Confidence in Protein Interaction Networks

Lyuba Ventsislavova Bozhilova

Pembroke College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas Term 2019

Acknowledgements

First and foremost, I would like to thank my supervisors, Charlotte Deane and Gesine Reinert, for their guidance. I have been incredibly privileged to have them as mentors.

Our industrial collaborators Alan Whitmore and Jonny Wray brought much appreciated enthusiasm and a fresh perspective to every meeting. Garrett Morris played a key role in the interdisciplinary training I received and provided valuable support and advice. I am also grateful to Mark Newman for some insightful discussions on network inference.

Working towards this thesis has been quite the ride and through it I have had the very best of travel companions in Fergus Boyles. His friendship made the fun parts all the more fun and the hard parts that bit easier. Other colleagues also helped along the way. Luis Ospina Forero taught me two great lessons: that it is often important to do things just the way you like and to always keep a stash of chocolate on my desk. Jinwoo Leem provided a well-timed break from DPhil life and reminded me how exciting the world out there is. Through it all, Javier Pardo-Diaz has been excellent both to work and to procrastinate with. Everyone else I had the pleasure of working with—especially members of the networks crew, cryptic crossword enthusiasts and SABS 2015—contributed to many fun and productive conversations.

My gratitude also goes to my family—to my father Ventsislav Bozhilov for understanding me so well, to my mother Lyubomira Shklifova for supporting me unconditionally, and to my brother Stefan Shklifov for being a brilliant partner in crime. Some family you are born in, and some you make yourself. Joseph Thorne puts up with me every time I hear a great science talk and then insist on telling him all about it. Without Caitlin Kennedy I doubt I would have even started a DPhil, let alone finished one. She continues to keep me (mostly) sane.

Thank you all.

Abstract

Protein interaction networks are a commonly used tool in bioinformatics, e.g. for the purposes of gene function prediction or drug target identification. They are built from often heterogeneous and error-prone protein–protein interaction data. In this thesis we study the effects of data uncertainty on the structure of protein interaction networks and on downstream network analysis.

Some databases provide confidence scores for protein–protein interactions, and networks are built from the data after a minimum score cut-off, or threshold, is applied. We study the effects of threshold choice on network structure. We argue that robust, biologically-relevant network analysis results should be replicated across networks obtained at different thresholds, and develop a methodology for quantifying this robustness in the context of node metrics. Our results indicate that the same node metrics are robust across a range of protein interaction networks, but are not necessarily robust in synthetic networks.

We further investigate uncertain networks as a possible approach to incorporating confidence scores explicitly into network analysis. Uncertain networks are a way of conceptualising the difference between the "true" network of biologically-relevant protein–protein interactions and the observed scored data. We show that any inference on the structure of the "true" network is strongly influenced by assumptions made about the dependence—or lack thereof—between edges in the scored network.

Finally, we focus on networks constructed from gene co-expression data. Gene co-expression can be measured in a number of different ways. Moreover, when networks are constructed, different thresholds can be applied to the co-expression values. It is not always clear which network construction method should be preferred. We develop a software package, COGENT, designed to aid network construction choice without the need for external validation data.

Contents

List of Figures

List of Abbreviations

AC . . . . . Adenylyl cyclase.

ADP . . . . . Adenosine diphosphate.

ATP . . . . . Adenosine triphosphate.

avg. . . . . . Average.

BN-PAGE . . Blue native polyacrylamide gel electrophoresis.

cAMP . . . . . Cyclic adenosine monophosphate.

coIP . . . . . Coimmunoprecipitation.

DNA . . . . . Deoxyribonucleic acid.

ERGM . . . . Exponential random graph model.

FDR . . . . . False discovery rate.

FEP . . . . . Fixed edge partition.

FNR . . . . . False negative rate.

GDP . . . . . Guanosine diphosphate.

GPCR . . . . . G protein coupled receptor.

GTP . . . . . Guanosine triphosphate.

HSP . . . . . Heat shock protein.

i.i.d. . . . . . Independent and identically distributed.

LOUD . . . . . Leave-one-out-difference.

mRNA . . . . Messenger ribonucleic acid.

MS . . . . . Mass spectrometry.

NA . . . . . Not available.

NMR . . . . . Nuclear magnetic resonance.

ORF . . . . . Open reading frame.

PIN . . . . . Protein interaction network.

PKA . . . . . Protein kinase A.

PPI . . . . . Protein-protein interaction.

pre-mRNA . . Precursor messenger ribonucleic acid.

PRS . . . . . Positive reference set.

REP . . . . . Random edge partition.

RNA . . . . . Ribonucleic acid.

RNA-seq . . . Ribonucleic acid sequencing.

rRNA . . . . . Ribosomal ribonucleic acid.

SBM . . . . . Stochastic block model.

st. dev. . . . . Standard deviation.

TAP . . . . . Tandem affinity purification.

TAP-MS . . . Tandem affinity purification followed by mass spectrometry.

TEV . . . . . Tobacco etch virus.

TF . . . . . Transcription factor.

Y2H . . . . . Yeast two-hybrid.

List of Symbols

K_d	Dissociation constant.
G, H	Networks.
V	Node set of a network; usually a gene or protein set.
E	Edge set of a network; usually a set of gene or protein interactions.
N	Number of nodes in a network.
$e(G)$	Number of edges in a network.
p	Network density.
A	Adjacency matrix of a network.
D	Degree matrix of a network.
$f(\cdot)$, $g(\cdot)$	Functions.
θ, μ	Thresholds.
$\Gamma(v)$	Neighbourhood (i.e. set of neighbours) of a node v in a network.
$deg(v)$	Degree of a node v in a network.
$l(u, v)$	Distance between two nodes u and v in a network.
$ego_i(v)$	Ego network around node v and containing all nodes u within distance i of v, $l(u, v) \leq i$, and all edges between these nodes.
$\Gamma_i(v)$	The node set of $ego_i(v)$.

Without a past you can't have a future.

