Karlsruher Institut für Technologie

Analysis of human tissue-specific protein-protein interaction networks

Master thesis
of

Patrick Flick

at the Faculty of Computer Science

Primary reviewer:	Prof. Dr. Alexandros Stamatakis
Secondary reviewer:	Juniorprof. Dr. Henning Meyerhenke
Advisors:	Dr. Tomas Flouri
	Francesco Gatto

Bearbeitungszeit: 15. November 2013– 14. Mai 2014

KIT – Universität des Landes Baden-Württemberg und nationales Forschungszentrum in der Helmholtz-Gemeinschaft **www.kit.edu**

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Karlsruhe, May 14, 2014

Abstract

Proteins are the core machinery of all living cells and protein interactions determine the inner workings of life itself. Insights into the nature of these interactions are important for learning about how and why cells work. The interactions between all proteins in a cell compose a so-called *protein-protein interaction (PPI) network*, in form of a graph. Not all proteins are present in all cell and tissue types, hence protein interactions are restricted to cell and tissue types where both interacting proteins exist. These tissue dependent interactions form *tissue-specific PPI (TSPPI) networks*.

In this thesis, we construct and analyze *TSPPI* networks from different data sources. We follow the goal to gain insights into the structure of interactions as well as into the properties of specific groups of proteins inside the TSPPI networks. To that end, we implement an analysis pipeline and develop efficient analysis algorithms, which operate on our graph representation for *TSPPI* networks. Moreover, we study the basic properties of *TSPPI* networks and investigate properties of certain classes of proteins. Then, we provide a method to identify proteins which gain in importance by cellular specialization. Furthermore, we re-evaluate prior research results on a large set of *TSPPIs* and demonstrate that some previous conclusions have to be reconsidered. Finally, we employ clustering algorithms with the objective to identify tissue-specific functional modules within *TSPPIs*. In addition to using available clustering methods, we pursue two more approaches.

Contents

Chapter 1

Introduction

1.1 Thesis structure

In this Chapter, we introduce the background and common terms and concepts used throughout this thesis. First, we explain the biological basics of protein-protein interaction networks and tissue specific expression patterns. We then introduce concepts from graph theory and complex network analysis.

Chapter 2 summarizes related work and prior research on the analysis of tissue-specific protein-protein interaction networks. Additionally, we coarsely lay out the differences between the various data sources, which we make use of in this thesis.

In Chapter 3 we explain our analysis pipeline, the basic properties of the datasets used, and the algorithms that we developed and implemented.

Next, we briefly describe the implementation of our analysis pipeline in Chapter 4.

In Chapter 5 we then evaluate the performance of our algorithms. Furthermore, we explain the analysis performed on the tissue-specific interaction networks and show the results obtained.

Finally, in Chapter 6 we summarize our work, draw conclusions and outline possible future work.

1.2 Biological background

This section is primarily designed for readers without a biological background. We will shortly explain the important biological terms and concepts needed for this thesis. If you feel confident with the concept of proteins and protein-protein interactions, you may skip the following few sections and continue with section 1.2.5.

First of all, in Section 1.2.1 we define the term *protein* and discuss how proteins relate to DNA, mRNA, and genes. Next in Section 1.2.2, we explain the concept of *protein-protein interactions* and give a few examples. We then describe how the interactions define a *protein-protein interaction network* (Section 1.2.3). Furthermore, the concepts of tissue specificity and protein expression are elaborated in Section 1.2.4, and finally, we explain how these are used to define *tissue-specific protein-protein interaction networks* (Section 1.2.5).

1.2.1 Proteins

Proteins are large macromolecules which can perform a variety of biological and chemical functions. They are the core machinery of all living cells and are responsible for DNA replication, signaling, metabolism, the inner and outer structure of cells, transporting other proteins and substances throughout the cell, and many other tasks. All proteins consist of the same basic building blocks called *Amino Acids*, only 20 different amino acids make up all proteins. The sequence of these amino acids is coded for by genes, which are subsequences of the DNA/Genome of the cell [2].

1.2.2 Protein-protein interactions

In order to fulfill their function, proteins interact with other substances (molecules, ions, DNA, . . .) or other proteins. Proteins interact in numerous different contexts and with different outcomes. Some proteins activate or deactivate other proteins by binding to them or by (de-)phosphorylating them. In the process of phosphorylation, a phosphate group is added (or removed) from a protein, which turns the protein on or off. Other proteins bind to each other, creating so-called protein-complexes. These have important roles in the entire cell, for instance in DNA replication. Another class of proteins bind to each other to create structural complexes which give the cell its 3-dimensional structure. Yet other proteins pass on signals by interacting with source and destination proteins in so-called signaling pathways. Transcription factors are proteins that bind to DNA to activate the transcription process (i.e., the expression) of a gene. This activation often requires multiple transcription factors to interact and bind to the DNA together [2].

All of these interactions, including many more, are the central part of the functioning of the cell. Understanding these core interactions is vital to understanding the inner workings of life itself.

A number of different approaches have been followed towards reconstructing the interactions between proteins (see 2.1). The literature comprises studies that demonstrate the presence of single or few interactions. Others use high-throughput experiments to find if there exist pairwise interactions between a large set of query proteins. Yet other studies use computational modeling to determine which proteins may bind to each other based on their (predicted) structural properties.

1.2.3 Protein-protein interaction networks

We represent protein-protein interactions as an undirected and unweighted graph $G(V, E)$, where the set of nodes V are the proteins. An edge $(p_1, p_2) \in E$ is present *iff* there is an interaction between the two proteins p_1 and p_2.

Since there are multiple sources for protein interactions from different studies and databases, there are also different graphs representing the interactions found in these studies.

Such a graph, representing protein-protein interactions, is commonly called a *protein-protein interaction network*, or *PPI network* for short.

1.2.4 Protein expression and tissue-specificity

Proteins are created from genes in a process called *gene expression*. In this process, initially genes are *transcribed* from the DNA into short RNA sequences called *mRNA (messenger RNA)*. The *mRNA* sequence is then *translated* into amino acid sequences, which then fold up to form proteins.

There is a variety of different factors that influence, whether or not, a gene is *transcribed* in the first place. If a gene is not transcribed or not translated, it is *non-expressed*. In this case, the protein will never be created.

The cells in a human (or most other multicellular organism for that matter) are highly specialized for certain tasks. There are nerve cells, blood cells, muscle cells, skin cells, liver cells, and numerous other types. However, the underlying code of the cells in form of the DNA is identical throughout the whole organism. The different cell types utilize different sets of proteins for their specific function. In nerve cells for instance, numerous membrane bound receptor proteins (located on the outer surface of the cell) are expressed. Many of these proteins are not found in any other cell types.

Henceforth, we will label a protein as *expressed* or *non-expressed* simply relating to whether a protein is present in a specific cell or not.

The set of proteins that are expressed varies among cell types and tissues. Some proteins are expressed only in one or very few cell types, while others are expressed in a all, or most cell types. The former are called *tissue-specific* proteins, while the latter are named *housekeeping* proteins. Examples for housekeeping proteins are proteins that are vital to the survival of the cell, such as DNA replication proteins or proteins that are active in the core metabolism of the cell.

Given an ordered set of all proteins, we define the protein expression of a given tissue or cell type as a binary vector $\vec{e_t} = (e_{t0}, e_{t1}, \ldots, e_{t(n-1))}$ where $e_{ti} = 1$ *iff* protein p_i is expressed in tissue t and $e_{ti} = 0$ otherwise.

A variety of different data sources for protein expression in human tissue and cell types are freely available. The expression data stems from different experimental methodologies. The protein expression can be measured directly, or indirectly, by measuring the *mRNA* abundance. The *mRNA* abundance can be measured using DNA arrays or by sequencing RNA (*RNAseq*) using next generation sequencing platforms (see Section 2.2 for details). Protein expression can be directly measured by using antibody essays [61]. This method however requires extensive manual annotations, which are not required for neither RNA based expression data from DNA arrays nor *RNAseq*.

A drawback of *mRNA* based expression data is, that *mRNA* abundance is not always highly correlated with protein abundance [31].

1.2.5 Tissue-specific PPIs

A *tissue specific protein-protein interaction network (TSPPI)* is defined as the subgraph of a protein-protein interaction network that contains only proteins that are expressed in a specific tissue [10] [17].

Given the graph $G(V, E)$ of a PPI network and an expression vector $\vec{e_t}$ for a tissue t, we construct the according tissue specific PPI network by creating a subgraph $G_t(V_t, E_t)$ of G, which only contains expressed proteins as nodes. Hence:

$$V_t = \{p \in V | e_{tp} = 1\} \tag{1.1}$$

and

$$E_t = \{(p_i, p_j) \in E | e_{tp_i} = 1 \wedge e_{tp_j} = 1\} \tag{1.2}$$

In order to construct a *TSPPI*, we therefore need a PPI network as a template and expression data for each protein. Since an expression dataset provides protein expression data for multiple tissues, we can construct multiple TSPPIs (one per tissue) for each PPI and expression dataset.

1.3 Complex networks

We will shortly introduce random graphs and *scale-free* graphs in this section, since PPI networks closely resemble such graphs. Note that throughout this work, we will use *graphs* and *networks* as synonyms.

1.3.1 Random graphs

Random graphs were introduced independently by Erdös & Rényi (1959) [23] and by Gilbert (1959) [27]. In Gilbert's model, a random graph is defined as $G(n, p)$, where n is the number of nodes in the graph and p is the probability of any possible edge to exist. Since there are a total of $\frac{n(n-1)}{2}$ possible edges in a graph with n nodes, the random graph $G(n, p)$ has an expected number of

$$\langle m \rangle = \mathbb{E}[m] = p \cdot \frac{n(n-1)}{2} \tag{1.3}$$

edges.

The degree distribution $P(k)$ of a random graph is defined as the probability distribution over the degrees of the graph's nodes. Similarly, the degree distribution $P(k)$ of an actual instance of a graph is given by the fraction of nodes having degree k. The degree distribution of the $G(n, p)$ random graph model is given by the binomial distribution:

$$P(k) = \mathbb{P}[deg(v) = k] = \binom{n-1}{k} p^k (1-p)^{n-1-k} \tag{1.4}$$

for any node v [9].

To fit the degree distribution of an instance of a graph $G(n, m)$ by a binomial degree distribution, the edge probability p is estimated as:

$$p = \frac{2m}{n(n-1)} \tag{1.5}$$

Exponential degree distribution Some graphs follow an exponential degree distribution [26], whose degree distribution is given by the exponential distribution [24], [45]:

$$P(k) = \lambda e^{-\lambda k} \tag{1.6}$$

where λ is estimated by $\hat{\lambda} = \frac{1}{\bar{k}}$ and $\bar{k}$ is the average degree.

1.3.2 Scale-free graphs

A multitude of real world graphs are not random graphs as discussed above. In fact, so-called *scale-free* networks are common. These graphs include the world wide web, science collaboration networks, phone call networks, and others [1].

Scale-free networks are defined by their degree distribution, which in contrast to random networks follows a power-law distribution [4]:

$$P(k) \sim k^{-\alpha} \tag{1.7}$$

If normalized for degrees starting from one, this yields:

$$P(k) = (\alpha - 1)k^{-\alpha} \tag{1.8}$$

To fit a power-law distribution to an observed degree distribution, the α parameter is estimated using a *Maximum Likelihood Estimator*.

As we will see later (Section 3.3), protein-protein interaction (PPI) networks also exhibit a power-law distribution and can thus be classified as complex networks with *scale-free* properties.

The numerous high degree nodes in *scale-free* graphs are called *hubs* [5]. These play a central role in the networks and are inherent to the *scale-free* model.

1.3.3 Graph properties

Besides obvious graph properties, such as the number of nodes, edges and the interaction degrees as well as the degree distribution, we are also going to look at topological properties of graphs. We will introduce those below.

Clustering Coefficient

A *clustering coefficient* measures to which degree the local neighborhood of nodes is inter-connected, thus how much nodes tend to cluster together.

The *local clustering coefficient* measures this in terms of triangles. For each node u, its *local clustering coefficient* is defined as:

$$C_u = \frac{2L_u}{deg(u)(deg(u) - 1)} \tag{1.9}$$

where T_u is the number of triangles at u, i.e., the number of combinations of neighbors v and w of u for which an edge $\{v, w\}$ exists in the graph.

The *global clustering coefficient* can be defined in different ways. One common definition is:

$$C = \frac{number\ of\ closed\ triplets}{number\ of\ triplets} \tag{1.10}$$

where a triplet consists of three nodes that are connected by at least two edges. If all three nodes are connected by a total of three edges, the triplet is called *closed*. Another definition of a *global clustering coefficient* is the average local clustering coefficient over all nodes, hence:

$$C = \frac{\sum_{u \in V} C_u}{|V|} \tag{1.11}$$

Centrality

A centrality measure of a node in a network is a number representing the *importance* of that node in the network. This could imply many things and a variety of different centrality measures have been proposed.

The *degree centrality* of a node is equivalent to its degree in the graph. In this thesis, we analyze the degree distributions and degree centralities of different classes of proteins in PPI networks.

The *betweenness centrality* of a node u is defined relative to the number of shortest paths that pass through this node. More specifically, the betweenness centrality is given by:

$$g(u) = \sum_{s \neq u \neq t} \frac{\sigma_{st}(u)}{\sigma_{st}} \tag{1.12}$$

where σ_{st} is the number of shortest paths from s to t and $\sigma_{st}(u)$ is the number of shortest paths from s to t passing through the node u. In a PPI network, a node with a high betweenness centrality can be interpreted either as a bottleneck in protein signaling pathways or as a protein which is involved in many different pathways. This is why we consider such a protein as important in PPI networks.

The *closeness centrality* of a node is a measure of the distances from this node to all other nodes. There are relatively short paths to all other nodes from a node with a high closeness centrality. We consider this measure to be less meaningful for a protein in a PPI graph compared to the betweenness centrality. We deem the number of pathways in which a protein is involved in (i.e., the betweenness centrality) more important than the distances to other proteins.

The *eigenvector centrality* is yet another centrality measure. For this method, the centralities of the nodes are the values of the eigenvector for the biggest eigenvalue of the adjacency matrix of the graph. We consider the *eigenvector centrality* to be less interpretable in PPI networks.

In our analysis of PPI and TSPPI networks, we therefore focus mainly on the *degree* and *betweenness centralities*.

Chapter 2

Related work and data sources

2.1 Protein-protein interaction networks

In this Section we will shortly describe the different *protein-protein interaction (PPI)* networks that are used for our analysis and the experimental methodologies used to obtain them.

2.1.1 Yeast two-hybrid

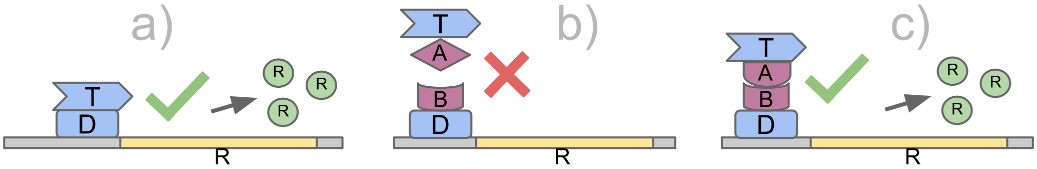


Figure 2.1: The *Yeast two-hybrid* method: *a)* original state (no *Y2H* screening), *b)* A and B do not interact in *Y2H* screening, *c)* A and B do interact in *Y2H* screening.

Yeast two-hybrid (Y2H) screening is a method used for finding if two proteins interact within a yeast cell. Initially, an artificial, circular DNA segment called a *plasmid* is inserted into a yeast cell. The plasmid contains multiple genes, including the two query genes A and B (also called the *prey* and *bait*), which are tested for interaction. We depict the *Y2H* method in Figure 2.1. Additionally, there is a reporter gene R and a combined transcription factor $T - D$. If this factor is available in this form, the D part will bind to the DNA close to the reporter gene R and the T part will promote transcription (see part *(a)* in Figure 2.1). Thus, the reporter gene R will be expressed and can be measured. However, if T and D are not connected, then R will not be expressed. In the artificial plasmid the gene for

$T - D$ is split and combined with A and B as: $T - A$ and $B - D$. Now only if the proteins A and B interact, the complex $T - A - B - D$ will form and the reporter gene R can be expressed *(c)*. If A and B do not interact, then R is not expressed *(b)*. The reporter gene R is selected in a way that facilitates measuring of R [40].

The group at the *Center for Cancer Systems Biology (CCSB)* of the *Dana-Farber Cancer Institute* has created a PPI network by using high-throughput Y2H screening [53] [63] [64]. For the newest release, they have run Y2H experiments on $13,000$ genes testing for all possible interactions. Of this $13,000 \times 13,000$ matrix, they found a total of approximately $14,000$ interactions. This PPI network is referred to as the *Human Interactome 2012*. From here on, we will refer to this network as the *CCSB HI-2012* or simply as *HI-2012*.

2.1.2 Protein complexes

A *protein complex* consists of multiple proteins that bind together into a single entity. These commonly occur in the cell and perform various tasks (see Section 1.2.2). The proteins in such a complex are represented as a densely connected clique in the PPI graph.

Havugimana *et al.* (2012) [32] extracted and purified protein complexes from human cells. Then, they fragmented the protein complexes and used mass spectrometry to identify the proteins in each complex. The resulting PPI network consists of many densely connected clusters with few links in between them.

Throughout this work, we will refer to this network by the name of the first author or as the *protein complexes* network.

2.1.3 Literature curated PPIs

There are various research groups that curate databases of protein-protein interactions published in the peer-reviewed literature. The *IMEx* consortium organizes these groups by providing a set of standards for curation, annotation, and publication of protein-protein interactions [47]. The PPI databases that are part of the *IMEx* consortium are *DIP* [55], *IntACT* [33], *MINT* [65], *I2D* [12], *MatrixDB* [16], *MBInfo* [46], *UniProt* [20], *MPIDB* [28], and *InnateDB* [42].

All these databases can be easily accessed through *PSICQUIC (Proteomics Standards Initiative common query interface)* interfaces [3]. In this thesis, we combine all those networks into one PPI network. We will henceforth refer to this network as *IMEx PSICQUIC* or *IMEx*.

2.1.4 Composite PPI networks

Another approach that is taken, is to combine the protein-protein interactions from multiple sources and experiments into one large PPI network and score the edges using some quality score. This approach is taken for instance by the *STRING* [25] and *HIPPIE* [56] PPI networks. Here, we use the *STRING* PPI network, since this networks is more established in the scientific community.

Bossi and Lehner (2009) created their own composite network by combining several sources into one joint PPI network [10]. Since we are re-evaluating their results in this thesis (see Section 5.2.1), we will also be using their PPI network throughout our analyses. We will refer to this network by the name of the first author: *Bossi*.

2.2 Protein expression

There are two fundamentally different approaches to quantify protein expression. One approach measures the messenger RNA (mRNA) transcribed from the genes. The alternative method is to measure protein abundance directly.

More high-throughput studies measuring mRNA compared to measuring protein abundance have been performed, although mRNA abundance does not always correlate highly with protein abundance [31].

We will now shortly describe the different expression data sources that we use in this thesis.

2.2.1 DNA microarray chips

DNA microarray chips are used to measure mRNA abundance. To do so, the mRNA of the cell is first transcribed into complementary DNA fragments. The microarray chip's surface has oligonucleotide DNA probes attached to it, to which the sample DNA fragments bind. Each gene has an associated region on the chip. The mRNA abundance of different genes is then measured by the quantity of DNA fragments binding to each part of the chip.

Su *et al.* (2004) [59] used DNA microarrays for a high-throughput study of the expression of human genes in over 80 different tissues and cell-types. We will refer to this expression dataset by its name, *Gene Atlas*, from here on.

2.2.2 RNA sequencing

Another method to measure mRNA abundance is to use next-generation sequencing (NGS) to directly sequence the mRNA. The sequenced reads are mapped to the genome and the number of reads mapping to each protein-coding gene are counted. This count is then normalized into a metric called *RPKM* (reads per kilobase per million).

In this thesis, we use the data from two studies which have sequenced mRNA and provided normalized expression. Krupp *et al.* (2012) [37] created the *RNAseq Atlas* which provides normalized mRNA expression for 11 different tissues. The RNAseq data for their database originates from a study by Castle *et al.* (2010) [14]. The second source of RNAseq data is the *Illumina Body Map 2.0* provided by the *European Bioinformatics Institute (EBI)* on *ArrayExpress* [54], which is a public database of experimental results in functional genomics. The *Body Map* provides normalized mRNA expression data for a total of 16 human tissues.

2.2.3 Antibody annotation

Uhlen *et al.* (2005, 2010) [61] [62] performed a high-throughput antibody essay for over 15,000 human proteins and over 80 different cell- and tissue-types. For each protein, one or more antibodies are used to determine whether the protein is present in a cell or not. For each cell- and tissue-type and antibody essay, microscopy images are evaluated and annotated by domain experts. The observed protein expression is categorized into four categories: *"None", "Low", "Medium"*, and *"High"* and then further classified into four levels of reliability based on antibody specificity, conflicting results and previously published results in the peer-reviewed literature. The resulting database is called the *Human Protein Atlas (HPA)* and freely available via the WWW.

We use this expression database in our analysis, but only use those expression values which are annotated by either "Medium" or "High" reliability. We will refer to this dataset as *HPA*. A large fraction of expression values is still annotated as unreliable, hence the *HPA* expression dataset is considerably smaller than the mRNA based datasets (see also Section 3.2). We will therefore also conduct analyses using the whole *Human Protein Atlas* database (without filtering for reliability). We denote this dataset as *HPA All* from here on.

2.3 Tissue-specific PPI networks

Since the release of the Gene Atlas data by Su *et al.* (2004) [59] a number of different research results on the analysis of tissue-specific interaction networks have been published. In the following, we will summarize those that are most relevant to our research.

Bossi and Lehner (2009) [10] created a tissue-specific PPI network, by combining a number of different PPI networks and annotating the proteins with expression data from the *Gene Atlas*. They then analyzed their tissue-specific PPI network, with particular emphasis on the network properties of tissue-specific as well as housekeeping proteins. We reproduce a substantial fraction of their analyses and show that their results remain valid only for some, not all, of the PPI and expression datasets (see Section 5.2.1).

Emig and Albrecht (2011) [21] and Emig *et al.* (2011) [22] re-evaluated some results from Bossi and Lehner [10] using RNAseq instead of *Gene Atlas* expression data. They observed that RNAseq expression data shows that many more proteins are universally expressed than when considering expression data from the *Gene Atlas*. They also show that, some results from Bossi and Lehner do not hold when considering RNAseq data. We will re-evaluate the results from these studies systematically for multiple PPI networks and multiple expression datasets.

Lopes *et al.* (2011) analyzed the characteristics of PPI networks [41]. They calculated various network statistics such as the average betweenness, the clustering coefficient, diameter, average shortest paths, and others and concluded that all PPI networks they analyzed are topologically similar. They further constructed tissue-specific PPIs and noted that these are considerably smaller than the whole PPI networks ($1\% - 25\%$). They used the tissue-specific PPI networks to show that proteins related to viral infections and responses show better *functional enrichment* in the tissue-specific PPIs than they do in the whole PPIs.

Functional enrichment analysis is a method to score the similarity of a set of genes. To that end, genes are annotated with terms describing their biological function. One possible annotation scheme are *Gene-Ontology* terms, which we will describe in more detail in Section 3.6. The similarity of genes is then determined from the similarity of the terms, which the genes are annotated by. The *functional enrichment* score for a set of genes reflects the similarity of the query genes in contrast to a background population of genes. Refer to Section 3.6 for more details.

Lin *et al.* (2009) [39] constructed tissue-specific PPIs using the *Human Protein Reference Database (HPRD)* for protein interactions. Their analysis demonstrates that, housekeeping proteins exhibit higher interaction degrees and higher betweenness centralities than randomly selected nodes inside the tissue-specific networks. However, most of their results are not statistically significant. The higher interaction degree of housekeeping proteins was also shown by Bossi and Lehner (see above).

Barshir *et al.* (2013) published the *TissueNet* database of tissue-specific protein-protein interactions in humans [6], which integrates a collection of PPI networks with three expression datasets. These are the same expression datasets that are used within this work: the *Gene Atlas* [59], the *Human Protein Atlas* [61] and the *Illumina Body Map* RNAseq data. The *TissueNet* database is provided through a web-interface, which can be queried for protein identifiers. It will then return all interaction partners with annotations for each tissue. This local neighborhood is rendered into a visual graph representation. However, Bashir *et al.* do not provide any detailed analyses of graph properties of their integrated network.

2.4 Graph analysis and clustering

The *NetworKit* toolkit by Staudt *et al.* (2014) [58] implements multiple graph analysis and clustering algorithms for parallel shared-memory systems. They provide novel parallelizations approaches of established clustering algorithms and use *OpenMP* thread-based parallelism for the implementation.

OpenMP (Open Multi-Processing) is a standardized interface for shared-memory multiprocessing, which provides compiler directives for parallelization and synchronization of code sections and loops. *OpenMP* uses thread based parallelism and supports the languages *C/C++* and *FORTRAN*.

