Keene KL, Mychaleckyj JC, Smith SG, Leak TS, Perlegas PS, Langefeld CD, et al. Association of the distal region of the ectonucleotide pyrophosphatase/phosphodiesterase 1 gene with type 2 diabetes in an African-American population enriched for nephropathy. Diabetes. 2008;57(4):1057–1062. Available from: https://doi.org/10.2337/db07-0886.

[66] De Cosmo S, Minenna A, Zhang YY, Thompson R, Miscio G, Vedovato M, et al. Association of the Q121 variant of ENPP1 gene with decreased kidney function among patients with type 2 diabetes. Am J Kidney Dis. 2009;53(2):273–280. Available from: https://doi.org/10.1053/j.ajkd.2008.07.040.

[67] Lorenz-Depiereux B, Schnabel D, Tiosano D, Häusler G, Strom TM. Loss-of-function ENPP1 mutations cause both generalized arterial calcification of infancy and autosomal-recessive hypophosphatemic rickets. Am J Hum Genet. 2010;86(2):267–272. Available from: https://doi.org/10.1016/j.ajhg.2010.01.006.

[68] Hall PA, Jung K, Hillan KJ, Russell SE. Expression profiling the human septin gene family. J Pathol. 2005;206(3):269–278. Available from: https://doi.org/10.1002/path.1789.

[69] Qi M, Yu Q, Liu S, Jia H, Tang L, Shen M, et al. Septin1, a new interaction partner for human serine/threonine kinase aurora-B. Biochem Biophys Res Co. 2005;336(3):994–1000. Available from: https://doi.org/10.1016/j.bbrc.2005.06.212.

[70] Kato Y, Uzawa K, Y, Yokoe H, Shibahara T, Tanzawa H. Expression profiling the human septin gene family. Int J Oncol. 2007;31(5):1021–1028. Available from: https://doi.org/10.3892/ijo.31.5.1021.

[71] Shenoy AR, Wellington DA, Kumar P, Kassa H, Booth CJ, Creswell P, et al. GBP5 promotes NLRP3 inflammasome assembly and immunity in mammals. Science. 2012;336(6080):481–485. Available from: https://doi.org/10.1126/science.1217141.

[72] Wehner M, Herrmann C. Biochemical properties of the human guanylate binding protein 5 and a tumor-specific truncated splice variant. FEBS J. 2010;277(7):1597–1605. Available from: https://doi.org/10.1111/j.1742-4658.2010.07586.x.

[73] Li S, Huang S, Peng SB. Overexpression of G protein-coupled receptors in cancer cells: Involvement in tumor progression. Int J Oncol. 2005;27(5):1329–1338. Available from: https://doi.org/10.3892/ijo.27.5.1329.

[74] Iida R, Ueki M, Yasuda T. Identification of Rhit as a novel transcriptional repressor of human Mpv17-like protein with a mitigating effect on mitochondrial dysfunction, and its transcriptional regulation by FOXD3 and GABP. Free Radical Bio Med. 2012;52(8):1413–1422. Available from: https://doi.org/10.1016/j.freeradbiomed.2012.01.003.

[75] Fujii H, Shao L, Colmegna I, Goronzy JJ, Weyand CM. Telomerase insufficiency in rheumatoid arthritis. PNAS. 2009;106(11):4360–4365. Available from: https://doi.org/10.1073/pnas.0811332106.

[76] Stern JL, Zyner KG, Pickett HA, Cohen SB, Bryan TM. Telomerase Recruitment Requires both TCAB1 and Cajal Bodies Independently. Mol Cell Biol. 2012;32(13):2384–2395. Available from: https://doi.org/10.1128/MCB.00379-12.

[77] Koetz K, Bryl E, Spickschen K, O'Fallon WM, Goronzy JJ, Weyand CM. T cell homeostasis in patients with rheumatoid arthritis. Proc Natl Acad Sci U S A. 2000;97(16):9203–9208. Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC16846/.

[78] Triola G. The Protein Lipidation and Its Analysis. J Glycom Lipidom. 2011;S2(001). Available from: https://doi.org/10.4172/2153-0637.S2-001.

[79] Voeltz GK, Rolls MM, Rapoport TA. Structural organization of the endoplasmic reticulum. EMBO Rep. 2002;3(10):944–950. Available from: https://dx.doi.org/10.1093%2Fembo-reports%2Fkvf202.

[80] Buttgereit F, Welhing M, Burmester GR. A New Hypothesis of Modular Glucocorticoid Actons: Steroid Treatment of Rheumatic Diseases Revisited. Arthritis Rheum-US. 1998;41(5):761–767. Available from: https://doi.org/10.1002/1529-0131(199805)41:5%3C761::AID-ART2%3E3.0.CO;2-M.

[81] Hohensinner PJ, Goronzy JJ, Weyand CM. Telomere Dysfunction, Autoimmunity and Aging. Aging Dis. 2011;2(6):524–537. Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3295061/.

[82] Wang A F Lerman, Herrmann J. Dysfunction of the ubiquitin-proteasome system in atherosclerotic cardiovascular disease. Am J Cardiovasc Dis. 2015;5(1):83–100. Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4447079/.

[83] Toyomoto M, Ishido S, Sugimoto H, Kohsaka H. Anti-arthritic effect of E3 ubiquitin ligase, c-MIR, expression in the joints. Int Immunol. 2011;23(3):177–183. Available from: https://doi.org/10.1093/intimm/dxq470.