— Michael Ende, *The Neverending Story*

1 Introduction

Contents

1.1 Motivation

The cell is the basic structural and functional unit of life. All known living organisms are comprised of cells. Some, like bacteria, consist of a single cell, while others can be formed of many cells, which share genetic material but can nevertheless

be heterogeneous. The body of an adult woman, for example, consists of about 2.1×10^{13} cells of different shapes and sizes (Sender et al. 2016).

Each cell contains millions of molecules, which have diverse functions (Alberts et al. 2013). Of these, proteins account for the majority of a cell's dry mass and play a dominant role in its functionality (Milo 2013). Proteins generally do not carry out their function in isolation, but rather interact with each other in order to complete complex cellular processes (Hartwell et al. 1999). A key step in studying these cellular processes is therefore understanding how proteins interact with each other and with their environment.

The concept of an interaction between two or more proteins is fluid. Often, two proteins are understood to interact if they physically bind to each other, and if such an event is thought to have some biological significance (De Las Rivas and Fontanillo 2010). However, proteins can interact indirectly as well. The human ribosome, for example, consists of eighty proteins and four rRNA molecules. Not all of these bind directly to each other (Khatter et al. 2015). However they all "interact" in order for protein translation to be carried out in the ribosome. This type of indirect, biologically relevant interaction is typically known as *functional association* between proteins. Other types of functional associations also exist.

Depending on how they are defined, protein–protein interactions (PPIs) can be detected experimentally in a number of different ways (Rao et al. 2014). They can also be computationally predicted (Galperin and Koonin 2000). Each way of detecting or inferring PPIs is subject to experimental error. These heterogeneous PPI data are often aggregated in different curated databases, which may control or quantify the uncertainty associated with the interactions (Klingström and Plewczynski 2010).

Protein interaction data can be used to investigate a particular protein or biological pathway of interest in detail. These data are often represented as networks. Network analysis of PPI data can be employed to study cellular processes as a complex system and at a larger scale (Barabasi and Oltvai 2004). If a social network describes people and the different relationships they build with each other, then

a protein interaction network (PIN) is a representation of proteins and how they interact. Once such a network is constructed, a range of network analysis techniques can be employed to understand how proteins collaborate in the cell.

However, there is uncertainty associated with protein interaction data. For example, protein–protein binding assays are carried out in non-native conditions, and while a positive experimental outcome is indicative of a biologically relevant interaction, such an interaction is only inferred, rather than being directly observed. Moreover, experimental error (i.e. noise) can result in missing or spurious interactions. The uncertainty and noise in experimental protein interaction data are an often overlooked problem of protein interaction network analysis. Typically, a single network is constructed from the data and is analysed under the implicit assumption that it is, in a sense, "correct". This thesis explores the ways in which uncertainty in the data can affect network analysis and its biological interpretability. We investigate the effect of data pre-processing on network construction, as well as strategies for incorporating uncertainty explicitly into PINs.

We begin this chapter with an overview of protein structure and function, as well as of protein interactions. We focus on commonly used techniques for protein–protein interaction detection and discuss their properties. We then discuss how protein interaction data is recorded, and how the uncertainty associated with it is handled during data curation and quantified by widely used databases. We subsequently consider different ways in which protein interaction networks are constructed, as well as analysed. While errors are rarely explicitly studied in the context of PINs, we discuss error and uncertainty in other network analysis applications. Finally, we give an overview of the remainder of this thesis.

1.2 Proteins and protein interactions

1.2.1 From genes to proteins

The genetic information carried by each cell, including information about what proteins can be produced, is stored in deoxyribonucleic acid (DNA). DNA is generally formed by two complementary chains of nucleic acids, which coil around each other

to form a double helix (Watson, Crick, et al. 1953). Each DNA strand is a polymer, consisting of a sequence of nucleotides bound together.

In DNA, information about proteins is stored in *genes*. A protein-coding gene is a sequence of nucleotides responsible for the production of a particular protein. The production of a protein, also known as *protein biosynthesis* happens in two main stages—first, the protein-coding part of the gene is transcribed, or copied, onto messenger ribonucleic acid (mRNA), and then the mRNA is translated into protein (Lengyel and Söll 1969). A single gene may be transcribed and translated multiple times, and result in multiple copies of the same protein. We call this phenomenon *gene expression*. Gene expression will vary in different cell types and under different experimental conditions (e.g. M. D. Robinson et al. 2010).

A protein-coding gene consists of an open reading frame (ORF) surrounded on each side by a regulatory sequence (Gerstein et al. 2007). Regulatory sequences control gene expression, while the open reading frame contains the code required to make the protein. Proteins are polymers made of amino acids. There are twenty naturally occurring amino acids. Each amino acid of a protein is encoded by a triplet of nucleic bases (*a codon*) in the ORF of the corresponding gene.

When an ORF is transcribed onto precursor mRNA, its entire sequence is copied. This sequence will typically consist of both coding and non-coding regions (*exons* and *introns*, respectively), and needs to be processed prior to translation. Precursor mRNA undergoes a series of post-transcriptional modifications to mature mRNA ready for translation (Machnicka et al. 2012). One of these is *splicing*, i.e. the removing parts of the mRNA. Splicing results in the removal of non-coding introns, but can also result in removing a subset of the exons in the molecule. This means that the same pre-mRNA can result in different mature mRNAs, which are then translated into different proteins. This process is known as alternative splicing, and is of particular importance in antibody production (Nilsen and Graveley 2010; Yabas et al. 2016).

Following post-translational modification, the mature mRNA is translated into protein by the ribosome. Translation involves processing codons sequentially and

synthesizing the corresponding amino acid chain (*polypeptide*). This chain then folds, to give the protein its three-dimensional structure (Gething and Sambrook 1992). It may also be subjected to a series of post-translational modifications, including phosphorylation, linkage to other proteins, and protein splicing (Mann and Jensen 2003). This means that, while often a gene is assumed to correspond to a single protein—the so-called "one gene–one polypeptide" hypothesis—in fact a single gene can result in different, albeit highly related proteins (X. Yang et al. 2016).

