Network Analysis of Heart Infarction Mice Transcriptomes

Caroline Jachmann
Bachelor thesis

Department of Computer Science
Algorithmic Bioinformatics

Submission:	10.01.2019
Supervisor:	Prof. Dr. Gunnar Klau
Second Assessor:	Prof. Dr. Martin Lercher
Advisor:	Eline van Mantgem

Declaration

I hereby confirm that this thesis is my own work. I have documented all sources and tools used. Any direct or indirect quote has been marked as such clearly with specification of the source.

Düsseldorf, January 09, 2019 Caroline Jachmann

Abstract

With the start of the postgenomic era, the general perception of genes started to shift dramatically. Now that entire genomes were sequenced but still didn't explain the broad spectrum of diversity of biological systems, researchers started to focus on different aspects: how genes are regulated, how they are transcribed and translated into proteins or how these proteins interact with others. Also, the technology used for analysing DNA and proteins has made big progress up to the point where entire genomes and proteomes can be studied at once. As a direct consequence, we are now swimming in a wealth of biological data from high-throughput experiments. In recent years, a field called functional genomics developed with the goal to make use of the thousands of gigabytes of data that is now available in order to answer biological questions regarding genetics on a genome-wide scale.

In this work we identify functional modules of genes that play a role in the healing process after a heart attack. We combine the differential gene expression data from mice transcriptomes that arose from a heart infarction experiment done by the university hospital of Düsseldorf with the protein-protein interaction databases STRING and BioGRID. First, we preprocess the data sources so that they can be linked together. Then we conduct a network analysis for the purpose of finding genes which on the one hand are influenced by a heart attack on an expressional level and on the other hand are interconnected with each other and thus possibly contributing to the same cellular function. Afterwards we transform the results into a format that can be used with the Cytoscape application eXamine for the sake of portraying the resulting networks in an interactive environment. With this method we found three differentially expressed regions in the STRING network with the gene expression levels from heart tissues (both damaged and undamaged) 3 hours and 24 hours after a heart attack. The findings are yet to be interpreted from a biological point of view.

Contents

1 Introduction

Heart infarction, also known as acute myocardial infarction or heart attack, is caused by damage to or the death of heart muscle tissue due to decreased blood flow (ischemia) through one or more of the coronary arteries [1]. The restoration of the blood flow (reperfusion) can also deal additional damage. Despite 90 percent of the risk factors being influenceable, for example cigarette smoking, exercise or obesity, myocardial infarctions are one of the leading causes of death in the developed world [2]. Being interested in the systemic and myocardial adaptations to an infarction, a team consisting of cardiologists from the university hospital Düsseldorf (UKD) and scientists from the biomedical research center (BMFZ) carried out an experiment in 2017 on two groups of mice. An infarction was simulated by opening their chests and manually tightening the coronary arteries to decrease the blood flow. In 2018, the experiment was repeated on two control groups that only underwent the preliminaries of the operation. In both cases, the transcriptome (i.e. the RNA) was extracted from several tissues and sequenced into reads. For one control group and one infarction group, this happened 3 hours after the operation, the RNA of the other two groups was read after 24 hours. Then the gene expression levels were compared in four groups (see also Figure 1):

- Group 1: Comparisons within the myocardial infarction groups (either between different tissues or between different times)
- Group 2: Control group 3h vs myocardial infarction group 3h
- Group 3: Control group 24h vs myocardial infarction group 24h
- Group 4: Comparisons within the control groups (either between different tissues or between different times)

The pipeline introduced in this thesis combines the data provided by those experiments together with protein-protein interaction databases. Thus, groups of genes that contribute to the same cellular function involved in the healing process after a myocardial infarction (in further chapters referred to as *functional modules*) can be detected by being differentially expressed regions of the protein-protein interaction networks [3], [4].

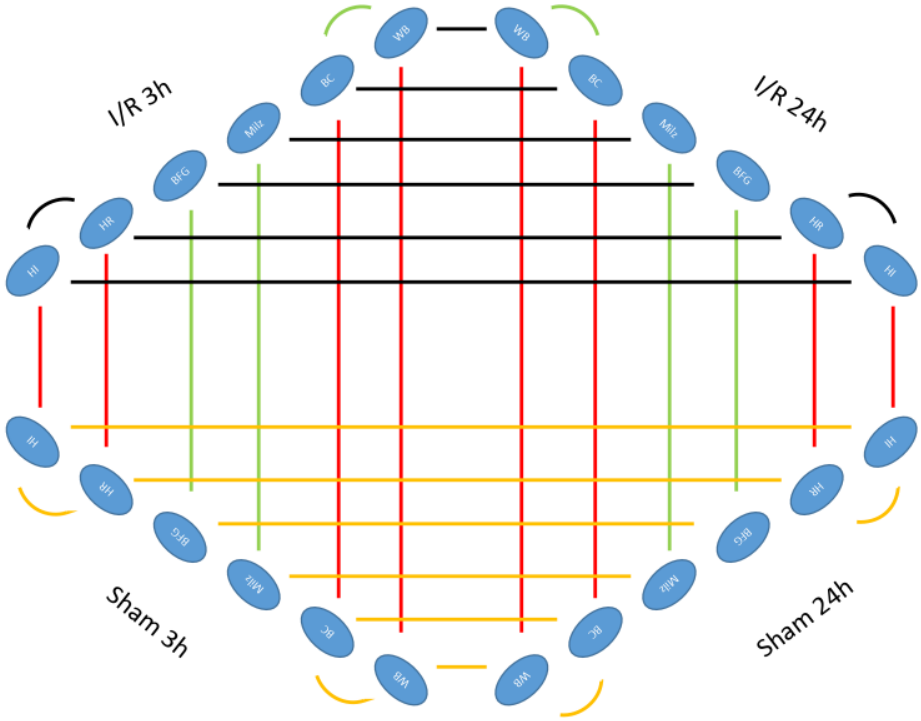

Figure 1: Experimental setup. The myocardial infarction mice are labeled with I/R = Ischemia/Reperfusion, the control groups are called the "Sham" groups. HI = heart infarction tissue, HR = heart remote tissue, BFG = brown adipose tissue, Milz = spleen, BC = blood cells, WB = whole blood. Black lines: experiment performed in 2017, red lines = new and highest priority, green lines = new and mediocre priority, yellow lines = new and low priority.

1.1 Mice transcriptome data sets

The data sets provided for this thesis contain the results of the differential gene expression analysis carried out on the RNA data via the CLC Genomics Workbench by Qiagen. The sequenced RNA reads are first mapped to a reference genome. From this mapping, the reads are categorized and assigned to the genes, and expression values for each gene are calculated [5]. In order to find genes that are differentially expressed, for example when comparing the heart tissue from the control group to the one from the I/R group, the CLC Workbench offers multiple statistical methods with different requirements to the input data [6]. They all assess how significant the expression level differences are for each gene by computing their p-values, a statistical value between 0 and 1 - the closer the p-

value to 0, the more significant the difference. The proportion of the difference is stored in the fold change value. For example, a fold change of 2 in the data set "Vergleich-2_S_HI-3h vs HI3h.xlsx" means that the corresponding gene was twice as much expressed in the heart infarction tissue ("HI") of the infarction group that was read 3 hours after the operation ("3h") than in the heart infarction tissue of control group ("S" for sham). An exemplary entry of a data set is shown in chapter 2.2.1 and we go into further detail on the p-values in chapter 2.3.1.

1.2 Protein-protein interaction networks

To manage all the knowledge gathered over the years, various projects in the form of databases have emerged, whether for general protein sequences, genetic diseases or protein-protein interactions, the latter being of great interest in this thesis. A protein-protein interaction (PPI) is defined as either a direct binding process or an indirect process, for example by sharing a substrate in a metabolic pathway, by regulating each other transcriptionally, or by participating in larger multi-protein assemblies [7]. The set of all interactions within an organism is called its interactome and varies greatly in sizes between species. For example, the interactome of humans is currently estimated to consist of around 600.000 interactions between around 1500 proteins, while the interactome of the nematode C. elegans was assessed to include circa 240.000 interactions between 2600 proteins [8]. There are many PPI networks that differ in how they provide the data, how they define interactions and how they compute their networks. For this thesis we will focus on the following public databases:

- BioGRID, the Biological General Repository for Interaction Datasets, holding information on 1,658,808 protein and genetic interactions collected from 68,215 publications [9].
- STRING, an abbreviation for "Search Tool for the Retrieval of Interacting Genes/Proteins", a protein-protein interaction network featuring 1380 million interactions between 9.6 million proteins in 2031 organisms. It computes the *probability* that two proteins interact based on seven so called channels: Conserved neighborhood, gene fusions, phylogenetic co-occurrence, co-expression, database imports, large-scale experiments and literature co-occurrence [10].

1.3 Methods and resources

The backbone of the workflow is formed by a Snakemake [11] pipeline, supplemented with scripts written in R and Python. The PPI networks are provided

by BioGRID [9] and STRING [10]. The network analysis is done with the integer linear software Heinz [12], which itself makes use of IBM ILOG CPLEX [13], a commercial software used for solving optimization models. To enrich the scored modules, the KEGG pathway database [14] and the Gene Ontology annotations [15] are used. Gene identifiers and GO Terms are collected from the Ensembl database [16] via the data mining tool biomaRt [17] for R scripts and pybiomart [18] for python scripts. Computations were done on the high-performance server of the "Centre for Information and Media Technology" (ZIM) at the University of Düsseldorf (Germany) with 16 Intel(R) Xeon(R) E5-2667 v4 (3.20 GHz) and 503 GB RAM.