[84] Wang H, Guo J, Jiang J, Wu W, Chang X, Zhou H, et al. New genes associated with rheumatoid arthritis identified by gene expression profiling. Int J Immunogenet. 2017;44(3):107–113. Available from: https://doi.org/10.1111/iji.12313.

Appendices

A1 Genes Previously Associated with RA

Table A1 gives an overview of 53 genes found to be associated with RA in previous studies.

Table A1: List of 53 genes previously associated with rheumatoid arthritis.

HLA-DRB1[14]	*PTPN22* [17]	*TRAF1* [17]	*CTLA4* [17]	*IRF5* [17]
STAT4 [17]	*CCR6* [17]	*PADI4* [17]	*IL23R* [17]	*PTPN22* [17]
SAP30BP [84]	*LPIN2* [84]	*DHRS3* [84]	*TTC38* [84]	*IRAK1* [59]
TLE3 [59]	*RASGRP1* [59]	*IL6R* [59]	*IRF8* [59]	*AIRD5B* [59]
IKZF3 [59]	*RUNX1* [59]	*POU3F* [59]	*RCAN1* [59]	*ANKRD55* [59]
RBPJ [59]	*CCL21* [59]	*CD40* [59]	*MMEL1* [59]	*AFF3* [59]
REL [59]	*GIN1* [59]	*DNASE1L3* [59]	*IL2RB* [59]	*DDX6* [59]
SPRED2 [59]	*TAGAP* [59]	*IL2RA* [59]	*CD2* [59]	*CD28* [59]
PTPRC [59]	*KIF5A* [59]	*PRKCQ* [59]	*FCGR2A* [59]	*PRDM1* [59]
IL2-IL21 [59]	*TRAF6* [59]	*TYK2* [59]	*FCRL3* [6]	*CD244* [6]
BLK [6]	*TNFAIP3* [6]	*SLC22A4* [6]		

A2 Genes Used for Signal Plot and Signal Stability Analysis

The following list of Ensembl IDs identify the 8 genes used for the signal plot and signal stability analysis:

- ENSG00000223972.4
- ENSG00000238009.2
- ENSG00000227232.4
- ENSG00000233750.3
- ENSG00000268903.1
- ENSG00000248527.1
- ENSG00000188976.6
- ENSG00000131591.13

A3 Stability of σ_6^{338} Computed 100 Times for 28 Gene Pairs

Fig. A1 displays boxplots of the final σ_{ij} values for 28 gene pairs *ij* obtained from a series of 100 repeated computations of σ per gene pair, using the parameters $S_{max} = 200$, $N = 338$ and $o = 6$. The gene pairs were all combinations of pairs of the genes given in Appendix A2 and chosen so that a variety of correlations were represented.

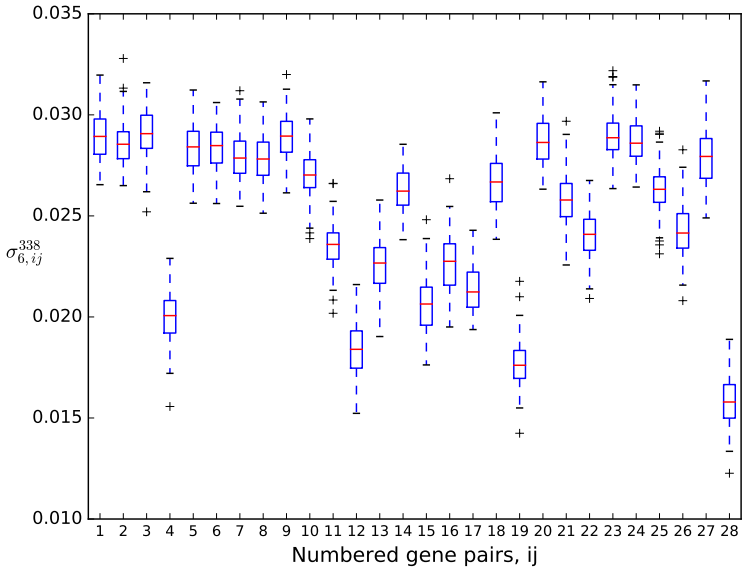


Figure A1: Boxplots of the final σ_{ij} values obtained from 100 repeated runs of my script on 28 different gene pairs with various correlations represented. Each plot represents one gene pair. Parameters: $S_{max} = 200$, $N = 338$, $o = 6$.

A4 Scripts for Computation of Gene Pair Correlation and Correlation Variability in a Data Set

The two following scripts were used for the calculation of Spearman correlation coefficients of the gene expression of all gene pairs in a data set, and for the sub-sampling and computation of correlation variability in the data set as the standard deviation of the mean of sub-sample correlations.

The first script was used if the sub-samples consisting of 7 data points were allowed to have up to 1, 2 or 3 data points overlap between them, and the standard deviation was calculated separately for the three degrees of permitted overlap respectively. The second script was used if the same sized sub-samples were allowed up to 6 data points overlap.

Both scripts take input files on the following format: The first (index) column contains G gene names, the first (header) row contains N sample names, the remaining rows and columns contain N gene expression measurements for each of the G genes. No entries should contain NaN, None or be empty, or the relevant gene could get an error message in the output file instead of correlation and variability measurements.