The reverse is also true. *Gene duplication* is the mechanisms by which a section of DNA which includes a gene is copied during DNA replication (Magadum et al. 2013). The two copies will then start to accumulate mutations independently of each other, which means that eventually, they may result in two distinct, but similar proteins. This process of duplication, followed by divergence due to mutation, has been used to model protein interactions (Ispolatov et al. 2005).

Once translated, proteins vary greatly in shape and size and are responsible for performing a wide range of cellular functions, such as DNA repair, metabolic reaction catalysis, and stimulus response. The specific task, or tasks, which a protein performs in the cell are determined by its three-dimensional structure (Zhang and S.-H. Kim 2003).

1.2.2 Protein interactions

One of the key aspects to protein function is the ability of a protein to interact with other proteins. Protein–protein interactions can occur in different ways and on different time scales. They can be loosely split in two categories—*physical interactions*, and *functional associations*.

Physical interactions

Physical interactions take place when two or more proteins bind to each other. Protein binding occurs when a large number of weak physico-chemical interactions are formed between the surfaces of two proteins. Often, protein–protein interfaces, i.e. the interacting surfaces, are characterised and studied with respect to their

amino acid sequence and composition (Yan et al. 2008). Physical protein–protein interactions can be split in two categories—*stable* and *transient* (S. Jones and Thornton 1996).

Stable binding occurs between the subunits of a stable or permanent protein complex. Such a complex can be formed of copies of the same protein, in which case it is the complex is called a homo-oligomer, or of copies of a number of different proteins, in which case it is called a hetero-oligomer (Nooren and Thornton 2003).

Indeed, the ribosome is one such complex, consisting both of proteins and rRNA molecules. ATP synthase is another example (Boyer 1997). It is a mitochondrial membrane protein complex which acts as a catalyst in the synthesis of adenosine triphosphate (ATP) from adenosine diphosphate (ADP). ATP synthase is formed of two regions, which are in turn made of several subunits each (see Figure 1.1). The F_O region rotates under a proton gradient in order to effectively power ATP synthesis carried out by the F_1 region.

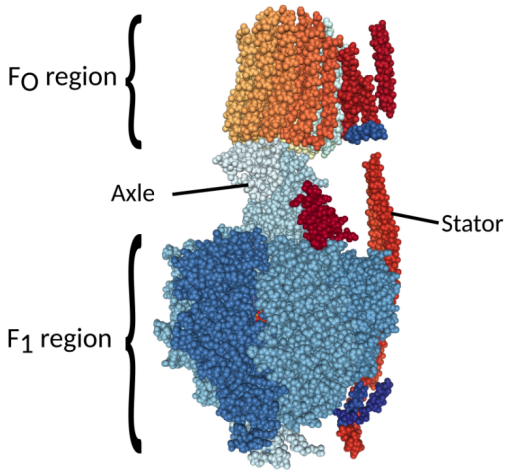

Figure 1.1: ATP synthase. ATP synthase of *Paracoccus denitrificans* (PDB code 5DN1). Subunits are shown in different colours. The F_O region is located in the cell membrane, and the F_1 region is intracellular. The two are connected via the axle and the stator. This figure was generated using NGL Viewer (Rose et al. 2018).

The different subunits of ATP synthase need to stay bound, i.e. to form stable interactions, to each other in order to form a working complex capable of ATP synthesis. However, not all biological functions are dependent on such stable

binding. Transient interactions occur when proteins associate and disassociate *in vivo* (Perkins et al. 2010).

Transient interactions can be further classified as weak or strong, depending on their lifespan (or dissociation constant K_d). Weak transient interaction are formed and broken continuously, and are characterised by K_d in the micromolar range (Acuner Ozbabacan et al. 2011). An example of weak transient binding is lysin dimerisation. Lysins are phage enzymes which can cleave the host cell wall and cause cell lysis. Lysins can exist both as a monomer, i.e. a single unbound molecule, or as a dimer, i.e. two molecules bound together (Shaw et al. 1995). The lysin monomer has been shown to be the more active of the two, while the lysin dimer is eliminated more slowly by the host cell (Grishin et al. 2019).

Strong transient interactions are characterised by a longer life-span than weak interactions, with K_d in the nanomolar range. They typically require a trigger, such as chemical modification, conformational change or colocalisation (Acuner Ozbabacan et al. 2011).

An example of a strong transient interaction is the interaction between the G_α and $G_{\beta\gamma}$ subunits of the G protein. G proteins are a signal transducers involved in a range of signalling pathways (Neves et al. 2002). G proteins are heterotrimeric, and consist of a G_α, G_β, and G_γ subunit. The G_β and G_γ subunits form a stable complex, $G_{\beta\gamma}$. G proteins function by converting guanosine triphosphate (GTP) to guanosine diphosphate (GDP). They are typically activated by a G protein coupled receptor (GPCR), which is bound to the G_α subunit and triggers the conformational change required for the exchange of GDP to GTP. Once bound to GTP, the G_α subunit dissociates from $G_{\beta\gamma}$ and both become active and can affect downstream signalling. When the GTP on G_α is hydrolyzed to GDP, the original complex is restored (Stewart and Fisher 2015). GTP binding acts as a trigger for the dissociation of the complex, while GTP to GDP hydrolysis triggers binding.

While the classification of physical PPIs into stable interactions, and strong or weak transient interactions is useful, many PPIs do not belong to just one interaction type (Nooren and Thornton 2003). Physiological or experimental conditions can

also have an effect. For example, the stability of an interaction can be disrupted by a temperature increase (e.g. Palleros et al. 1994) or the presence of a ligand (e.g. Moellering et al. 2009).

Whether stable or transient, strong or weak, the physical binding between two proteins is a concrete, well-defined concept, which can be experimentally tested. Functional associations, on the other hand, are a way of conceptualising protein interactions without requiring a specific mechanism of action.