2 Network analysis pipeline

2.1 Snakemake

A widespread problem in biological research is the reproducibility of results. Results can be influenced by many environmental factors that can be easily overlooked, which makes replicating them difficult and sometimes even impossible. An article published 2017 in the journal Nature about an experiment on the lifespan of worms describes how it took scientists from three different labs a year to set up a system so that under the same defined circumstances, each lab would get almost the same results. Strict measurements were taken; worm incubators were bought at the same time and from the same seller, and even the way on how to pick up the worms was standardized [19].
This challenge is not limited to the field of biology. While computer scientists may not need to control things like picking up worms correctly, they do need to specify a different kind of environment. When analyzing data with external tools and sources, the recording of the workflow and of which versions were used is key to making the process reproducible due to the non-static nature of databases and software. For this purpose, several workflow managers have been created. For this thesis we use Snakemake [11], a pythonic manager comparable to GNU Makefiles [20].
Snakemake enables us to split the workflow into smaller steps called rules. Each rule creates an output either from scratch or from one or more input files by executing a script or a shell command. A rule can be given a configuration file in which the environment gets specified (e.g. which python version to use when running a python script). The target files can either be specified in the command line or in the first rule, which as a best practice is called the "all" rule. Snakemake automatically checks in which order the rules must be executed to generate the target files, recreating the workflow as a directed acyclic graph (see Figure 2).

Figure 2: Part of the DAG created by Snakemake. The full DAG can be found in the appendix.

It computes the target files as efficiently as possible. If the output files of a rule already exist, the step only gets executed if one of the input files is newer than one of the output files or one of the input files will be updated by another job [20]. Another useful addition Snakemake brings to the table is the use of named wildcards. For example, if the target file is called "comparisons/controlgroup1.txt" and the output of a rule is called "comparisons/{sample}.txt", Snakemake will automatically replace the "sample" wildcard with the value "controlgroup1". We will come back to these features at the evaluation (chapter 4).

2.1.1 Requirements

To be able to run the pipeline that is available on `github.com/AlBi-HHU/cardio_network_analysis`, the following packages and software need to be installed:

- Heinz: build from `https://github.com/ls-cwi/heinz`, must be set as a PATH variable
- eXamine: build from `https://github.com/ls-cwi/eXamine`
- miniconda: `https://conda.io/miniconda.html`

- Snakemake: install via `conda install bioconda-utils snakemake`
- pybiomart: install via `pip install pybiomart`

2.1.2 Workflow - an Overview

The process can be roughly divided into the steps that are depicted in Figure 3.

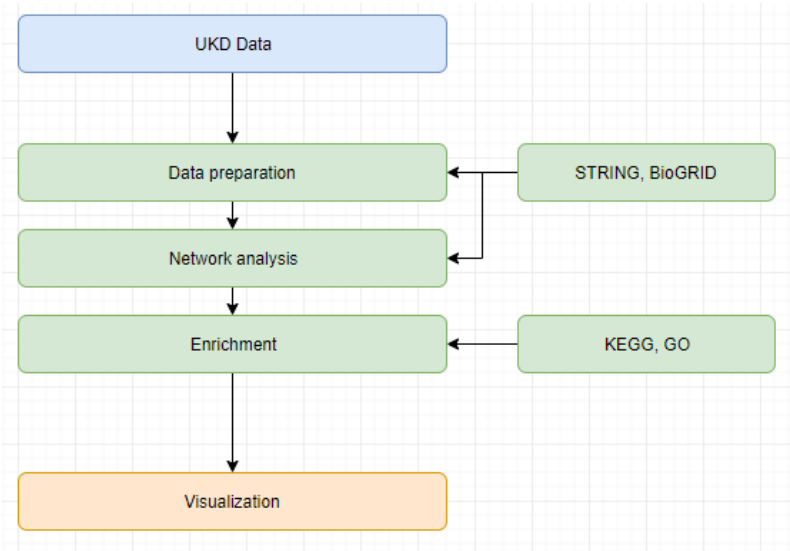

Figure 3: Steps of the workflow

Before analyzing the data, we need to prepare both the UKD data and the protein-protein networks (chapter 2.2). By combining them we are able to perform network analysis, possibly finding groups of genes that could be of interest for further research on myocardial infarction (chapter 2.3). To ease the interpretation of the results, we include additional external sources (chapter 2.4) and finally visualize the findings using eXamine (chapter 2.5).

2.2 Preparation

2.2.1 UKD Data

The differential gene expression analysis data was provided in the form of 32 xlsx formatted files, split up into four groups of comparisons (see chapter 1). Each file consists of 48710 entries and holds the information depicted in Table 1.

Feature ID	p-value	fold change	Bonferroni	FDR p-value correction
0610006L08Rik	1	1	1	1
Aagab	4,17E-03	-1,48	1	8,65E-02

Table 1: Example entries of one of the UKD data sets

The data set does not only feature the computed p-values and fold changes for each gene, but also statistically corrected values. As we will work with the raw data and do some statistical correction later on, the first step is to remove the last two columns. For reasons explained in chapter 2.3.1, we need to filter entries with a p-value smaller than 0.98. Additionally, the whitespace in the file titles need to be removed due to Snakemake falsely assigning wildcard values because of space bars.

2.2.2 STRING Database

Via the official STRING website [10] it is possible to download different PPI networks of many organisms. For this thesis we use one that provides scored links between the proteins of the house mouse (*mus musculus*) with additional information on the channel scores (see Table 2).

protein1	protein2	textmining	...	combined_score
ENSMUSP0(..)1	ENSMUSP0(..)72868	129	...	216
ENSMUSP0(..)1	ENSMUSP0(...)5531	111	...	157

Table 2: First entries of the STRING data set. For the sake of clearness only one channel is portrayed.

STRING uses different identifiers than the UKD. To get the corresponding MGI symbols to the given Ensembl protein identifiers, we implemented a search query to the Ensembl database with biomaRt.
In the database, interactions are stored as directed edges, so for one interaction

there are two entries in the database - one from protein one to protein two and one from protein two to protein one, both with the same combined score. To improve runtimes and the readability of the results, we transformed the two directed edges into one undirected edge. For further steps we only use tendentially more significant interactions with a combined score bigger or equal to 700 and an experimental score of at least 300. We also took out all edges incident to Ubc, a gene encoding a ubiquitin precursor which is associated with numerous cell reactions, because it connected too many independent nodes in the resulting network with its 194 edges.

2.2.3 BioGRID Database

The BioGRID database [9] provides gene identifiers that match with the UKD data, so they can almost directly be used as an input for our network analysis. Only slight changes needed to be made in terms of removing additional information and also transforming directed edges into undirected ones. Similar observations to Ubc in STRING were made with the genes Fancd2 and Eed (see Figure 4) in the BioGRID network. The corresponding edges were also taken out (Fancd2: 1678 edges, Eed: 1177 edges).

UniprotID_A	UniprotID_B
SMAD2	Rasd2
SMAD2	Rab34

Table 3: Example entry for BioGRID, simplified

2.3 Network analysis

Now that the data is preprocessed, we are able to analyze it. First we model the p-values into an already known distribution to tell which part of the observations is actually significant (the "signal" component) and which part is only noise in the data (chapter 2.3.1 and 2.3.2). Then we perform the actual network analysis with Heinz, which computes our desired modules of genes (chapter 2.3.4).

2.3.1 P-value distributions

In statistics it is common to work with hypotheses. While the null hypothesis H_0 often describes the absence of any correlation or effect, what really is expected or tried to prove is formulated in the alternative hypothesis H_1. Translated to our

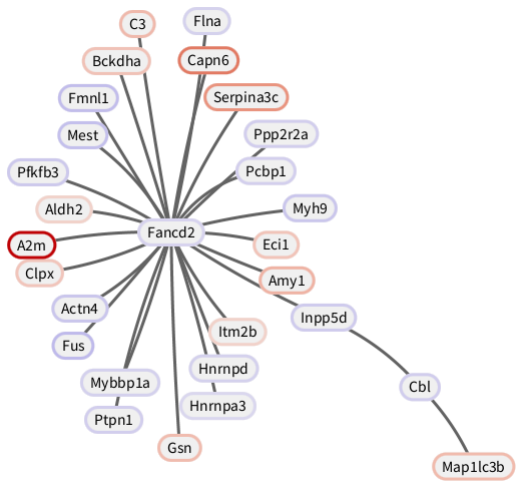

Figure 4: Resulting module without taking Fancd2 out

differential gene expression experiment, our null hypothesis is "The gene expression levels are the same in both groups". The alternative hypothesis is "The gene expression levels are different in group 1 and group 2". The computed p-values tell us how likely the observed expression levels are under the assumption that the null hypothesis was true. If the null hypothesis is true, the aggregated p-values should follow a uniform distribution [21]. If the null hypothesis is false (i.e. if there are observable changes in gene expression rates between the compared groups), the p-values should deviate from a uniform distribution and in this case, they should accumulate near 0 due to their corresponding genes being expressed at either significantly higher or lower levels.
When looking at the p-value plots for the first time, none of the distributions seemed to fit any of the expectations. The plots were distorted by a big peak at values between 0.99 and 1 (see Figure 5). The cause for this might be the software used to compute the p-values out of the RNA reads - some programs set the p-value of genes that have no reads at all to 1. After filtering out all entries with a p-value bigger than 0.98, some plots coming from comparisons between the two control groups still looked odd, but in total 26 out of 32 distribution plots looked as expected and similar to the one displayed in Figure 6. Figure 5 and 6 also feature quantil quantil plots (or qq-plots) that are used to compare two distributions. In this case, the qq-plots compare the estimated BUM distributions (see chapter 2.3.2) to the observed distributions of the p-values. The closer the dotted line is to the diagonal, the more similar the two distributions are.