For graph analysis, the *NetworKit* implements algorithms to determine the exact and approximate local and global clustering coefficients, and a collection of different centrality methods, including the betweenness centrality. The betweenness centrality is computed using Brandes' algorithm [11], which is a sequential method based on breadth-first search (BFS) and runs in $O(m \cdot n)$ in unweighted graphs where n is the number of nodes and m is the number of edges.

Furthermore, the authors of the *NetworKit* toolkit provide parallel clustering algorithms, among which are the *parallel label propagation (PLP)* algorithm based on the label propagation method by Raghavan *et al.* (2007) [49] and the *parallel Louvain method (PLM)* based on the algorithm by Blondel *et al.* (2008) [8].

We will make use of, and provide modified forms of these algorithms, for our analysis of tissue-specific protein-protein interaction networks.

Chapter 3

Methods and Datasets

3.1 Data pipeline

To ensure easy reproducibility of all results, we implemented an automated analysis pipeline. The input to this pipeline are the data files in the form they were downloaded from the various sources. The data pipeline then processes the data by transforming itto a common format and then executes all analyses on all datasets. In this section, we will describe the different processing steps, i.e., the stages of the pipeline.

The protein-protein interaction networks and the expression data sets are published in different formats. The first steps of the automated analysis are therefore to import all the data into a unified database and transform it into a common, shared representation.

The stages for importing and processing a protein-protein interaction network are roughly the following:

1. Import raw file into the SQL database
2. (optional) Filter edges by their reliability
3. Map gene identifiers into a common gene identifier system
4. Normalize tables and graphs into the common representation

Analogously, the tissue expression datasets are imported and processed into a common format using the following steps:

1. Import the raw file into the SQL database
2. Filter unreliable or empty data points
3. Map gene identifiers into a common gene identifier system

4. Normalize the table into the common format
5. Remove/merge duplicates
6. Classifying expression values into *expressed* or non-expressed

In the following we will describe some of these steps in more detail.

3.1.1 Common data format

In order to jointly analyze and combine data sets, we first define a common representation for PPI networks and expression data sets.

Common identifiers We use *HGNC (HUGO Gene Nomenclature Committee)* [29] identifiers as common gene identifiers. Some data sets use protein identifiers (STRING) or transcript identifiers (RNAseq data), which are more specific than gene identifiers, since one gene can result in different transcript or protein variants due to splicing. However, most of the data sets use gene identifiers, which can not be mapped back to the more specific protein or transcript identifiers. Therefore, we map all identifiers to the more general gene identifiers, at slight loss of specificity of the data.

PPI networks We represent protein-protein interaction networks as edge lists of gene identifiers. An edge in a PPI network represents an interaction between two proteins. Therefore, such a network is an undirected graph. The format of the *STRING* network saves each edge twice, once for each orientation. For an edge $\{u, v\}$, we only save one orientation: $(\min(u, v), \max(u, v))$, to avoid redundancy.

Expression data Expression datasets contain gene expression levels for genes in different tissues or cell types. One way to represent this data would be as a two-dimensional matrix, where one dimension is given by the genes and the other by the tissues or cell-types. However, because we are using a relational database in our implementation, we model expression data as a table with one column for each: the gene identifier, the tissue, and the expression value.

3.1.2 Gene identifier mapping

The different datasets use distinct gene identification systems. Table 3.1 shows the identifier systems used by the PPI networks and the expression datasets.

PPI	**Identfier**
Bossi & Lehner	Ensembl Gene
CCSB HI-2012	HGNC
Havugimana *et al.*	Uniprot
PSICQUIC IMEx	Uniprot
STRING	Ensembl Protein

Expression dataset	**Identfier**
Illumina BodyMap	Ensembl Gene
GeneAtlas	Chip annotation IDs
Human Protein Atlas	Ensembl Gene
RNAseq Atlas	Entrez & HGNC

Table 3.1: Different gene identifier systems used by the PPIs (top table) and expression data sets (bottom table).

We choose *HGNC (HUGO Gene Nomenclature Committee)* [29] gene identifiers to consistently name genes, since this identifier system is well curated and well established. This identifier system is also referred to as the *official gene names* [35].

For mapping most identifiers to *HGNC* identifiers, we download, import, and merge two identifier mapping tables: one from *BioMart* [36] and the other from the HGNC website at `genenames.org`.

The merged mapping table is then used to map the identifiers in a dataset from its identifier system to HGNC gene identifiers. Since the mapping tables do not guarantee a one-to-one mapping, all duplicates need to be merged.

3.1.3 Merging duplicates

As a result of mapping identifiers from one identifier system to the other, multiple distinct identifiers can be mapped to a single identifier. This is typically the case when protein or transcript identifiers are mapped to gene identifiers, since one gene can give rise to multiple different proteins and transcripts due to alternative splicing. It is thus not only a technical issue but also biologically relevant.

For PPI networks, this is equivalent to merging multiple distinct vertices into one. Thus we simply remove duplicate edges after mapping identifiers.

In an expression dataset such duplicates can result in different expression values associated with a single gene in a single tissue. We merge duplicate expression values by only keeping the maximum expression value. We argue that, taking the maximum value is a good choice, since if multiple variants of a gene (e.g., in form of different splicing variants) are expressed, that gene is expressed at least as much as any of it's variants.

3.1.4 Classifying expression values

Once all data is imported and brought into a common representation, the expression data is classified into *expressed* and *non-expressed*.

The *Human Protein Atlas* supplies discrete expression levels. For *Staining*, these are “Negative”, “Weak”, “Moderate”, and “Strong”, while for *APE* combined scoring the provided levels are: “None”, “Low”, “Medium”, and “High”. We classify a gene to be expressed whenever it is not “Negative” or “None”.

For the *Gene Atlas* expression dataset published by Su *et al.* [59], we use a fixed threshold of ≥ 100 to classify a gene as expressed. This threshold value is also used by the authors of *TissueNet* [6] and others [66].

For both RNAseq datasets (Illumina Body Map 2.0 and RNAseq Atlas) we use a fixed threshold of ≥ 1.0 RPKM for classifying a gene as being expressed [6].

Dataset	**Classification threshold**
Gene Atlas	$>= 100$
Human Protein Atlas	not "None" nor "Negative"
RNAseq Atlas	≥ 1.0
Illumina Body Map 2.0	≥ 1.0

Table 3.2: Our classification thresholds for the different expression data sources

In Table 3.2 we show the classification criteria for the different expression datasets, and Figure 3.1 illustrates the cumulative distribution of expression values for each expression data set. The plots also show how many gene-tissue combinations are classified as non-expressed and expressed for the given thresholds (blue dotted lines).

The cumulative frequency at the classification threshold value is equal to the fraction of gene-tissue combinations that are classified as *non-expressed.* This number is printed in the plots of Figure 3.1 on the left side of the dotted, horizontal line. These numbers show, that the thresholds do not necessarily result in the same fractions of genes classified into expressed and non-expressed. Most notable is the percentage of gene-tissue combinations classified as non-expressed within the *Gene Atlas* data.

3.2 Expression datasets

In this section we will calculate some simple statistics about the different expression datasets, including the number of genes and tissues and their expression pro-

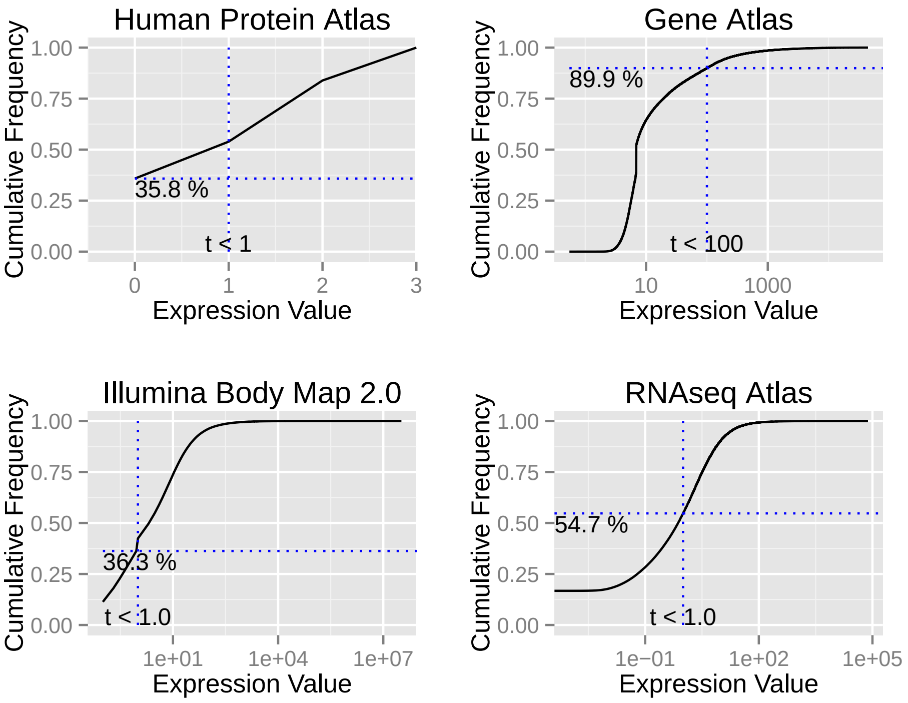


Figure 3.1: This figure shows the cumulative distribution of expression values for the four different expression data sets. The blue dotted lines show the cutoffs, including a percentage of how many gene-tissue combinations are classified as non-expressed. The plot for the Human Protein Atlas uses a simple mapping of “Negative”, “Weak”, “Moderate”, and “Strong” to the numerical values $\{0, 1, 2, 3\}$.

files. Furthermore, we will explain the classification of proteins into housekeeping and tissue-specific genes.

3.2.1 Basic properties

In Table 3.3, we give the sizes of the expression data sets in terms of the number of genes and tissues or cell-types covered. The *Gene Atlas* and the *Human Protein Atlas* have data from over 80 different cell-types. The RNAseq datasets are more restrictive in the number of tissues covered with 11 in the *RNAseq Atlas* and 16 tissues by the *Illumina Body Map*. The *GeneAtlas* and *RNAseq* Atlas cover most genes, while the *Human Protein Atlas* and the *Body Map* cover only around 75$\%$ of genes. The restricted *Human Protein Atlas* data, i.e., the subset of the *HPA* which is rated as being reliable, is much smaller, with only 3836 genes.

Dataset	Number of Genes	Number of Tissues
Gene Atlas	18444	84
Human Protein Atlas	3836	83
Human Protein Atlas all	15061	83
RNAseq Atlas	21399	11
Illumina Body Map 2.0	14404	16

Table 3.3: Sizes of the expression data sets after import and identifier mapping.

3.2.2 Tissue expression

In Figure 3.2, we show the tissue expression for each gene of each expression dataset. The tissue expression is defined as:

$$tissue\_expression(gene) = \frac{number\ of\ tissues\ where\ gene\ is\ expressed}{total\ number\ of\ tissues}$$

The genes in the figures are sorted by their tissue expression values, resulting in a monotonic graph. We observe that, especially the *Gene Atlas*, has a more narrow distribution with many not genes being expressed at all. Note that, the data for this plot is already using the relaxed threshold of ≥ 100 for classifying expression in the *Gene Atlas*. The *Human Protein Atlas* shows more evenly distributed tissue expression values for all genes. These differences in tissue expression have an influence when classifying genes into tissue specific and housekeeping proteins.

3.2.3 Tissue-specific and housekeeping proteins

Tissue-specific (TS) proteins are proteins that are expressed in only a few tissues, while *housekeeping (HK)* proteins are proteins that are at the core of the cell machinery and thus are expressed in all or almost all of the cells. We later analyze the properties of these two classes of proteins inside of protein-protein interaction networks. To that end, we must first define a way to identify genes as *TS* or *HK*.

The classifications of tissue-specific proteins by Chang *et al.* [15] and Greco *et al.* [30] are based on the underlying expression values from microarray analyses. They define a tissue-selectivity score based on the distribution of expression values of a gene across the different tissues. However, we want to find a simple method that we can use on the already classified gene expression (i.e., the binary gene expression values) so that it can be used universally across all our expression datasets (note that, the *HPA* does not supply numeric expression values at all). Bossi and Lehner [10] define tissue-specific and housekeeping genes in terms of fixed ranges of tissues. For example, they classify genes expressed in 1 up to

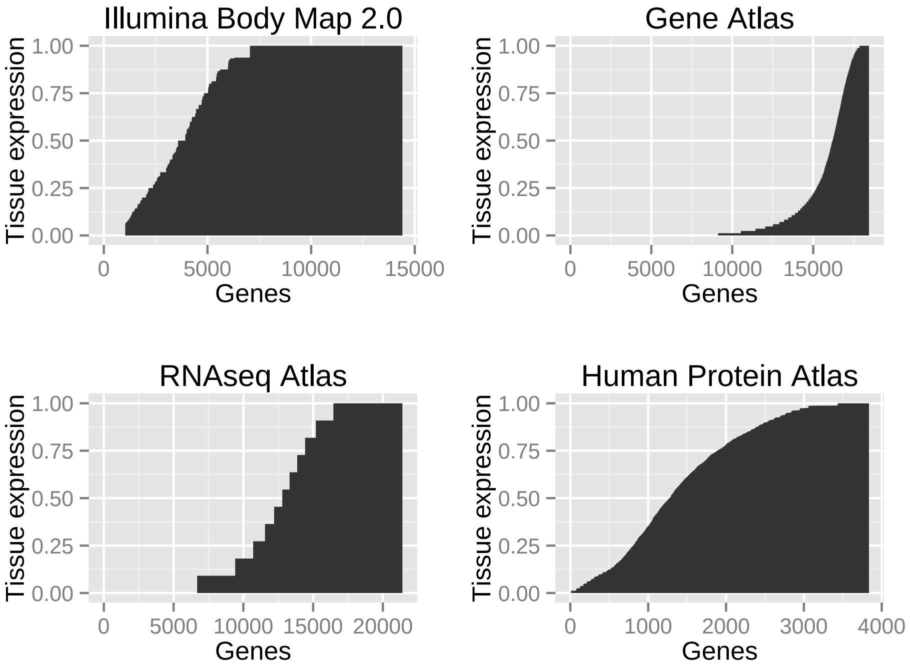


Figure 3.2: This figure shows the tissue expression distribution for all expression datasets, i.e., in how many tissues each gene is expressed. The tissue expression is given by the number of tissues a gene is expressed in divided by the number of total tissues. The genes are displayed sorted by their respective tissue expression levels.

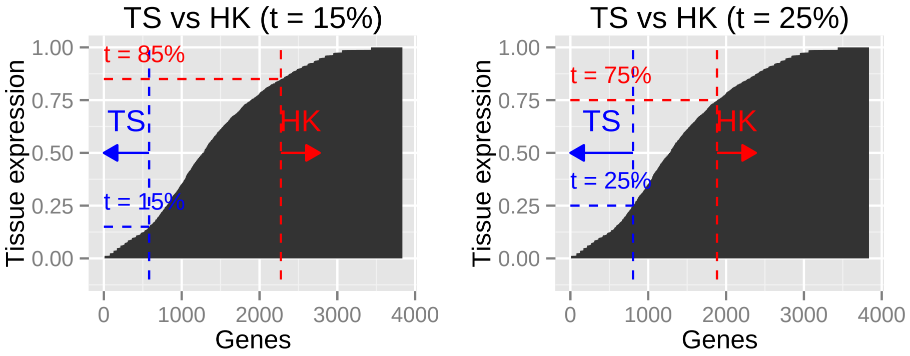


Figure 3.3: This figure shows how we classify proteins into *tissue-specific* and *housekeeping* based on their tissue expression. Here we show the classification for the *Human Protein Atlas* data for threshold values $t \in \{0.15, 0.25\}$.

10 out of 79 tissues as tissue-specific and similarly $71 - 79/79$ as housekeeping

genes.

We take a similar approach by using thresholding. For a threshold t (e.g., 10%) we define all proteins as *tissue-specific* if they are expressed in at most t percent of all tissues of the expression dataset. Likewise, we define all proteins that are expressed in at least $(100 - t)$ percent oh tissues as *housekeeping*. Figure 3.3 illustrates this thresholding concept on the *Human Protein Atlas*.

By using the same threshold t for classifying both *TS* and *HK*, it becomes more feasible to explore the results of our later analyses (for example in Section 5.2.1) for different *TS* and *HK* classifications (i.e., by varying the single parameter t).

The current definition of tissue-specific proteins is problematic, since proteins which are never expressed in any tissues are still classified as tissue-specific. Especially the *Gene Atlas* and *RNAseq Atlas* datasets contain many such proteins. To solve this problem, we only consider proteins that are expressed in at least one tissue for the classification. We depict an example of a classification for all datasets and a classification threshold of $t = 15\%$ in Figure 3.4. This shows that the number of genes classified into either group heavily depends on the dataset used.

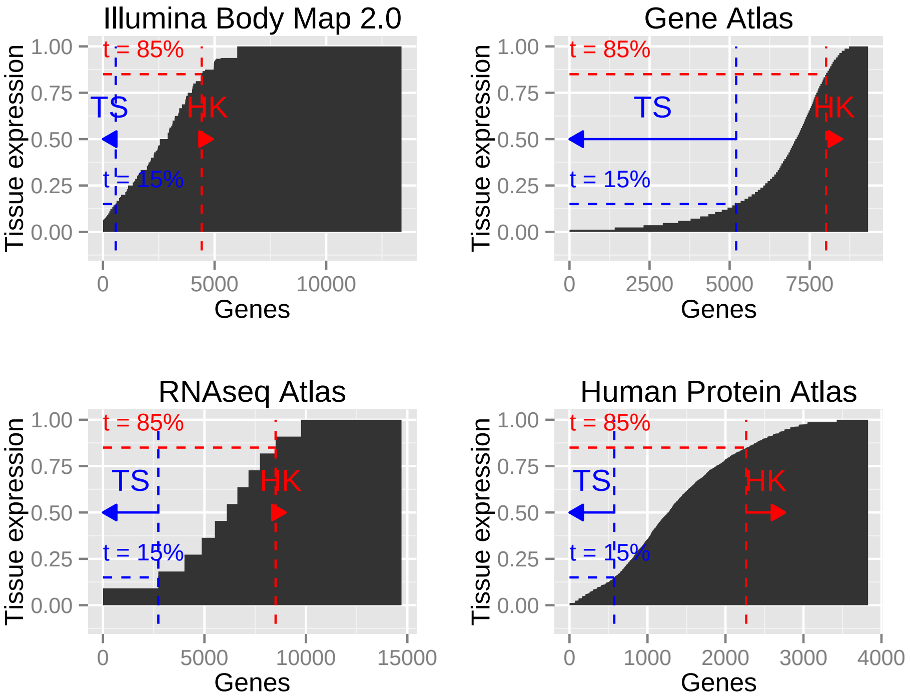


Figure 3.4: Here we show the classification into *tissue-specific* and *housekeeping* for all expression datasets and a classification threshold of $t = 15\%$.

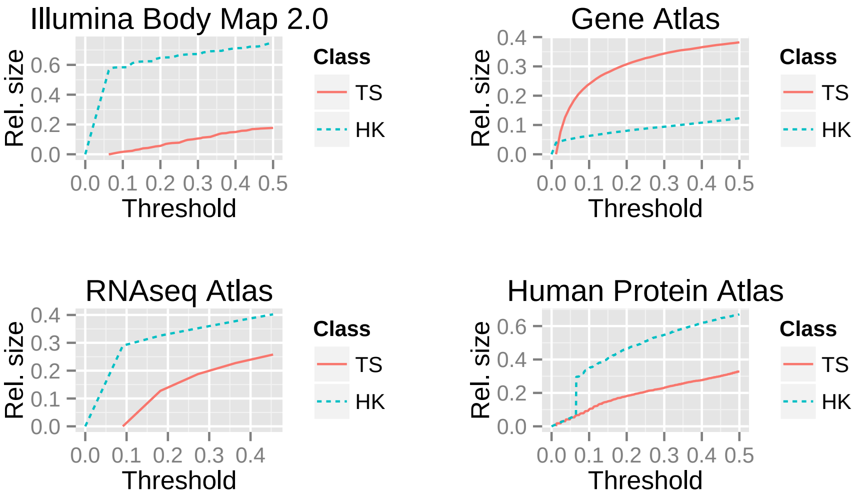


Figure 3.5: The relative sizes of the classes TS and HK are given for various classification thresholds. The relative size is defined as the number of genes in the class divided by the number of total genes in the expression dataset.

To further illustrate this point, Figure 3.5 shows the relative number of tissue-specific and housekeeping proteins for various settings of the threshold t. We vary the threshold in the range $[0, 0.5]$, since any higher threshold would classify proteins into both classes TS and HK. As Figure 3.5 shows, the housekeeping class is twice as large as the size of the tissue-specific class for three out of the four expression datasets. For the *Gene Atlas* however, approximately 3 times as many proteins are classified as tissue-specific than as housekeeping proteins.

3.2.4 Expression data “core”

The tissue-specific expression datasets can be represented as a 2D matrix, where the rows represent the proteins, and the columns represent the tissues. Each value in the matrix is an expression value. If the expression data is classified into expressed and non-expressed, this matrix is a binary matrix.

Some of the expression datasets have missing values for some combinations of proteins and tissues. The *Human Protein Atlas* has most missing values, due to missing experiments. For some of the tissues less than half of the proteins have been measured for expression in the current version of *HPA*. There exist also proteins that have been measured in but a few of the tissues.

When generating tissue-specific PPI subnetworks, missing data poses a problem: how do we handle the nodes of the global PPI for which there is no data available? Do we assume that it is expressed or not-expressed, and based on that, will the node be part of the subnetwork or not?

In order not to make any assumption, we only use those proteins and tissues that, when combined, have no missing data. We define such a combination of proteins and tissues as the *core* of the expression data set. In terms of a 2D binary matrix (where 0 corresponds to missing data and 1 to data being available), we are looking for a combination of rows and columns such that the sub-matrix given by these rows and columns only contains 1s.

Ideally, we would like to find a combination of proteins and tissues that maximize the data points covered. If n is the number of rows of a *core* and m is the number of columns of a *core*, we want to maximize $n \cdot m$, i.e., find a maximum area *core* of the expression data set.

We are not aware of any algorithm solving this problem. Moreover, an exhaustive search for an optimal solution requires all possible choices of rows and columns to be checked. This has exponential complexity: $O(2^n \cdot 2^m)$.

We therefore implement a greedy algorithm to approximately solve this problem. For a subset of tissues, we keep only those genes that are expressed in all those tissues. To select a good subset of tissues, we first define the gene coverage of a tissue as the fraction of genes that have an expression value in that tissue. We then rank tissues according to their gene coverage and iterate over them in this order and add one tissue at a time to the subset. For each tissue that is added, we count the genes that have expression values in all of those tissues. We select the subset of tissues that yield the maximum number of values covered, i.e.,:

$$\max((number\ of\ tissues) \times (number\ of\ proteins)) \tag{3.1}$$

3.3 Properties of PPI networks

PPI	n	m	$\bar{k}$	$\max(k)$	Connected Components
Bossi	9974	77197	15.48	298	79
STRING	15867	304524	38.38	2065	67
IMEx	10937	58011	10.52	714	77
Havugimana	2989	13885	9.29	185	65
HI-2012	4298	13943	6.37	313	134

Table 3.4: Basic properties of the five PPI networks used in our analysis. $\bar{k}$ is the average node degree and $\max(k)$ is the maximum degree in the graph.

3.3.1 Network sizes

We consider five different PPI networks in our analysis pipeline (see Section 2.1.4). The various PPI networks differ in size and their network properties (see Table 3.4). The composite *STRING* PPI network is by far the largest network with over 300 thousand interactions between a total of 15867 unique proteins. The two PPI networks resulting from single experiments (*CCSB*'s *Human Interactome 2012* and the protein complexes network from Havugimana *et al.*) yield the smallest networks. Both networks consist of approximately 13900 interactions. However, the interactions are restricted to 2989 unique proteins in the protein complexes network, while the *HI-2012* network reports these interactions between a total of 4298 unique proteins. The reason for this difference is apparent when the different experimental methodologies for finding interactions is considered. The protein complexes network consists of experimentally found protein complexes, where all proteins in these complexes are connected to all other proteins. This results in fewer proteins being more densely connected. The number of interactions in the composite networks by Bossi and Lehner and by the *IMEx* consortium are approximately 58,000 and 77,000 respectively between close to 10,000 unique proteins.

None of the PPI networks are connected, moreover, each network consists of over 60 connected components that do not interact with each other.

3.3.2 Degree distribution

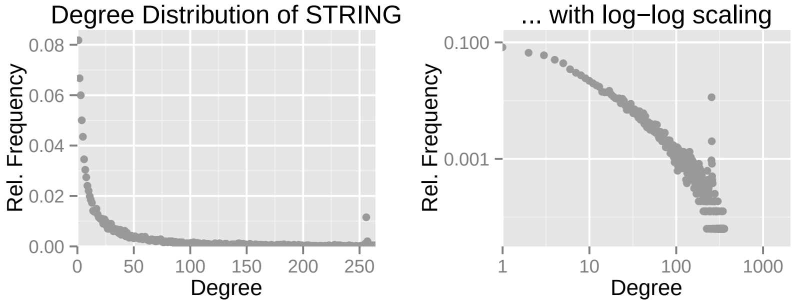


Figure 3.6: The degree distribution of the *STRING* PPI network on a linear scale (left) and a log-log scale (right).