Allowed sub-sample overlap from 1 to 3 data points

```
from __future__ import division
import numpy as np
import pandas as pd
import math, random
import itertools
from scipy import stats
from multiprocessing import Pool

#############################################################################
#
# This script takes a gene expression data set as input, and computes
# Spearman correlation coefficients and variance/standard deviation of
# the mean (SDOM) of the correlations of sub-samples in the data set.
# The sub-samples have maximum overlap of 1, 2 and 3 data points, and
# SDOM/variances are found for all three options.
#
# Three parameters has to to be filled in by the user:
# The name of the input file (inputfile), the name of the output
# (outputfile), and the number of cores on the machine (nCores).
# If variance should be written
# to file instead of SDOM, this has to be changed inside the
# genepairFunction.
#
#############################################################################

inputfile = 'WholeBlood338.txt'
outputfile = 'CorrSDOM.txt'
nCores = 24

#genepairFunction calculates Spearman correlation + mean correlation
# and SDOM/var for permitted overlaps of sub-samples set to 1,2 and 3.
def genepairFunction(genepair):
    #list to fill with results for genepair
    resList=[]
```

```
    resList.append(str(genepair))

    #Assign rows for the genes in genepair to g1 and g2.
    g1 = gedTable.loc[genepair[0]]
    g2 = gedTable.loc[genepair[1]]

    #Calculate Spearman for genepair across all columns
    RHO, ptot = stats.spearmanr(g1,g2)

    resList.append(str(RHO))

    #each k gives a new level of permitted overlap between the sub-samples
    for k in range(3):
        #Stores all pairs/triplets/fours of columns drawn together and
    used
        usedPairs = {}

        #Number of subset correlations calculated for this genepair
        N = 0
        #Sum of calculated corr. values
        corrSum = 0
        #Sum of squared calculated corr. values
        sqrSum = 0
        #Standard deviation of the mean, initial value
        SDOM = 0
        #Mean correlation start value
        corrMean = 0
        #Initial discard rate - what fraction of pickLists are discarded
        discRate=0
        #Number of discarded pickLists
        disc=0
        #tot = total number of pickLists (discared and used)
        tot=0


        while (discRate < 0.99 and N <= 199)  or tot < 20:

            pickList = []

            tot = tot + 1
            #Lists to fill with measurements for gene pair x,y
            xList = []
            yList = []

            #Put seven unique and random col indexes in pickList
            while len(pickList) < 7:
                pickIndex = int(random.uniform(0, len(gedTable.columns)-1)
    )

                if pickIndex not in pickList:
                    pickList.append(pickIndex)

            #Make unique pairs/triplets/fours of the seven indexes
            if k == 0:
                currentPairs = list(itertools.combinations(pickList,2))
            elif k == 1:
                currentPairs = list(itertools.combinations(pickList,3))
```

```
        else:
            currentPairs = list(itertools.combinations(pickList,4))

        #Fill xList and yList with measurements corresponding to the
two
        #genes and the columns given in pickList
        for i in pickList:

            xList.append(gedTable.loc[genepair[0]][i])
            yList.append(gedTable.loc[genepair[1]][i])

        #Count "pairs" (pairs/triplets/fours) in current pickList that
 have previously been drawn
        usedCount = 0
        for pair in currentPairs:
            if pair in usedPairs:
                usedCount += 1

        #If xList and yList contains enough nonzeros and pickList has
no
        #pair/triplet/four in common with previous draws, continue:
        if np.count_nonzero(xList)>4 and np.count_nonzero(yList)>4 and
 usedCount == 0:

            #Add all combinations of used "pairs", (pairs, triplets or
 fours) to currentPairs dict:
            if k == 0:
                for pair in currentPairs:
                    usedPairs[(pair[0], pair[1])] = True
                    usedPairs[(pair[1], pair[0])] = True

            elif k == 1:
                for pair in currentPairs:
                    usedPairs[(pair[0], pair[1], pair[2])] = True
                    usedPairs[(pair[0], pair[2], pair[1])] = True
                    usedPairs[(pair[1], pair[0], pair[2])] = True
                    usedPairs[(pair[1], pair[2], pair[0])] = True
                    usedPairs[(pair[2], pair[1], pair[0])] = True
                    usedPairs[(pair[2], pair[0], pair[1])] = True
            else:
                for pair in currentPairs:
                    usedPairs[(pair[0], pair[1], pair[2], pair[3])] =
True
                    usedPairs[(pair[0], pair[1], pair[3], pair[2])] =
True
                    usedPairs[(pair[0], pair[2], pair[1], pair[3])] =
True
                    usedPairs[(pair[0], pair[2], pair[3], pair[1])] =
True
                    usedPairs[(pair[0], pair[3], pair[1], pair[2])] =
True
                    usedPairs[(pair[0], pair[3], pair[2], pair[1])] =
True

                    usedPairs[(pair[1], pair[0], pair[2], pair[3])] =
True
```

```
                    usedPairs[(pair[1], pair[0], pair[3], pair[2])] =
True
                    usedPairs[(pair[1], pair[2], pair[0], pair[3])] =
True
                    usedPairs[(pair[1], pair[2], pair[3], pair[0])] =
True
                    usedPairs[(pair[1], pair[3], pair[0], pair[2])] =
True
                    usedPairs[(pair[1], pair[3], pair[2], pair[0])] =
True

                    usedPairs[(pair[2], pair[0], pair[1], pair[3])] =
True
                    usedPairs[(pair[2], pair[0], pair[3], pair[1])] =
True
                    usedPairs[(pair[2], pair[1], pair[0], pair[3])] =
True
                    usedPairs[(pair[2], pair[1], pair[3], pair[0])] =
True
                    usedPairs[(pair[2], pair[3], pair[0], pair[1])] =
True
                    usedPairs[(pair[2], pair[3], pair[1], pair[0])] =
True

                    usedPairs[(pair[3], pair[0], pair[1], pair[2])] =
True
                    usedPairs[(pair[3], pair[0], pair[2], pair[1])] =
True
                    usedPairs[(pair[3], pair[1], pair[0], pair[2])] =
True
                    usedPairs[(pair[3], pair[1], pair[2], pair[0])] =
True
                    usedPairs[(pair[3], pair[2], pair[0], pair[1])] =
True
                    usedPairs[(pair[3], pair[2], pair[1], pair[0])] =
True

            #Calculate Spearman correlation from xList, yList and
update var/SDOM
            rho, pvalue = stats.spearmanr(xList, yList)

            N = N + 1

            corrSum = corrSum + rho
            sqrSum = sqrSum + rho**2
            corrMean = corrSum/N

            #Calculate variance and SDOM of correlation distribution
            var = (sqrSum - N*(corrMean**2))
            SDOM = math.sqrt(var/N)
            print 'Computing CorrVar data for genepair ' + str(
genepair) + '.'

        else:
            disc = disc + 1
        discRate = disc/tot
```

```
        #Add correlation mean, SDOM and number of sub-samples N to resList
    .
        #If variance is wanted as output instead of SDOM, change SDOM to
    var.
        resList.append(str(corrMean))
        resList.append(str(SDOM))
        resList.append(str(N))

    return resList

#Open gene expression dataset and import as DataFrame.
ged = open(inputfile, 'r+')
gedTable = pd.read_table(ged, sep='\t', header=0 , index_col=0,
    lineterminator='\n')

print 'Headers added.'


#List of all unique pairs of genes
genepairs = list(itertools.combinations(gedTable.index, 2))


pool = Pool(processes = nCores)
result = pool.map(genepairFunction, genepairs)
pool.close()
pool.join()

#Create output file corrRes.txt and add headers + content
resFile = open(outputfile,'w')
resFile.write('\t'+'RHO\t'+'Rho1\t'+'SDOM1\t'+'N1\t'+'Rho2\t'+'SDOM2\t'+'
    N2\t'+'Rho3\t'+'SDOM3\t'+'N3\n')

for res in result:
    resFile.write('\t'.join(res)+'\n')

resFile.close()
ged.close()
```