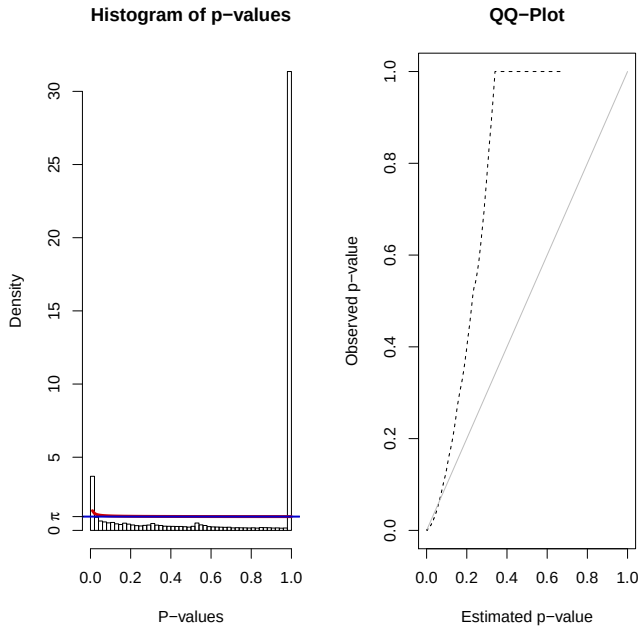

Figure 5: Distribution of p-values from 2 S HI 3h vs HI 3h before applying filter

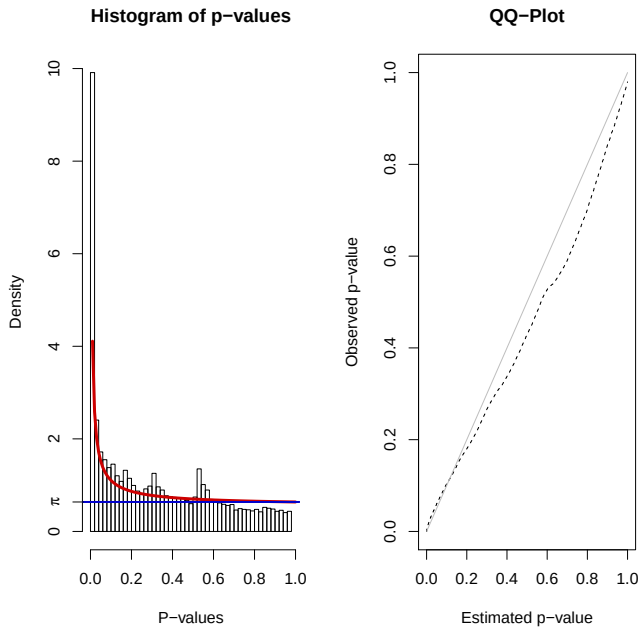

Figure 6: Distribution of p-values from 2 S HI 3h vs HI 3h after applying filter

2.3.2 Beta uniform mixture model

After applying the filter, the p-value distributions can now be described with a mixture of the uniform(0,1) distribution (the "noise" component) and a special case of the beta distribution B(a,1) (the "signal" component). Put together we score the following beta uniform distribution B(a, b) [4] by setting

$$f(x) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1}(1-x)^{b-1}$$

where Γ denotes the gamma function. The B(a, b) distribution can be simplified between $0 < x \leq 1$ to

$$f(x \mid a, \lambda) = \lambda + (1-\lambda)ax^{a-1}$$

where a denotes the shape parameter of the beta distribution and λ denotes the mixture parameter of the uniform distribution. Those two variables can be obtained by calling the fitbumModel function from the BioNet package. Their further use is described in chapter 2.3.3. The fitbumModel function also creates the qq-plots that can be seen in Figures 5 and 6. They compare the B(a, b) distribution to the observed p-value distribution and the almost straight line in the second plot verifies the use of the BUM model to describe the actual distributions.

2.3.3 Controlling the false discovery rate

When applying hypothesis tests multiple times like it is the case with testing the expression levels of the entire genome of mus musculus, false discoveries will inevitably accumulate. If we set the p-value threshold below which we declare a gene to be significantly expressed to a fixed value, we would have no information on how many genes we wrongfully declare as significant (false positive or type 1 error) compared to the number of how many genes are considered significant in total. This rate is also called the false discovery rate (FDR). To bring this under control, we can use the computed values a and λ to derive a more sophisticated threshold τ that makes sure a fixed FDR α will not be exceeded.
First we obtain the horizontal threshold π with $\pi = \lambda + (1-\lambda)a$ and then the vertical threshold τ with

$$\tau(\alpha) = \left(\frac{\pi - \alpha\lambda}{\alpha(1-\lambda)}\right)^{1/(a-1)}$$

as can be read in [22]. This leaves us with the partition depicted in Figure 5, holding the following information:

- Section A: observations that are rightfully declared as significant (true positives).

- Section B: observations that are wrongfully declared as not significant (false negatives).
- Section C: observations that are wrongfully declared as significant (false positives).
- Section D: observations that are rightfully declared as not significant (true negatives).

Due to the possibility of setting the FDR for Heinz manually, we are able to scale the resulting subnetwork. For broader, but possibly less meaningful modules because of a higher tolerance for false positives, we can set the FDR higher. For smaller modules that mostly consist of significant genes, we can set a small FDR. To score manageable modules with around 10–20 nodes specifically for the given data sets, the FDR should be set between 1e-10 and 1e-20. The values used to compute the modules for the transcriptome data sets can be found in the configuration file in the github repository and will be applied automatically when running the pipeline.

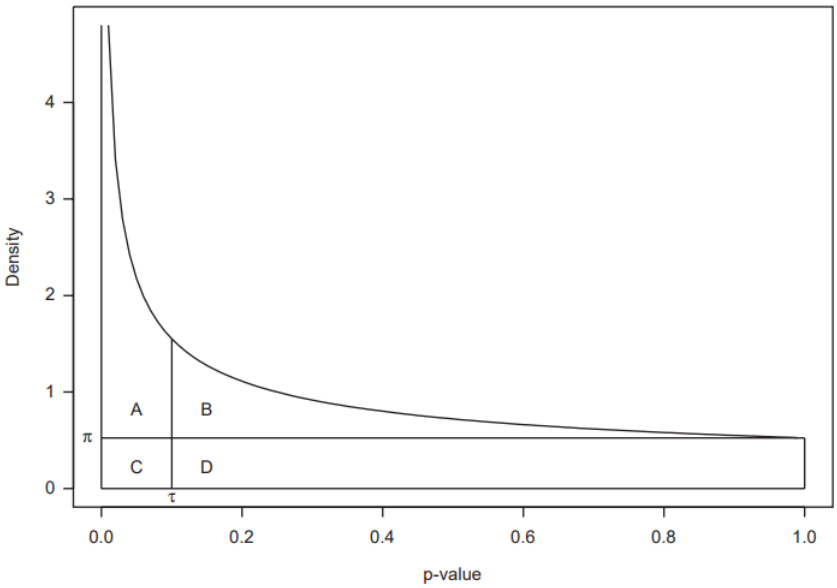

Figure 7: Partitioning of the BUM model. Source: [22]

2.3.4 Heinz

Our goal is to find groups of genes that are differentially expressed and interact with each other. For this purpose we use Heinz (heaviest induced subgraph). It needs the following input:

- a list of the nodes, in this case the genes with their p-values
- a list of the edges, in this case the interactions between the genes/proteins given by the PPI
- λ and a
- the upper bound α for the FDR

Heinz assigns an adjusted log likelihood ratio score to each node given by the following function:

$$S^{FDR}(x) = log(\frac{ax^{a-1}}{a\tau^{a-1}}) = (a-1)(log(x) - log(\tau(FDR)))$$

where x is the p-value and $\tau(FDR)$ is the threshold which is obtained with the method described in chapter 2.3.3. This way, p-values smaller than τ will be assigned a positive score whereas p-values bigger than τ will be given a negative score. When finding the set of connected nodes whose combined score is maximal, the scoring function makes sure that genes deemed insignificant (i.e. got a negative score) will only be included in the module if they pave the way to nodes or subgraphs that compensate the negative weight and contribute to a bigger net profit.

From an algorithmic point of view, our problem can be formulated as the Maximum-Weight Connected Subgraph Problem (MWCS):

Definition (MWCS). Given a connected, undirected vertex-weighted graph $G = (V, E, w)$ with weights w, find a connected subgraph $T = (V_T, E_T)$ of G with $V_T \subseteq V$, $E_T \subseteq E$, that maximizes the score $w(T) = \sum_{v \in V_T} w(v)$

In our case, the graph G is the PPI network, the weights w are the scores assigned to the nodes by Heinz and the connected subgraph T is the module we are searching for.

In order to find the best-scoring subgraph in acceptable runtimes without cutting back on optimality, Heinz transforms the MWCS instance into an instance for the prize-collecting Steiner tree problem (PCST).

Definition (PCST). Given a connected undirected vertex- and edge-weighted graph $G = (V, E, c, p)$ with vertex profits p and edge costs c, find a connected subgraph $T = (V_T, E_T)$ of $G, V_T \subseteq V, E_T \subseteq E$, that maximizes the profit

$$p(T) = \sum_{v \in V_T} p(v) - \sum_{e \in E_T} c(e)$$

How it is transformed is described in detail in [4]. Heinz makes use of the algorithm introduced by [23] which finds the optimal solution within short runtimes with the help of integer linear programming.

2.4 Enrichment

Up until now we computed networks of genes that for the biggest part were differentially expressed. The only biological information those networks hold is binary in the form of the edges. If and only if the proteins the genes encode interact with each other, there is an edge between them. To make the interpretation of these networks easier, we will enrich them with additional external data as described in chapter 2.4.1 and 2.4.2.

2.4.1 KEGG pathways

The Kyoto Encyclopedia of Genes and Genomes (KEGG) [14] aims to represent entire biological systems like cells or organisms by storing data divided into four groups:

- Genomic information - Genomes, proteins
- Chemical information - Biochemical reactions, enzymes
- Systems information - Pathways, structural complexes
- Health information - Diseases, drugs

For the enrichment we use the systems information database for pathways which can be accessed through the online interface or the biopython package Bio.KEGG.REST. Specifically for mus musculus, KEGG provides 326 manually drawn pathways such as glycolysis, the citrat cycle or the pentose phosphate pathway with a description, a list of involved genes and references. The resulting table built by the Snakemake pipeline is depicted in Table 4.

ID	Description	genes
path:mmu00010	Glycolysis / Gluconeogenesis	Hk2, Hk3, Hk1, Gck, (...)