Functional associations

Physical binding is not necessary for a meaningful functional interaction between two proteins. Above, we discussed ATP synthase as a protein complex which consist of a number of protein subunits bound together. However, it is not the case that every subunit of ATP synthase binds to every other subunit—most of the subunits of the F_O region are separated in space from the subunits of the F_1 region (Figure 1.1). Yet, the complex functions as a whole in order to carry out ATP synthesis, meaning that all of its subunits "interact" to carry out the same task. Functional associations describe relationships between two or more proteins which are involved in the same cellular process but which do not necessarily bind to each other (Marcotte et al. 1999).

Two distant subunits of the same complex are associated via a series of intermediate physical interactions with other subunits. Two proteins can also be functionally associated without other protein intermediaries. Like G proteins, adenylyl cyclase (AC) takes part in a number of signalling pathways (Simonds 1999). Upon binding to the G_α subunit of a G protein, AC catalyses the conversion of ATP to cyclic adenosine monophosphate (cAMP). The resulting cAMP acts as an intracellular messenger—it can travel from AC, which is on the cell membrane, and activate intracellular proteins such as protein kinase A (PKA), which in turn triggers downstream cell response (Lodish et al. 2008). While AC does not physically bind PKA, there is a clear meaningful relationship between the two via cAMP. Detecting such relationships can help us understand protein function.

Unlike physical interactions, functional associations do not have a single, clearly defined method of action. The two types of interactions are therefore detected through different experimental and computational techniques. Binding assays are commonly employed to identify physical interactions, while a range of genetic assays and computational tools can be used to infer functional associations. Each of these techniques is subject to different types of experimental error, and is usually suited to detecting only certain types of interactions. This may result in heterogeneous errors and biases in protein interaction data.

1.3 Protein interaction data and networks

1.3.1 Detecting and inferring protein–protein interactions

Detecting physical binding

There exist a range of experimental techniques for the detection of physical binding between two proteins. Some are designed for screening many possible interactions at a time, potentially at the cost of experimental accuracy (high-throughput), while others are more accurate but cannot be used for big systematic screens (low-throughput). The two most widely used high-throughput techniques for detecting physical binding are yeast two-hybrid screens (Y2H) (Fields and Song 1989) and tandem affinity purification followed by mass spectrometry (TAP-MS) (Rigaut et al. 1999; Puig et al. 2001). Low throughput techniques include coimmunoprecipitation, blue native polyacrylamide gel electrophoresis (BN-PAGE), nuclear magnetic resonance (NMR) spectroscopy and X-ray crystallography (Miernyk and Thelen 2008; Rao et al. 2014).

Yeast two-hybrid experiments (Fields and Song 1989) are carried out in genetically modified yeast strains where the DNA binding domain of a transcription factor (TF) is fused to one of the proteins of interest (the 'bait' protein), while the activation domain of the same TF is fused to the other protein (the 'prey'). If the two proteins interact, then the two domains are brought in close proximity, the TF becomes functional, and reporter gene transcription is initiated, so that an increase in the reporter gene expression can be observed. This technique has been

widely used and can be fully automated and is therefore also easily scaled. However, Y2H often suffers from high false positive and false negative rates (H. Huang, Jedynak, et al. 2007). The functionality of the proteins of interest may be altered by the fusion to a TF domain, for example because of misfolding or because the TF domain may obstruct the protein's binding site, leading to potential false negative outcomes. Furthermore, in its standard form Y2H is limited to proteins which can enter the nucleus, although variations exist (e.g. see Snider et al. 2010 for membrane protein Y2H). Since in Y2H the bait and prey proteins are highly expressed and co-localised in the cell nucleus, interactions which may be physically possible, but are not necessarily biologically meaningful, can be detected, leading to false positives. This happens for example when the two proteins of interest are typically expressed in different cell types, or different sub-cellular compartments (Van Criekinge and Beyaert 1999). Error rates in Y2H assays will vary with the experimental protocol. Overall, false discovery rates (FDR) per single screen have been estimated in the range 9.9%–17% and false negative rates (FNR) in the range 28%–51% across different model organisms (H. Huang and J. S. Bader 2009). Due to the high error rates, it is common to only report interactions observed multiple times (e.g. Rolland et al. 2014). This reduces the false positives in the dataset, at the cost of an increased false negative rate.

Unlike Y2H, which tests for an interaction between two specifically chosen proteins, a single *tandem affinity purification* assay can provide evidence for multiple preys binding to a single bait (Rigaut et al. 1999; Puig et al. 2001). In TAP-MS the bait protein is fused with a TAP tag, which consists of two proteins used for two affinity purification steps (originally Protein A and a calmodulin binding peptide), separated by a tobacco etch virus (TEV) protease cleavage site. The fused protein is expressed in a cell line and allowed to interact with other proteins. The cells then undergo lysis and two purification steps, in order to isolate the bait protein and its interaction partners. The components of the resulting elution are identified using mass spectrometry. Unlike Y2H, TAP-MS allows the bait protein to be expressed and interact with other proteins in near-native conditions. However, fusion to a

TAP tag may still alter the protein's function. Different TAP tags can be used (P. Kaiser et al. 2008). In addition to affecting the bait protein's structure and function, the choice of TAP tag and consequently purification protocol introduces a trade-off between false positive and false negative rates: stringent purification minimises contaminants, but is also likely to wash away weak transient interactions. Additional error is accumulated in the mass spectrometry (MS) phase. In most cases, the proteins in the final elution are broken into shorter peptides, which are then identified via MS. The interacting proteins are then inferred by the identified peptides. However, proteins may share peptides, which makes correctly identifying them difficult and in particular introduces false positives (Dunham et al. 2012). Spectral counts from the MS phase can be used to score identified interactions, so that a final interaction data set is constructed by fixing the estimated false positive rate (e.g. Choi et al. 2011). Reported estimates of the false negative rate of TAP-MS are around 15% (Edwards et al. 2002).