Allowed sub-sample overlap of 6 data points

```
from __future__ import division
import numpy as np
import pandas as pd
import math, random
import itertools
from scipy import stats
from multiprocessing import Pool
from tqdm import tqdm
import datetime

#############################################################################
#
# This script computes Spearman correlation coefficients of all gene pairs
# in a gene expression data file, draws up to 200 sub-samples of 7
# measurement samples, rejects any sub-samples with more than 1, 2 and 3
```

```
# data points overlap and computes the standard deviation of the mean
# of the gene pair correlations from all valid sub-samples.
#
# Four parameters has to be changed for each run:
# FilenameIn, FilenameOut, usedMachines and thisMachine (see SETTINGS
# after genepairFunction).
#
# The number of sub-samples are set to 200, but can be changed inside
# genepairFunction. If the variance is to be found instead of SDOM, this
# must also be changed inside the genepairFunction.
#
#############################################################################


#genepairFunction finds the Spearman correlation coefficient,
#mean correlation & standard dev. of the mean for sub-samples with
#up to 6 data points overlap.

def genepairFunction(genepair):

  #list to fill with results for genepair
  resList=[]
  resList.append(str(genepair[0:2]))

  try:
    #Calculate Spearman for genepair across all columns, and add to result
    RHO, ptot = stats.spearmanr(gedTable.loc[genepair[0]],gedTable.loc[
     genepair[1]])
     resList.append(str(RHO))
  except:
    resList.append("RHO_ERROR")

  try:

    try:
      #Stores all drawn pickLists (sorted)
      usedPickLists = set()

      #Number of subset correlations calculated for this genepair
      N = 0
      #Standard deviation of the mean, initial value
      SDOM = 0
      #Initial discard rate - what fraction of pickLists are discarded
      discRate=0
      #Number of discarded pickLists
      disc=0
      #tot = total number of pickLists (discared and used)
      tot=0

      #variance
      var=0

      #Variables for variance calculation, Welford's algorithm
      M = 0
      M2 = 0
      S = 0
      #Empty list to fill with current sub-sample
```

```
      pickList = [None]*7

      while (discRate < 0.99 and N <= 149)  or tot < 50:

        tot = tot + 1

        #Put seven unique and random col indexes in pickList
        pickList = random.sample(xrange(0,gedLen-1),7)

        #Check if current sub-sample is allready used, and if so: if it
    contains more than 2 zeros. If not, continue:
        if frozenset(pickList) not in usedPickLists:
          if np.count_nonzero(gedTable.loc[genepair[0]][pickList])>4 and
    np.count_nonzero(gedTable.loc[genepair[1]][pickList])>4:
            #Add pickList to usedPickLists
            usedPickLists.add(frozenset(pickList))

            #Calculate Spearman correlation from xList, yList and update
    avg., variance etc.
            rho, p = stats.spearmanr(gedTable.loc[genepair[0]][pickList],
    gedTable.loc[genepair[1]][pickList])

            N = N + 1

            #Welford's algorithm running variance
            delta = rho - M
            M = M + delta/N
            delta2 = rho - M
            M2 = M2 + delta*delta2
            var = M2/N

        #discard if sub-sample did not meet criteria. count discard
    percentage
          else: disc = disc +1
          discRate = disc/tot
        else:
          disc = disc + 1
        discRate = disc/tot

      #Add correlation mean, SDOM and number of sub-samples N to resList
      resList.append(str(M))
      resList.append(str(math.sqrt(var/N)))
      resList.append(str(N))
    except:
      resList.append("UNKNOWN_SUBSAMPLING_ERROR")

    return resList

def wrapMyFunc(arg):
  return arg, genepairFunction(arg)

def update((pair,ans)):
  result[pair[2]] = ans
  pbar.update()


def runChunk(intervalEnds,i,nProcesses):
```

```
  pool = Pool(processes = nProcesses)

  #Perform computations if interval is not used before and add used
    interval
  #to check-file.
  for pair in genepairs_modified[int(intervalEnds[i]):int(intervalEnds[i
    +1])]:
    pool.apply_async(wrapMyFunc, args = (pair,), callback = update)

  pool.close()
  pool.join()

#################################################################
#SETTINGS: Filenames for input data and output file is set here:

filenameIn = 'RActrl.txt'
filenameOut = 'RActrl_CorrSDOM_.txt'

#################################################################

#Open gene expression dataset and import as DataFrame.
ged = open(filenameIn, 'r+')
gedTable = pd.read_table(ged, sep='\t', header=0, index_col=0,
    lineterminator='\n')
gedLen = len(gedTable.columns)

print "Go!"

#List of all unique pairs of genes + number in the list