Next, we will take a look at the degree distributions of the PPI graphs. The degree distribution is the observed distribution of vertex degrees in the graph (see Section 1.3). Figure 3.6 shows the degree distribution of the *STRING* network. At first glance, the degree distribution appears to follow an exponential distribution.

However, an exponential degree distribution would not allow for as many high degree nodes as we observe for this and other PPI networks.

Subsequently, we fit three graph models to the observed degree distributions of all PPI networks using Maximum Likelihood Estimation. The Erdös-Rényi random graph model has a binomial degree distribution (see Section 1.3), this model does not fit the observed degree distributions. Additionally, we fit an exponential distribution and a power-law distribution to the degree distribution. Since the power law becomes most prevalent in the "heavy tail" of degree distributions, Clauset *et al.* [18] introduced a method that estimates a point x_{min} in the distribution which acts as a cutoff value. This cutoff is found by optimizing the Kolmogorov-Smirnov goodness-of-fit statistic. The power-law is then fitted only to values $\geq x_{min}$. Figure 3.7 shows the observed degree distribution and the fitted distributions for all PPI networks.

Model	Statistic	Bossi	STRING	IMEx	Havu.	HI-2012
Binomial	χ^2	1.6e+08	Inf	2.1e+06	2.1e+06	4e+04
	p	0	0	0	0	0
Exponential	χ^2	1.4e+04	1.3e+04	1.2e+04	2.7e+03	5.8e+03
	p	0	0	0	0	0
Power-law	χ^2	4.2e+02	2.6e+03	8.8e+02	1.7e+02	6e+02
	p	6.8e-87	0	6.8e-186	1.4e-33	1.7e-126
Power-law	χ^2	1.8e+02	6.2	69	4.6	64
(tail)	p	4.7e-33	**0.72**	2.2e-11	**0.87**	8e-11

Table 3.5: Results of the χ^2 test for goodness of fit for the PPIs degree distributions and different models.

Furthermore, we test how well the different models fit the degree distribution using Pearson's χ^2 *goodness-of-fit* test [48]. The result of this test is shown in Table 3.5. None of the degree distributions fits a binomial (Erdös-Rényi random graph) or exponential degree distribution ($p = 0$ in all cases). However, even the power-law does not fit the degree distributions of any of the networks ($p \leq 1.4 \cdot 10^{-33}$ in all cases). Only by maximizing the fit using the method from Clauset *et al.* [18], we observe two significant fits to the estimated power-law tail distribution. The *STRING* PPI networks and the protein complexes network from Havugimana *et al.* are the only two PPI networks that fit the scale-free model. However, even though none of the networks fit the full power-law distribution, the χ^2 statistic is smaller (meaning better fit) for this distribution than for either the binomial or the exponential distribution models.

We conclude from these results, that none of the PPI networks perfectly resemble the power-law distributions of *scale-free* networks. Still, these networks

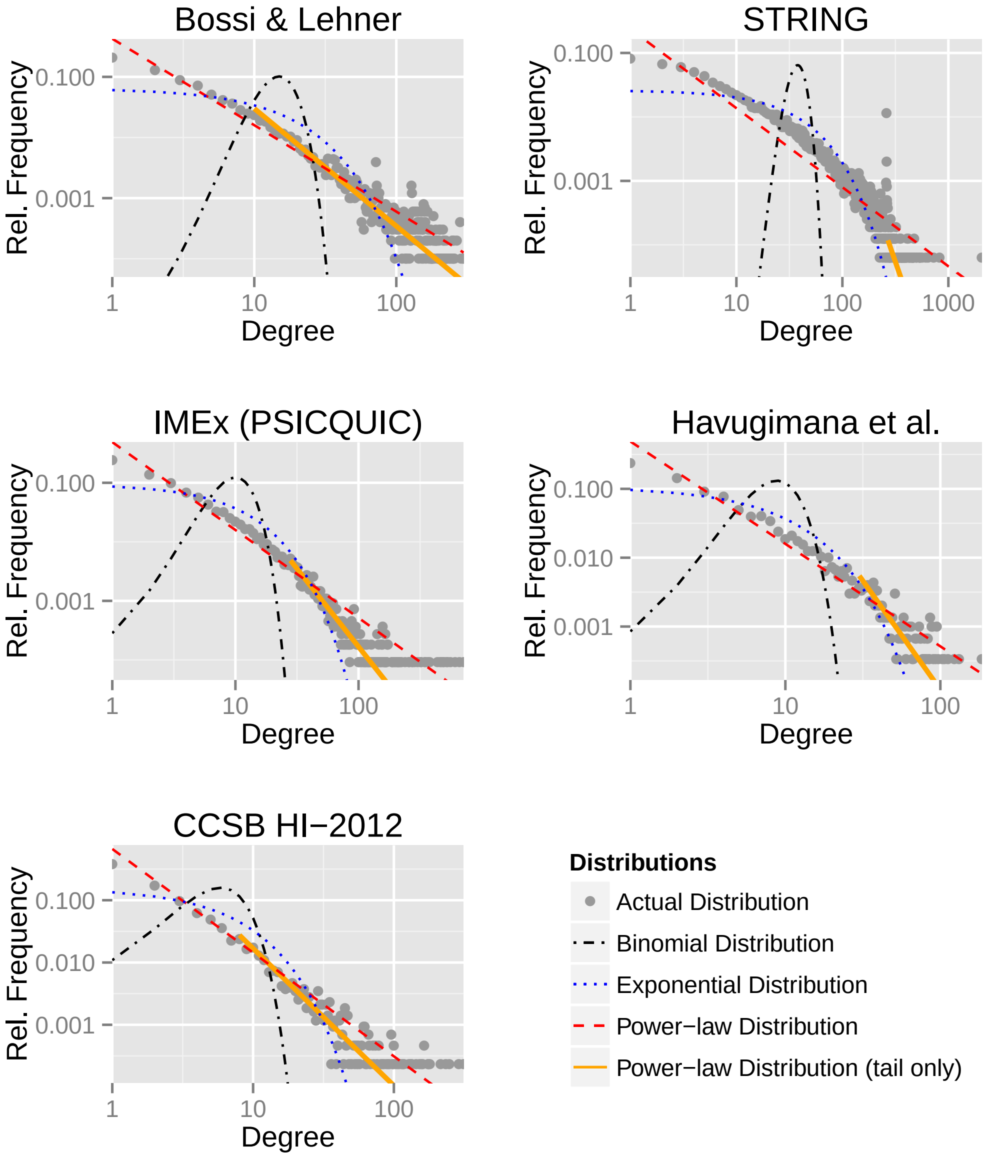


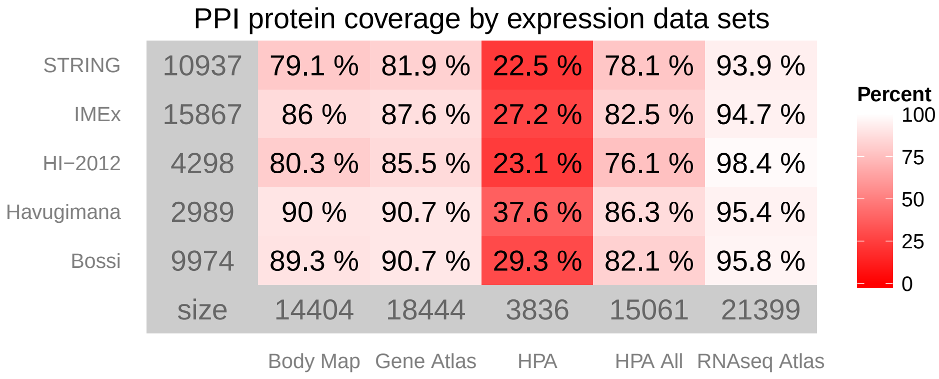
Figure 3.7: This figure shows the degree distributions of the PPI networks on log-log plots. The four different models are fitted to the data and plotted as different lines (see legend of this figure).

all exhibit a “heavy tail” in their degree distribution and thus have a large number of *hubs*, which is a prevalent property of *scale-free* and *complex* networks. We will thus consider the PPI networks to be *scale-free*-like.

3.4 Tissue specific PPIs

A tissue specific PPI is a sub-graph of a PPI network, in which a vertex is only present if it is expressed in the tissue. Similarly, an edge is only present if both of the interacting proteins are expressed in the tissue. We construct such tissue specific PPIs by combining the graph with expression data from an expression dataset. Each protein/node is labeled with its tissue expression vector, which is a binary vector indicating in which tissues the protein is expressed.

3.4.1 Expression coverage

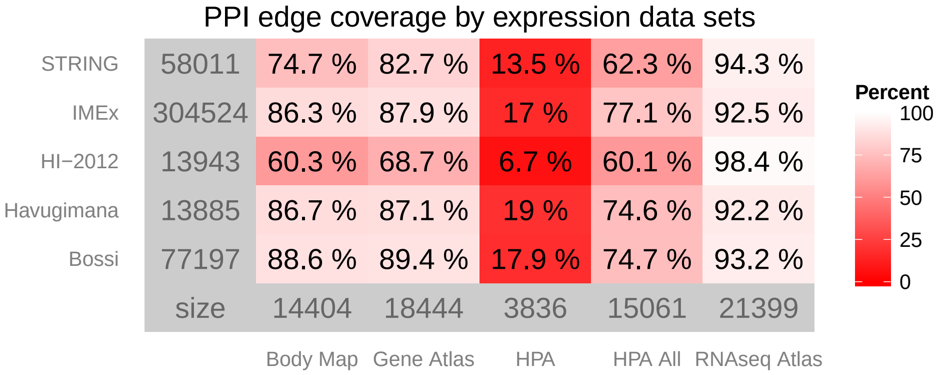

PPI protein coverage by expression data sets

		Body Map	Gene Atlas	HPA	HPA All	RNAseq Atlas
STRING	10937	79.1 %	81.9 %	22.5 %	78.1 %	93.9 %
IMEx	15867	86 %	87.6 %	27.2 %	82.5 %	94.7 %
HI−2012	4298	80.3 %	85.5 %	23.1 %	76.1 %	98.4 %
Havugimana	2989	90 %	90.7 %	37.6 %	86.3 %	95.4 %
Bossi	9974	89.3 %	90.7 %	29.3 %	82.1 %	95.8 %
	size	14404	18444	3836	15061	21399

Figure 3.8: The percent of nodes (i.e., proteins) in the PPI networks for which the expression datasets contain expression values.

When we combine the PPI networks with the expression datasets, not every protein in the PPI network is annotated with an expression value inside the expression dataset. This is because the expression datasets do not necessarily provide expression values for all possible proteins. For some of our later applications and analyses, we need every node in the network to be annotated. Proteins that have no annotation in the expression datasets are simply removed from the network. The tissue-specific subnetworks are therefore smaller than the original full PPI networks.

We are thus interested, how much smaller the networks will become. The table in Figure 3.8 shows the percentage of proteins in each PPI network that have an expression value associated with them. All expression sets but *HPA* cover about 80% of proteins in the PPI networks. Especially, the *RNAseq Atlas*, which is the biggest expression dataset, covers more than 93% of proteins of all PPIs. Since the *HPA* expression dataset is the smallest one (it only contains values that are scored as reliable), it also shows the lowest coverage percentages, dropping as low as 22.5%.



PPI edge coverage by expression data sets

		Body Map	Gene Atlas	HPA	HPA All	RNAseq Atlas
STRING	58011	74.7 %	82.7 %	13.5 %	62.3 %	94.3 %
IMEx	304524	86.3 %	87.9 %	17 %	77.1 %	92.5 %
HI−2012	13943	60.3 %	68.7 %	6.7 %	60.1 %	98.4 %
Havugimana	13885	86.7 %	87.1 %	19 %	74.6 %	92.2 %
Bossi	77197	88.6 %	89.4 %	17.9 %	74.7 %	93.2 %
	size	14404	18444	3836	15061	21399

Percent
100
75
50
25
0

Figure 3.9: The percent of edges in the PPI networks for which the expression datasets contain expression values for both, connected proteins.

Since an edge in a PPI can only be present when both interacting partners are preset, the percentages of edges that remain in the networks are lower than those we showed before. Still, for most expression datasets more than 60% of edges are covered for all PPIs. For the small *HPA* dataset however, the network size shrinks substantially to less than 7% of the original network size (see the table in Figure 3.9).

3.4.2 Sizes of tissue specific subnetworks

In the previous section, we considered whether two interacting proteins have expression data associated to them. The graphs that contain only those nodes are already smaller than the whole PPIs. However, in the tissue specific sub-graphs, an edge exists only if both interacting proteins are expressed in the same, specific tissue.

Lopes *et al.* (2011) [41] constructed tissue-specific networks in the same way we do. They found that the size of the specific subnetworks is in the range $1-25\%$ of the original network size. In our case, the size of the subgraphs depends on the

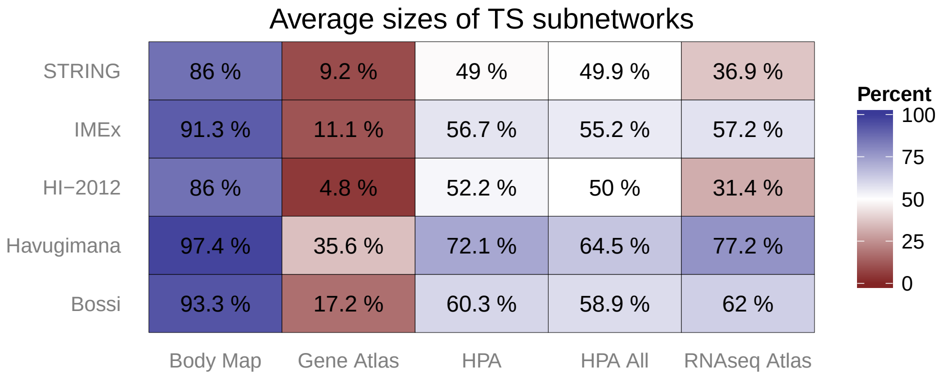


Figure 3.10: This figure shows the average size (in number of edges) of tissue-specific subnetworks relatively compared to the full graph.

expression data: some resulting sub-graphs are considerably smaller, while others decrease only slightly in size. In Figure 3.10, we show the average number of expressed edges of the tissue-specific subnetworks relative to the total number of edges. The resulting relative size differences depend to a large degree on the expression data set. Over 80% of all edges in all PPI networks are expressed when the *Illumina Body Map* is used. Using the *Gene Atlas* on the other hand results in at most 35% of edges remaining. Nevertheless, the number of expressed edges depends also on the PPI network, with most edges being expressed in the protein complex network from *Havugimana et al*.

3.5 Algorithms for analysis of tissue-specific PPI networks

3.5.1 Graph representation of tissue-specific PPIs

For our analysis of tissue-specific PPI graphs we use and extend the graph data-structure and the algorithms implemented in the *NetworKit* toolkit [58] (see Section 2.4). A tissue-specific graph is represented as a single graph in *NetworKit*, where we also label each node with a binary expression vector. For a given protein (i.e., a vertex), this expression vector contains a binary value for each tissue, stating whether the protein is expressed in each tissue.

This graph data-structure implicitly defines a set of subgraphs of the full PPI graph. For each tissue, the tissue-specific subgraph is defined as the graph containing only nodes whose expression value is 1 for that tissue (see also Section

1.2.5).

3.5.2 Local clustering coefficient

The *local clustering coefficient* was previously introduced in Section 1.3.3. For computing the local clustering coefficient, the number of *triangles* around each node has to be determined. A triangle around a node u consists of three distinct nodes including u itself, that are all connected to each other.

The *NetworKit* toolkit [58] implements an *OpenMP* parallel algorithm to determine the local clustering coefficients. The computation of the local clustering coefficients of nodes in the graph are independent of each other. Therefore, the computation can be parallelized across the nodes in the graph. The algorithm implemented by the *NetworKit* toolkit is shown in Algorithm 3.1. We use the **`parallel for`** keywords to represent *OpenMP* parallelized for loops.

```
parallel for u in V:
  triangles = 0
  for (u, v) in E:
    for (v, w) in E:
      if (u, w) in E:
        triangles += 1
  cc[u] = triangles / (deg(u) * (deg(u) - 1))
```

Algorithm 3.1: The local clustering coefficient method in *NetworKit*, where the graph is given by $G(V, E)$ with nodes V and edges E.

This algorithm extends all paths of length two from the source node u. The run time for a given node u (i.e., the inner part of the loop) thus depends not only on the degree of u, but also on the degrees of all neighboring nodes. The run time for a single iteration of the outer-most loop is thus given by:

$$T_{cc}(u) = O\left(deg(u) \times \sum_{(u,v)\in E} deg(v) \right) \tag{3.2}$$

where u is the node for which the local clustering coefficient is calculated.

In our analysis pipeline we calculate the local clustering coefficients for all tissue-specific subgraphs (TSPPIs) for each combination of PPI and expression dataset. For each tissue, a new subgraph is instantiated and the analysis is executed in the subgraph. The whole analysis runs for multiple minutes. Here, we propose an alternate algorithm in order to decrease the run-time. The idea is to calculate the local clustering coefficients for all tissues simultaneously using the tissue-specific graph representation. This eliminates the need to create new graph objects for each tissue-specific subgraph which is analyzed.

First, we propose to change the *NetworKit* algorithm so that only direct neighbors of u are used for the calculation of the local clustering coefficient of u. This algorithm goes through all possible combinations of neighbors and then checks whether the two neighbors are connected by an edge. The resulting algorithm is shown in Algorithm 3.2 and the run time for a single node u (i.e., a single iteration of the outer-most loop) is given by:

$$T_{cc'}(u) = O\left(deg(u)^2\right) \tag{3.3}$$

```
parallel for u in V:
  B = neighbors(u)
  triangles = 0
  for i in 1..(|B|-1):
    for j in (i+1)..|B|:
      if (B[i], B[j]) in E:
        triangles += 1
  cc[u] = 2 * triangles / (deg(u) * (deg(u) - 1))
```

Algorithm 3.2

We show the increased performance of this algorithm compared to the original *NetworKit* implementation for our datasets in Section 5.1.

Tissue-specific subgraphs Next, we change the algorithm further so that we will not have to create a new subgraph for each tissue. Instead, we use the graph representation of the tissue-specific graphs directly and thus operate only on a single graph and on the expression vectors of the nodes. We observe that, if a triangle exists in the full PPI graph, then it exists in a tissue-specific subgraph if and only if all three nodes are expressed in that tissue. Thus we define a *triangle expression vector* as the boolean *AND* of the expression vectors of the three nodes. For each triangle in the graph, this triangle expression vector states in which tissue it is fully expressed and therefore present in the tissue specific graph.

To calculate the local clustering coefficients for a single node for all tissues, we thus simply need to add the triangle expression vectors for all triangles in the global PPI graph around that node. We show the resulting algorithm in 3.3. Since a single node can have different clustering coefficients for different tissues, the local clustering coefficients per node are also a vector. These vectors of clustering coefficients are computed from the number of triangles in each tissue and the number of maximum possible triangles for each node and for each tissue.

In Section 5.1, we illustrate that this algorithm performs considerably better compared to creating a new graph instance for each tissue and running either Algorithm 3.1 or 3.2 on all subgraphs.

```
parallel for u in V:
  B = neighbors(u)
  # vector of size equal to number of tissues
  triangles = [0,...,0]
  for i in 1..(|B|-1):
    for j in (i+1)..|B|:
      if (B[i], B[j]) in E:
        # vector boolean AND
        t_expr = expr[u] and expr[B[i]] and expr[B[j]]
        # vector addition:
        triangles += t_expr
  # vector ops
  cc[u] = 2 * triangles / (deg(u) * (deg(u) - 1))
```

Algorithm 3.3: our algorithm to simultaneously calculate the local clustering coefficients in all tissue-specific subnetworks.

3.5.3 Betweenness centrality

The *NetworKit* toolkit [58] implements Brandes algorithm [11] for computing the betweenness centrality. This algorithm performs a breadth-first search (BFS) and then a backwards accumulation in the BFS search tree for every node in the network. Since a BFS runs in $O(m)$, Brandes algorithm has a runtime of $O(n \cdot m)$ time, where n is the number of nodes in the graph and m is the number of edges.

For our later analysis of tissue-specific subnetworks, we calculate the betweenness centrality for all proteins in all tissues for all combinations of PPIs and expression datasets. This operation takes considerable time. Our goal is to adapt this algorithm in a similar manner as we did for the local clustering coefficient, i.e., run the analysis for all tissues simultaneously without the need to create new graph instances for each subnetwork. That algorithm is run on the global graph once and the results for all tissues are computed simultaneously using vector representations.

However, this approach does not work for the BFS, since the BFS queue and the search paths diverge in case a node is missing in one subnetwork, but not in another. We therefore revert to running Brandes algorithm on each single tissue-specific subgraph independently.

The implementation of Brandes algorithm in *NetworKit* is sequential (i.e., not parallelized). Since the subgraphs for all tissues are independent from each other in regards to computing the betweenness centrality, we can easily parallelize the computation across all the tissues. We use *OpenMP* to parallelize the outer loop which loops over all tissues of a tissue-specific graph.

Furthermore, we use two approaches to calculate the betweenness centrality for all tissue-specific subgraphs of one PPI. The first method we use, is to create new graph instances for each tissue and then run the *NetworKit* implementation on those graphs. This method does not make use of the tissue-specific graph representation. The algorithm for this approach is shown in Algorithm 3.4.

```
def tissue_betweenness(TSG)
  parallel for tissue in TSG.tissues():
    G = TSG.subgraph(tissue)
    bw[tissue] = NetworKit.betweenness(G)
```

Algorithm 3.4: The parallelization over tissues for calculating the betweenness centrality for all tissues in the tissue-specific graph given by TSG.

Additionally, we implemented modified version of the algorithm from *NetworKit* which works directly on the tissue-specific graph representation. Only a few changes are necessary: first we start the BFS only on nodes which are expressed in the current tissue. Additionally, whenever an edge is expanded in the BFS, we have to also check whether the target node is expressed in the current tissue. Only when the node is expressed, it will be inserted into the BFS queue. We show the performance of these two methods in Section 5.1.

3.6 Scoring clusters based on Gene Ontology

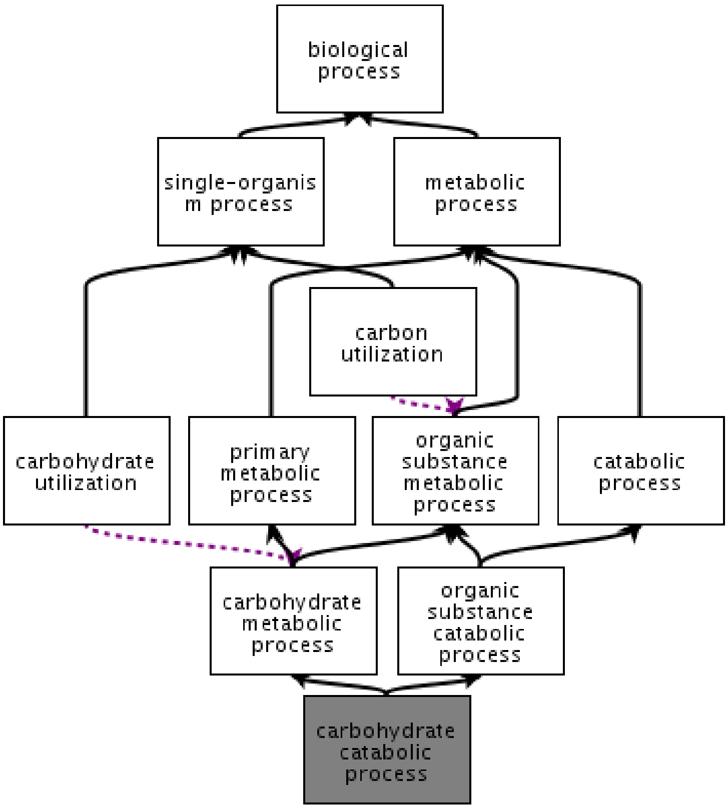
Given a cluster in a clustering of a protein-protein interaction network, we would like to be able to find out how "good" the cluster is. We would like to be able to say that a cluster is "good", if the genes within the cluster are biologically more similar to each other than they are to other genes not contained in that cluster.

To that end, we will use *Gene Ontology* to calculate the similarity between genes and subsequently we will score clusters based on this gene similarity. We will introduce *Gene Ontology* in Section 3.6.1 and then explain *semantic similarity* measures in 3.6.2 and finally show how we efficiently implemented the semantic scoring of clusters in 3.6.3.

3.6.1 Gene Ontology

A *Gene Ontology* defines a set of standardized terms (so called *GO-Terms*) to describe the function, processes, and localization of proteins and gene products [19]. These terms are grouped into 3 distinct *namespaces*: *molecular function*, *biological process*, and *cellular component*. In each of these groups, the terms are organized hierarchically in a *directed, acyclic graph (DAG)*. This hierarchy of terms is continuously updated and published by the *Gene Ontology Consortium*.