Table 4: First entry of the KEGG pathway data

Now we want to find out if there are any overrepresented pathways in our modules. It is not desirable to directly map the genes to the pathways they are involved in as we would include them even if they have only a small overlap with our module. In order to find pathways that are significantly represented, we use a statistical method called Fisher's exact test. Given two binary variables, this

test provides information on how those variables are associated with each other. In this case, we want to test if the variable A: "The gene is part of the computed module" and variable B: "The gene is part of the pathway" are associated. First, we set up a contingency table as seen in Table 5 by counting the number of genes that are located in the four disjunct areas showed in Figure 8.

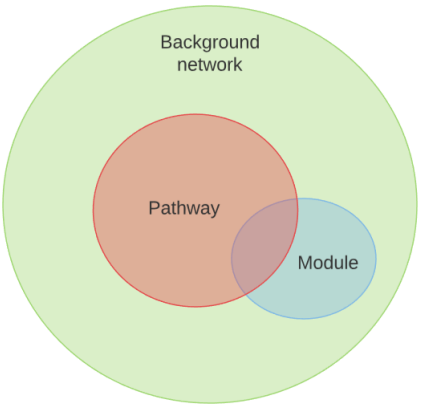

Figure 8: Set partition. The background network consists of all genes in the UKD data set with a p-value smaller or equal to 0.98

	in module	**not in module**	**row sum**
in pathway	a = 1	b = 1	a + b = 142
not in pathway	c = 2	d = 17420	c + d = 17422
column sum	a + c = 3	b + d = 17561	a + b + c + d = 17564

Table 5: Contingency table example

Under the null hypothesis H_0: A and B are not associated, it is proven that the distribution of the figures in the contingency table follow a so called hypergeometric distribution. With this information we are able to compute the probability of the actual observation now [24]:

$$p = \frac{(a+b)!(c+d)!(a+c)!(b+d)!}{(a+b+c+d)!\ a!\ b!\ c!\ d!}$$

In contrast to other hypotheses tests, the Fisher test assesses the *exact* probability of the observation by looking at all possibilities where an equally or more extreme distribution is achieved, taking advantage of the fixed column and row sums.

This test is applied to every pathway-module pair, resulting in a lot of p-values and the need to control the amount of false positives. In this case, we use a statistical correction method called the Bonferroni correction. The Bonferroni correction does not aim at the *rate* of false positives, instead it makes sure that the probability of having *at least one* false positive (also called the familywise error rate, FWER) does not exceed the limit α by declaring pathways as significantly represented (i.e. rejecting the null hypothesis) only if their p-value is smaller than α/m, with m being the total number of pathways which were deemed significant [25]. We choose $\alpha = 0.05$.

2.4.2 Gene Ontology terms

Having the same goal as KEGG, but a different approach, the Gene Ontology consortium [15] provides data in the form of so called GO Terms which can be split up into three categories:

- Molecular function, e.g. GO term GO:001530: lipopolysaccharide binding
- Biological process, e.g GO term GO:0034097: response to cytokine
- Cellular component, e.g. GO term GO:0045202: synapse

Every GO term has a unique identifier and a description of the function, process or component it stands for. The terms can be arranged in a directed acyclic graph with three root nodes (i.e. the categories), and with every step down the tree the GO terms get more and more specific - e.g. from "regulation of biological process" down to "positive regulation of eye pigmentation". As we have seen with Ubc, a gene can have many functions in a biological system. Thus, a gene can be part of multiple GO terms. To compute the enrichment, we download all GO terms that have at least one associated gene that can be found in mus musculus via biomaRt (in general, GO terms are species-agnostic). Then we filter out double entries so that each term has one entry only and add the ancestors from the DAG to the entry via the GO.db. Analogous to the KEGG pathways, we only want to include GO terms that are significantly represented. By calling the runTest function of the topGO R package, we compute up to 20 GO terms with the Fisher's exact test for each category. Then we trace back which genes from our module are part of which significant GO terms and store the information in a format that can be used by eXamine (chapter 2.5).

2.5 Visualization

For the purpose of portraying the results, a standalone version of the Cytoscape app eXamine is used [26]. eXamine is a tool that can be used for a set-oriented visual analysis approach for annotated modules. The pipeline splits the computed module and its associated enrichment up into the following files:

- proteins.nodes: List of the nodes of the module together with their score, their fold change and a hyperlink to the corresponding genecards website.
- interactions.links: List of the edges.
- go_and_kegg_annotations: List of KEGG pathways and GO terms, includes a short description and in case of GO terms also information on which category it belongs to (biological process, function, or component) and a hyperlink to the corresponding geneontology website.
- go_and_kegg.memberships: Links the proteins to the GO terms and KEGG pathways they are part of.

If more than one module is shown at the same time, it is possible to make them visible by creating the modules in the file modules.annotations and link the belonging nodes in modules.memberships. As we will highlight the functional modules dynamically according to their GO terms and KEGG pathways, we only have one hard coded module containing every node.

With those files, eXamine will portray the computed modules as seen in Figure 9. The color of the nodes corresponds to their associated p-value or fold change, depending on the settings. On the right side, the user can hover over the KEGG pathways or the GO terms to highlight the involved genes or click on them to mark them permanently with color. When clicking directly on a gene node, the user gets redirected to the genecards website with additional information, e.g. other aliases or expression levels. eXamine will load every directory that is inside the folder named "data-sets". A list of the available modules can be viewed by clicking on the button on the left side at the bottom.

3 Results

Due to the high deviation from the beta uniform distribution, six data sets comparing gene expression levels between the control groups could not be analyzed with our approach. We chose to focus on the comparisons out of the remaining 26 data sets that hold information on how gene expression levels changed in the direct heart infarction tissue and in the remote, undamaged heart tissue:

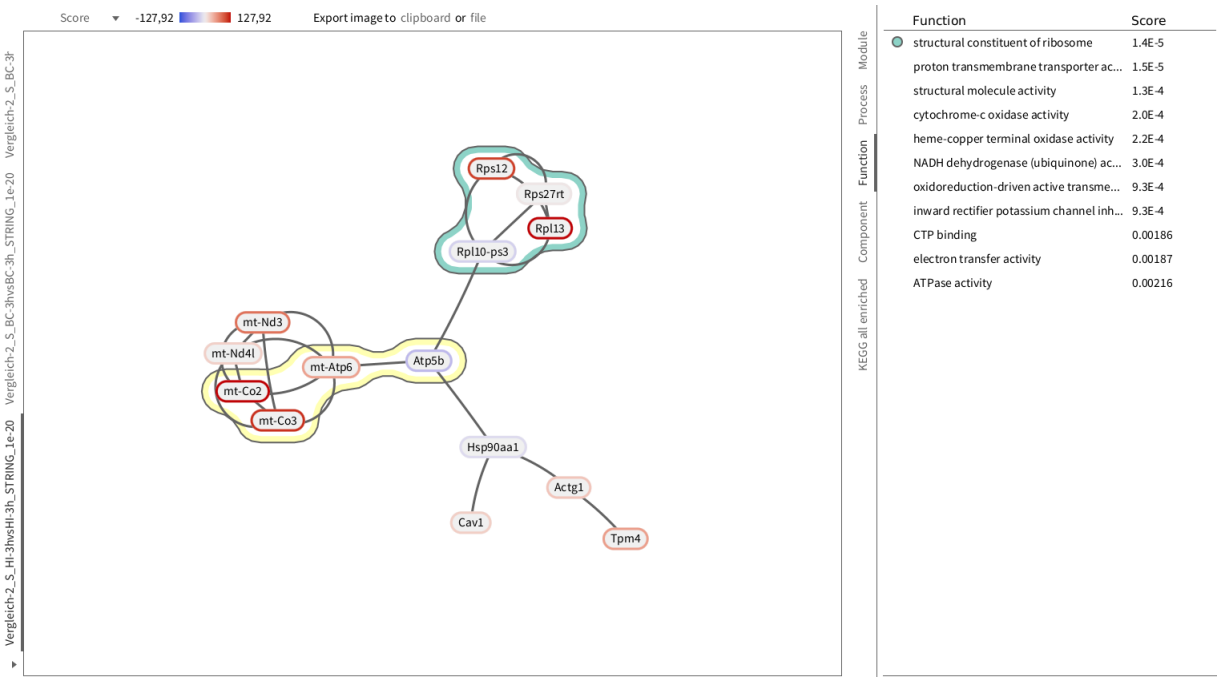

Figure 9: Visualization of the results in eXamine

- Comparison group 1:
  - Heart infarction tissue 3h vs heart infarction tissue 24h
  - Heart infarction tissue 3h vs heart remote tissue 3h
  - Heart infarction tissue 24h vs heart remote tissue 24h
  - Heart remote tissue 3h vs heart remote tissue 24h
- Comparison group 2:
  - Heart infarction tissue (Sham) 3h vs. heart infarction tissue 3h
  - Heart remote tissue (Sham) vs. heart remote tissue 3h
- Comparison group 3:
  - Heart infarction tissue (Sham) 24h vs. heart infarction tissue 24h
  - Heart remote tissue (Sham) 24h vs. heart remote tissue 24h

Most of the computed modules unfortunately do not show any striking results. If they included any functional modules, they would have been visible by a high degree of interconnectivity and by being part of the same GO terms or KEGG pathways. Instead, the associated KEGG pathways and GO terms are mostly

spread all over the networks without significant connections between the nodes that belong to them.

Nevertheless, when looking at the modules computed with the STRING background network, three functional modules stand out in all four sets that compare the heart tissue (damaged, HI, and undamaged, HR) from the control groups to the I/R groups (see Figure 10).

- GO:0003735: Structural constituent of ribosome (yellow).
  Definition: The action of a molecule that contributes to the structural integrity of the ribosome.
  Includes: Rps27rt, Rps12, Rpl3 (sometimes Rpl9-ps6, Rpl35, Rpl13).
- GO:0008137: NADH dehydrogenase (ubiquinone) activity (violet).
  Definition: Catalysis of the reaction: NADH + H^+ + ubiquinone = NAD^+ + ubiquinol.
  Includes: Ndufa12, mt-Nd3, mt-Nd4l.
- GO:0005743: Mitochondrial inner membrane (blue).
  Definition: The inner, i.e. lumen-facing, lipid bilayer of the mitochondrial envelope. It is highly folded to form cristae.
  Includes: Ndufa12, Atp5b, mt-Atp6, mt-Co3, mt-Co2.