Coimmunoprecipitation (coIP) is an affinity purification method which uses antibody binding in order to identify protein complexes. Antibodies are proteins which bind with high specificity to a particular protein or peptide target (antigen) as part of the immune response system. Immunoprecipitation is a technique used to isolate a specific protein from a protein sample, such as solution or cell lysate (Bonifacino et al. 1999). It works by introducing antibody-coated beads to the protein sample. The antibodies on the beads will bind specifically to the antigen of interest, and can then be extracted and washed. Different types of beads and washing protocols exist (Kaboord and Perr 2008). CoIP consist of applying immunoprecipitation to proteins which are known or suspected to be part of a stable complex. In this case, the entire complex is bound to the beads via the specific antibody-antigen interaction, and its remaining subunits can be identified via MS (Mann, Hendrickson, et al. 2001). Two-step coIP, where protein G-coated beads are used to remove any impurities extracted by the antibody-coated beads, can be used for higher quality results (Sciuto et al. 2018).

Another way to isolate complexes from cell and tissue homogenates, as well as from cell membranes, is *blue native polyacrylamide gel electrophoresis* (Schägger and von Jagow 1991; Wittig et al. 2006). Gel electrophoresis is a technique by which a mixture of molecules are separated by their size and charge. This is done by suspending them into a porous gel, which is subjected to an electric field. The electric field causes the charged molecules to move, with smaller molecules travelling faster and farther through the pores of the gel than larger ones. This results in bands on the gel, where molecules of the same size cluster. Gel electrophoresis can be used to separate DNA and RNA molecules (Aaij and Borst 1972) as well as proteins. BN-PAGE is a charge shift gel electrophoresis technique optimised for membrane protein complexes—the proteins are coated in Coomassie Brilliant Blue dye to introduce a negative charge, which is then used to separate them through electrophoresis. A stable complex should travel through the gel without dissociating and concentrate in a single band. However, binding to the dye may cause protein complexes to dissociate into their subunits, leading to false negative outcomes, especially in smaller and less abundant complexes (Eubel et al. 2005).

The techniques described above are most often used to detect stable or strong transient interactions, since the protein preparation protocols can often disrupt weaker protein binding. *NMR spectroscopy* can provide information about protein structure (Clore and Gronenborn 1998), as well as about the interaction between different proteins (Vaynberg and Qin 2006). Since samples consist of proteins in solution, NMR spectroscopy can be used to identify weak interactions through chemical shift mapping or hydrogen–deuterium exchange experiments (Qin et al. 2001). In particular, NMR can be used to determine interaction surfaces (e.g. Vaynberg, Fukuda, et al. 2005).

While NMR can provide structural information about proteins and the physical interactions between them, currently the highest resolution three-dimensional data comes from X-ray crystallography. The Protein Data Bank (Berman et al. 2000), which is the single largest repository of biological macromolecular structures, contains over 140,000 structures obtained through X-ray crystallography, compared

to about 13,000 structures obtained through NMR and about 4,000 obtained through electron microscopy (*data accessed December 2019*). X-ray crystallography is performed by first crystallising the molecule or complex of interest, and then radiating the crystal with an X-ray beam. The resulting diffraction pattern is transformed to an electron density map, to which the structure is then fitted. This approach has been used to solve protein structures since the late 1950s (Kendrew et al. 1958). It is also used to determine the exact binding mode of complexes—e.g. see Rasmussen et al. 2011 for a GPCR bound to a G protein.

In addition to being experimentally determined, physical binding can also be computationally predicted. Motivated by the availability of high-quality structure data, this is often interpreted as a binding surface prediction task—given two proteins, with potentially solved three-dimensional structures, can we predict where and how well they are likely to bind? This problem can be tackled, for example, through molecular docking (e.g. Smith and Sternberg 2002), by calculating amino acid pair propensities based on known complexes (e.g. Hamer et al. 2010), or by identifying co-evolving residues (e.g. Cong et al. 2019).

Inferring functional associations

Functional associations can be inferred entirely computationally, or using both computational tools and experimental data. Since many methods work on the gene or RNA level, rather than on the protein level, they are often called *genetic interactions* (Mani et al. 2008). The most widely used type of data to infer functional associations is gene expression data (D'haeseleer et al. 2000). Synthetic lethality screens are also commonly used (Tong, Evangelista, et al. 2001). Alternative approaches involve the study of DNA sequence evolution over time and across species—including gene neighbourhoods (Dandekar et al. 1998) and gene fusion events (Marcotte et al. 1999).

Gene expression profiling is the process of measuring the amount of gene product in the cell at a particular time point or under a particular set of experimental conditions (Lowe et al. 2017). The measurement made is of mRNA abundance,

rather than protein abundance. The underlying assumption is that if an increase in mRNA production is measured, then there should exist a corresponding increase in protein synthesis. However, the relationship between mRNA and protein abundance is non-linear, making such inference open to debate (Edfors et al. 2016).

There exist two main ways of measuring mRNA—through DNA microarray analysis (Schena et al. 1995) and through RNA sequencing (RNA-seq) (Z. Wang et al. 2009). A microarray is a glass slide to which a series of single-stranded DNA *probes* are attached (Pease et al. 1994). These probes are complementary to mRNA strands in the sample. As mRNA strands bind to their corresponding complementary probe, their abundance can be measured. Microarrays are a relatively fast and inexpensive, albeit inaccurate method for measuring mRNA abundance. Advances in sequencing techniques mean mRNA molecules can now be directly sequenced rather than identified using probes. This so called RNA-seq approach provides higher accuracy (Zhao et al. 2014). While microarray analysis and RNA-seq are frequently performed on homogenised tissue, it has recently become possible to perform single cell RNA-seq (Eberwine et al. 2014; Saliba et al. 2014).

Gene expression data provides a snapshot of the amounts of different gene products in the cell. It is commonly used in order to identify differences between healthy and diseased cells (Golub et al. 1999; Segal et al. 2004)—if a gene is over-expressed in a diseased cell compared to the control, then we can infer that the corresponding protein may be involved in or affected by the disease pathway.