In addition to the standardized vocabulary in form of GO-Terms, the *Gene Ontology Consortium* publishes gene product annotations. These associate each protein/gene with a set of GO-Terms describing its molecular function, the biological processes the protein is involved in, and the localization of the protein (i.e., its cellular components).


biological process
single-organism process
metabolic process
carbon utilization
carbohydrate utilization
primary metabolic process
organic substance metabolic process
catabolic process
carbohydrate metabolic process
organic substance catabolic process
carbohydrate catabolic process

QuickGO - http://www.ebi.ac.uk/QuickGO

Figure 3.11: This figure depicts the GO-Term hierarchy of the term "carbohydrate catabolic process". We created this figure with the *QuickGO* online GO browser on May 16th 2014 [7]

Take, for example, the protein *Glucose-6-phosphate dehydrogenase (G6PD)*, which is an essential part of the cell's metabolic process. This protein is annotated with 11 distinct GO-terms (AmiGO2 [13] queried on May 16th 2014), one of which is the GO-term *GO:0016052*: "carbohydrate catabolic process". This term is part of the *biological process* namespace. The term and its ancestors are shown in Figure 3.11. This example depicts the GO-DAG graph structure, which shows the ancestors of the "carbohydrate catabolic process" term, including the GO-terms: "metabolic process" and "carbon utilization".

The full GO-DAG consists of a total of 38,738 GO-Terms with 64,457 edges (see Table 3.6). The *biological process* namespace is the largest of the three distinct namespaces with 25,637 nodes and 47,588 edges.

namespace	**terms (nodes)**	**edges**
biological process	25637	47588
molecular function	9678	11729
cellular component	3423	5140
total	38738	64457

Table 3.6: The number of GO-Terms and edges in the GO-DAG for each namespace

3.6.2 Semantic similarity

In order to be able to define similarity between genes, we will first summarize previous methods for determining *semantic similarity* for GO-Terms. We will then go on to describe a method to combine GO-Term similarities in order to calculate a functional similarity between genes.

Semantic similarity between two GO-terms relates to how similar these two terms are. Different similarity measures have been proposed, e.g., by Reisnik [50], [51] and Lin [38].

These two similarity measures are based on the information content of GO-Terms, which is defined in the following paragraphs.

Frequency of GO-Terms First, we need to define the frequency and probability of a GO-Term. The gene product annotation data provided by the *Gene Ontology Consortium* associates each gene with multiple GO-Terms. Inversely, each GO-Term can be mapped to multiple genes, the number of which shall be given by $gene\_no(t)$ for the GO-Term t. The frequency $f(t)$ of a term t is then defined as the sum of the number of genes mapping to itself or any of its decedents in the DAG. The frequency is thus given by:

$$f(t) = gene\_no(t) + \sum_{t' \in decendents(t)} gene\_no(t') \tag{3.4}$$

Probability of GO-Terms The probability of a GO-Term is defined as its relative frequency, i.e., its frequency divided by the frequency of the root node of the DAG. By this definition, the probability of the root node equals 1, while the probability of all other nodes is between 0 and 1. The probability $p(t)$ of a term t is therefore given as:

$$p(t) = f(t)/f(root) \tag{3.5}$$

Information content Analogous to the definition of Information in Information Theory, the *Information Content (IC)* of a term is defined as [50]:

$$IC(t) = -\log_{10} p(t) \tag{3.6}$$

Similarity measures Based on this definition of *Information Content*, Resnik defines the similarity between two terms as the maximum Information Content of all common ancestors of these two terms [50]. Thus:

$$sim_R(t_1, t_2) = \max_{t' \in CA(t_1,t_2)} IC(t') \tag{3.7}$$

where $CA(t_1, t_2)$ is the set of all common ancestors of t_1 and t_2.

Lin on the other hand, defines the similarity of two GO-Terms as the maximum ratio between the Information content of the common ancestors and of the terms themselves [38]. From the definition of the frequency and thus Information Content follows that Lin's similarity measure takes values in the range between 0 and 1, while Resnik's measure is not bounded. Lin's similarity measure for two terms is given by:

$$sim_L(t_1, t_2) = \max_{t' \in CA(t_1,t_2)} \frac{2 \cdot IC(t')}{IC(t_1) + IC(t_2)} \tag{3.8}$$

Based on the measures by Resnik and Lin, Schlicker *et al.* proposed what they call *relevance similarity* [57]. They also take the overall probability - or rather specificity - of the common ancestor terms into account. Their measure is given by:

$$sim_{rel}(t_1, t_2) = \max_{t' \in CA(t_1,t_2)} \left(\frac{2 \cdot IC(t')}{IC(t_1) + IC(t_2)} (1 - p(t')) \right) \tag{3.9}$$

Similarity of proteins So far similarity was only defined for pairs of GO-Terms. A protein however can have multiple annotations (more than one GO-Term). Schlicker *et al.* define the *BPScore* similarity measure for two proteins as the maximum *relevance similarity* between all pairs of GO-Terms (one term from the annotation of each gene) restricted to terms from the *biological process* namespace [57].

$$BPScore(g_1, g_2) = \max_{\substack{t_1 \in A_{BP}(g_1) \\ t_2 \in A_{BP}(g_2)}} (sim_{rel}(t_1, t_2)) \tag{3.10}$$

where $A_{BP}(g)$ is the annotation for the gene product g, i.e., a set of GO-Terms, but restricted to terms belonging to the *biological process* namespace.

3.6.3 Efficiently scoring clusters

Scoring clusters Based on the above similarity measure between proteins we can now score clusters. The goal of such a semantic similarity score is to assign clusters a high score if the proteins within the cluster are more similar to each other than they are to other proteins. Similarly, we want to give clusters a low score if the proteins in the cluster are not semantically more similar to each other than to other proteins.

We define a cluster's semantic similarity score as the average of all pairwise $BPScore$s between all proteins in the cluster minus the average of all pairwise $BPScore$s between all protein pairs where one protein is part of the cluster and the other is not. More formally:

$$score_{Cl}(C) = \operatorname*{mean}_{\substack{g1, g2 \in C \\ g1 \neq g2}} (BPScore(g1, g2)) - \operatorname*{mean}_{\substack{g1 \in C \\ g2 \in \overline{C}}} (BPScore(g1, g2)) \tag{3.11}$$

where C is a cluster in the form of a set of proteins and $\overline{C}$ is the set complement of C, i.e., all proteins that are not in the cluster C. Since the $BPScore$ is always in the range $[0, 1]$, the difference of two average $BPScore$ values will be in the range $[-1, 1]$.

Complexity For a cluster of size k from a total population of n proteins, the $BPScore()$ function has to be evaluated $\frac{k(k-1)}{2}$ times for the first mean and $k(n-k)$ times for the second mean for a total of $O(nk)$ times. For all clusters in a clustering this will be in $O(n^2)$.

Furthermore, the computation of a single *BPScore* requires evaluation of the sim_{rel} function for each pair of GO-Terms that the genes are annotated by. For each pair of GO-Terms, the GO-DAG has to be traversed to find the common ancestor with maximum information content. The evaluation of clusters is thus a very time consuming process.

In the following, we show how to speed this process up by approximately two orders of magnitude. We do not solve this by introducing a new algorithm, but by optimizing the similarity computation and by introducing lookup tables, thus shifting the majority of the computations into a single pre-processing step.

fastSemSim Mina *et al.* implemented an extensive software framework *fastSemSim* for "fast and easy evaluation of semantic similarity measures on GO annotations" [44], [43]. We used their software to score clusters in PPI network clusterings. *fastSemSim* scores a single cluster of 500 proteins in 43.1 seconds. Clustering a PPI network with *NetworKit* takes at most 10% of that time for our largest PPI network. Thus *fastSemSim* is too slow for our purposes (see also table 3.7.

Speeding up the scoring To evaluate clusters more efficiently, we implemented our own methods. To this end, we re-implemented the scoring functions from Schlicker *et al.* (2006) [57]: sim_{rel} and $BPScore$. Note that these are also implemented in *fastSemSim* among many other scoring techniques. We however only concentrated on these two measures.

We based our implementation on the *goatools* software [60]. Specifically, we modified the graph data structure of the GO DAG graph used in the software to allow for more efficient global lookups of GO-Terms, their edges, children, parents, ancestors and decedents.

sim_{rel} **lookup table** First, we observe that the *relevance similarity* sim_{rel} only depends on the GO DAG graph structure and the gene product annotations. Especially, it does not depend on the PPI graph structure or the specific clusters. Therefore, we can pre-compute the sim_{rel} scores once and reuse them for all scorings. The similarities are calculated for all GO-Terms and then saved as single-precision floating point values of 4 bytes each. Single precision representation is sufficient, since the total precision is bounded by the comparatively small total number of GO-Terms and genes in the GO-DAG. Since there are 25,637 distinct GO-Terms in the *biological process* namespace (see Table 3.6), thus the lookup table would have a size of $25,637^2 \cdot 4$ bytes $\approx 2.5\ GiB$. However, far from all GO-Terms are used in actual gene product annotations. Some are only internal nodes representing more abstract categories, and no genes are directly annotated by these. Saving the score of only those GO-Terms that appear in the annotation, we are left with a total of $9,392$ terms. This yields a lookup table with a size of $9,392^2 \cdot 4$ bytes $\approx 336\ MiB$, which can easily be kept in main memory. Using this lookup table for scoring a cluster of 500 genes, we observe a speedup of 3.44 compared to re-computing the similarity score every time.

$BPScore$ **lookup table** Analogously, the $BPScore$ measure for scoring genes in a pairwise fashion (where each gene is annotated with multiple GO-Terms) depends only on the GO-DAG graph and the gene annotations. Thus, we apply a pre-computed lookup table here as well. In order to reduce the memory usage and pre-computation time, we restrict ourselves to genes that are annotated with at least one GO-Term from the *biological process* namespace. Note that genes with no such annotation will always have a $BPScore$ of 0, thus there is no need to pre-compute and save these $BPScores$. There are a total of $15,116$ such genes, therefore the $BPScore$ lookup table has a size of $871MiB$. Note, that this lookup table is computed only once (taking approximately 5000 seconds), saved, and then reused every time a clustering is scored.

Further reducing the complexity When scoring a cluster C of size k using the $BPScore$ lookup table, the following steps are performed:

1. internal sum = $\sum_{g\in C}\sum_{i\in C, i\neq g} BPScore[g,i] \qquad \rightarrow O(k^2)$
2. external sum = $\sum_{g\in C}\sum_{j\in\overline{C}} BPScore[g,j] \qquad \rightarrow O(nk)$
3. calculate mean for each: divide by number of summands
4. subtract the means

However, we observe that for any gene product g:

$$\sum_{j\in\overline{C}} BPScore[g,j] = \sum_{i\in G} BPScore[g,i] - \sum_{i\in C} BPScore[g,i] \tag{3.12}$$

where G is the set of all genes and the sum $\sum_{i\in G} BPScore[g,i]$ is the sum over the row g in the lookup matrix/table. Therefore, in addition to the matrix $BPScore[i,j]$, we save a vector containing the sum of each row. This eliminates the $O(nk)$ step in the scoring computation and we are left with $O(k^2)$ table lookups and additions for scoring a cluster of size k.

Runtime evaluation Using this $BPScore$ lookup table, we observe a 46-fold speedup compared to using the sim_{rel} lookup table. This yields a total speedup of $113\times$ compared to not using any lookup tables. In Table 3.7 we show the run times for scoring clusters of various sizes using *fastSemSim*, our implementation without any pre-computed lookup tables, our implementation using the sim_{rel} lookup table and finally using the $BPScore$ lookup table.

n	fastSemSim	our impl	sim_{rel} lookup table	$BPScore$ lookup table
10	1.78 s	0.0246 s	0.0025 s	6.87e-05 s
100	2.56 s	0.747 s	0.3044 s	0.0037 s
250	13.56 s	3.95 s	1.637 s	0.0236 s
500	43.13 s	12.60 s	5.148 s	0.1108 s
1000	161.5 s	47.17 s	18.70 s	0.3726 s

Table 3.7: Timing results for calculating the pairwise $BPScore$ for $n \in \{10, 100, 250, 500, 1000\}$ genes with fastSemSim, our Implementation and our implementation using the two levels of lookup tables. The timings were all run single threaded on a Intel(R) Core(TM) i5-3570 system with 8 GiB of main memory.

3.7 Clustering of PPIs

Chen and Wang (2012) have used the CFinder clustering method to identify clusters of functionally related proteins in tissue-specific PPIs [17].

We use clustering algorithms to find clusters or modules of proteins in tissue-specific PPIs with the goal of uncovering specific functional modules. Furthermore, we investigate whether clusters found in tissue-specific networks score higher in terms of their *BPScore* than clusters found in the global/non-specific PPI networks.

3.7.1 Clustering and Modularity

A clustering (or partition) of a graph is defined as a set of disjunct sets of nodes which cover the whole graph. For a clustering $\zeta \subset 2^V$ of a graph $G(V, E)$ this means that for each pair of clusters $C_1, C_2 \in \zeta$ where $C_1 \neq C_2$ that $C_1 \cap C_2 = \emptyset$ and further that:

$$\bigcup_{C \in \zeta} C = V \tag{3.13}$$

Modularity The aim of a clustering algorithm, in our context, is to maximize the modularity of the clusters in a graph. The modularity measures the relation of the number and weight of edges within a cluster versus those edges between clusters. To define the modularity more formally, first the coverage of ζ for a graph $G(E, V)$ is defined as [58]:

$$coverage = \frac{\sum_{C \in \zeta} \sum_{e \in E(C)} \omega(e)}{\sum_{e \in E} \omega(e)} \tag{3.14}$$

where $E(C)$ are all edges internal to the cluster $C \subset V$ and $\omega(e)$ is the edge-weight of an edge e. Similarly, the expected coverage is defined as:

$$expected\_coverage = \frac{\sum_{C\in\zeta}(\frac{1}{2}\sum_{v\in C}\omega(v))^2}{(\sum_{e\in E}\omega(e))^2} \tag{3.15}$$

where $\omega(v)$ is the weighted degree of v, which is the sum of the edge-weights over all adjacent edges of v. The modularity is then defined as the difference between the coverage and the expected coverage [58].

Modularity of a cluster Using this definition of modularity, we define the modularity of a cluster C in a clustering as:

$$m(C) = \frac{\sum_{e\in E(C)}\omega(e)}{\sum_{e\in E}\omega(e)} - \frac{(\frac{1}{2}\sum_{v\in C}\omega(v))^2}{(\sum_{e\in E}\omega(e))^2} \tag{3.16}$$

This definition allows us to score each cluster by its contribution to the total modularity score of a clustering, since:

$$modularity(\zeta) = \sum_{C\in\zeta} m(C) \tag{3.17}$$

3.7.2 Clustering algorithms

In this study, we use the *Parallel Label Propagation* and *Parallel Louvain Method* clustering algorithms, which are implemented in the *NetworKit* toolkit [58] (see also Section 2.4).

Similar to our modifications to the algorithms for finding the local clustering coefficients and the betweenness centrality, we modify the *PLP* algorithm to work directly on the tissue-specific graph representation. The modified algorithm clusters all tissue-specific graphs simultaneously by taking advantage of the binary vector representation of the per protein tissue-expression.

Instead of using a single label for each node, we use a vector of labels with one label per tissue. Initially, the labels are all initialized to the same value. In each iteration a node changes its label to the most common label in its neighborhood. Since the adjacent edges of a node differ from tissue to tissue, we do this selection separately for each tissue. Therefore, while running the label propagation algorithm, a node might become labeled with distinct labels.

Using the original *PLP* algorithm, we have to create a subgraph for each tissue and run the algorithm on each subgraph. Hence, there is an outer loop across all tissues. In our modified *PLP* algorithm we have moved this loop into the process for each node. However, contrary to our algorithm for the local clustering coefficient, we can not take as much advantage of the vector representations. This modified algorithm performs worse than running the original on newly created subgraphs (see Section 5.1.3).

3.7.3 Identifying important modules

The clusterings returned by the community-detection algorithms cover all nodes and thus proteins in the graph. However, we are only interested in the "good" clusters. Since a "good" clustering is one which maximizes the modularity, we rank the clusters of a clustering by their contribution to the total modularity as defined in equation 3.16. We arbitrarily choose the 20% highest ranking clusters as "good" clusters. We later show in Section 5.4, that these clusters have significantly higher *BPScores* than the remaining 80% of clusters.

In order to determine the semantic and functional similarity score of each of those clusters, we use the Gene-Ontology based *BPScore*, previously introduced in Section 3.6. We use this score for testing the results of our clustering approach. If a cluster has a BPScore above zero, its proteins are semantically more related to each other than they are to other proteins. This might indicate that we have successfully identified clusters of semantically and functionally related proteins.

We test this for all tissue-specific PPIs and different clustering algorithms (PLP and PLM) and different parameter settings for those algorithms. We show the results in Section 5.4.

We furthermore investigate, whether clusters found in tissue-specific networks obtain higher BPScores than clusters in global PPI networks. We repeat this for all combinations of PPIs and expression datasets and run the clustering algorithms with a set of different parameters. The results are shown in Section 5.4.

3.7.4 Edge weighting

The clustering algorithms and the definition of modularity take into account the weights of the edges in the graph. So far the full PPI graph and the tissue-specific subnetworks are all unweighted graphs. Thus, all the edge weights default to 1.0 in the implementations of the clustering algorithms in *NetworKit*. In addition to the unweighted graphs, we construct weighted PPI graphs by combining PPIs with expression data. We base the edge weights on the expression profiles of the interacting proteins. We evaluate the clusters found in those graphs using the same clustering algorithms which we used for the previous analysis.

We try two different methods for weighting the edges of a PPI based on the expression data. As a first approach, we use a scaled correlation and secondly we base the weight on the co-expression of the interacting proteins.

Correlation weight For any edge (p_1, p_2) in the PPI graph, we determine the Pearson correlation coefficient $corr(\vec{e_{p_1}}, \vec{e_{p_2}})$ of the expression vectors for p_1 and p_2. The expression vector $\vec{e_p}$ for a protein is a binary vector, which contains for each tissue whether this protein is expressed or not. The value of the correlation coefficient lies in the range $[-1, 1]$. We scale the correlation into the range $[0, 1]$ and assign this as the weight of the edge:

$$\omega((p_1, p_2)) := \frac{1}{2} \cdot (1 + corr(e_{p_1}, e_{p_2})) \tag{3.18}$$

Two proteins with a perfect negative correlation are never expressed together. An edge between those proteins will be assigned a weight of zero, which agrees to the fact that these proteins can never interact.

Co-expression weight The second edge weight is based on the number of tissues in which both interacting proteins are simultaneously expressed. We call this number the co-expression count of two proteins and denote it as $coexpr\_count(p_1, p_2)$ for two proteins p_1 and p_2. Additionally, let $expr\_count(p)$ for any protein p be the number of tissues in which this protein is expressed. Given these, we define the co-expression edge weight as:

$$\omega((p_1, p_2)) := \frac{coexpr\_count(p_1, p_2)}{max\,(expr\_count(p_1), expr\_count(p_2))} \tag{3.19}$$

Chapter 4

Implementation

In this chapter, we will describe our implementation of the pipeline used for data import, management and analysis. More Specifically, we will describe the technologies and programming languages used in each part and we will lay out the general software structure.

Our implementation can be roughly categorized into three main components. First, the data import and normalization procedures, then the tissue-specific graph analysis, and last data analysis and visualization. We illustrate the overall layout of and functionality of these components in Figure 4.1.

4.1 Data pipeline

The first part of the implementation consists of the data importing and normalization steps. We described the data processing steps in Section 3.1.

We implement the data import and processing using *Python*. For a unified data format and storage we use the SQL database *SQLite*. We store the PPIs, the expression data sets, the ID mapping tables, and the analysis results in the same SQL database.

In order to abstract the common processing steps for PPIs and expression datasets, we define an object-oriented class hierarchy (see figure 4.2). Since each PPI and expression dataset is stored as a SQL table, we define a `TableManager` as a common base class which implements common functions for managing SQL tables. Two abstract classes `PPI` and `ExpressionSet` inherit from `TableManager`. These two classes implement the common processing steps for PPIs and expression datasets.

Each PPI is implemented as a class inheriting from `PPI`. In the class for each PPI only PPI specific functions are implemented, such as importing the downloaded files in their raw format and filtering by reliability. The implementations

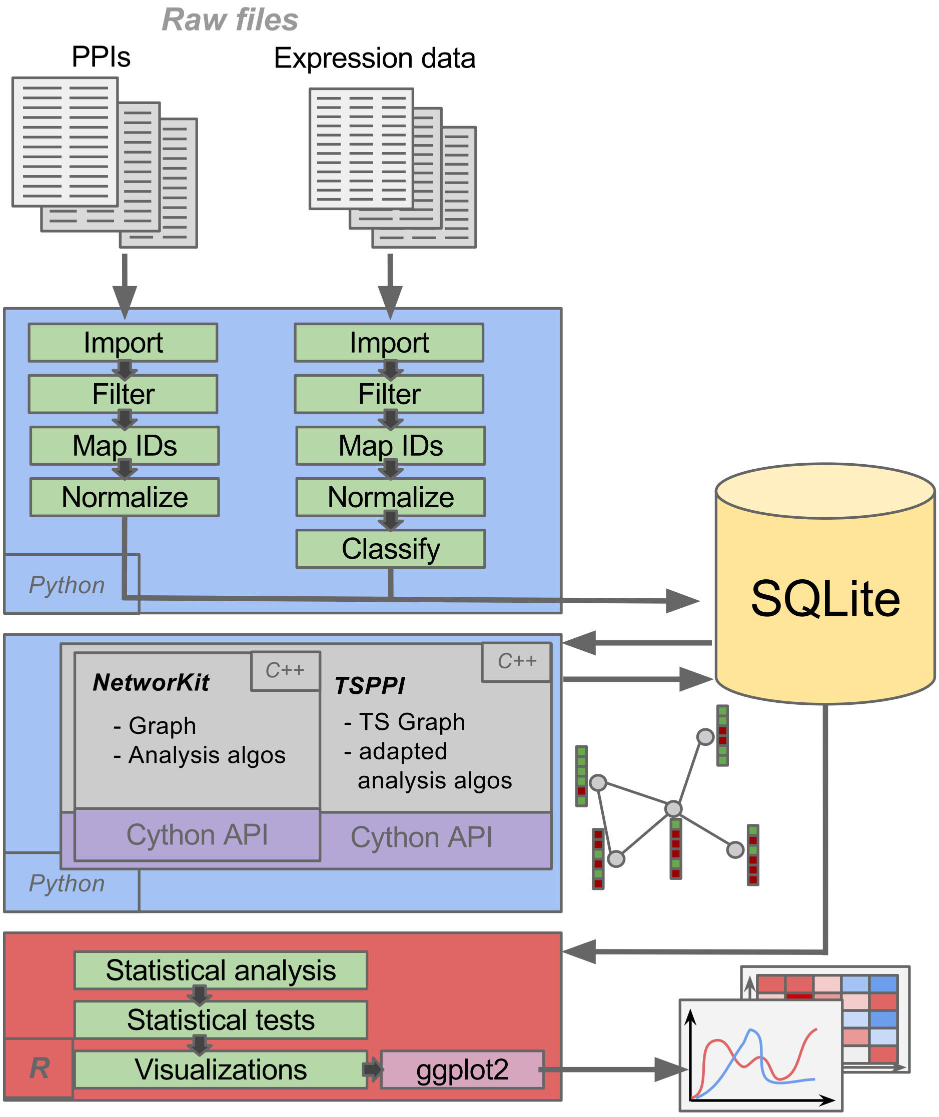


Figure 4.1: Overview of all components of our implementation.

for normalization and ID mapping are shared across all instances.

We implement a class for each expression dataset in an analogous way. Here again, only those functions which are expression set specific are implemented and all other functions are implemented in `ExpressionSet` and thus shared across all instances.

We chose to implement the pipeline in this object-oriented and modular fashion, to allow for easy extensibility. A new PPI or expression dataset can be added to the pipeline by inheriting from the corresponding base class and then overwrit-

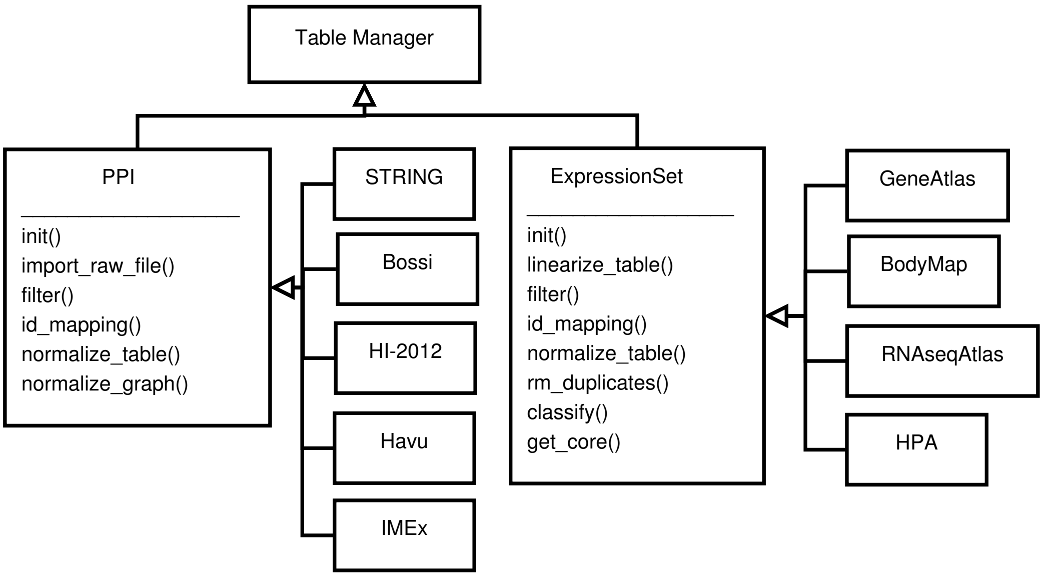


Figure 4.2: Class diagram of the data import and processing pipeline.

ing only those functions which are specific to the newly added class.