genepairs_modified = []
genepairsTot = list(itertools.combinations(gedTable.index, 2))

#List the numbers of cores in correct order on the machines that can be
    used
nCores = [48, 24, 24, 16 ,16, 16, 16]

#################################################################################

#SETTINGS:
#List the numbers (zero indexing) of all used machines in "usedMachines",
#e.g. if using machine 2,3 and 4, set: "usedMachines = [2,3,4]"
#
#Then set "thisMachine" equal to the machine the script is running on,
#e.g. if the script is on machine 3, set "thisMachine = 3"

usedMachines = [0,1,2,3,4,5,6]
thisMachine = 3

#################################################################################

#genepairFunction is run  in parallell on the given machine for the
#defined number of gene pairs, the progress bar is updated,
#the logfile with completed chunks + errorlog is updated,
#and the results are written to intermediate backup files and result file.

#################################################################################
```

```
#The genepairs are distributed on the used machines based on the number
#of cores on the respective machines
totalCores = 0
for machine in usedMachines:
  totalCores = totalCores+nCores[machine]
nGenePairsTot = len(genepairsTot)

pairsPerCore = np.ceil(nGenePairsTot/totalCores)

pairIt = 0
genepairsForMachines = []
for machine in usedMachines:
  genePairsInterval = []
  genePairsInterval.append(pairIt)
  pairIt = np.ceil(pairIt+pairsPerCore*nCores[machine])
  pairIt = min(pairIt,nGenePairsTot)
  genePairsInterval.append(pairIt)
  genepairsForMachines.append([genePairsInterval])

genePairsIntervalToRun = genepairsForMachines[thisMachine]
genePairsIntervalToRun = genePairsIntervalToRun[0]
genepairs = genepairsTot[int(genePairsIntervalToRun[0]):int(
    genePairsIntervalToRun[1])]

print "Adding indexes to all genepairs..."

L = len(genepairs)
print "Number of genepairs: "+str(L)+" (interval: "+str(
    genePairsIntervalToRun[0])+" to "+str(genePairsIntervalToRun[1])+")"


#Add number to each gene pair for use in progress tracking
for i in xrange(L):
  genepairs_modified.append((genepairs[i][0], genepairs[i][1],i))

#Make file to save which chunks are finished
try: #testing to see if the file exists first
  test = open('usedChunks.txt', 'r')
  test.close()
except: #if not, the file is made and initalized
  a = open('usedChunks.txt', 'w')
  a.write('The following intervals are successfully processed:\n')
  a.close()

#Make output file and add header
filenameBase = filenameOut.split(".")
filenameBase = filenameBase[0]
filenameAll = filenameBase+"_ALL_RESULTS.txt"

resFileAll = open(filenameAll, 'w')
resFileAll.write('\t'+'RHO\t'+'Rho1\t'+'SDOMrand\t'+'S\n')
resFileAll.close()

print "Caculating Corr/SDOM data..."

#Initiate progress bar
pbar = tqdm(total = L)
```

```
result = [None]*L

#nChunks is the number of chunks at which each result is saved,
#i.e. "save points"

nChunks = 100
#setting up chunk intervals
intervalEnds = np.ceil(np.linspace(0, L, num = nChunks + 1))

#Create error log
dateAndTime = datetime.datetime.now()
errorFile = open("errorLog.txt",'w')

for i in range(len(intervalEnds)-1):
  chunk = str(intervalEnds[i]) + " to " + str(intervalEnds[i+1])

  #Check if interval is already in the file of used chunks
  chunkUsed = False
  usedChunksFile = "usedChunks.txt"
  a = open(usedChunksFile, 'r')
  lines = a.readlines()

  for line in lines:
    lineWithoutNewline = line.split("\n")
    lineWithoutNewline = lineWithoutNewline[0]
    if lineWithoutNewline == chunk:
      chunkUsed = True
  a.close()

  #If the chunk interval has not already been run, then run it:
  chunkUsed = False
  if chunkUsed == False:
    runChunk(intervalEnds,i,nCores[thisMachine])

    #try writing to file; if it doesn't work,
    #write the chunk to the error log and give error message
    try:
      filenameOutForInterval = filenameBase+str(int(intervalEnds[i]))+".
    txt"
      resFileChunk = open(filenameOutForInterval,'w')
      resFileAll = open(filenameAll,'a')
      noneCounter = 0
      for res in result[int(intervalEnds[i]):int(intervalEnds[i+1])]:
        if res is None:
          noneCounter = noneCounter+1
        else:
          resFileChunk.write('\t'.join(res)+'\n')
          resFileAll.write('\t'.join(res)+'\n')
      if noneCounter>0:
        errorFile.write("Number of None in chunk: "+chunk+"\n")
        errorFile.write(str(noneCounter)+"\n\n")
      resFileChunk.close()
      resFileAll.close()

      b = open(usedChunksFile, 'a')
      b.write(chunk + '\n')
```

```
        b.close()
      except:
        print 'Error in gene pair interval '+chunk+" (see errorFile.txt)"
        errorFile.write("Error in chunk:"+chunk+"\n")
        errorFile.write(str(res)+"\n\n")


errorFile.close()
pbar.close()

ged.close()

print 'Done!'
```