Similarly, two genes being expressed at similar rates under a range of experimental conditions may indicate that their products are involved in performing the same task (Stuart et al. 2003). This is known as *gene co-expression analysis*. Typically, the expression profiles between every pair of genes are compared and are given a similarity score. Similar profiles can be indicative of shared function. Similarity can be measured in a range of ways, including Pearson correlation coefficient (e.g. Guttman et al. 2011), and mutual information (e.g. Butte and Kohane 1999). Since co-expression measures are continuous rather than binary, they can be used to provide a measure of uncertainty (or alternatively confidence)

in interactions. However, co-expression values can be adversely affected by sample size—too few samples may give unreliable values (Ballouz et al. 2015), while too many may obfuscate a signal which is specific to certain experimental conditions (Cosgrove et al. 2010).

A different method of identifying functional associations is through a synthetic lethality screen. A gene is considered essential if knocking it out, i.e. introducing a deletion mutation so it can no longer be expressed, makes the cell no longer viable. Such mutations have been systematically introduced in yeast (Winzeler et al. 1999), and it has been shown that as many as 73% of yeast genes are non-essential (Giaever et al. 2002). Rather than introducing single-gene knock-out, Tong, Lesage, et al. 2004 systematically knock out pairs of genes. If a knock-out of two genes, which are separately non-essential, results in cell death, then these two genes are said to have a *synthetic lethal interaction*. Such interactions have been shown to be biologically relevant (see Novick et al. 1989 for an early example). Gene interactions can also be identified through synthetic lethality data in a way similar to co-expression analysis. Barido-Sottani et al. 2019 argue that if two genes have similar synthetic lethality profiles (as opposed to similar gene expression profiles), i.e. if they are co-lethal with the same partners, then they likely perform similar function and therefore interact.

The approaches described above rely on experimental data, which is labour-intensive and often has poor reproducibility. Therefore, purely computational methods for predicting functional associations are desirable. The paradigm dictating that protein sequence determines protein structure, which in turn determines protein function, suggests that it should be possible to infer functional associations solely through sequence information, either on the protein or on the gene level. Indeed, Cong et al. 2019 take such an approach to predict physical binding. They use co-evolving protein residues to identify likely protein interaction interfaces.

Another way of using information about mutations, or evolution, in order to infer functional rather than physical interactions is through *gene fusion* events (Enright et al. 1999). A fusion event occurs when two previously separate genes are fused to form a single, longer gene. This mechanism is of particular importance in

cancer (Rabbitts 1994; Mitelman et al. 2007). However, it can also be illustrative of genetic interaction—Enright et al. 1999 show that if a pair of genes in one genome are observed to be fused in another genome, then they are more likely to belong to the same complex or functional pathway.

Cross-species sequence analysis can also be used to identify potential functional associations. Dandekar et al. 1998 show that a conservation in gene order across bacterial species is indicative of shared function. Overbeek et al. 1999 argue that the same is true for preserved clusters of genes, i.e. that if a group of genes appear close to each other in the genomes of several species, then they are more likely to functionally related.

So far we have outlined a number of commonly used ways to detect or predict different types of protein–protein interactions. It is beyond the scope of this thesis to provide an exhaustive list. It is estimated that there exist around 650,000 physical PPIs in human (Stumpf et al. 2008), while BioGRID, one of the largerst PPI repositories, currently only contains around 400,000 (Oughtred et al. 2018). A range of interaction detection methods will undoubtedly play a role in identifying these missing interactions, as well as validating already published ones. In the following section we discuss how protein interaction data is stored and shared, as well as how it is curated.

1.3.2 Databases

There exist a number of publicly available databases of protein–protein interactions, as well as of supporting data such as gene expression data and synthetic lethality data. These databases vary greatly in their content—some focus on a single organism, or a single experimental technique, while others integrate data from multiple sources. How the data is curated is also handled differently. Different databases can therefore vary greatly in quality and content (J. K. Huang et al. 2018). The three PPI databases which have been used throughout this thesis are STRING (Szklarczyk, Franceschini, et al. 2014; Szklarczyk, Gable, et al. 2018), HitPredict (López et al. 2015) and BioGRID (Chatr-Aryamontri et al. 2017).

STRING

As outlined above, there exist a number of techniques for detecting or predicting protein–protein interactions. They can be used to detect different types of interactions, and are subject to different kinds of errors and biases. Therefore, interaction data cannot be expected to perfectly capture the true biological state, but are rather noisy observations of it. STRING attempts to quantify the uncertainty associated with each observed or inferred interaction based on the nature, quality and quantity of the supporting evidence (Szklarczyk, Gable, et al. 2018). Each interaction in STRING is reported with a confidence score. It is intended that researchers choose the level of confidence they require of the data and effectively threshold on these scores, using only high-scoring interactions for their analysis.

STRING contains both physical and functional protein interactions derived from in-house predictions and homology transfers, as well as taken from a number of externally maintained databases. Principal data contributors are the IMEx consortium (Orchard et al. 2012) for physical binding data and ProteomeHD (Kustatscher et al. 2019) for gene co-expression data. STRING is, to our knowledge, one of the largest and most diverse public PPI databases, with the most recent version (STRING v.11) containing approximately 3.1 billion interactions across 24.6 million proteins from 5090 organisms.

Each interaction recorded in STRING is assigned a score between zero and one, which is an estimate for the likelihood that the interaction exists given the available evidence. STRING contains both physical binding and functional association data. Interaction evidence is split across seven channels. Gene co-expression data is analysed through the *coexpression* channel, and low- and high-throughput biochemical interaction experiments, such as Y2H or TAP-MS, are analysed through the *experimental* channel. Manually curated data is handled separately through the *database* channel. Sequence-based functional association predictions are analysed through the *neighbourhood*, *cooccurrence* and *fusion* channels, depending on the exact prediction method. Finally, the *textmining* channel contains interactions inferred from an in-house text-mining pipeline. Supporting evidence for each

interaction is scored in a channel-specific way, with missing evidence corresponding to a channel score of zero. So, for example, if the interaction for two proteins A and B is supported by a Y2H experiment and high gene co-expression, but has not been previously published, then the AB interaction entry would have a positive *experimental* and *coexpression* score, and a *textmining* score of zero. Each channel can contain information both specific to the organism A and B belong to, and information about interacting orthologs from other species. A single combined score is calculated for each interaction, under the assumption that channels are independent of each other and with an *a priori* probability of two randomly chosen proteins interacting set to $p = 0.041$ (Von Mering et al. 2005). The final combined scores range between 0 and 1, but only scores of 0.15 and above are reported.