4.2 Analysis

4.2.1 Graph analysis

For analyzing the PPI and tissue-specific PPI graphs, we use the *NetworKit* toolkit (see also Section 3.5).

We implement a tissue-specific graph class `TsPPI` using *C++*. This class contains a `NetworKit::Graph` instance for the PPI network. The per protein tissue expression is implemented as a node label consisting of a binary vector. We use *boost's* `boost::dynamic_bitset` as implementation for the binary vector. Among others, this class overloads operators for *binary or*, *binary and*, and methods to count the number of bits that are set to `1`. We use these functions in our implementations of the graph algorithms for tissue-specific PPIs (local clustering coefficients and betweenness).

We further implement *C++* functions to load and construct the tissue-specific PPIs from the SQL database. We use the *SQLiteC++* wrapper to access the database from within *C++* [52]. Since *NetworKit* uses integer vertex identifiers, we also implement a mapping of gene names and tissues to integers. After analysis of the tissue-specific graphs, we map these integers back to the original gene names, prior to storing the results of the analysis into the SQL database.

Furthermore, we implement a *cython* Python interface for our implemented classes and algorithms, which is compatible with *NetworKit's cython* Python interface. We use this interface from within Python scripts in order to analyze all possible tissue-specific graphs for all pairs of PPIs and expression datasets.

4.2.2 Final data analysis and visualizations

After all graphs have been analyzed and the results have been stored in the SQL database, we use *R* for the final data analysis and visualization. Most of the plots and graphs in this thesis are generated using the *R ggplot2* library.

Chapter 5

Evaluation and discussion

5.1 Performance evaluation

5.1.1 Local clustering coefficient

We introduced a modified version of the *NetworKit* algorithm for computing local clustering coefficients in Section 3.5.2. Additionally, we provided a new algorithm to compute local clustering coefficients for all tissues simultaneously using the tissue-specific graph representation, in which each node is annotated with an expression vector. Here, we will evaluate the performance of the three methods for calculating the local clustering coefficients in all tissue-specific subnetworks.

The first algorithm, we evaluate, is the original implementation from the *NetworKit* toolkit. This algorithm extends all paths of length 2 from the current node. The second algorithm instead goes through all possible choices of two distinct neighbors. We call this method *Neighbor combinations* in the figure below. These two algorithms both work on a single graph data-structure. Thus a new subgraph has to be created for each tissue. The final algorithm we evaluate, is our proposed algorithm which operates directly on the tissue expression vectors. This algorithm is denoted as *Tissue expr. vectors* below.

We ran the experiments on an Intel(R) Core(TM) i5-3570 system with 8 GiB of main memory and four physical cores. Each algorithm was run with 4 *OpenMP* threads. We ran the algorithms on all 25 combinations of PPI networks and expression datasets. In Figure 5.1, we show the results for the 5 instances that have the highest runtime for the *NetworKit* method. These five instances are also the largest among all 25 in terms of the number of edges multiplied with the number of tissues.

The benchmark results show that the *Neighbor combinations* algorithm performs by a large margin better than the algorithm implemented in *NetworKit.* We observe the largest speedup (> 14) for the *STRING* and *Gene Atlas* combination.

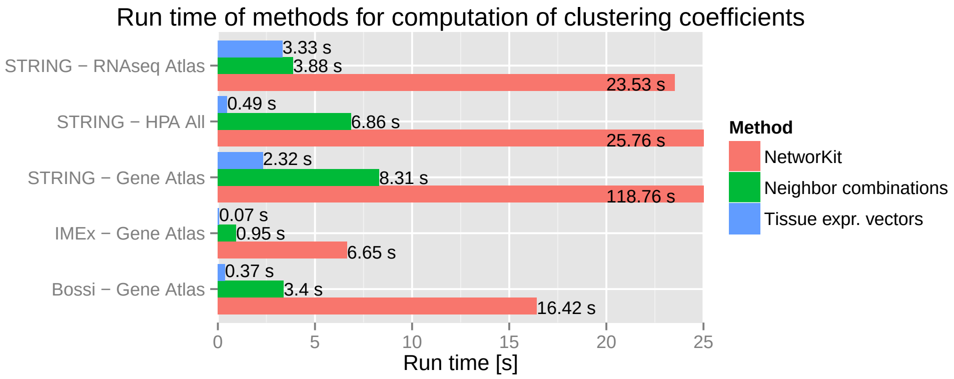


Figure 5.1: In this figure we show the results of benchmarking the three different methods for calculating the local clustering coefficients in all tissue-specific subnetworks for the five largest combinations of PPI networks and expression datasets.

Furthermore, we find that the algorithm that operates directly on the expression vectors achieves another considerable speedup over the *Neighbor combinations* method. However, the speedup between these two methods varies from as low as ≈ 1 all the way up to > 10. The results for all 25 combinations of PPI networks and expression datasets are printed in the appendix in Table A.1.

Method	**Full.Runtime**
NetworKit	211.1 s
Neighbor combinations	31.0 s
Tissue expr. vectors	7.6 s

Table 5.1: The total runtime for calculating the local clustering coefficients in all tissues for all 25 combinations of PPI networks and expression data sets.

In Table 5.1, we show the total cumulative runtime for executing the algorithms on all PPI networks and expression data sets. These runtimes illustrate the large margin of improvement over the *NetworKit* implementation.

5.1.2 Betweenness centrality

Next, we will evaluate the performance of the two methods for calculating the betweenness centrality for all tissue-specific subnetworks. The two methods were introduces in Section 3.5.3.

Figure 5.2 shows the running times of the two methods for the 8 largest instances of all PPI networks and expression combinations. We ran these experiments on the same test system as for the clustering coefficients. Both methods were run with *OpenMP* parallelism with 4 threads.

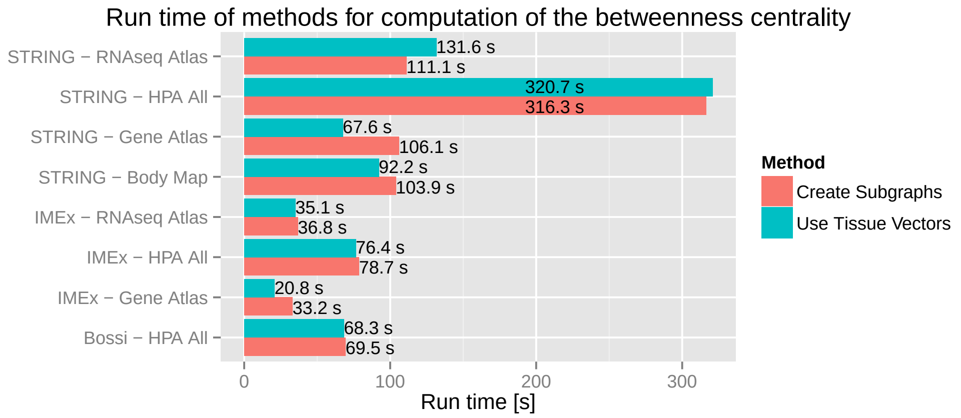


Figure 5.2: The run times of the two different methods for the betweenness centrality for the largest 8 PPI and expression instances.

We observe mixed results: the method that creates new subgraphs for each tissue performs better in some, but not all cases. For the *STRING* PPI combined with the *Gene Atlas* expression set, the second method performs better by a factor over $1.5\times$.

Method	**Sequential**	**Parallel**	**Speedup**
Create Subgraphs	2612.9 s	1029.4 s	2.54 ×
Use Tissue Vectors	3124.7 s	971.0 s	3.22 ×

Table 5.2: The total run time for calculating the betweenness centrality in all tissues for all 25 combinations of PPI networks and expression datasets.

In Table 5.1 we show the run time accumulated across all PPI networks and expression datasets for the cases when they are run sequential and in parallel using 4 *OpenMP* threads. When running the algorithms sequential, we observe that the *Tissue Vectors* method takes approximately 1.2 times longer. For the parallel execution, however, this method reaches a speedup of $3.22\times$, which is larger than the $2.54\times$ speedup obtained with the *Subgraph* method. Due to the higher speedup, the total parallel runtime is smaller for the *Tissue Vectors* method. The better speedup could be attributed to data remaining in a shared chache, since the *Tissue Vectors* method uses only one instance of the tissue-specific graph representation, whereas the first method creates a new graph for each iteration.

Table A.2 in the appendix shows the run time for all PPIs and expression dataset combinations for both methods.

5.1.3 Parallel Label Propagation

We implemented an adapted version of the *Parallel Label Propagation (PLP)* algorithm to use the tissue-specific graph representation. We show the resulting runtime for some of the PPI and expression combinations in figure 5.3.

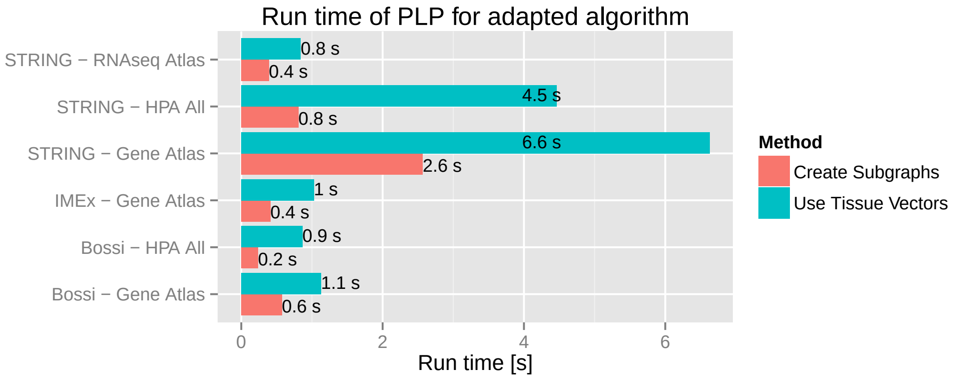


Figure 5.3: The run times of the adapted PLP algorithm compared to running the original *NetworKit* implementation on each subnetwork separately.

The results indicate, that the algorithm adapted for the tissue-specific graph representation performs less well than the original *NetworKit* implementation. We observe up to a factor $4\times$ difference in runtime. We therefore conclude, that in this case there is no gain from adapting the algorithm to run on our tissue-specific graph representation.

We show the runtime results for all combinations of PPIs and expression datasets in Table A.3.

5.2 Benchmarking prior results

5.2.1 Analysis and Results

Bossi and Lehner (2009) have come to multiple conclusions about tissue specific PPI networks [10] (see Section 2.3). Their findings are based on a composite network, which they constructed from various sources. We will refer to their PPI network as *Bossi* (see Section 2.1.4). Bossi and Lehner constructed a tissue-specific PPI network by annotating their composite PPI with expression data from the *Gene Atlas* expression data set. In their study, Bossi and Lehner analyzed the properties of tissue-specific and housekeeping proteins in their PPI graph.

Emig and colleagues showed that some of the results from Bossi and Lehner cease to remain true when using RNAseq expression data in place of the *Gene Atlas* [21] [22].

In this section we will reconstruct Bossi and Lehner's findings and evaluate their validity on all 25 combinations of PPI networks and expression data sets, including those originally used by Bossi and Lehner and by Emig *et al.* .

Interaction degrees of tissue specific proteins

The first reported finding by Bossi and Lehner is, that tissue-specific proteins make fewer interactions than more widely expressed proteins. Bossi and Lehner retain only those edges in their PPI, for which the interacting proteins are co-expressed in at least one tissue. We will thus do the same in this analysis. We further define the *tissue specificity* of a protein as the number of tissues in which that protein is expressed. In Figure 5.4 we plot the protein interaction degree against tissue specificity of the proteins for the *Bossi* PPI network and the *GeneAtlas* expression data set. We observe that tissue-specific proteins make fewer interactions than more widely expressed proteins. This is the same result that Bossi and Lehner showed in their study.

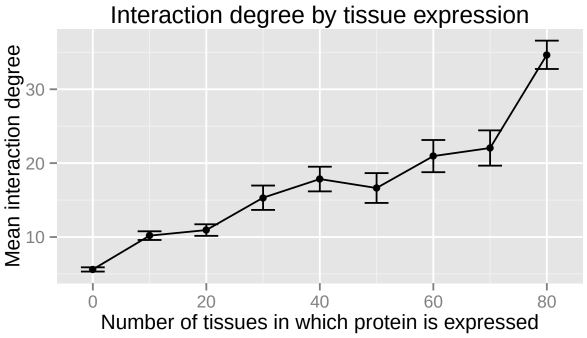


Figure 5.4: This shows the mean protein interaction degree for varying tissue specificity. The x-Axis represents the number of tissues a protein is expressed in. The error bars are one standard error. This data is based on the *Bossi* PPI network and the *GeneAtlas* expression data.

This trend is, however, not observable for all combinations of PPI networks and expression data sets. Consider for example the results when combining the *CCSB HI-2012* PPI network with the *Human Protein Atlas* expression data (Figure 5.5). With this particular combination, there is no clear trend observable. The larger error bars are explainable by the relatively small size of this PPI ($n =$

991 proteins). The previously shown combination of (*Bossi* with *Gene Atlas*) has almost ten times that size ($n = 9048$).

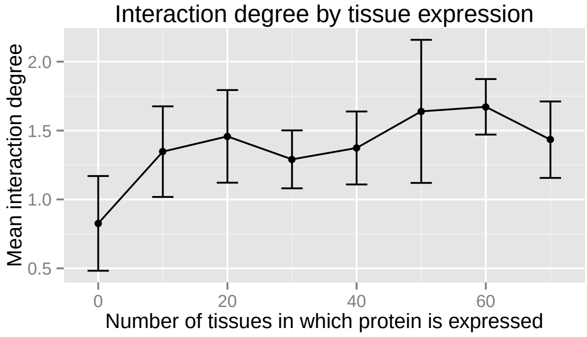


Figure 5.5: This shows the mean protein interaction degree for varying tissue specificity. The x-Axis represents the number of tissues a protein is expressed in. The error bars are one standard error. This data is based on the *CCSB HI-2012* PPI network and the *HPA* expression data.

However, for the *STRING* PPI network annotated by the *Illumnia Body Map* RNA expression data, which has a total size of $n = 15078$, yet another trend appears. The proteins that are expressed in most tissues still have a higher interaction degree than all other proteins, but the preceding trend appears to be inverse to what Bossi and Lehner initially observed (Figure 5.6).

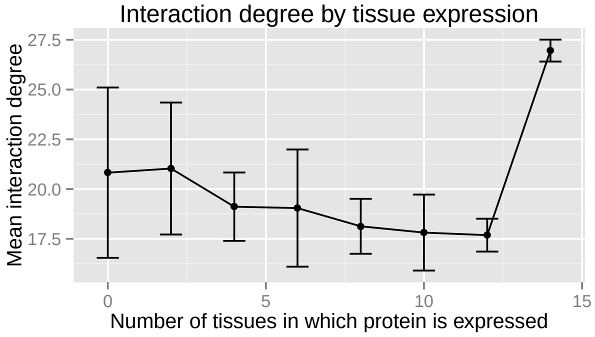
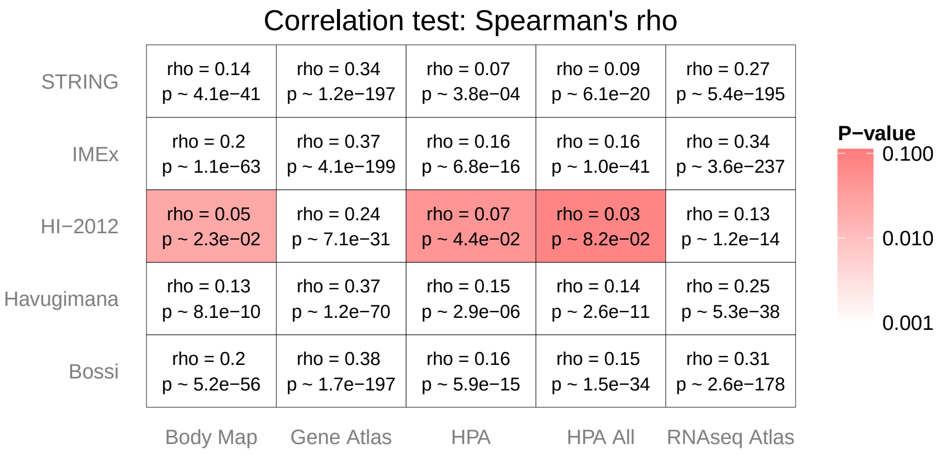


Figure 5.6: This shows the mean protein interaction degree for varying tissue specificity. The x-Axis represents the number of tissues a protein is expressed in. The error bars are one standard error. This data is based on the *STRING* PPI network and the *Illumnia Body Map* RNAseq expression data.

Plots for all combinations of PPI networks and expression data sets are printed in the appendix (see Figure A.1).

Next, we calculate the correlation between a protein's interaction degree and it's tissue specificity in order to show the trend more systematically for all PPI and expression dataset combinations. To test for this correlation, we use *Spearman's* correlation test. For all combinations of PPI networks and expression datasets, we calculate Spearman's ρ and the p-value for the H_0 hypothesis of ρ being zero (see Figure 5.7). In all cases we find a positive correlation ($\rho > 0$).

However, the majority of correlation coefficients are relatively small ($\rho < 0.2$ for 16 out of 25 combinations). In one case (*HI-2012* combined with *HPA All*), we observe no significant p-value ($p > 0.05$).



Correlation test: Spearman's rho

	Body Map	Gene Atlas	HPA	HPA All	RNAseq Atlas
STRING	rho = 0.14 p ~ 4.1e−41	rho = 0.34 p ~ 1.2e−197	rho = 0.07 p ~ 3.8e−04	rho = 0.09 p ~ 6.1e−20	rho = 0.27 p ~ 5.4e−195
IMEx	rho = 0.2 p ~ 1.1e−63	rho = 0.37 p ~ 4.1e−199	rho = 0.16 p ~ 6.8e−16	rho = 0.16 p ~ 1.0e−41	rho = 0.34 p ~ 3.6e−237
HI−2012	rho = 0.05 p ~ 2.3e−02	rho = 0.24 p ~ 7.1e−31	rho = 0.07 p ~ 4.4e−02	rho = 0.03 p ~ 8.2e−02	rho = 0.13 p ~ 1.2e−14
Havugimana	rho = 0.13 p ~ 8.1e−10	rho = 0.37 p ~ 1.2e−70	rho = 0.15 p ~ 2.9e−06	rho = 0.14 p ~ 2.6e−11	rho = 0.25 p ~ 5.3e−38
Bossi	rho = 0.2 p ~ 5.2e−56	rho = 0.38 p ~ 1.7e−197	rho = 0.16 p ~ 5.9e−15	rho = 0.15 p ~ 1.5e−34	rho = 0.31 p ~ 2.6e−178

Figure 5.7: For each combination of PPI networks and expression data sets, Spearman's ρ and the according p-value is shown. Larger p-values are color coded in red.

Tissue-specific and housekeeping proteins

Prior work uses different definitions for whether a protein classifies as tissue-specific (TS), housekeeping (HK) or neither.

Bossi and Lehner [10] fixed the definition of *tissue specific* to be all proteins that are expressed in $\leq$ 10 out of the total of 79 tissues, corresponding to a percentage threshold of approximately 13%. As for defining *housekeeping* proteins, Bossi and Lehner use various definitions and show that their results remain relatively consistent across definitions. For example, they define *housekeeping* proteins to be expressed in $\geq$ 71 out of 79 tissues in one case and to be expressed in all $79/79$ tissues in another case (corresponding to a percentage threshold of approximately 89% or 100% respectively). Additionally, they use varying classification thresholds for the *Gene Atlas* expression data and consider expression data from microarray and EST studies by Zhu *et al.* [66] for the definition of *housekeeping*.

The RNAseq data used by Emig and colleagues [21] consists of expression data for merely 15 different tissues. They define a protein as *tissue-specific* if it is expressed in at most two tissues and as *housekeeping* if it is expressed in at least 14 out of 15 tissues. This corresponds to percentual thresholds of 13.3% and $100 - 13.3\%$.

Since we want to compare the results across our collection of all PPI networks and expression data sets, we define *tissue-specific* and *housekeeping* proteins according to percentage thresholds as previously explained in Section 3.2.3: for a threshold of t (e.g., 15%) we define those proteins as *tissue-specific*, which are expressed in at most t percent of total tissues. Accordingly, we define all proteins expressed in at least $(100\% - t)$ percent of total tissues as *housekeeping*.

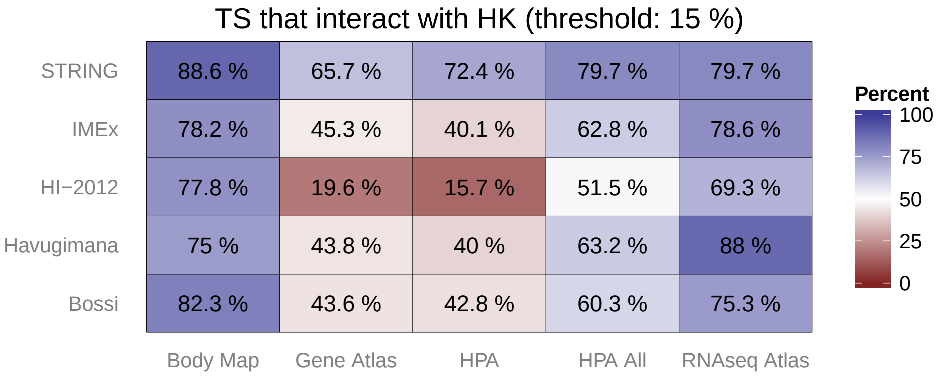
Interactions partners of tissue-specific and housekeeping proteins

Bossi and Lehner further investigated the interaction partners of tissue-specific and housekeeping proteins [10]. They found that most of the tissue-specific proteins interact with at least one housekeeping protein. Furthermore, they found that most of the housekeeping proteins interact directly with one or more non-housekeeping proteins.

In order to evaluate these results for the collection of PPIs and expression sets, we calculate the percentage of *tissue-specific* proteins that interact directly with *housekeeping* proteins, i.e., the fraction of TS proteins that have at least one interacting partner in HK. We find for various thresholds $t \in \{10, 12.5, 15, 20, 50\}$ that Bossi and Lehner's findings remain true for most but not all combinations of PPI networks and expression datasets (Figure 5.8). Especially for the high confidence Y2H network *HI-2012* combined with either the *GeneAtlas* or *Human Protein Atlas* expression datasets, less than 20% (for at least $t \leq 15\%$) of tissue-specific proteins interact with housekeeping proteins. However, the percentage of TS proteins interacting with HK proteins is especially high for both RNAseq based expression datasets combined with any PPI network, as well as for the *STRING* PPI network combined with any expression dataset.

In Figure 5.9 we show how the percentage values depend on the threshold parameter t. In this figure, each pair of PPIs and expression datasets is plotted as a single line. We observe that the percentage of TS interacting with HK is slightly increasing when the threshold parameter (and thus the size of the TS and HK classes) is increased. However, no matter how the threshold t is chosen, the results span a large range, and Bossi and Lehner's findings remain valid for most, but not all PPI and expression sets.

Next, we analyze the interactions of housekeeping proteins with non-housekeeping



	Body Map	Gene Atlas	HPA	HPA All	RNAseq Atlas
STRING	88.6 %	65.7 %	72.4 %	79.7 %	79.7 %
IMEx	78.2 %	45.3 %	40.1 %	62.8 %	78.6 %
HI−2012	77.8 %	19.6 %	15.7 %	51.5 %	69.3 %
Havugimana	75 %	43.8 %	40 %	63.2 %	88 %
Bossi	82.3 %	43.6 %	42.8 %	60.3 %	75.3 %

Figure 5.8: For each combination of PPI networks and expression data sets the percentage of tissue specific (TS) proteins that interact directly with housekeeping (HK) proteins is given for a threshold of $t = 15\%$ used for the classification into tissue specific and housekeeping.

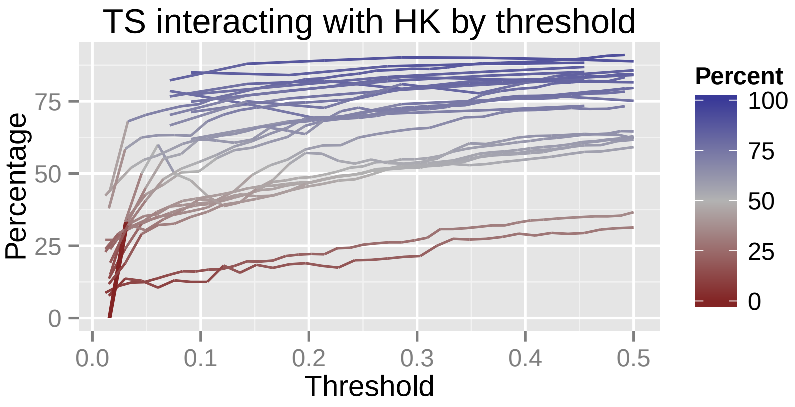


Figure 5.9: The percentage of TS that interact directly with HK plotted for different choices of the threshold parameter t. Each PPI and expression set combination is drawn as a single line.