The first two are biological function GO terms, the latter belongs to the cellular component category. All of the corresponding genes were expressed at higher levels in the I/R groups than in the control groups. As the NADH dehydrogenase is a complex of the electron transport chain in the inner mitochondrial membrane, two of the three modules hint to a higher activity of mitochondria in both the damaged and the undamaged tissue after a myocardial infarction. A possible reason for this is increased demand for ATP (the energy currency in cells) that can be funneled into other cell processes (e.g. cell growth or proliferation) in order to reestablish the tissue functions. Further analysis by the experts could not be done within the time frame of the thesis but will be done in the near future.

The findings could not be verified with the modules computed with BioGRID as they do not show a big similarity to the ones that used STRING, the only overlap are the two genes Rpl13 and Rps12. Modules coming from the same data set but used a different network showed little overlap in general. Only 34 out of 432 computed GO terms and KEGG pathways were found both in the module computed by STRING and by BioGRID. The most shared GO terms and KEGG pathways can be found in the modules for heart infarction tissue 3h vs heart remote tissue 3h (24 out of 77), heart infarction tissue 24h vs heart remote tissue 24h (6 out of 47) and heart infarction tissue 3h vs heart infarction tissue 24h (3 out of 63). The rest of the modules can be found in the appendix.

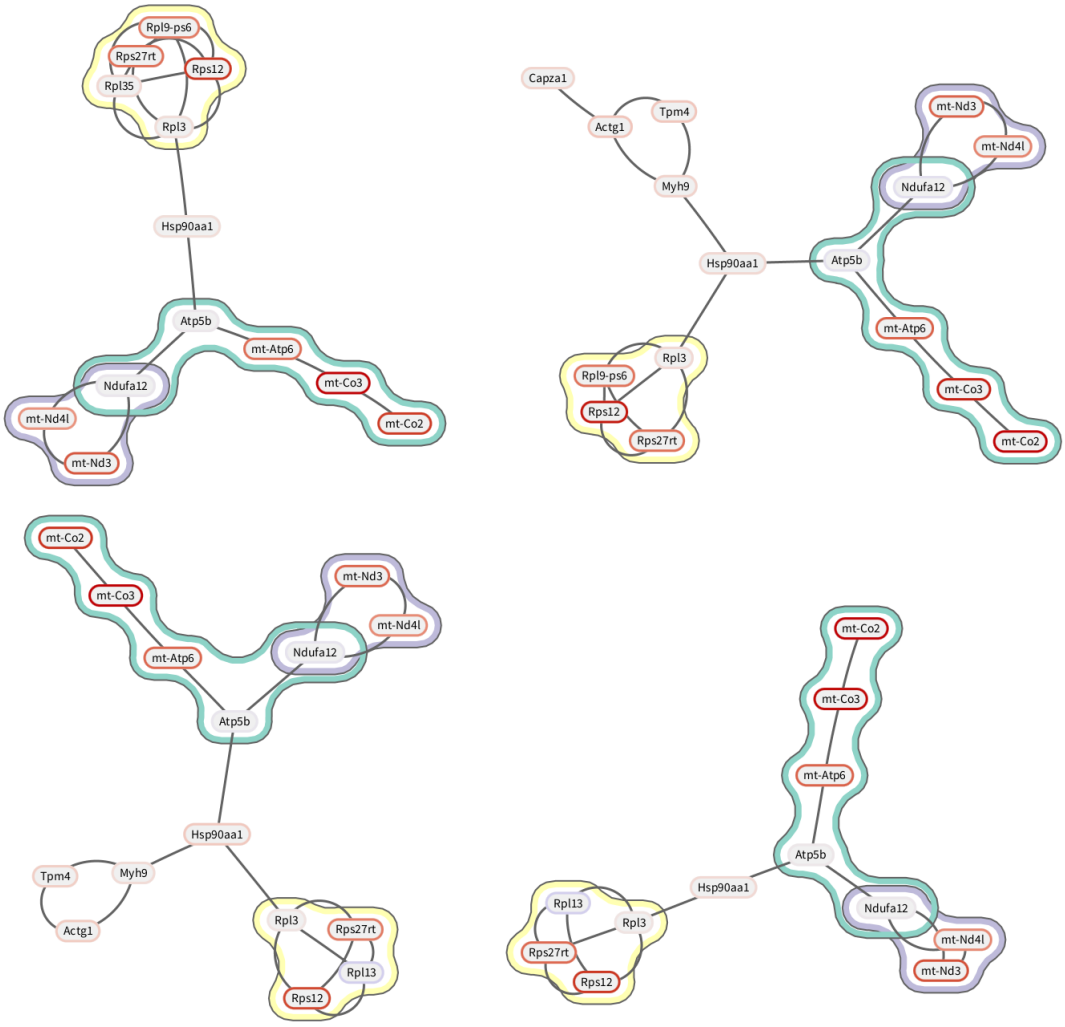

Figure 10: Functional modules with STRING. Top left: 3_S_HR-24hvsHR-24h with FDR 1e-10, top right: 3_S_HI-24hvsHI-24h with FDR 1e-10, bottom left: 2_S_HI-3hvsHI-3h with FDR 1e-20, bottom right: 2_S_HR-3hvsHR-3h with FDR 1e-20

4 Evaluation

4.1 Efficiency

By implementing the workflow with Snakemake, we achieve a high level of efficiency. Snakemake makes sure that in order to compute the target files, only the necessary actions get taken. While this effect may be negligible when running the analysis for the first time, the reuse of already generated files like the STRING interactions greatly decreases runtime later on. The automatization also helps to

deal with the 32 data sets, as manually conducting the analysis so many times would be both time consuming and error-prone. In addition to the configuration files that are assigned to the rules, there is a general configuration file that stores parameters such as the names of the experiments, the FDRs and networks. This separation allows the user to customize the workflow without changing the code. The data sets on which the network analysis should be performed can be selected here, as well as the FDR which can be set individually for each data set. Therefore it is possible to execute the workflow on multiple data sets with different FDRs in one run. If only interim results are of interest, for example for debugging purposes, superfluous steps can be avoided by setting the target files accordingly.

4.2 Reproducibility

The pipeline computes the analysis from scratch and without the need of manual interference. If a Snakemake rule makes use of an external package, the version that is used is recorded in a configuration file. We also tried to prescribe the versions of the external data sources. The versions of STRING and BioGRID are defined in the download link. Ensembl, the data source we got the GO Terms from, also provides archived versions that can be accessed directly by stating the desired version as a parameter in the biomaRt search query. This works for all versions except for the newest, so in order to ensure consistent results in the future, the parameter has to be set once there is a new update. KEGG on the other hand does not have any archived versions of their pathway lists and instead features an online protocol that records title changes or added pathways [27]. We refrained from controlling the version here because of the slow rate of change (in 2018, 12 changes were recorded) and the small probability that those changes would influence results. If that was the case, it would still be possible to compare old results with new ones with only little effort by taking the protocol into account.

4.3 Comparison to IPA

Qiagen, the company that provides the software used for the differential gene expression analysis, also offers a program for network analysis itself called Ingenuity Pathway Analysis (IPA). The basic principles are the same, it combines the results from the DGE analysis with external sources like public databases and computes highly represented pathways, associated diseases or cellular functions. However, there are differences that come with both advantages and disadvantages. First of all, IPA is a commercial product, so in order to use it a license needs to be bought. While our pipeline is freely available on github, it depends on the IBM software ILOG CPLEX which is also commercial, so it is only free to use if a

license for CPLEX already exists. On the other hand, in order to make profit out of the software, Qiagen needs to keep the code private, making it impossible to retrace the exact steps or to adapt it to special needs. This also has the effect that the actual analyses are not executed on the local computer, instead the input data is uploaded to the Qiagen server and after a short time, the results get sent back. However, as long as the external data sources like STRING and KEGG have already been downloaded, our implemented network analysis works offline. The size of the input file is also restricted, requests for analyses on more than 8000 genes are denied. For the sake of satisfying this rule, genes may need to be taken out before the actual analysis, for example by removing entries above a certain p-value or fold change threshold. In our case, the pipeline was successfully tested with input files with around 48000 entries out of which around 17000 were taken into account in the core analysis with Heinz. The biggest advantage IPA poses in contrast to our pipeline is the amount of data sources it uses for its analyses. While our pipeline only takes one source per run (either STRING or BioGRID), IPA provides multiple sources that can be chosen to be used together in a single analysis, making the results more informed and reliable. Moreover, IPA has a lot of features our pipeline does not offer, like choosing whether the analysis should be depending on the fold change or on the p-values, adding own pathways, displaying the resulting networks in a cell or opting out indirect protein-protein interactions.

To summarize, our pipeline represents an easy to use method for network analyses that can be retraced step by step and if needed can even be alternated. However the results seem to be heavily depending on the PPI network that is used. IPA provides more reliable results by combining several networks and if used correctly the additional features contribute to a more specific analysis.

4.4 Limitations

Special care is required when interpreting the modules computed with the STRING background PPI due to the protein interactions only being *predicted*. BioGRID provides more reliable information by only reporting experimentally verified interactions [28], but this does not mean that the network is completely free of fault. For example, BioGRID contains data from papers reporting high throughput screens [29], a type of experiment that was called out for having a high false discovery rate and only representing snapshot interaction information [30]. Moreover the results can not be immediately applied to humans. Mice have been established as main models to study human biology because of genetic and physiological similarities, but they are less reliable as models of human disease because the networks linking genes to disease are likely to differ between the two species [31]. Also, the pipeline does not necessarily compute *functional* modules

as can be seen in the modules computed with BioGRID. Unfortunately they do not contain any significant biological context, perhaps because of key genes filtered out in preprocessing steps or being slightly below the threshold. Despite all the problems tackled in the evaluation, we believe that the implemented network analysis is a useful tool in order to gain a different kind of insight into differential gene expression data and the results can guide the direction in which further research could go.