A5 Script for Filtering Two Correlation/Variance Files for Gene Pairs not in Common

The following script was used for filtering out gene pairs that the two gene pair/correlation/variance text files did not have in common:

```
import pandas as pd

#############################################################################

#This script takes in two gene pair correlation/variance text files,
#removes all gene pairs that the two files do not have in common,
#and writes the filtered results to two new text files. All gene pairs
#in the new files are listed in the same order, so that the same genes
#are listed first and last. The lists are not sorted alphabetically
#unless they were already on a sorted format, but this can easily be
#done using the "sort" command in bash/Linux terminal.

#############################################################################


#First input file and name of the filtered output file
file1 = 'RActrl_clean.txt'
file2 = 'Corr_Var_RA_jointsyn.txt'

#Second input file and name of the filtered output file
file3 = 'RA_Full_Flt.txt'
file4 = 'RActrl_Full_Flt.txt'

#Open the first file as dataframe
f1 = open(file1, 'r+')
df1 = pd.read_table(f1, sep='\t', header=None, index_col=None,
    lineterminator='\n')

firstPairs = {}
commonPairs = {}

#Add all pairs of the first input file to a dictionary
for row in df1.itertuples(index = False):
    pair = str(row[0]) + '\t' + str(row[1])
    firstPairs[pair] = True

#Open the second input file
f2 = open(file2, 'r+')
df2 = pd.read_table(f2, sep='\t', header=None, index_col=None,
    lineterminator='\n')

#Add pairs common to both input files to a dictionary of common pairs
for row in df2.itertuples(index = False):
    pair1 = str(row[0]) + '\t' + str(row[1])
    pair2 = str(row[1]) + '\t' + str(row[0])

    if (pair1 in firstPairs) or (pair2 in firstPairs):
        commonPairs[pair1] = True

out1 = open(file3, 'w')
```

```
#Write filtered content of file one to new file
for row in df1.itertuples(index = False):
    pair1 = str(row[0]) + '\t' + str(row[1])
    pair2 = str(row[1]) + '\t' + str(row[0])

    if pair1 in commonPairs:
        out1.write(str(row[0]) + '\t' + str(row[1]) + '\t' + str(row[2]) +
     '\t' + str(row[3]) + '\n')

    if pair2 in commonPairs:
        out1.write(str(row[1]) + '\t' + str(row[0]) + '\t' + str(row[2]) +
     '\t' + str(row[3]) + '\n')

out1.close()

out2 = open(file4, 'w')

#write filtered content of file two to new file
for row in df2.itertuples(index = False):
    pair = str(row[0]) + '\t' + str(row[1])

    if pair in commonPairs:
        out2.write(str(row[0]) + '\t' + str(row[1]) + '\t' + str(row[2]) +
     '\t' + str(row[3]) + '\n')

out2.close()


f1.close()
f2.close()
```

A6 Script for Node Homogeneity and Ratios of C-, S- and D-Scores per Node

The following script was used for computing the ratios of C-, S- and D- scores, as well as the node homogeneity, textitH, for all nodes in the CSD network:

```
import pandas as pd

#Open CSD network (three columns: Gene1 Gene2 CSD-type)
network = 'CSDSelection.txt'
f = open(network, 'r')

gglink = pd.read_table(f, sep='\t', header=None, index_col=None,
    lineterminator='\n')

genelist = []

#Go through all genepairs and add all gene names to genelist once per gene
for row in gglink.itertuples(index = False):
    if str(row[0]) not in genelist:
        genelist.append(str(row[0]))

    if str(row[1]) not in genelist:
        genelist.append(str(row[1]))

columns = ['C','S','D']

#Make dataframe with CSDratios for each gene
df = pd.DataFrame(index = genelist, columns = columns)
df = df.fillna(0)

#Iterate through gglink, count the number of C, S and D links per gene
#and add number to respective columns in df
for row in gglink.itertuples(index = False):
    if str(row[3]) == 'C':
        df.at[str(row[0]), 'C'] += 1
        df.at[str(row[1]), 'C'] += 1

    elif str(row[3]) == 'S':
        df.at[str(row[0]), 'S'] += 1
        df.at[str(row[1]), 'S'] += 1

    elif str(row[3]) == 'D':
        df.at[str(row[0]), 'D'] += 1
        df.at[str(row[1]), 'D'] += 1

#Find the node degree and divide the elements of the C, S and D column by
    the respective degree
df['Sum'] = df.sum(axis=1)

df['C'] = df['C']/df['Sum']
df['S'] = df['S']/df['Sum']
df['D'] = df['D']/df['Sum']

#Find node homogeneity
df['H'] = df['C']**2 + df['S']**2 + df['D']**2
```

```
#Remove degree-column and write to file
df.drop('Sum', axis = 1, inplace = True)

df.to_csv('CSDratios.txt', sep='\t', header = True, index = True)

f.close()
```

A7 Signal Plot for σ_3^{338}

Figure A2 shows the plot of the correlation variability, σ_3^{338}, for 28 gene pairs as a function of the number of sub-samples from which it is calculated, S. The wholeblood gene expression data set with 338 data points per gene was used with an allowed sub-sample overlap of 3 data points.