(non-HK) proteins. Non-housekeeping proteins are all those proteins which are not in the housekeeping class. Bossi and Lehner's results suggest that approximately 90% (and at least 80%) of housekeeping proteins interact with non-housekeeping proteins. According to our findings, this is not always the case. Here we consider proteins that are expressed in at least 90% of all tissues ($t = 10\%$) as housekeeping (Figure 5.10). Especially for the *Illumnia Body Map* expression data the results differ strongly from what Bossi and Lehner found: paired with 4 of the

5 PPI networks the achieved percentage results are below 50%. The highest percentages and thus the strongest results in favor of Bossi and Lehner's findings are achieved for the *Gene Atlas* expression dataset, the same dataset that Bossi and Lehner used.

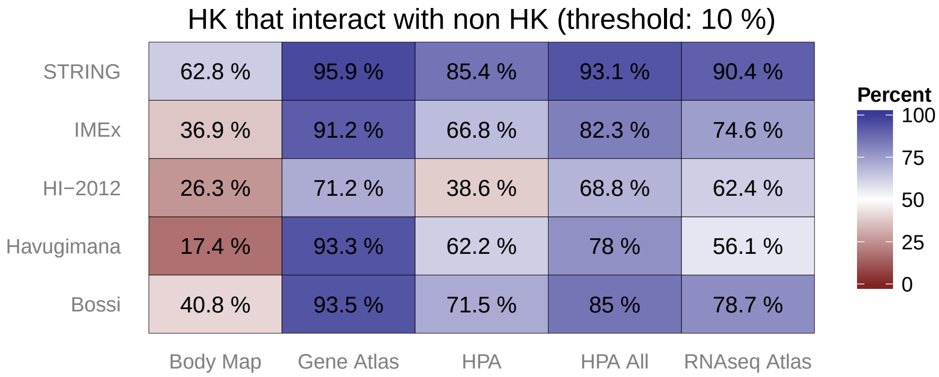


Figure 5.10: For each combination of PPI networks and expression datasets the percentage of housekeeping (HK) proteins that interact directly with non-housekeeping (non-HK) proteins is given for a threshold of $t = 10\%$ used for the classification of housekeeping proteins.

Furthermore, we show the trends of these results for various values of the threshold parameter t in Figure 5.11. No matter how t is chosen, the range of the percentages of HK interacting with non-HK proteins spans almost all possible values depending on which PPI and expression set is chosen. This shows that our specific choice of t is irrelevant for the our results.

Putting prior results into perspective: random expectations

In this section we have so far shown, that prior results are not always reproducible for all PPI networks and all expression datasets. In fact, the conclusions drawn in those studies depend on the networks and expression sets chosen for analysis.

In the following, we are going one step further, and demonstrate that the ranges of results achieved for the various combinations of PPI networks and expression datasets depend mostly on the network's degree distribution. Especially, the exact assignments of the housekeeping and tissue-specific protein classes to proteins in the network are (mostly) irrelevant for the reported results.

Expected number of interactions Let the PPI graph be given by $G = (V, E)$ with $n = |V|$ and $m = |E|$. Further, let $K \subset V$ be a random subset of fixed size

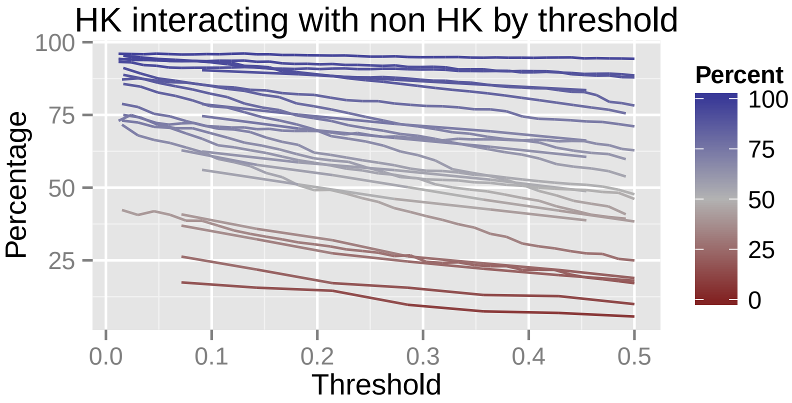


Figure 5.11: The percentage of HK that interact directly with non HK plotted for different choices of the threshold parameter t. Each PPI and expression set combination is drawn as a single line.

$k = |K|$. For each node $u \in V$ and given its degree d_u, we define the random variable X_u to represent the number of outgoing edges from u that connect to any node in K. Note that the value of X_u is bound by the degree d_u of u such that:

$$P(0 \leq X_u \leq d_u) = 1 \tag{5.1}$$

Hypergeometric distribution We assume all edges (u, v) to be equally likely for all v. This corresponds to a random graph model. Therefore, the probability mass function for $P(X_u = i)$ is given by the *hypergeometric distribution*, which is used when sampling from a finite population without replacement. When sampling from a population of size N with a total of M successes, then the probability of drawing exactly m successes in a sample of size n is given by:

$$P(X = m) = \frac{\binom{M}{m}\binom{N-M}{n-m}}{\binom{N}{n}} \tag{5.2}$$

This equation is the probability mass function of the *hypergeometric distribution*.

In our case we are sampling all possible edges for a node u, thus the population size is given by the number of nodes $n = |V|$ or, when explicitly excluding self-loops, by $n - 1$. We call a draw successful, if the edge connects to the subset $K \subset V$, therefore the number of successes in the population is given by $k = |K|$ and the number of draws is given by the node's degree d_u. We are interested in the probability that there is any edge (u, v) connecting to K, i.e., the probability that at least one edge from u connects to a node in K. This probability is given by:

$$P(X_u \geq 1) = 1 - P(X_u = 0) \tag{5.3}$$

the right side of which does not require the evaluation of the cumulative distribution function.

Note that this model depends on the degree of node u, but does not take the degree distribution into account for the set K, since all target nodes are assumed to be equally likely. This model is therefore not completely accurate, nevertheless we show that it can be used to predict the outcome of the analysis of the previous section.

We apply this model to calculate the expected number of tissue specific proteins interacting with housekeeping proteins, as well as the expected number of housekeeping proteins interacting with non-housekeeping proteins.

These problems can be formalized in the following manner: For two randomly chosen subsets $K \subset V$ and $L \subset V$ with $K \cap L = \emptyset$ and fixed sizes $k = |K|$ and $l = |L|$, what is the expected number of nodes in L that interact with nodes in K? Using the previously stated probability of a single node u to interact with at least one node from K, the expected number of nodes in L that interact with at least one node in K is then given by:

$$\left(\sum_{u \in V} P(X_u \geq 1) \right) \cdot \frac{l}{n} \tag{5.4}$$

Note that this sum further assumes, that the probability of each node connecting into K is independent from each other, i.e., the degrees of the nodes in K are again disregarded. However, we will show in the following that this model still results in good predictions.

Tissue Specific proteins interacting with housekeeping proteins We set K to be the set of all housekeeping proteins and L respectively to the set of tissue specific proteins. We then use equation 5.4 to calculate the expected number of tissue-specific proteins that interact with housekeeping proteins.

Note that according to the above formulation, the calculated expected number of proteins will only depend on the degree distribution of the PPI network and on the number of proteins inside the protein classes (i.e., the number of tissue specific and the number of housekeeping proteins). Particularly, it does not depend on the actual mapping of these labels to nodes in the network or the actual network structure.

We use the same threshold $t = 15\%$ as used before to determine the sizes of K and L for each expression dataset and then use the degree distribution of each PPI network to calculate the expected number nodes in L that interact with nodes in K. We plot the resulting percentages for all combinations of PPI networks and expression datasets in Figure 5.12. The results are strikingly similar to the results found by Bossi and Lehner, that we reproduced before (Figure 5.8).

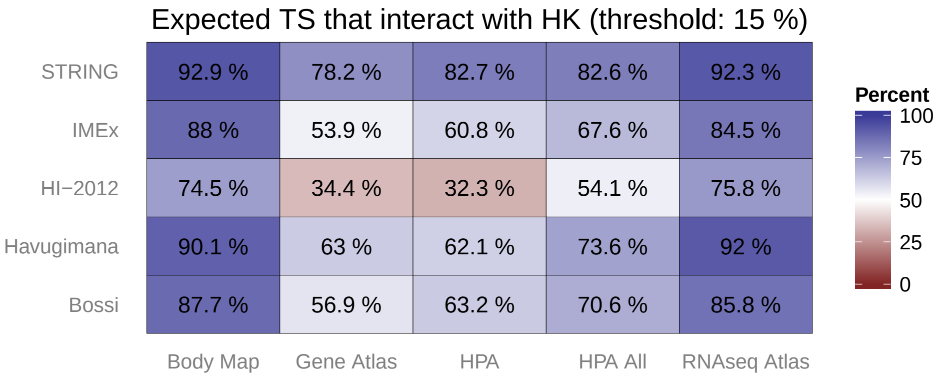


Figure 5.12: For each combination of PPI networks and expression data sets the expected percentage of tissue specific (TS) proteins that interact directly with housekeeping (HK) proteins is given for a threshold of $t = 15\%$ used for the classification into tissue specific and housekeeping. The expectation is calculated by only taking the network's degree distribution and the number of tissue specific and housekeeping proteins into account.

To show the similarity between the expected and the actually observed data, we plot these two results against each other (figure 5.13). This figure compares the actually observed percentages with the percentages that are predicted by our model. Each combination of PPI networks and expression datasets is represented by a single point. The actual percentage of TS interacting with HK is used as the x-coordinate and the predicted, expected percentage is the y-coordinate. Given a perfect model, the points would all lie exactly on the diagonal and give rise to a perfect correlation of $\rho = 1.0$. Our simplified model slightly over estimates the interaction percentages, however the trend is still accurately predicted. The Pearson correlation coefficient between the estimated and real values is $\rho = 0.933$ when using a threshold parameter of $t = 15\%$.

We have previously shown that tissue-specific proteins have lower interaction degrees than housekeeping proteins (see above). Our model does not take this into account, but slightly overestimated the percentage of interactions of tissue-specific proteins and housekeeping proteins.

We show how the predicted results change by accounting for the degree distribution of tissue-specific proteins in our prior analysis. Figure 5.14 shows that this leads to a better fit, especially the expected percentages are no longer overall bigger than the actual values. The correlation coefficient increases to $\rho = 0.954$

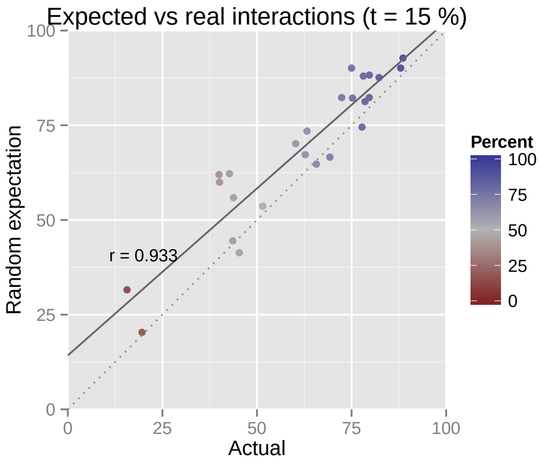


Figure 5.13: The calculated expected percentage of tissue specific proteins interacting with housekeeping proteins (y-axis) are plotted against the actually observed percentage (x-axis). The threshold for classification into tissue specific and housekeeping is set to $t = 15\%$.

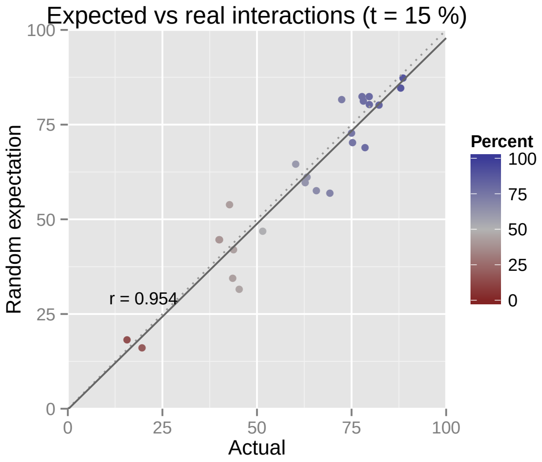


Figure 5.14: The calculated expected percentage of tissue specific proteins interacting with housekeeping proteins (y-axis) are plotted against the actually observed percentage (x-axis). The degree distribution of the tissue specific proteins are taken into account for calculating the probabilities. The threshold for classification into tissue specific and housekeeping is set to $t = 15\%$.

Housekeeping proteins interacting with non-housekeeping proteins We repeat the same analysis for finding the expected percentage of housekeeping proteins that interact with non-housekeeping proteins. The results are very similar to what we found for the interaction between tissue-specific and housekeeping proteins (Figure 5.15). We find that the Pearson correlation is even higher for this case with $\rho = 0.975$. Additionally, the predicted/expected results are closer to the actual results than previously seen for the tissue specific interactions.

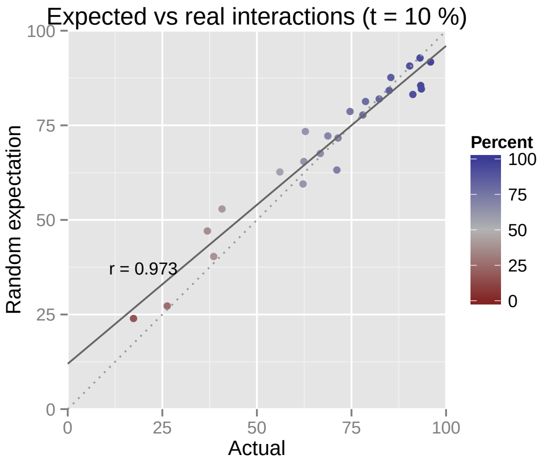


Figure 5.15: The calculated expected percentage of housekeeping proteins interacting with non-housekeeping proteins (y-axis) are plotted against the actually observed percentage (x-axis). The threshold for classification into tissue specific and housekeeping is set to $t = 10\%$.

Centrality of housekeeping genes

Next, we are going to consider the importance of housekeeping and tissue-specific genes in the PPI network topology. We use centrality measures (see 1.3.3) to quantitatively characterize proteins in the PPI networks according to their importance in the network.

Lin *et al.* (2009) [39] analyzed tissue-specific PPI networks and found that housekeeping proteins have significantly higher centrality than other proteins for both the *degree centrality* and *betweenness centrality*. Since the *degree centrality* refers to nothing else but the degree of a node, this analysis result is identical to the first result found by Bossi and Lehner, which we showed earlier in this chapter. We have already evaluated this result and also found that housekeeping proteins have significantly higher interaction degrees for most PPIs and expression datasets. Therefore, we will focus on the betweenness centrality of housekeeping and tissue-specific proteins in this section.

First, we consider the betweenness of housekeeping proteins in the full PPI networks. Throughout this analysis, we will use a threshold of $t = 10\%$ for classification of genes into housekeeping and tissue-specific. Given the betweenness centrality of all proteins in a network, we calculate the differences of the mean and their significances for housekeeping proteins versus randomly sampled proteins. We closely follow the analysis procedure by Lin *et al.* (2009). We then calculate the *z-score* of the mean of housekeeping proteins within the sets of randomly sampled proteins. The *z-score* is defined as the distance from the mean in units of the standard deviation. Therefore a *z-score* over 1.96 or below -1.96 would signify a significant difference with $p = 0.05$.

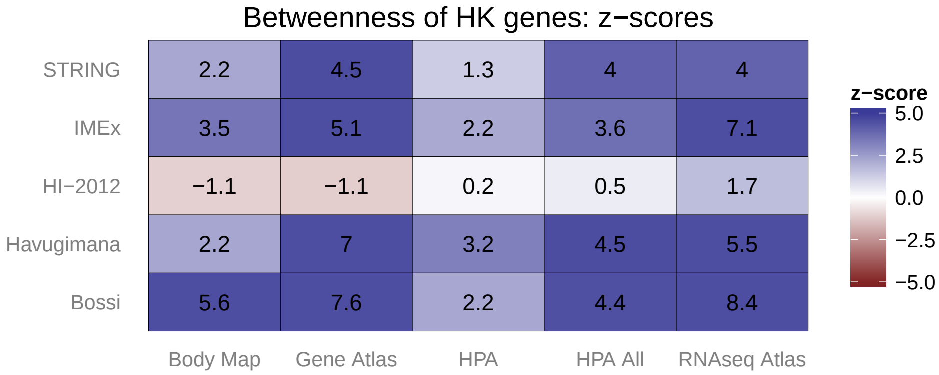


Figure 5.16: The *z-scores* of the betweenness of housekeeping proteins compared to randomly sampled sets of proteins from the whole population of proteins.

We find that housekeeping proteins have significantly higher betweenness centralities for many of the PPI and expression dataset combinations (19 out of 25 with $z > 1.96\sigma$ and thus $p < 0.05$, see figure 5.16). The remaining $6/25$ combinations are all lie within the 95% confidence intervals, and only two (*HI-2012* combined with the *Illumina Body Map* or *Gene Atlas*) have z-scores smaller than zero.

For tissue-specific proteins, we find that all z-scores are negative (see figure 5.17). However, only $9/25$ of combinations result in significantly smaller betweenness centrality scores ($z < -1.96 \Rightarrow p < 0.05$). Therefore, most cases show no significant differences, despite the apparent trend.

We have so far analyzed the betweenness centralities of the whole PPI network. Next, we will analyze the betweenness scores of housekeeping and tissue-specific proteins in all tissue-specific subgraphs. We now have to consider the betweenness scores of proteins for each tissue, each of which generates a different subgraph. We therefore show the ranges of resulting *z-scores* and their mean (see Figure 5.18). Note that this mean does not necessarily have any statistical meaning, but simply helps to demonstrate the distribution of z-scores. We find for

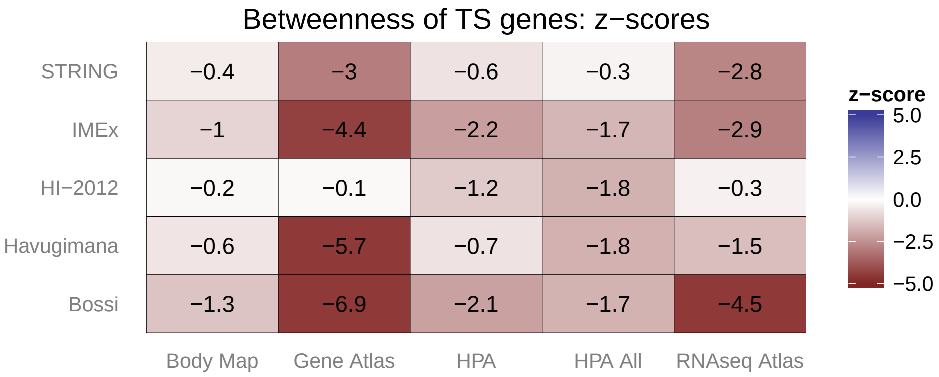


Figure 5.17: The *z-scores* of the betweenness of tissue-specific proteins compared to randomly sampled sets of proteins from the whole population of proteins.

housekeeping genes (Figure 5.18) that a majority of z-scores is bigger than zero, yet we observe less overall significant differences. Merely 6 out of 25 PPI and expression set combinations yield exclusively significant differences in betweenness scores in all of their specific subnetworks (the minimum of the range is ≥ 1.96).

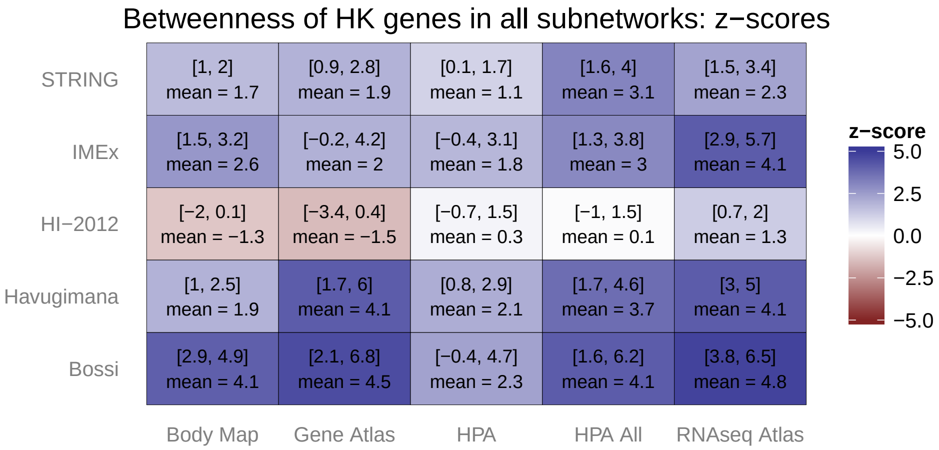
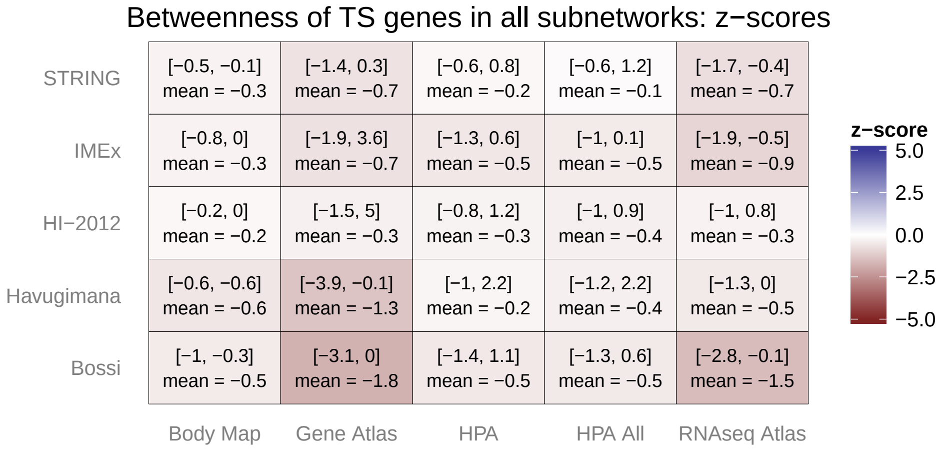


Figure 5.18: The range and mean of *z-scores* of the betweenness of housekeeping proteins compared to randomly sampled sets of proteins for the specific subnetworks for all tissues.

The betweenness centralities of tissue-specific proteins in the tissue-specific subnetworks show little deviation from the means of randomly sampled proteins (see Figure 5.19). In all but three cases, none of the z-scores are significant. In the other three cases, only a subset of the subgraphs have significantly smaller betweenness centralities.



	Body Map	Gene Atlas	HPA	HPA All	RNAseq Atlas
STRING	[−0.5, −0.1] mean = −0.3	[−1.4, 0.3] mean = −0.7	[−0.6, 0.8] mean = −0.2	[−0.6, 1.2] mean = −0.1	[−1.7, −0.4] mean = −0.7
IMEx	[−0.8, 0] mean = −0.3	[−1.9, 3.6] mean = −0.7	[−1.3, 0.6] mean = −0.5	[−1, 0.1] mean = −0.5	[−1.9, −0.5] mean = −0.9
HI−2012	[−0.2, 0] mean = −0.2	[−1.5, 5] mean = −0.3	[−0.8, 1.2] mean = −0.3	[−1, 0.9] mean = −0.4	[−1, 0.8] mean = −0.3
Havugimana	[−0.6, −0.6] mean = −0.6	[−3.9, −0.1] mean = −1.3	[−1, 2.2] mean = −0.2	[−1.2, 2.2] mean = −0.4	[−1.3, 0] mean = −0.5
Bossi	[−1, −0.3] mean = −0.5	[−3.1, 0] mean = −1.8	[−1.4, 1.1] mean = −0.5	[−1.3, 0.6] mean = −0.5	[−2.8, −0.1] mean = −1.5

Figure 5.19: The range and mean of *z-scores* of the betweenness of tissue-specific proteins compared to randomly sampled sets of proteins for the specific subnetworks for all tissues.

Overall, we observe in accordance with the study by Lin *et al.* (2009), that housekeeping proteins tend to have higher betweenness centralities than randomly selected proteins. Additionally, tissue-specific proteins tend to have smaller betweenness centralities than observed by random. However, these results are not significant for many combinations of PPIs and expression datasets.

5.2.2 Discussion

Previous results regarding the interactions between tissue-specific and housekeeping proteins have been in disagreement [10] [21] [22]. These studies have thus come to opposing conclusions about the role of tissue-specific proteins and their interactions in the specialization of cells. However, these studies have only looked at one PPI network and used two different expression datasets for their analysis. We have shown that the results depend in large part on the PPI and expression dataset analyzed. Moreover, we demonstrated that the contradicting results about the interaction partners of TS and HK proteins do not stem from the actual identity of tissue-specific and housekeeping proteins in the network, but are predictable using a simple random model. We argue that this demonstrates an impossibility to reject the null hypothesis that there is a significant difference between the interactions of TS or HK proteins and randomly chosen proteins.