5 Outlook

Concerning the mouse transcriptome data, the network analysis could be improved by making it prejudiced in favor of genes that are already known or expected to be involved in the healing process after a myocardial infarction. Perhaps then the analysis will include key genes that are apparently missing in many of the computed modules. For further research on the biological reasons behind the results, the findings were given to the cardiologists from the university hospital that also conducted the experiment.
One improvement to the general pipeline would be the implementation of a feature that lets the user set the desired number of nodes the resulting network should have, for example by iterating the process until the set number is at least approximated. The scaling of the module through the FDR makes sense on a statistical level, but when conducting the analysis for the very first time setting the value is a shot in the dark and most of the time results in modules that are either too big or too small. To find good FDRs for the UKD data sets, we used binary search to slowly approach a value where a manageable module size arose, but doing this for all 32 data sets was time consuming.
To overcome the dependency from the PPI networks, we can imagine that using meta databases such as APID could help. Those meta databases unify and combine the knowledge gathered from several other primary databases (such as BioGRID). We refrained from using APID because of the nonexistence of documentation regarding updates or changes in the data base, but perhaps there is a way to use meta databases without having to cut off replicability. An addition that is already planned is the expansion of the workflow so that it does the differential gene expression analysis itself instead of using the already computed p-values and fold changes.

6 Acknowledgements

A big, grateful Thank You goes out to Gunnar Klau for taking time for me even with a full schedule and for making this thesis possible to begin with, to Eline van Mantgem for the help with the setup, the proof reading and the emotional support, to Philip Spohr for explaining the server related things, to Patrick Petzsch for patiently leading me through IPA, and to Karl-Erich Köhrer for answering all of my questions about the biological things.