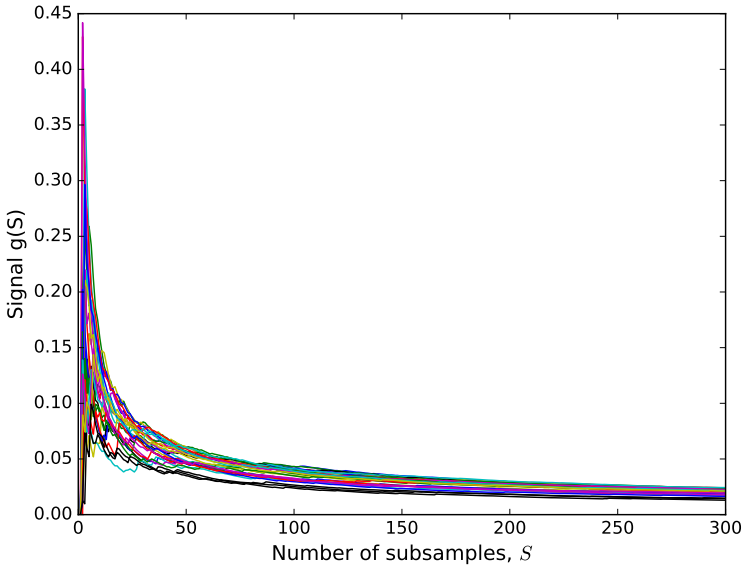


Figure A2: The function *g(S)* shows the σ_3^{338} signal for 28 gene pairs as a function of the number of sub-samples from which it is calculated, S. Each graph represents one gene pair.

A8 Networks with Alternative Importance Values

Figs. A3 and A4 give a visualization of the CSD networks obtained with importance levels of $p = 10^{-4}$ and $p = 10^{-6}$ respectively. The links are colored according to their link type (C-type links are blue, S-type links are green and D-type links are red). The network obtained with $p = 10^{-4}$ has 7554 nodes and $36\,873$ links distributed over 178 components. The giant component contains 7101 nodes, and the network has an average degree of $k_{avg} = 9.8$. The network obtained with $p = 10^{-6}$ has 426 nodes and 360 links distributed over 98 connected components. The biggest component has 104 nodes, and the average degree of the network is $k_{avg} = 1.6$.

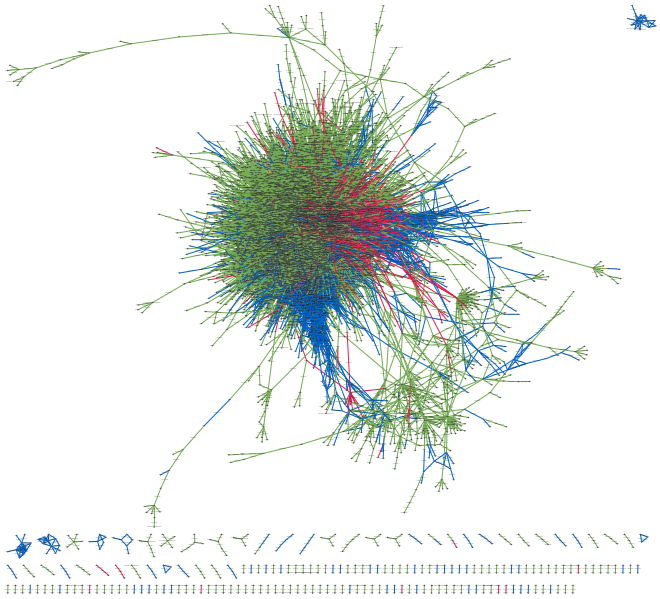


Figure A3: Visualization of the CSD network obtained with an importance level $p = 10^{-4}$. The network has 7554 nodes and $36\,873$ links distributed over 178 components. The giant component contains 7101 of those nodes, and the network had an average degree of $k_{avg} = 9.8$. The links are colored according to their link type: C-type links are blue, S-type links are green and D-type links are red.

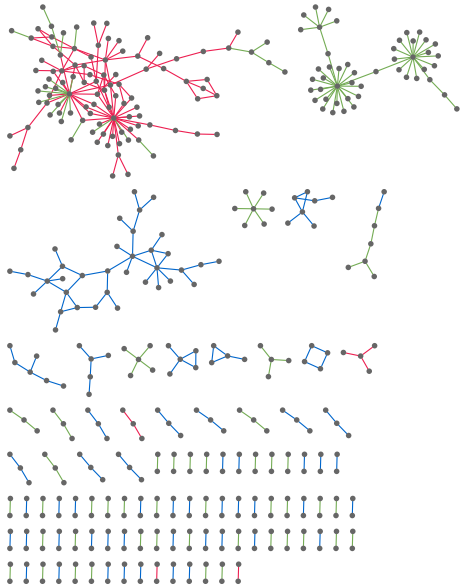


Figure A4: Visualization of the CSD network obtained with an importance level $p = 10^{-6}$. The network has 426 nodes and 360 links distributed over 98 connected components. The biggest component contains 104 of those nodes, and the average degree of the network is $k_{avg} = 1.6$. The links are colored according to their link type: C-type links are blue, S-type links are green and D-type links are red.