Furthermore, we have shown that the results by Lin *et al.* (2009) [39] are reproducible for only a subset of all PPI networks and expression datasets. Even though only some cases result in significant higher betweenness centrality for housekeeping proteins, we observe a trend of housekeeping proteins taking more central positions in the network. Tissue-specific proteins tend to be less central than expected by random. However, for the tissue-specific subnetworks, TS proteins become less distinguishable from randomly selected proteins. Interestingly, the betweenness centrality of HK proteins remains more conserved in tissue-specific subnetworks.

Our results put prior biological conclusions into perspective. We argue, that the previously observed properties of tissue-specific and housekeeping proteins and the conclusions drawn by those studies do not necessarily hold a biological meaning, since the choice of data sources largely determines the outcome of the analysis.

5.3 Gained importance of tissue-specific proteins

So far we have found that tissue-specific proteins show a tendency towards less important roles in the interaction networks. However, are there proteins that have gained in importance by evolving to be tissue-specific? If yes, which biological processes are these proteins involved in? To find out, we analyze the centrality of proteins in the PPI graphs and compare it with all the according tissue specific subgraphs. We continue to use a classification threshold of $t = 10\%$ for the classification of proteins into HK and TS.

5.3.1 Analysis

First we identify tissue-specific proteins that have a higher betweenness centrality score in a tissue-specific subnetworks than in the full PPI network. For these we calculate the factor of increase in betweenness. We do this for all combinations of PPI networks and expression datasets. In order to achieve a higher reliability of our results, we merge the proteins identified in all those combinations and keep only those resulting from at least two different PPI networks and at least two different expression data sets. Therefore, results that stem from only one PPI but are consistent across different expression datasets will be filtered out, since at least two different PPI networks have to support the result.

This procedure identifies 122 tissue-specific proteins that show an increase in their betweenness centrality score in the tissue-specific subnetworks compared to the global PPI network. We further reduce this number by considering only those proteins for which the betweenness increases by at least a factor of two, which leaves a total of 39 proteins.

We use functional annotation and enrichment analysis on the previously identified proteins to determine which biological processes these proteins are involved in, and if multiple of these proteins are involved in similar processes. To facilitate this analysis we use *DAVID (Database for Annotation, Visualization and Integrated Discovery)* [34] [35] and concentrate on the *GO (Gene Ontology)* terms of the *biological process (BP)* namespace (see also Section 3.6).

GO Term	Term Name	Gene Count	P-Value
GO:0032501	multicellular organismal process	20	5.93E-03
GO:0048731	system development	16	4.33E-04
GO:0048856	anatomical structure development	16	1.05E-03
GO:0048513	organ development	13	1.12E-03
GO:0007275	multicellular organismal development	16	3.86E-03
GO:0032502	developmental process	16	9.65E-03
GO:0006810	transport	14	1.30E-02
GO:0051234	establishment of localization	14	1.41E-02
GO:0051179	localization	14	3.57E-02

Table 5.3: Significantly enriched GO-Terms in tissue-specific proteins with increased betweenness centrality in their respective tissue-specific subnetwork PPIs. The *Gene Count* column gives the number of genes that are enriched with the given GO Term as returned by *DAVID*.

5.3.2 Results

DAVID identifies 21 GO-terms that are significantly enriched ($p < 0.05$), nine of which have a gene count over 30% of the total number of genes used in the functional annotation (see Table 5.3). We group these terms into two groups. The terms within these groups have most genes in common, where the first group shares 13 genes in all its terms and the terms of the second group share all 14 genes.

The first group consists of terms relating to the developmental process for anatomical structure, organ development and multicellular organismal development. The terms of the second group relate to protein transport and protein localization in the cell.

All significant GO terms are printed in the appendix in table A.4.

5.3.3 Discussion

We have identified two groups of tissue-specific genes which have more important positions in their tissue-specific networks than in the global PPI networks. The first group of genes is functionally enriched in developmental processes, which might hint that our method successfully identifies genes which hold important roles in multicellular specialization. These genes may have an important role in the evolution towards multicellular organisms, since these genes profit from cellular specialization by taking a more central role in the cellular interactions.

5.4 Graph clustering for identification of functional modules

5.4.1 Different clustering algorithms

We use the *Parallel Label Propagation (PLP)* and the *Parallel Louvain Method (PLM)* for clustering in all PPI networks (see Sections 2.4 and 3.7.2).

In Figure 5.20, we show the sizes of the clusters as the result of the different clustering algorithms and parameters for the *STRING* PPI. The *PLP* algorithm finds a large cluster of over 12 thousand proteins, which is a majority of all proteins in the network.

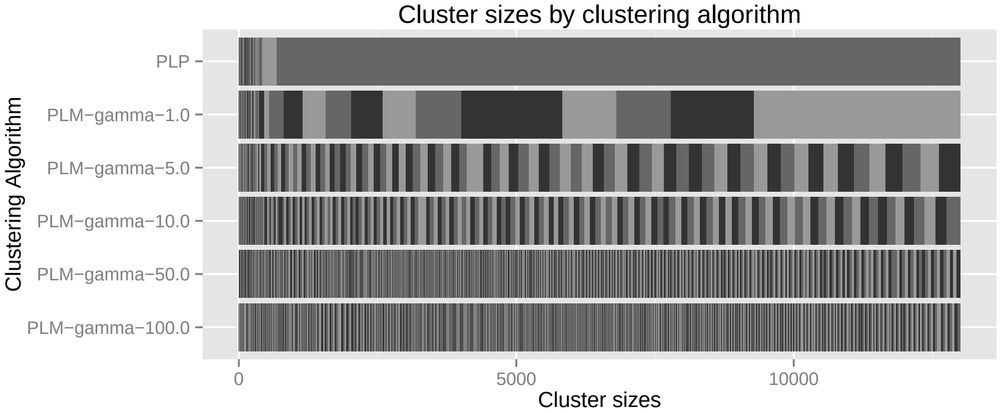


Figure 5.20: The sizes of clusters as identified by the different clustering algorithms and parameter settings for the *STRING* PPI.

When using the *PLM* algorithm for clustering, the sizes of the clusters is determined by the gamma parameter. For the default value of gamma=1.0, the modularity of the clustering is maximized. For this parameter setting, we find that, the largest cluster still contains over 3000 proteins and eight more clusters of size bigger than 500 are returned. Both algorithms *PLP* and *PLM* with default parameter settings return clusters which are too big for our purpose, since our goal is to identify functional meaningful clusters/modules. In our opinion, a cluster that contains a large fraction of all proteins, most likely can not lead to any specific, meaningful results. Since the *PLM* method is hierarchical in nature, we choose a larger value of gamma in order to get more fine grained clusters. Figure 5.20 shows an overview of the achieved cluster sizes for the different parameter values $\gamma \in \{1.0, 5.0, 10.0, 50.0, 100.0\}$.

For further analysis of clusters, we choose and restrict ourselves to the *PLM* algorithm with parameter $\gamma = 50.0$, because for this choice there are no clusters containing more than 100 proteins (thus all clusters have size $< 1\%$ of proteins) and fewer than 10% of proteins are contained in clusters with sizes smaller than 4. The latter being of importance, since we only use clusters with size greater than 3 within the upcoming analysis.

5.4.2 Identifying functional modules

Functional modules in the PPI graph are clusters of proteins which are functionally or semantically related. We use the clustering algorithms implemented in *NetworKit* to find clusters and then use the *BPScore* scoring method to evaluate the functional and semantic similarity of the proteins in each cluster. In the following we show the analysis methods and results for clustering in the full PPIs and compare those to clusters obtained in tissue-specific subnetworks.

Global PPIs

First, we use the *PLM* clustering algorithm with $\gamma = 50.0$ on the different PPI networks. As described in Section 3.7.2, we consider the top 20% of clusters ranked by their modularity as “good” clusters. In Table 5.4 we show the clustering results for the five PPI networks.

We score all resulting clusters using the *BPScore* explained in Section 3.6. Remember that this score states how much more similar (in terms of semantic similarity using GO-Terms) proteins within a given cluster are compared to all proteins. A positive value means that the proteins are more similar, while a negative score means that the proteins are less similar to each other than they are to the set of all proteins.

We apply a student’s t-test to test for significant differences of *BPScores* between the top 20% of clusters (i.e., the “good” clusters) and the remaining clusters. We observe that the top 20% of clusters score significantly higher in their *BPScore*

PPI	# Cl.	Avg. size	BP(+20%)	BP(−80%)	greater?	P-value
Bossi	643	13.78	0.27	0.15	TRUE	$< 1.7 \cdot 10^{-18}$
STRING	860	16.11	0.28	0.14	TRUE	$< 6.2 \cdot 10^{-24}$
IMEx	936	10.80	0.18	0.09	TRUE	$< 5.3 \cdot 10^{-13}$
Havugimana	308	7.42	0.26	0.11	TRUE	$< 4.3 \cdot 10^{-7}$
HI-2012	459	8.21	0.06	0.08	FALSE	0.29

Table 5.4: Result of clustering with *PLM* using $\gamma = 50.0$. Clusters are ranked according to their modularity and scored with *BPScore*. The columns show (2) the number of clusters, (3) the average size per cluster, (4) the average *BPScore* of the top 20% clusters ranked by their modularity, (5) the average *BPScore* of the remaining bottom 80%, (6) whether the "good" top 20% clusters have a higher average *BPScore* and finally (7) the significance of the difference of means of (4) and (5) using a t-test.

for $4/5$ PPIs. For the *HI-2012* the "good" clusters do not score higher, however no significant p-value is reached for this result.

Overall, these results illustrate that by using this clustering algorithm and by choosing the high modularity clusters, we can identify potential functional modules.

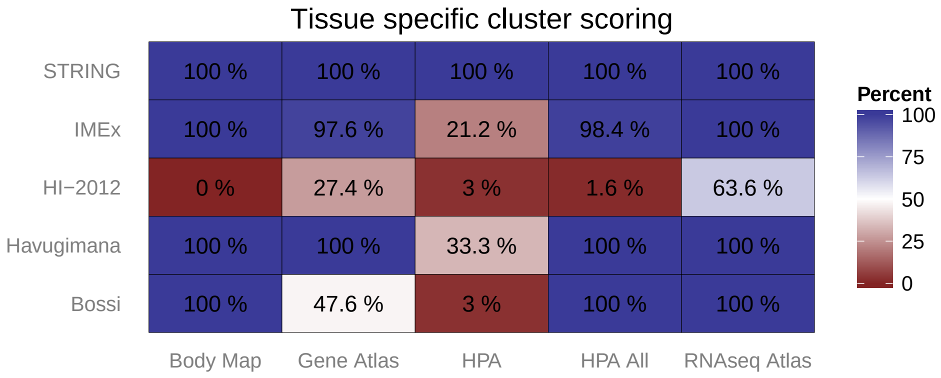
Tissue-specific networks

We run the same clustering algorithm on all tissue-specific subnetworks for all combinations of PPIs and expression datasets. For each tissue-specific subnetwork we test whether the top 20% of clusters have higher *BPScore*s than the remaining clusters using the t-test. We summarize the results in Figure 5.21, where we show the percentage of tissues in which the "good" clusters have significant ($p < 0.05$) higher *BPScore*s.

We observe similar results as seen for the whole PPIs. Overall, in almost all tissues ($> 97.6\%$), the top clusters score significantly higher than bottom 80% for a majority of PPI and expression data combinations $(16/25)$. The combinations where this does not remain true are mostly restricted to the *HI-2012* network and to the *HPA* expression dataset. This agrees with the results on the whole PPI networks above.

Tissue-specific versus full PPI networks

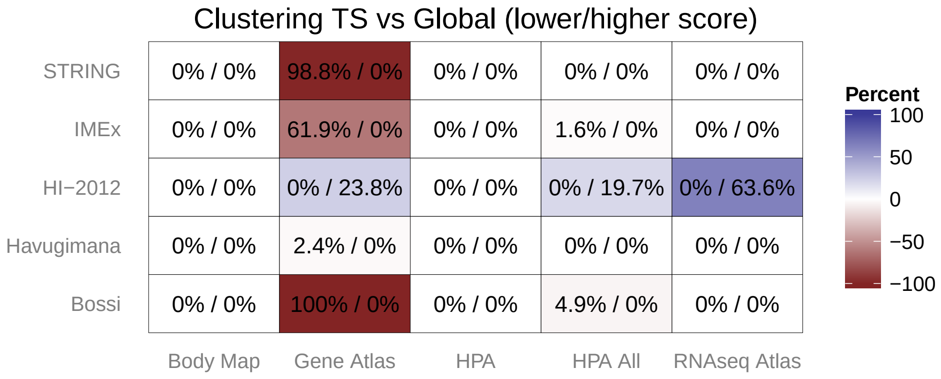
Next, we investigate whether clustering in tissue-specific subnetworks results in more functionally related (higher scoring) clusters compared to the clustering of the full PPI networks. For a given pair of a PPI and an expression dataset, we create all the tissue-specific subnetworks and the full PPI network. We run the



	Body Map	Gene Atlas	HPA	HPA All	RNAseq Atlas
STRING	100 %	100 %	100 %	100 %	100 %
IMEx	100 %	97.6 %	21.2 %	98.4 %	100 %
HI−2012	0 %	27.4 %	3 %	1.6 %	63.6 %
Havugimana	100 %	100 %	33.3 %	100 %	100 %
Bossi	100 %	47.6 %	3 %	100 %	100 %

Figure 5.21: For each combination of PPIs and expression datasets, this figure shows the percentage of tissues for which the top 20% clusters have significant higher *BPScore*s ($p < 0.05$) than the other 80% of clusters.

PLM clustering algorithm and then take the top 20% (ranked by the modularity) of clusters in each network. We then compare the *BPScore* of the clusters of each tissue-specific subnetwork with the clusters of the full PPI network. We use a t-test to determine if the clusters in the specific subnetworks score significantly lower or higher than the clusters of the full PPI network. Figure 5.22 gives an overview of the results. This figure shows the percentage of tissues in which the clusters exhibit significantly lower/higher *BPScore*s ($p < 0.05$).



	Body Map	Gene Atlas	HPA	HPA All	RNAseq Atlas
STRING	0% / 0%	98.8% / 0%	0% / 0%	0% / 0%	0% / 0%
IMEx	0% / 0%	61.9% / 0%	0% / 0%	1.6% / 0%	0% / 0%
HI−2012	0% / 0%	0% / 23.8%	0% / 0%	0% / 19.7%	0% / 63.6%
Havugimana	0% / 0%	2.4% / 0%	0% / 0%	0% / 0%	0% / 0%
Bossi	0% / 0%	100% / 0%	0% / 0%	4.9% / 0%	0% / 0%

Figure 5.22: For each combination of PPIs and expression datasets, this figure shows the percentages of tissues for which the top 20% clusters in the tissue-specific subnetworks have significant lower/higher *BPScore*s ($p < 0.05$) than the top 20% of clusters in the corresponding full PPI graph.

For a majority of cases $(16/25)$, we observe no significant differences in any tissue to either direction $(0\%/0\%)$. For the *Gene Atlas*, we find that the clusters identified in many tissue-specific subnetworks exhibit significantly lower scores than clusters found in the according full PPIs. Only for the *HI-2012* PPI, the clustering identifies higher scoring clusters in some of the tissue-specific subnetworks.

Edge weighting

We defined two methods for assigning weights to the edges of the PPI based on the expression data of the interacting proteins (see Section 3.7.4). Here, we evaluate whether clustering the weighted graphs yield better results compared to using the unweighted, full PPI graph.

For this analysis, we again choose the top 20% (ranked by modularity) of clusters in the clusterings of both the edge-weighted and the global PPI graphs. We compare the *BPScore*s of the clusters in these two graphs and run a student's t-test in order to test for significant differences in the mean of the cluster's scores. We run the clustering, scoring and tests for all combinations of PPIs and expression datasets.

Correlation weight Using the correlation weights, we find no significant differences in means for all of the tissue-specific PPIs. Therefore, this method does not produce any better results than using the unweighted, full PPI graphs. We show the full results including all p-values in the appendix in table A.5.

Co-expression weights We observe similar results for the co-expression weighted networks. For all but two combinations of PPIs and expression data sets, we find no significant differences between the means of the *BPScores* of the clusters. For two instances however, we find significant differences between the *BPScores* of the clusters in the weighted graph compared to the corresponding non-weighted PPI graph. We show these two instances in Table 5.5. We further observe, that for one of the instances the weighted graph has a significantly higher score, while for the second instance a significantly lower score is reached. A table of all results is printed in table A.6.

PPI	Expression	BP(edge-graph)	BP(global)	P-value
STRING	Gene Atlas	0.26	0.29	0.033
HI-2012	Gene Atlas	0.15	0.07	0.050

Table 5.5: Co-expression weighted PPIs for which we find significantly different BP-Scores of the clusters in the weighted graph compared to the corresponding non-weighted (global) PPI graph.

5.4.3 Discussion

We have used clustering algorithms with the goal of identifying functional modules in PPI networks. We evaluated the clusters using the GO-Term based *BP-Score* scoring method. Contrary to our expectation, we found that clustering in the more specific tissue-specific subnetworks did not generally result in higher scoring clusters. Thus, we were unable to identify more specific functional and semantically related modules in the specific subgraphs with our approach.

We furthermore tried a hybrid method, adding weights to edges in the PPI graphs based on the expression patterns of the interacting proteins. For this method as well, we were also unable to achieve better clustering results.

Overall, we were not able to beat the clustering results achieved by purely looking at the full PPI graphs. Adding tissue-specific expression data, did not yield any benefit.

Chapter 6

Conclusion and future work

In this chapter, we summarize our work with a focus on the results achieved and then outline future work.

6.1 Summary and conclusions

In this thesis, we analyzed human tissue and cell-type specific protein-protein interaction networks. For our analysis, we chose multiple different PPIs and protein expression datasets. In order to create tissue-specific PPIs, we combined those PPIs with the protein expression data to create a total of 25 different tissue-specific PPIs. Each of these was further subdivided into subgraphs for each tissue or cell-type.

We implemented an analysis pipeline for importing and converting all source datasets, and the subsequent automated analysis of the tissue-specific PPIs. We calculated the statistical properties of the expression datasets and the PPI networks. Among others, we demonstrated that the degree distributions of the PPIs most closely follow a power-law distribution. Furthermore, basic properties of the tissue-specific PPI networks were laid out.

For the analysis of tissue-specific PPI graphs, we developed and implemented custom versions of common analysis algorithms. These operate directly on our tissue-specific graph representation and run the analysis on the subgraphs for all tissues simultaneously. We adapted and implemented algorithms for the local clustering coefficient, the betweenness centrality and a version of the *Parallel Label Propagation (PLP)* clustering algorithm. We demonstrated substantial improvements of runtime for the clustering coefficients computation and minor improvements in the parallel runtime of the betweenness centrality. Our adaption of the PLP algorithm performed worse than the alternative of running the original algorithm succinctly on separately created graphs. Hence, we could report improvements only for two but not all algorithms.

We then re-evaluated the results of multiple prior studies on our collection of 25 tissue-specific PPI networks. We demonstrated, that the results of Bossi and Lehner (2009), Emig *et al.* (2011) and Lin *et al.* (2009) all depend on the exact combination of PPI network and expression dataset chosen for the analysis. We could not reproduce their results for all of these combinations. Furthermore, we gave a statistical model for the interactions between two randomly chosen groups or proteins in PPI graphs and showed that this model predicts the results from Bossi and Lehner to close accuracy. We conclude, that a null hypothesis of no significant differences between tissue-specific/housekeeping and randomly chosen proteins can not be rejected. Therefore, we argue that the conclusions drawn in their study are not necessarily biologically meaningful.

Next, we analyzed proteins in tissue-specific subnetworks and looked at those proteins which gained in centrality in the subgraphs compared to their position in the full PPI networks. We found that the proteins we identified with our method are functionally enriched in GO-Terms relating to developmental processes. We conclude that this method might prove useful in identification of proteins which have important roles in cellular-specialization and the developmental process.

Finally, we used clustering and community-detection algorithms from the *NetworKit* toolkit to identify tissue-specific functional modules/clusters. We ran the clustering methods on the full PPI graphs, all tissue-specific subnetworks and on weighted graphs, which we constructed by adding edge weights to PPIs based on the expression profiles of the interacting proteins. We hoped to be able to identify clusters of more specific and more functionally related proteins in the tissue-specific subnetworks or the weighted graphs. However, for neither of these approaches, were we able to demonstrate better results than those we achieved by clustering the full PPI graphs.

6.2 Future work

In this final section, we elaborate on possible extensions and future directions of the research conducted for this thesis.

For the analysis of gained centrality of proteins in tissue-specific networks, we used the betweenness centrality measure. It might be worth exploring other measures of node centrality in the tissue-specific graphs. Comparing the results from these alternative centrality measures might provide further insight into proteins involved in cellular specialization. Furthermore, algorithms to calculate other centrality measures on the tissue-specific graph representation could be developed, implemented and evaluated.

The *NetworKit* toolkit is still under active development, and since the start of this thesis additional clustering algorithms have been implemented into its framework. It could be interesting to run those algorithms on the tissue-specific graphs, which might yield better identification of functional modules in these graphs.

We observed that prior results and conclusions depend on the datasets which are used for the analysis. Current PPIs and expression datasets do not yet seem to agree with all their proposed protein interactions and protein expression profiles. It seems valuable to extend the analysis pipeline by adding implementations for the reproduction and automated re-evaluation of more previous research results. Adding more PPIs and expression datasets whenever new, more reliable data becomes available, would enable constant re-evaluation of prior results and conclusions. It might turn out to be helpful for the community, if this analysis pipeline and its results would be made publicly available in a format which enables easy navigation of up-to-date version of all results.

Appendix A

Appendix

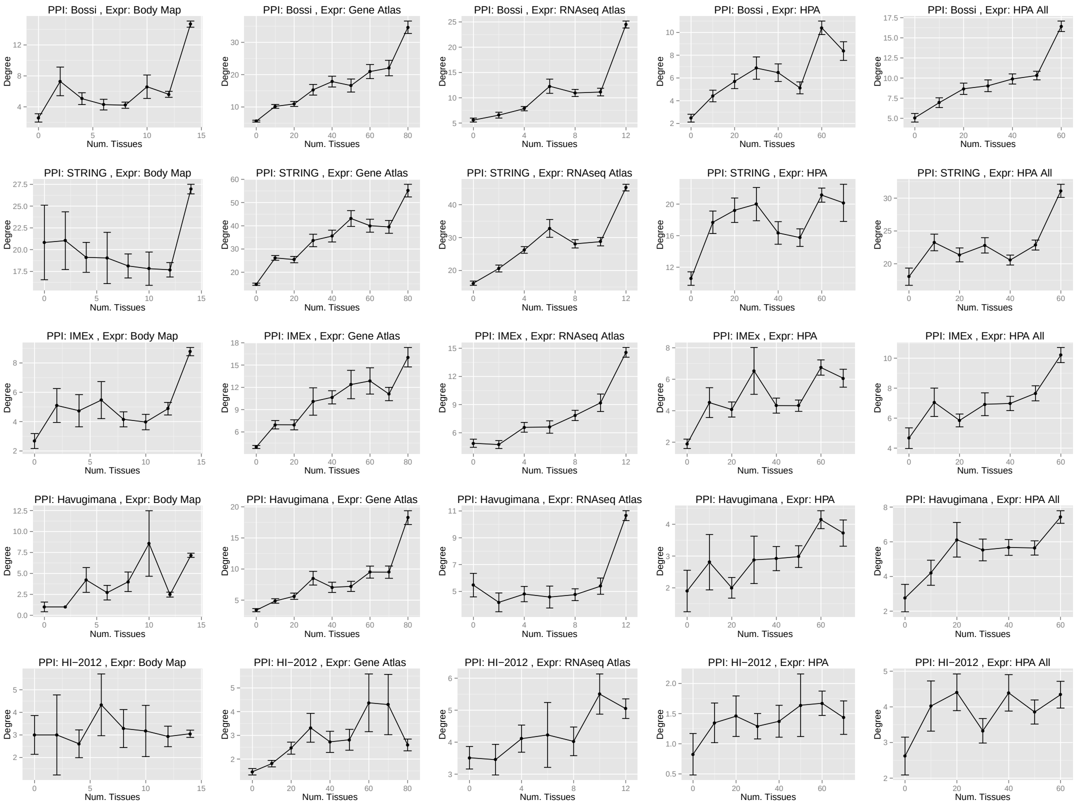


Figure A.1: For all PPIs and expression datasets, this shows the mean protein interaction degree for varying tissue specificity. The x-Axis represents the number of tissues a protein is expressed in. The error bars are one standard error.