A Appendix

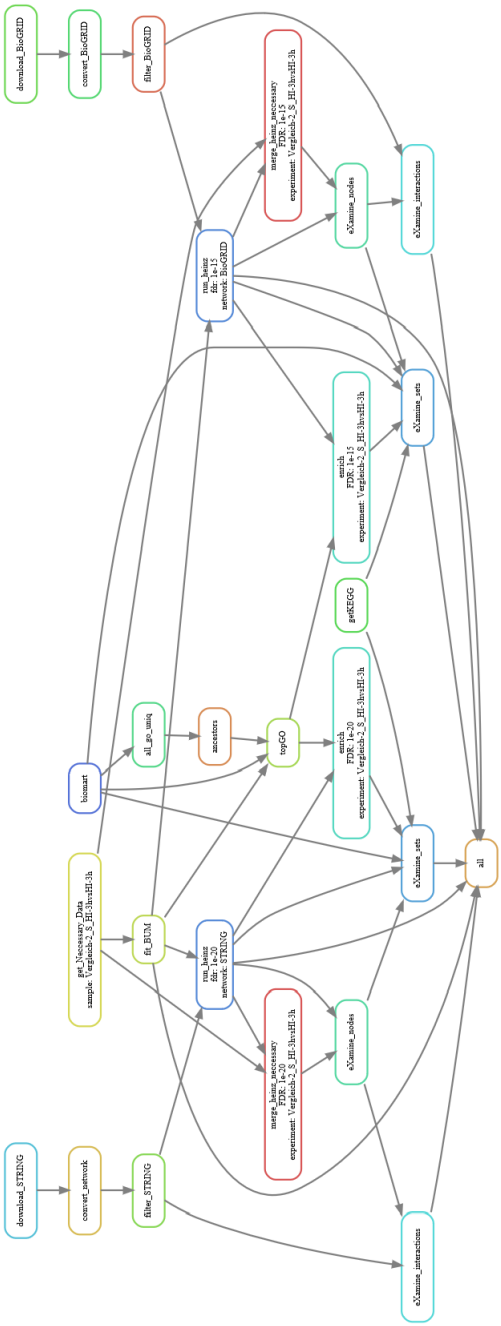

Figure 11: Full DAG created by Snakemake

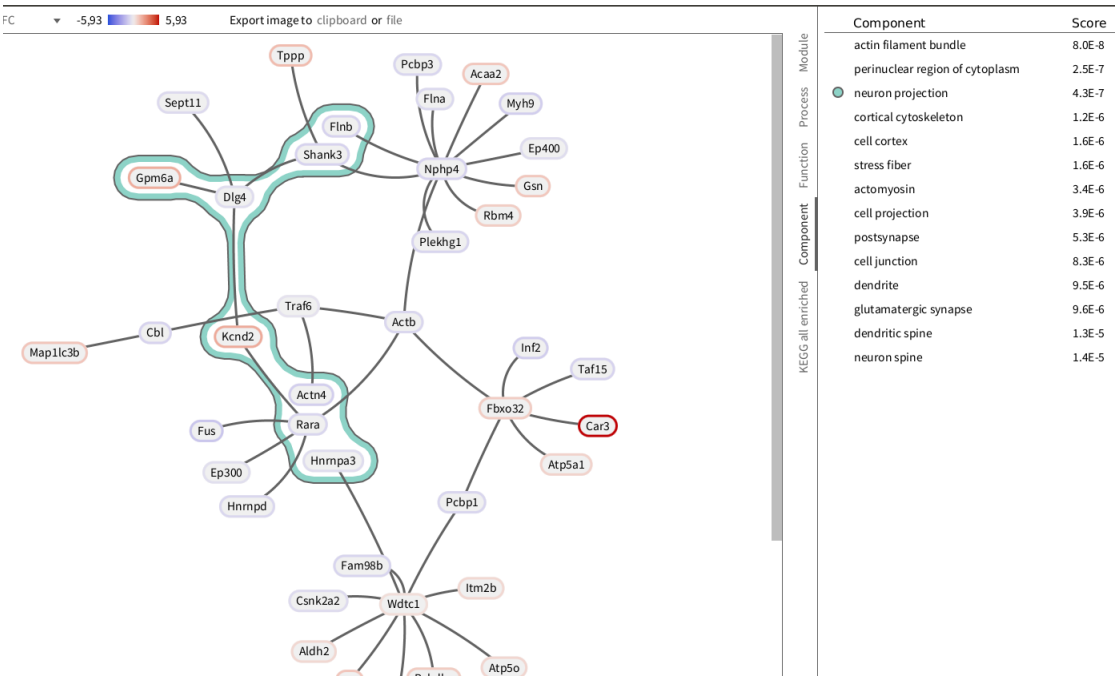

Figure 12: 1_HI-24hvsHR-24h_BioGRID, FDR 0.0012

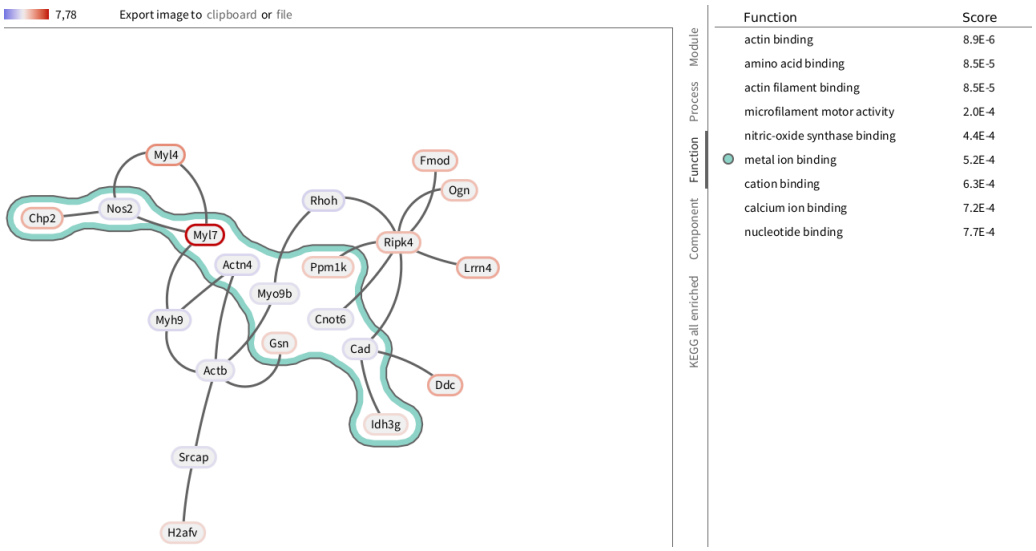

Figure 13: 1_HI-24hvsHR-24h_STRING, FDR 6e-4

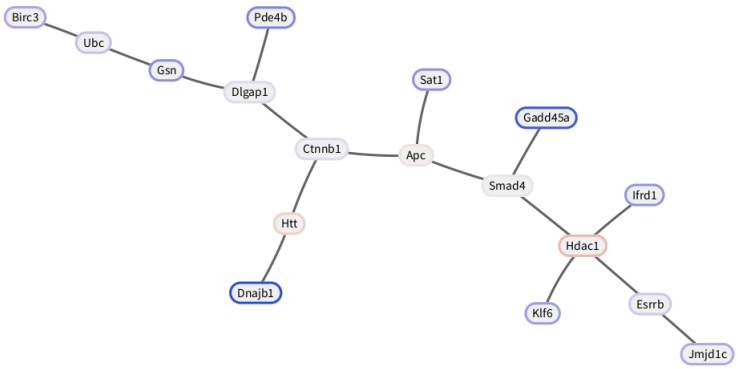

Figure 14: 1_HI-3hvsHI-24h_BioGRID, FDR 1e-15

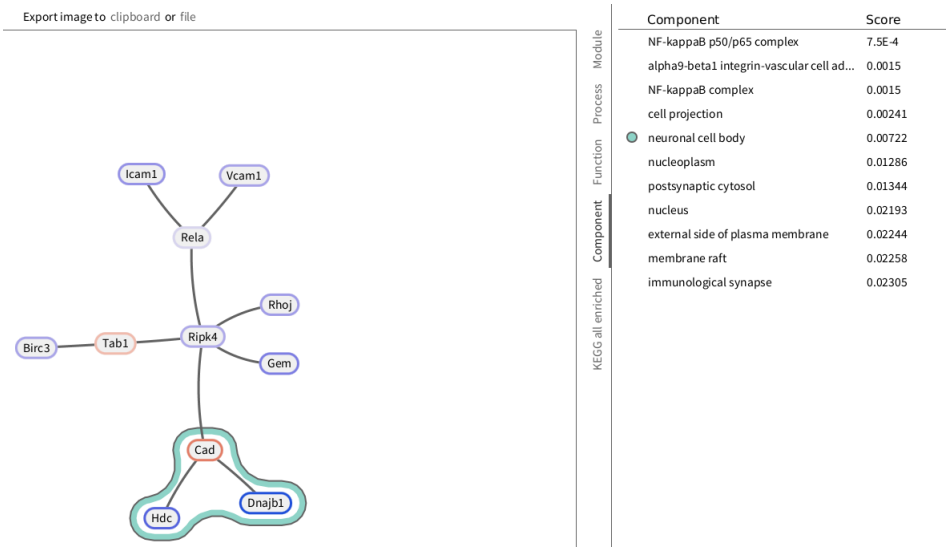

Figure 15: 1_HI-3hvsHI-24h_STRING, FDR 1e-15

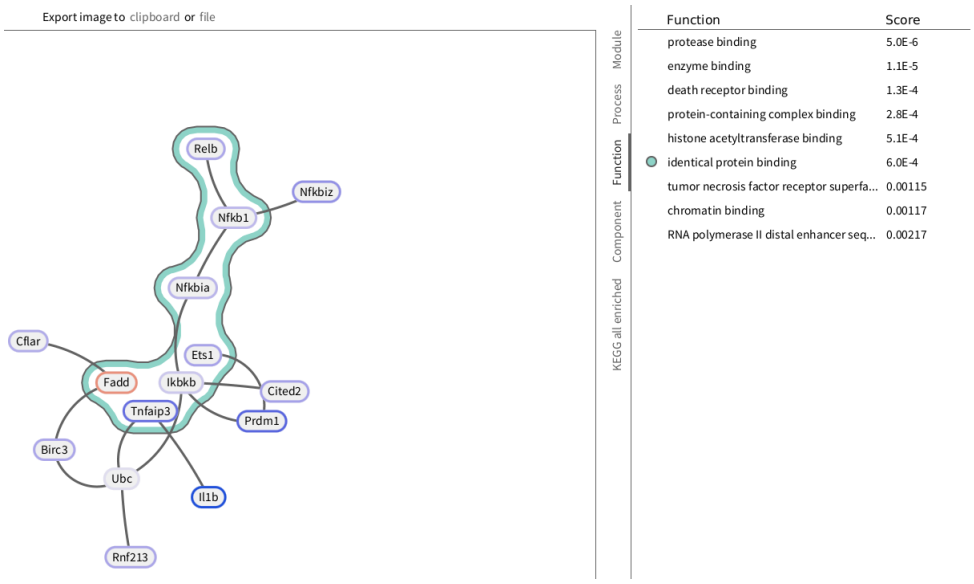

Figure 16: 1_HI-3hvsHR-3h_BioGRID, FDR 1e-6

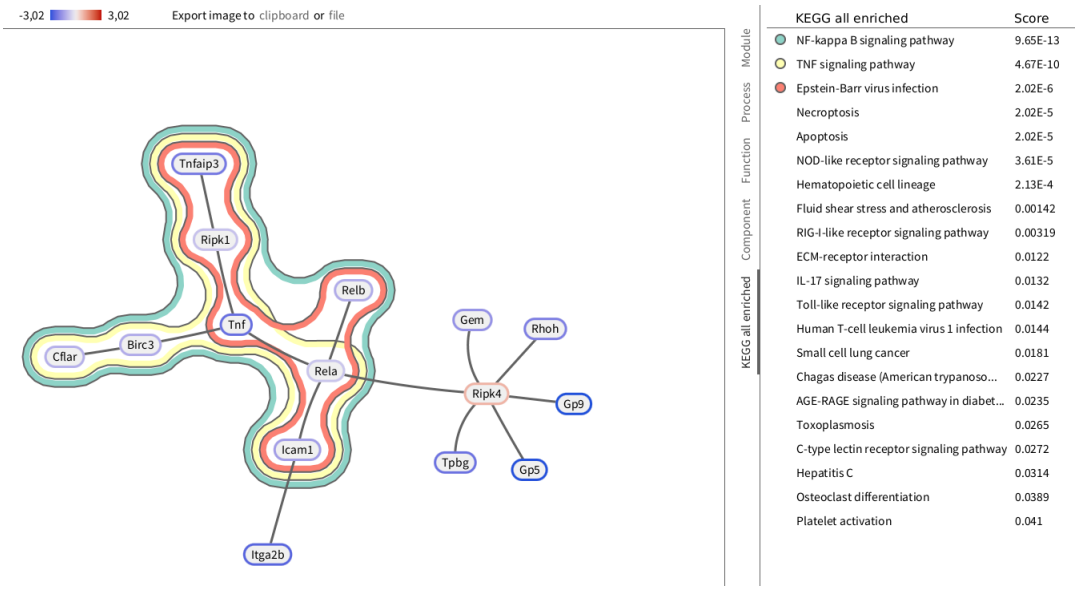

Figure 17: 1_HI-3hvsHR-3h_STRING, FDR 1e-7

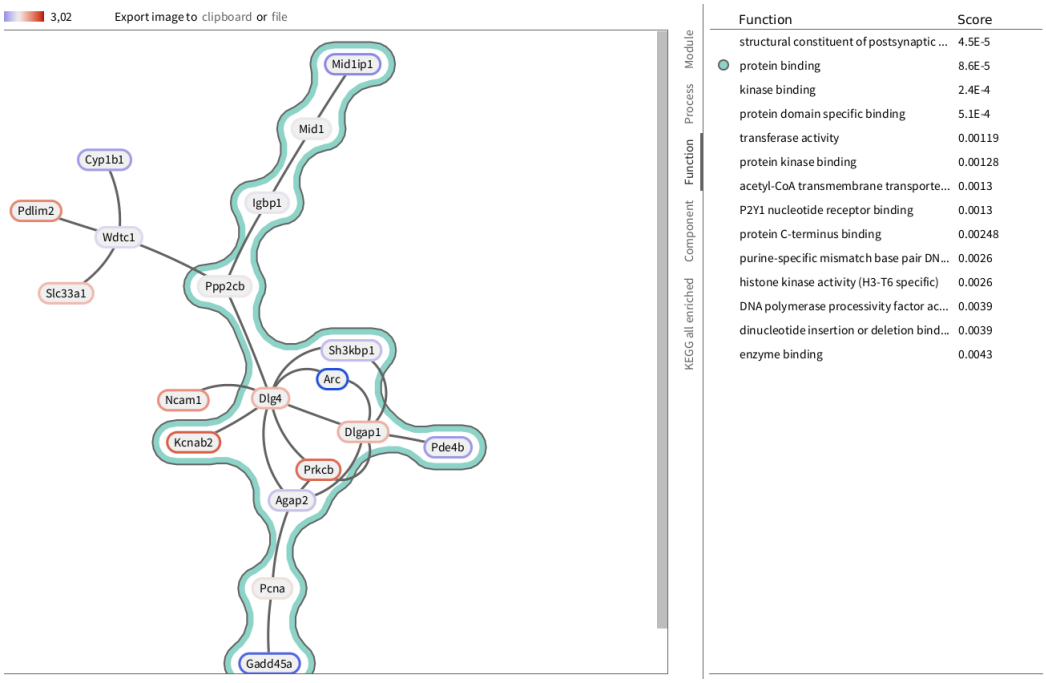

Figure 18: 1_HI-3hvsHR-3h_STRING, FDR 1e-7

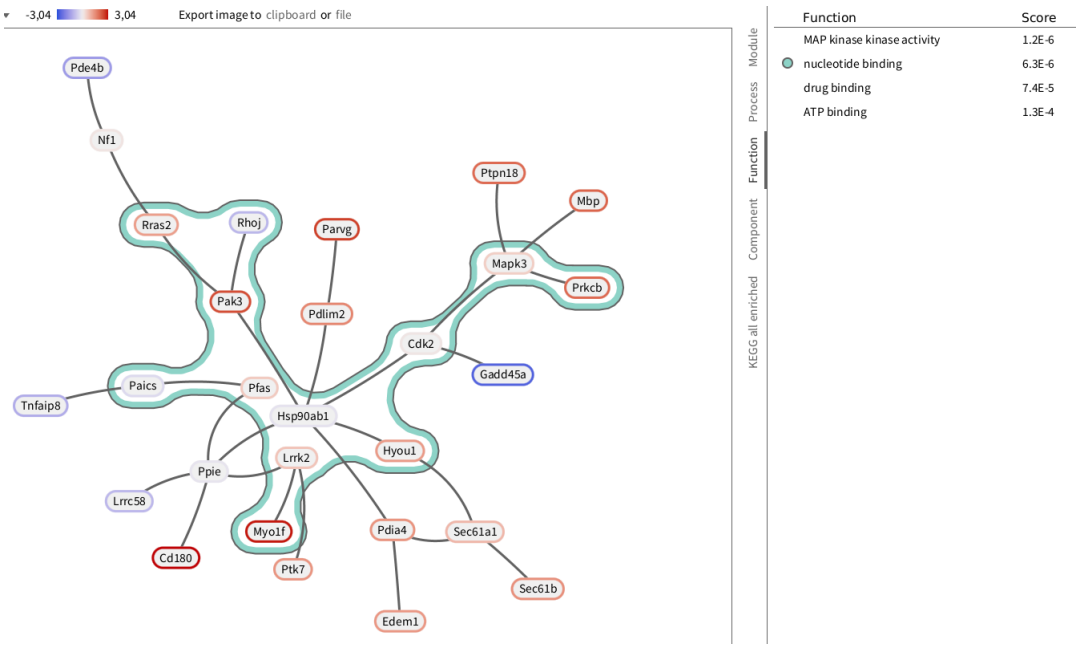

Figure 19: 1_HR-3hvsHR-24h_STRING, FDR 3e-7

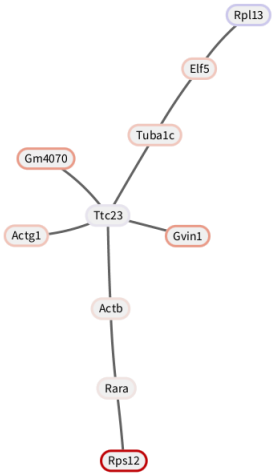

Figure 20: 2_S_HI-3hvsHI-3h_BioGRID, FDR 1e-15

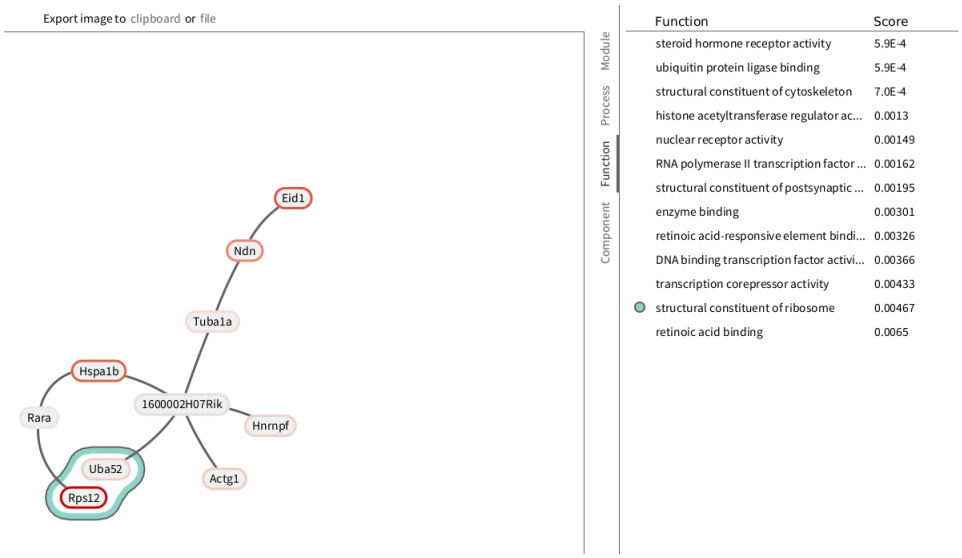

Figure 21: 2_S_HR-3hvsHR_BioGRID, FDR 1e-10

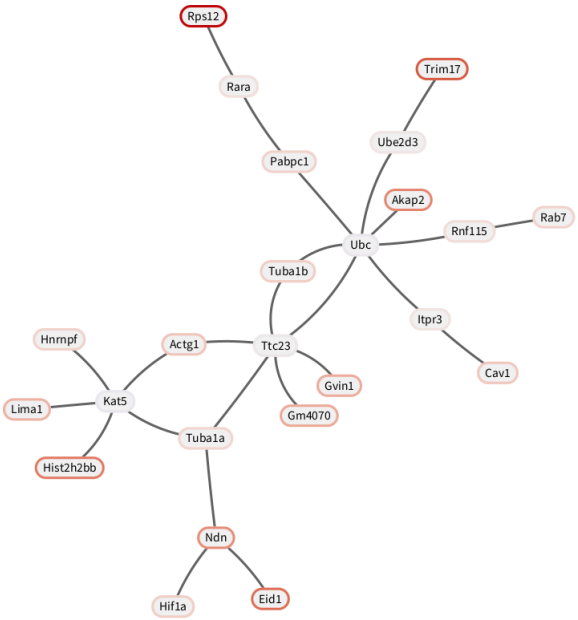

Figure 22: 3_S_HI-24hvsHI-24h_BioGRID, FDR 1e-8

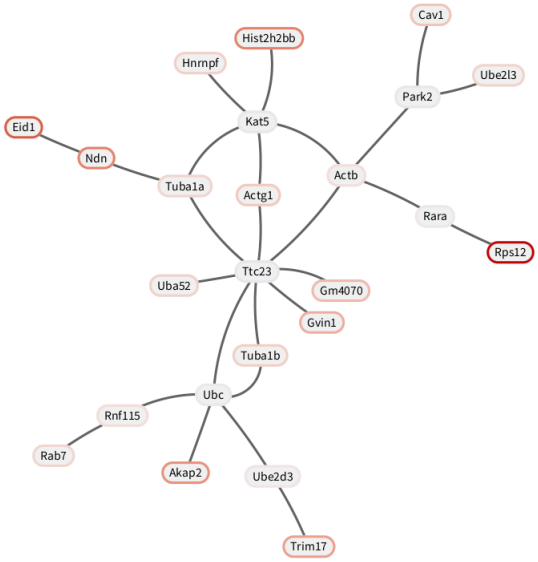

Figure 23: 3_S_HR-24hvsHR-24h_BioGRID, FDR 1e-6

References

[1] Jane Reece et al. "Campbell Biologie, 10. Auflage". In: Pearson, 2016, p. 1248.

[2] Oren J Mechanic and Shamai A Grossman. "Myocardial Infarction, Acute". In: (2017).

[3] Anais Baudot, Ouissem Souiai, and Christine Brun. "Network analysis and protein function prediction with the PRODISTIN Web site". In: *Bacterial Molecular Networks*. Springer, 2012, pp. 313–326.

[4] Marcus T Dittrich et al. "Identifying functional modules in protein-protein interaction networks: an integrated exact approach". In: *Bioinformatics* 24.13 (2008), pp. i223–i231.

[5] *CLC Genomics Workbench Manual*. URL: `http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/601/index.php?manual=RNA_Seq_analysis.html` (visited on 06/01/2019).

[6] *CLC Genomics Workbench Manual, Statistical analysis*. URL: `http://resources.qiagenbioinformatics.com/manuals/clcgenomicsworkbench/700/index.php?manual=Statistical_analysis_identifying_differential_expression.html` (visited on 06/01/2019).

[7] Christian von Mering et al. "STRING: a database of predicted functional associations between proteins". In: *Nucleic acids research* 31.1 (2003), pp. 258–261.

[8] Michael PH Stumpf et al. "Estimating the size of the human interactome". In: *Proceedings of the National Academy of Sciences* 105.19 (2008), pp. 6959–6964.

[9] *BioGRID*. URL: `https://thebiogrid.org/` (visited on 06/01/2019).

[10] *STRING*. URL: `https://string-db.org/` (visited on 06/01/2019).

[11] *Snakemake*. URL: `https://snakemake.readthedocs.io/en/stable/` (visited on 08/01/2019).

[12] Mohammed El-Kebi and Gunnar W. Klau. *Heinz - single species module discovery*. URL: `https://github.com/ls-cwi/heinz` (visited on 07/01/2019).

[13] IBM. *CPLEX IBM ILOG CPLEX Optimization Studio*. URL: `https://www.ibm.com/products/ilog-cplex-optimization-studio` (visited on 06/01/2019).