While STRING is a valuable resource of quality-assessed PPI data, it is important to note that the reported scores are subject to large disparities across different release versions, and may therefore not be reliable. We illustrate this through the interaction data for *Saccharomyces cerevisiae* (baker's yeast). STRING v.10.5 (Szklarczyk, Franceschini, et al. 2014) contains 1,003,567 interaction records for yeast, compared to 922,983 records in STRING v.11.0 (Szklarczyk, Gable, et al. 2018). However, the two versions only overlap on 563,641 interactions—439,962 of the interactions reported in STRING v.10.5 were subsequently omitted from the data base, and 359,342 new interactions were added in the new version. While the majority of the omitted interactions have low scores, as many as 49,416 (or 11.2% of the omitted set) had scores above 0.40, which STRING classifies as "medium confidence" (see Figure 1.2). The version change can also be observed in the scores of the overlapping 563,641 interactions (see Figure 1.3). Of the shared interactions, 14,856 or 2.6% have retained their score across both versions, and about 45% of the interactions have undergone a score change of less than 0.05. However, 9.8% of interactions have undergone a score change above 0.30, and 3.9% of interactions have undergone a score change above 0.425, which corresponds to half the confidence score region, as reported scores are between 0.15 and 1.00. It is not immediately clear what causes the large disparities across different STRING

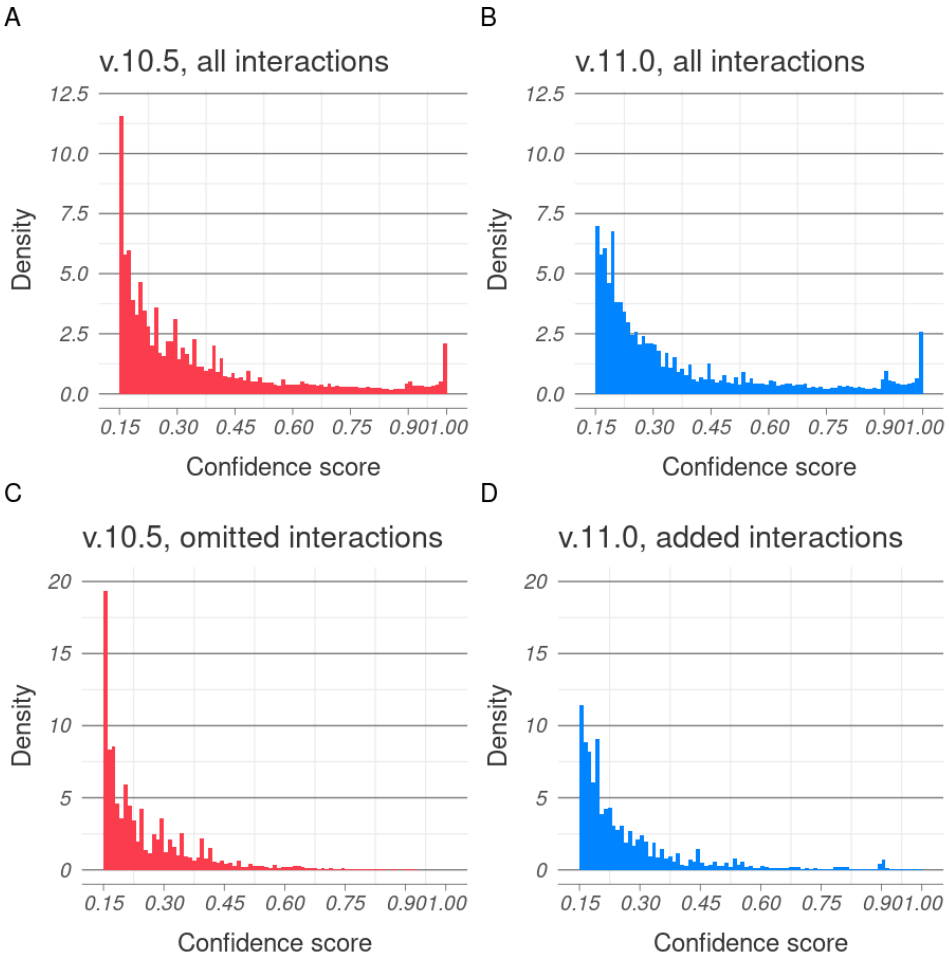

Figure 1.2: STRING version comparison. (A) Confidence scores for interactions in yeast in STRING v.10.5. (B) Confidence scores for interactions in yeast in STRING v.11.0. (C) Confidence scores of the interactions in v.10.5 which were not transferred to v.11.0. The majority of these were low, but medium and high-scoring interactions were also omitted. (D) Confidence scores for the interactions introduced in v.11.0. Like omitted interactions, added interactions have predominantly low scores. Overall, the peak of low scoring interactions around 0.15 is more pronounced in v.10.5 than it is in v.11.0.

versions, especially as all 6319 gene identifiers used in interaction records in v.10.5 are also present in the 6574 identifiers recorded in v.11.0. We hypothesise this is the result of new interaction data becoming available, combined with changes to the scoring procedures involved. We also note that similar large differences can be observed between previous database releases, which we report in the supplementary materials of our paper Bozhilova et al. 2019.