PPI	Expression	NetworKit	Neighbor Comb.	Tissue expr.
STRING	Gene Atlas	118.76 s	8.31 s	2.32 s
STRING	HPA All	25.76 s	6.86 s	0.49 s
STRING	RNAseq Atlas	23.53 s	3.88 s	3.33 s
Bossi	Gene Atlas	16.42 s	3.40 s	0.37 s
IMEx	Gene Atlas	6.65 s	0.95 s	0.07 s
STRING	Body Map	4.48 s	2.16 s	0.31 s
Bossi	HPA All	3.90 s	1.19 s	0.12 s
STRING	HPA	2.93 s	0.87 s	0.06 s
Bossi	RNAseq Atlas	2.26 s	0.89 s	0.25 s
IMEx	HPA All	1.85 s	0.56 s	0.03 s
Bossi	Body Map	0.97 s	0.49 s	0.11 s
IMEx	RNAseq Atlas	0.91 s	0.32 s	0.06 s
Havugimana	Gene Atlas	0.86 s	0.35 s	0.03 s
HI-2012	Gene Atlas	0.38 s	0.10 s	0.01 s
Bossi	HPA	0.30 s	0.12 s	0.01 s
IMEx	Body Map	0.28 s	0.15 s	0.02 s
IMEx	HPA	0.19 s	0.07 s	0.00 s
Havugimana	HPA All	0.17 s	0.07 s	0.01 s
HI-2012	HPA All	0.14 s	0.05 s	0.00 s
Havugimana	RNAseq Atlas	0.13 s	0.07 s	0.02 s
HI-2012	RNAseq Atlas	0.11 s	0.02 s	0.01 s
Havugimana	Body Map	0.06 s	0.06 s	0.01 s
Havugimana	HPA	0.02 s	0.03 s	0.00 s
HI-2012	Body Map	0.01 s	0.01 s	0.00 s
HI-2012	HPA	0.01 s	0.01 s	0.00 s

Table A.1: Benchmark results for different algorithms to compute the local clustering coefficients for all tissue-specific subnetworks of each combination of PPI and expression dataset.

PPI	Expression	Create Subgraphs	Tissue Vectors
STRING	HPA All	316.33 s	320.72 s
STRING	RNAseq Atlas	111.10 s	131.62 s
STRING	Gene Atlas	106.09 s	67.57 s
STRING	Body Map	103.89 s	92.18 s
IMEx	HPA All	78.73 s	76.43 s
Bossi	HPA All	69.47 s	68.33 s
IMEx	RNAseq Atlas	36.82 s	35.12 s
IMEx	Gene Atlas	33.17 s	20.78 s
Bossi	RNAseq Atlas	33.03 s	32.35 s
Bossi	Gene Atlas	32.73 s	23.13 s
IMEx	Body Map	31.78 s	28.04 s
Bossi	Body Map	27.12 s	24.06 s
STRING	HPA	15.61 s	17.22 s
Havugimana	HPA All	4.71 s	4.88 s
Bossi	HPA	4.71 s	5.06 s
IMEx	HPA	4.62 s	4.97 s
HI-2012	HPA All	4.57 s	4.78 s
Havugimana	Gene Atlas	3.61 s	3.91 s
Havugimana	RNAseq Atlas	2.82 s	2.77 s
HI-2012	RNAseq Atlas	2.44 s	2.35 s
Havugimana	Body Map	2.16 s	2.05 s
HI-2012	Gene Atlas	1.78 s	0.57 s
HI-2012	Body Map	1.43 s	1.40 s
Havugimana	HPA	0.62 s	0.65 s
HI-2012	HPA	0.10 s	0.09

Table A.2: Benchmark results of the two different methods for computing the betweenness centrality for all tissues of the tissue-specific networks. These are the results when running the *OpenMP* parallel version of the algorithms on 4 cores and threads.

PPI	Expression	NetworKit	TS Representation
STRING	Gene Atlas	2.569 s	6.626 s
STRING	HPA All	0.808 s	4.464 s
Bossi	Gene Atlas	0.575 s	1.129 s
IMEx	Gene Atlas	0.411 s	1.027 s
STRING	RNAseq Atlas	0.390 s	0.840 s
Bossi	HPA All	0.236 s	0.864 s
IMEx	HPA All	0.198 s	0.534 s
STRING	HPA	0.153 s	0.585 s
STRING	Body Map	0.122 s	0.204 s
Havugimana	Gene Atlas	0.087 s	0.179 s
Bossi	RNAseq Atlas	0.073 s	0.166 s
HI-2012	Gene Atlas	0.071 s	0.101 s
HI-2012	HPA All	0.064 s	0.111 s
Bossi	HPA	0.061 s	0.152 s
IMEx	RNAseq Atlas	0.054 s	0.091 s
IMEx	HPA	0.047 s	0.089 s
Havugimana	HPA All	0.046 s	0.099 s
Bossi	Body Map	0.044 s	0.066 s
IMEx	Body Map	0.031 s	0.046 s
HI-2012	RNAseq Atlas	0.018 s	0.039 s
Havugimana	HPA	0.018 s	0.016 s
Havugimana	RNAseq Atlas	0.015 s	0.028 s
Havugimana	Body Map	0.011 s	0.011 s
HI-2012	Body Map	0.010 s	0.013 s
HI-2012	HPA	0.010 s	0.007 s

Table A.3: Benchmark results for the adapted *PLP* algorithm working directly on the tissue-specific graph representation versus the original *NetworKit* implementation which is run on each tissue separately.

GO Term	Term Name	Count	P-Value
GO:0048731	system development	16	4.33E-04
GO:0048856	anatomical structure development	16	1.05E-03
GO:0048513	organ development	13	1.12E-03
GO:0044057	regulation of system process	6	1.19E-03
GO:0007155	cell adhesion	8	2.00E-03
GO:0022610	biological adhesion	8	2.02E-03
GO:0007275	multicellular organismal development	16	3.86E-03
GO:0048168	regulation of neuronal synaptic plasticity	3	3.98E-03
GO:0050804	regulation of synaptic transmission	4	5.35E-03
GO:0032501	multicellular organismal process	20	5.93E-03
GO:0051969	regulation of transmission of nerve impulse	4	6.64E-03
GO:0031644	regulation of neurological system process	4	7.41E-03
GO:0032502	developmental process	16	9.65E-03
GO:0051239	regulation of multicellular organismal process	8	9.90E-03
GO:0048167	regulation of synaptic plasticity	3	1.22E-02
GO:0006810	transport	14	1.30E-02
GO:0031099	regeneration	3	1.40E-02
GO:0051234	establishment of localization	14	1.41E-02
GO:0065008	regulation of biological quality	9	3.36E-02
GO:0051179	localization	14	3.57E-02
GO:0015669	gas transport	2	4.62E-02

Table A.4: Significantly enriched GO-Terms in tissue-specific proteins with increased betweenness centrality in their respective tissue-specific subnetwork PPIs. The *Gene Count* column gives the number of genes that are enriched with the given GO Term as returned by *DAVID*.

PPI	Expression	BP(edge-graph)	BP(global)	P-value
Bossi	Body Map	0.26	0.27	0.73
Bossi	Gene Atlas	0.26	0.27	0.79
Bossi	RNAseq Atlas	0.25	0.27	0.20
Bossi	HPA	0.25	0.27	0.43
Bossi	HPA All	0.24	0.26	0.48
STRING	Body Map	0.31	0.31	0.69
STRING	Gene Atlas	0.30	0.29	0.78
STRING	RNAseq Atlas	0.28	0.28	0.95
STRING	HPA	0.33	0.33	0.84
STRING	HPA All	0.28	0.28	0.99
IMEx	Body Map	0.18	0.18	0.91
IMEx	Gene Atlas	0.18	0.19	0.69
IMEx	RNAseq Atlas	0.18	0.19	0.64
IMEx	HPA	0.19	0.19	0.99
IMEx	HPA All	0.15	0.16	0.63
Havugimana	Body Map	0.24	0.22	0.62
Havugimana	Gene Atlas	0.27	0.26	0.91
Havugimana	RNAseq Atlas	0.26	0.27	0.80
Havugimana	HPA	0.36	0.32	0.63
Havugimana	HPA All	0.26	0.28	0.65
HI-2012	Body Map	0.09	0.10	0.79
HI-2012	Gene Atlas	0.07	0.07	0.82
HI-2012	RNAseq Atlas	0.07	0.07	0.94
HI-2012	HPA	0.06	0.13	0.51
HI-2012	HPA All	0.07	0.05	0.46

Table A.5: The mean BPScores of top 20% of clusters identified in the full PPI graph and the graph where edges are weighted with the correlation of the expression of the interacting proteins. The P-values are the significance levels of the two-sided student's t-test testing for a difference in means. None of the means are significantly different.

PPI	Expression	BP(edge-graph)	BP(global)	P-value
Bossi	Body Map	0.26	0.27	0.92
Bossi	Gene Atlas	0.23	0.27	0.08
Bossi	RNAseq Atlas	0.25	0.27	0.21
Bossi	HPA	0.24	0.27	0.42
Bossi	HPA All	0.22	0.26	0.10
STRING	Body Map	0.31	0.31	0.87
STRING	Gene Atlas	0.26	0.29	0.03
STRING	RNAseq Atlas	0.27	0.28	0.65
STRING	HPA	0.31	0.33	0.44
STRING	HPA All	0.27	0.28	0.43
IMEx	Body Map	0.18	0.18	1.00
IMEx	Gene Atlas	0.16	0.19	0.08
IMEx	RNAseq Atlas	0.17	0.19	0.29
IMEx	HPA	0.18	0.19	0.62
IMEx	HPA All	0.14	0.16	0.47
Havugimana	Body Map	0.24	0.22	0.69
Havugimana	Gene Atlas	0.28	0.26	0.72
Havugimana	RNAseq Atlas	0.26	0.27	0.78
Havugimana	HPA	0.33	0.32	0.84
Havugimana	HPA All	0.26	0.28	0.62
HI-2012	Body Map	0.09	0.10	0.80
HI-2012	Gene Atlas	0.15	0.07	0.05
HI-2012	RNAseq Atlas	0.09	0.07	0.29
HI-2012	HPA	0.03	0.13	0.23
HI-2012	HPA All	0.08	0.05	0.22

Table A.6: The mean BPScores of top 20% of clusters identified in the full PPI graph and the graph where edges are weighted with the number of tissues in which the two interacting proteins are btoh expressed in (normalized by the maximum number of tissues each protein is expressed in). The P-values are the significance levels of the two-sided student's t-test testing for a difference in means. None of the means are significantly different.

Bibliography

[1] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74:47–97, Jan 2002. `http://link.aps.org/doi/10.1103/RevModPhys.74.47`.

[2] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. *Molecular Biology of the Cell*. Garland Science, 5th edition, November 2007.

[3] Bruno Aranda, Hagen Blankenburg, Samuel Kerrien, Fiona SL Brinkman, Arnaud Ceol, Emilie Chautard, Jose M Dana, Javier De Las Rivas, Marine Dumousseau, Eugenia Galeota, et al. Psicquic and psiscore: accessing and scoring molecular interactions. *Nature methods*, 8(7):528–529, 2011.

[4] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999. `http://www.sciencemag.org/content/286/5439/509.abstract`.

[5] BY ALBERT-LÁSZLÓ BARABÁSI and Eric Bonabeau. Scale-free. *Scientific American*, 2003.

[6] Ruth Barshir, Omer Basha, Amir Eluk, Ilan Y. Smoly, Alexander Lan, and Esti Yeger-Lotem. The tissuenet database of human tissue protein-protein interactions. *Nucleic Acids Research*, 41(D1):D841–D844, 2013. `http://nar.oxfordjournals.org/content/41/D1/D841.abstract`.

[7] David Binns, Emily Dimmer, Rachael Huntley, Daniel Barrell, Claire O'Donovan, and Rolf Apweiler. Quickgo: a web-based tool for gene ontology searching. *Bioinformatics*, 25(22):3045–3046, 2009. `http://bioinformatics.oxfordjournals.org/content/25/22/3045.abstract`.

[8] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[9] Béla Bollobás. *Random graphs*. Cambridge studies in advanced mathematics ; 73. Cambridge Univ. Press, Cambridge, 2. ed., 5. print. edition, 2008. Previous ed.: London : Academic Press, 1985.

[10] Alice Bossi and Ben Lehner. Tissue specificity and the human protein interaction network. *Molecular systems biology*, 5(1), 2009.

[11] Ulrik Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25:163–177, 2001.

[12] Kevin R Brown and Igor Jurisica. Unequal evolutionary conservation of human protein interactions in interologous networks. *Genome biology*, 8(5):R95, 2007.

[13] Seth Carbon, Amelia Ireland, Christopher J. Mungall, ShengQiang Shu, Brad Marshall, Suzanna Lewis, the AmiGO Hub, and the Web Presence Working Group. Amigo: online access to ontology and annotation data. *Bioinformatics*, 25(2):288–289, 2009. `http://bioinformatics.oxfordjournals.org/content/25/2/288.abstract`.

[14] John C Castle, Christopher D Armour, Martin Löwer, David Haynor, Matthew Biery, Heather Bouzek, Ronghua Chen, Stuart Jackson, Jason M Johnson, Carol A Rohl, et al. Digital genome-wide ncrna expression, including snornas, across 11 human tissues using polya-neutral amplification. *PloS one*, 5(7):e11779, 2010.

[15] Cheng-Wei Chang, Wei-Chung Cheng, Chaang-Ray Chen, Wun-Yi Shu, Min-Lung Tsai, Ching-Lung Huang, and Ian C. Hsu. Identification of human housekeeping genes and tissue-selective genes by microarray meta-analysis. *PLoS ONE*, 6(7):e22859, 07 2011. `http://dx.doi.org/10.1371%2Fjournal.pone.0022859`.

[16] Emilie Chautard, Marie Fatoux-Ardore, Lionel Ballut, Nicolas Thierry-Mieg, and Sylvie Ricard-Blum. Matrixdb, the extracellular matrix interaction database. *Nucleic acids research*, 39(suppl 1):D235–D240, 2011.

[17] Gang Chen and Jianxin Wang. Identifying functional modules in tissue specific protein interaction network. In *Bioinformatics and Biomedicine Workshops (BIBMW), 2012 IEEE International Conference on*, pages 581–586, 2012.

[18] A. Clauset, C. Shalizi, and M. Newman. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703, 2009. `http://epubs.siam.org/doi/abs/10.1137/070710111`.

[19] Gene Ontology Consortium. The gene ontology (go) database and informatics resource. *Nucleic Acids Research*, 32(suppl 1):D258–D261, 2004. `http://nar.oxfordjournals.org/content/32/suppl_1/D258.abstract`.

[20] UniProt Consortium et al. Ongoing and future developments at the universal protein resource. *Nucleic acids research*, 39(suppl 1):D214–D219, 2011.

[21] Dorothea Emig and Mario Albrecht. Tissue-specific proteins and functional implications. *Journal of proteome research*, 10(4):1893–1903, 2011.

[22] Dorothea Emig, Tim Kacprowski, and Mario Albrecht. Measuring and analyzing tissue specificity of human genes and protein complexes. *EURASIP Journal on Bioinformatics and Systems Biology*, 2011(1):1–6, 2011.

[23] Paul Erdős and Alfréd Rényi. On random graphs. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.

[24] Paul Erdős and Alfréd Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5:17–61, 1960.

[25] Andrea Franceschini, Damian Szklarczyk, Sune Frankild, Michael Kuhn, Milan Simonovic, Alexander Roth, Jianyi Lin, Pablo Minguez, Peer Bork, Christian von Mering, et al. String v9. 1: protein-protein interaction networks, with increased coverage and integration. *Nucleic acids research*, 41(D1):D808–D815, 2013.

[26] Gourab Ghoshal and Albert-László Barabási. Ranking stability and super-stable nodes in complex networks. *Nature communications*, 2:394, 2011.

[27] E. N. Gilbert. Random graphs. *The Annals of Mathematical Statistics*, 30(4):1141–1144, 12 1959. `http://dx.doi.org/10.1214/aoms/1177706098`.

[28] Johannes Goll, Seesandra V Rajagopala, Shen C Shiau, Hank Wu, Brian T Lamb, and Peter Uetz. Mpidb: the microbial protein interaction database. *Bioinformatics*, 24(15):1743–1744, 2008.

[29] Kristian A. Gray, Louise C. Daugherty, Susan M. Gordon, Ruth L. Seal, Mathew W. Wright, and Elspeth A. Bruford. Genenames.org: the hgnc resources in 2013. *Nucleic Acids Research*, 41(D1):D545–D552, 2013. `http://nar.oxfordjournals.org/content/41/D1/D545.abstract`.

[30] Dario Greco, Panu Somervuo, Antonio Di Lieto, Tuomas Raitila, Lucio Nitsch, Eero CastrÃ©n, and Petri Auvinen. Physiology, pathology and relatedness of human tissues from gene expression meta-analysis. *PLoS ONE*, 3(4):e1880, 04 2008. `http://dx.plos.org/10.1371%2Fjournal.pone.0001880`.

[31] Dov Greenbaum, Christopher Colangelo, Kenneth Williams, and Mark Gerstein. Comparing protein abundance and mrna expression levels on a genomic scale. *Genome Biol*, 4(9):117, 2003.

[32] Pierre C Havugimana, G Traver Hart, Tamás Nepusz, Haixuan Yang, Andrei L Turinsky, Zhihua Li, Peggy I Wang, Daniel R Boutz, Vincent Fong, Sadhna Phanse, et al. A census of human soluble protein complexes. *Cell*, 150(5):1068–1081, 2012.

[33] Henning Hermjakob, Luisa Montecchi-Palazzi, Chris Lewington, Sugath Mudali, Samuel Kerrien, Sandra Orchard, Martin Vingron, Bernd Roechert, Peter Roepstorff, Alfonso Valencia, et al. Intact: an open source molecular interaction database. *Nucleic acids research*, 32(suppl 1):D452–D455, 2004.

[34] Da Wei Huang, Brad T Sherman, and Richard A Lempicki. Systematic and integrative analysis of large gene lists using david bioinformatics resources. *Nature protocols*, 4(1):44–57, 2008.

[35] Da Wei Huang, Brad T Sherman, and Richard A Lempicki. Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic acids research*, 37(1):1–13, 2009.

[36] Arek Kasprzyk. Biomart: driving a paradigm change in biological data management. *Database*, 2011:bar049, 2011.

[37] Markus Krupp, Jens U. Marquardt, Ugur Sahin, Peter R. Galle, John Castle, and Andreas Teufel. Rna-seq atlas - a reference database for gene expression profiling in normal tissue by next-generation sequencing. *Bioinformatics*, 28(8):1184–1185, 2012. `http://bioinformatics.oxfordjournals.org/content/28/8/1184.abstract`.

[38] Dekang Lin. An information-theoretic definition of similarity. In *ICML*, volume 98, pages 296–304, 1998.

[39] Wen-hsien Lin, Wei-chung Liu, and Ming-jing Hwang. Topological and organizational properties of the products of house-keeping and tissue-specific genes in protein-protein interaction networks. *BMC systems biology*, 3(1):32, 2009.

[40] H. Lodish. *Molecular Cell Biology*. W. H. Freeman, 2008. http://books.google.com/books?id=K3JbjG1JiUMC.

[41] Tiago JS Lopes, Martin Schaefer, Jason Shoemaker, Yukiko Matsuoka, Gabriele Neumann, Miguel A Andrade-Navarro, Yoshihiro Kawaoka, Hiroaki Kitano, et al. Tissue-specific subnetworks and characteristics of publicly available human protein interaction databases. *Bioinformatics*, 27(17):2414–2421, 2011.

[42] David J Lynn, Geoffrey L Winsor, Calvin Chan, Nicolas Richard, Matthew R Laird, Aaron Barsky, Jennifer L Gardy, Fiona M Roche, Timothy HW Chan, Naisha Shah, et al. Innatedb: facilitating systems-level analyses of the mammalian innate immune response. *Molecular systems biology*, 4(1), 2008.

[43] Marc Mino and T Sanavia. Fastsemsim: Fast semantic similarity over gene ontology annotations. http://www.eccb12.org/poster/accepted/. Poster presented at ECBB 2012, cited with permission from the author.

[44] Marco Mino. fastSemSim. https://sites.google.com/site/fastsemsim/home. https://sites.google.com/site/fastsemsim/home. Accessed: 2014-03-18.

[45] M. E. J. Newman, S. H. Strogatz, and D. J. Watts. Random graphs with arbitrary degree distributions and their applications. *Phys. Rev. E*, 64:026118, Jul 2001. http://link.aps.org/doi/10.1103/PhysRevE.64.026118.

[46] National University of Singapore. Mbinfo. http://www.mechanobio.info/. http://www.mechanobio.info/.

[47] Sandra Orchard, Samuel Kerrien, Sara Abbani, Bruno Aranda, Jignesh Bhate, Shelby Bidwell, Alan Bridge, Leonardo Briganti, Fiona SL Brinkman, Gianni Cesareni, et al. Protein interaction data curation: the international molecular exchange (imex) consortium. *Nature methods*, 9(4):345–350, 2012.

[48] Karl Pearson. X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *Philosophical Magazine Series 5*, 50(302):157–175, 1900. http://www.tandfonline.com/doi/abs/10.1080/14786440009463897.

[49] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3):036106, 2007.

[50] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007*, 1995.

[51] Philip Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *arXiv preprint arXiv:1105.5444*, 2011.

[52] Sebastien Rombauts. SQLiteC++. `https://srombauts.github.io/SQLiteCpp/`. `https://srombauts.github.io/SQLiteCpp/`. Accessed: 2014-01-21.

[53] Jean-François Rual, Kavitha Venkatesan, Tong Hao, Tomoko Hirozane-Kishikawa, Amélie Dricot, Ning Li, Gabriel F Berriz, Francis D Gibbons, Matija Dreze, Nono Ayivi-Guedehoussou, et al. Towards a proteome-scale map of the human protein–protein interaction network. *Nature*, 437(7062):1173–1178, 2005.

[54] Gabriella Rustici, Nikolay Kolesnikov, Marco Brandizi, Tony Burdett, Miroslaw Dylag, Ibrahim Emam, Anna Farne, Emma Hastings, Jon Ison, Maria Keays, et al. Arrayexpress update - trends in database growth and links to data analysis tools. *Nucleic acids research*, 41(D1):D987–D990, 2013.

[55] Lukasz Salwinski, Christopher S Miller, Adam J Smith, Frank K Pettit, James U Bowie, and David Eisenberg. The database of interacting proteins: 2004 update. *Nucleic acids research*, 32(suppl 1):D449–D451, 2004.

[56] Martin H Schaefer, Jean-Fred Fontaine, Arunachalam Vinayagam, Pablo Porras, Erich E Wanker, and Miguel A Andrade-Navarro. Hippie: Integrating protein interaction networks with experiment based quality scores. *PLoS One*, 7(2), 2012.

[57] Andreas Schlicker, Francisco S Domingues, Jörg Rahnenführer, and Thomas Lengauer. A new measure for functional similarity of gene products based on gene ontology. *BMC bioinformatics*, 7(1):302, 2006.

[58] Christian Staudt, Aleksejs Sazonovs, and Henning Meyerhenke. Networkit: An interactive tool suite for high-performance network analysis. *CoRR*, abs/1403.3005, 2014.

[59] Andrew I. Su, Tim Wiltshire, Serge Batalov, Hilmar Lapp, Keith A. Ching, David Block, Jie Zhang, Richard Soden, Mimi Hayakawa, Gabriel Kreiman, Michael P. Cooke, John R. Walker, and John B. Hogenesch. A gene atlas of the mouse and human protein-encoding transcriptomes. *Proceedings of the National Academy of Sciences of the United States of America*, 101(16):6062–6067, 2004. `http://www.pnas.org/content/101/16/6062.abstract`.

[60] Haibao Tang, Brent Pedersen, Aurelien Naldi, and Patrick Flick. goatools - tools for gene ontology. `https://github.com/tanghaibao/goatools`. `https://github.com/tanghaibao/goatools`.

[61] Mathias Uhlén, Erik Björling, Charlotta Agaton, Cristina Al-Khalili Szigyarto, Bahram Amini, Elisabet Andersen, Ann-Catrin Andersson, Pia Angelidou, Anna Asplund, Caroline Asplund, et al. A human protein atlas for normal and cancer tissues based on antibody proteomics. *Molecular & Cellular Proteomics*, 4(12):1920–1932, 2005.

[62] Mathias Uhlen, Per Oksvold, Linn Fagerberg, Emma Lundberg, Kalle Jonasson, Mattias Forsberg, Martin Zwahlen, Caroline Kampf, Kenneth Wester, Sophia Hober, et al. Towards a knowledge-based human protein atlas. *Nature biotechnology*, 28(12):1248–1250, 2010.

[63] Kavitha Venkatesan, Jean-Francois Rual, Alexei Vazquez, Ulrich Stelzl, Irma Lemmens, Tomoko Hirozane-Kishikawa, Tong Hao, Martina Zenkner, Xiaofeng Xin, Kwang-Il Goh, et al. An empirical framework for binary interactome mapping. *Nature methods*, 6(1):83–90, 2009.

[64] Haiyuan Yu, Leah Tardivo, Stanley Tam, Evan Weiner, Fana Gebreab, Changyu Fan, Nenad Svrzikapa, Tomoko Hirozane-Kishikawa, Edward Rietman, Xinping Yang, et al. Next-generation sequencing to generate interactome datasets. *Nature methods*, 8(6):478–480, 2011.

[65] Andreas Zanzoni, Luisa Montecchi-Palazzi, Michele Quondam, Gabriele Ausiello, Manuela Helmer-Citterich, and Gianni Cesareni. Mint: a molecular interaction database. *FEBS letters*, 513(1):135–140, 2002.

[66] Jiang Zhu, Fuhong He, Shuhui Song, Jing Wang, and Jun Yu. How many human genes can be defined as housekeeping with current expression data? *BMC Genomics*, 9(1), 2008. `http://www.biomedcentral.com/1471-2164/9/172`.