[14] *Kyoto Encyclopedia of Genes and Genomes*. URL: https://www.genome.jp/kegg/ (visited on 06/01/2019).

[15] *Gene Ontology Consortium*. URL: http://www.geneontology.org/ (visited on 06/01/2019).

[16] *Ensembl genome browser 94*. URL: https://www.ensembl.org/index.html (visited on 06/01/2019).

[17] *biomart*. URL: http://www.biomart.org/ (visited on 06/01/2019).

[18] *PyBiomart - A simple and pythonic biomart interface for Python*. URL: https://jrderuiter.github.io/pybiomart/index.html# (visited on 06/01/2019).

[19] Gordon J Lithgow, Monica Driscoll, and Patrick Phillips. "A long journey to reproducible results". In: *Nature News* 548.7668 (2017), p. 387.

[20] *Snakemake Tutorial*. URL: https://snakemake.readthedocs.io/en/stable/ (visited on 06/01/2019).

[21] Larry Wasserman. *All of statistics: a concise course in statistical inference*. Springer Science & Business Media, 2013.

[22] Stan Pounds and Stephan W Morris. "Estimating the occurrence of false positives and false negatives in microarray studies by approximating and partitioning the empirical distribution of p-values". In: *Bioinformatics* 19.10 (2003), pp. 1236–1242.

[23] Ivana Ljubić et al. "An algorithmic framework for the exact solution of the prize-collecting Steiner tree problem". In: *Mathematical programming* 105.2-3 (2006), pp. 427–449.

[24] Jenny V Freeman and Michael J Campbell. *The analysis of categorical data: Fisher's exact test*. URL: https://www.sheffield.ac.uk/polopoly_fs/1.43998!/file/tutorial-9-fishers.pdf (visited on 08/01/2019).

[25] Alexander Gordon et al. "Control of the mean number of false discoveries, Bonferroni and stability of multiple testing". In: *The Annals of Applied Statistics* 1.1 (2007), pp. 179–190.

[26] *eXamine - A set-oriented visual analysis approach for annotated modules that displays set membership as contours on top of a node-link layout*. URL: https://github.com/ls-cwi/eXamine (visited on 07/01/2019).

[27] *KEGG change history*. URL: https://www.genome.jp/kegg/docs/upd_map.html (visited on 06/01/2019).

[28] Nahid Safari-Alighiarloo, Mohammad Taghizadeh, and Mostafa Rezaei tavirani. "Protein-protein interaction databases: an overall view on interactome organization the nature of protein-protein interactions data". In: *International journal of analytical, pharmaceutical and biomedical sciences* 4 (Jan. 2015), pp. 15–23.

[29] *BioGRID Documentation*. URL: `https://wiki.thebiogrid.org/doku.php/high_throughput_screens` (visited on 06/01/2019).

[30] Bo Xu et al. "Reconstruction of the Protein-Protein Interaction Network for Protein Complexes Identification by Walking on the Protein Pair Fingerprints Similarity Network". In: *Frontiers in genetics* 9 (2018).

[31] Robert L Perlman. "Mouse models of human disease: An evolutionary perspective". In: *Evolution, medicine, and public health* 2016.1 (2016), pp. 170–176.

List of Figures

List of Tables