A9 The C-, S- and D-networks

Figs. A5, A6 and A7 visualize the separate C-, S- and D-networks, respectively. Apart from the smallest components which are moved closer to the giant component, the links are placed according to their original position in CSD network.

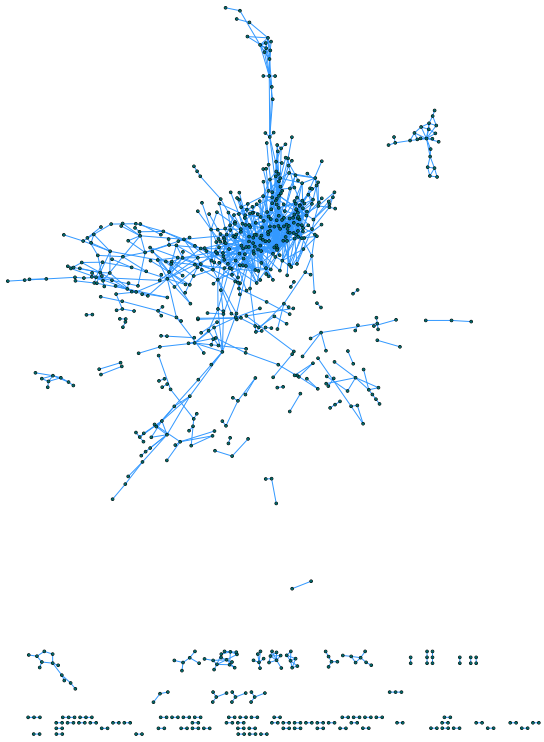


Figure A5: The C-network: A visualization of all C-type links and the nodes that they connect in the CSD network.

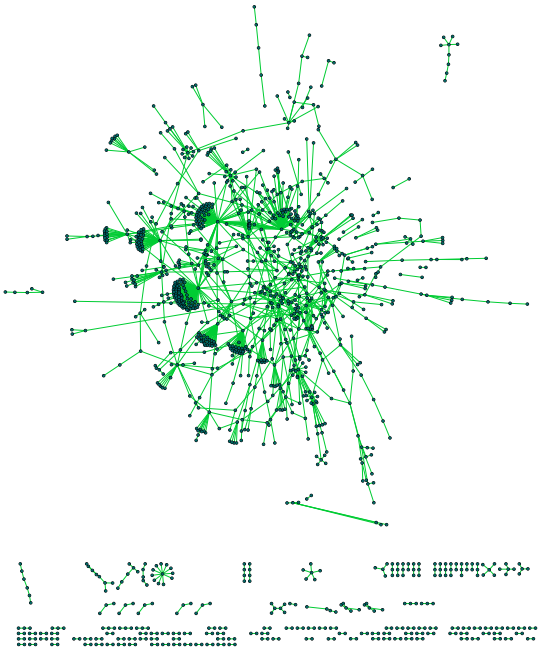


Figure A6: The S-network: A visualization of all S-type links and the nodes that they connect in the CSD network.

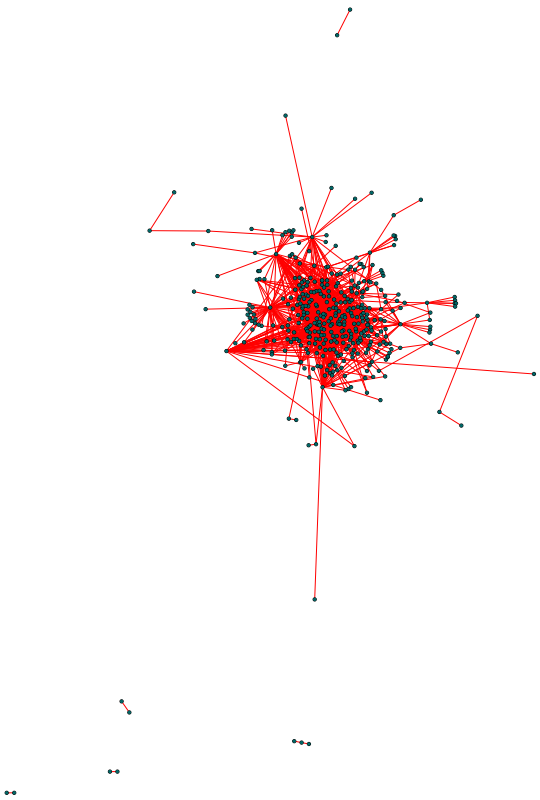


Figure A7: The D-network: A visualization of all D-type links and the nodes that they connect in the CSD network.

A10 Full Results of GO Enrichment

All biological process categories that were enriched (with a false discovery rate < 0.05) in the full CSD network, as well as in the the extracted C-, S- and D-networks, are given in four external tables that were too long to fit in this Appendix. The number of genes in the reference list that belongs to each category (#ref), the number of genes in the given network that belongs to each category (#genes), the expected number of genes to be found in a random selection of the same number of genes (Exp), the fold enrichment (FE), the raw p-value and the false discovery rate (FDR) are also found in these tables. The tables may be found at `https://goo.gl/PwV4sU`.