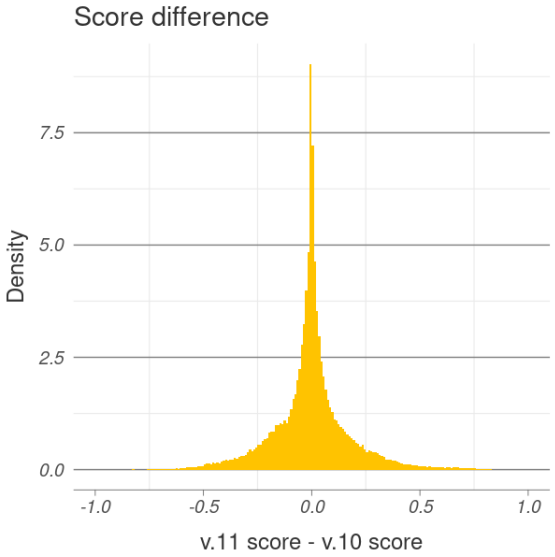

Figure 1.3: STRING score differences. Change in score for yeast interactions in STRING v.10.5 and v.11.0. While the mean change is close to zero, many interactions have been rescored. The score differences have mean -0.0015 and standard deviation 0.17.

HitPredict

Like STRING, HitPredict (López et al. 2015) is a database which aggregates interaction data and associates a confidence score to each interaction record. Unlike STRING, interactions in HitPredict are physical binding only. HitPredict is also far smaller than STRING, with a total of approximately 753,000 interactions across nearly 88,000 proteins from 124 species as of its August 2019 release.

Confidence scores in HitPredict are calculated and interpreted differently from STRING scores. Each interaction is assigned an annotation-based score and a method-based score, which are combined to an overall interaction score. The annotation score is calculated as a scaled likelihood ratio, based on structural information obtained from 3did (Mosca et al. 2013), functional annotations obtained from GO (Gene Ontology Consortium 2018) and homologous interactions obtained from HINTdb (Patil and Nakamura 2005). The scaling is done so that an annotation score above 0.50 corresponds to a likelihood ratio above 1, with the interpretation that the interaction is deemed more likely to exist than not. The method-based score quantifies the specific experimental evidence available, and is a combination of a publication score (*"How many times has the interaction been reported?"*), a

method score (*"What assays have been used?"*), and a type score (*"What is known about the type of the interaction?"*). The method-based score is a weighted average of these, calculated following Villaveces et al. 2015.

The two sub-scores—annotation-based, which captures indirect supporting evidence for each interaction, and method-based, which captures the available experimental evidence—are combined using the geometric mean to produce an overall interaction score. Unlike STRING scores, HitPredict scores are not meant to be interpreted as interaction likelihoods. HitPredict is also more stringent than STRING: while in STRING, interactions scoring above 0.70 are considered "high-confidence", the corresponding score threshold in HitPredict is 0.28. In both cases, a higher score corresponds to higher confidence in the data.

BioGRID

While other databases which aggregate and score data exist, one of the most widely used PPI data sources, BioGRID (Chatr-Aryamontri et al. 2017; Oughtred et al. 2018), contains interaction records which have been curated, but which have not been aggregated or systematically scored. BioGRID v.3.5.179 spans 1.75 million interactions across approximately 77,000 genes from 71 species. These interactions are both physical (corresponding to physical binding) and genetic (corresponding to functional associations).

Interaction records in BioGRID are based on experimental evidence only, and are not predicted. Multiple interaction records identifying the same interacting pair are often treated as independent support for the interaction, and such interactions can be used to form positive reference sets for benchmarking (e.g. Rolland et al. 2014). Data can be deposited to BioGRID and is internally curated.

Other PPI databases

Numerous other sources of PPI data exist. Some, like HuRI (Luck et al. 2019) are based on a single high-throughput experimental protocol, which has been employed to systematically study interactions in a particular system. Others contain exclusively predicted interactions, e.g. PIPs (McDowall et al. 2008).

Databases which aggregate data can be species-specific, such as HIPPIE for human PPIs (Alanis-Lobato et al. 2016), and SGD for yeast PPIs and other genomic and proteomic data (Cherry et al. 2011). There also exist dedicated efforts to record cross-species interactions, as host-pathogen interactions are of direct clinical interest (Ammari et al. 2016).

As discussed above, databases which aggregate PPI from primary sources will either provide confidence scores for recorded interactions or will curate their content in some other way. Interaction scores in STRING and HitPredict are entirely based on the quality of the available interaction evidence; GeneMANIA, in contrast, provides scores that are calculated in real time based on user input, in order to provide a measure of relevance, as well as confidence for each interaction (Franz et al. 2018).

Gene expression data

While resources like STRING use gene co-expression to infer protein–protein interactions, gene expression data is generally stored and distributed separately. Since experimental set-ups vary greatly, experiments are reported separately rather than aggregated. The Gene Expression Atlas is a database of manually curated and systematically re-analysed gene expression experiments from a range of organisms and experimental conditions (Papatheodorou et al. 2017). The curation and data processing pipeline employed by the Gene Expression Atlas means a lot of the datasets available have a reduced number of genes or samples (or both) compared to the raw data.

Gene expression data can be used to calculate gene co-expression in a number of different ways. While expression remains the standard form in which data is shared, some databases like COXPRESdb (Obayashi, Kagaya, et al. 2018) and ATTED (Obayashi, Aoki, et al. 2017) provide pre-calculated co-expression values.

Pathways and functional annotations

Binary interactions and high gene co-expression are both indicative of shared protein function. However, proteins often do not perform their function in isolated pairs. In Section 1.2.2 we discussed some of the interactions between GPCRs, G proteins, adenylyl cyclase and protein kinase A. All of these interactions play a role in cell signalling: upon detecting an adrenaline molecule, an adrenaline receptor (a type of GPCR) will activate its corresponding G protein, which will then carry the signal to AC, which will in turn indirectly activate protein kinase A, and by doing so will trigger downstream cell response (Hillis et al. 2012). This series of events is known as a *biological pathway*.

Many biological pathways have been identified. The epinephrine (i.e. adrenaline) pathway is a type of signalling pathway. Other commonly studied pathways are metabolic pathways and gene regulatory pathways (e.g. Caspi et al. 2006; Tu et al. 2006). Disease pathways are a way of conceptualising the cellular mechanism underlying disease (Y. Li and Agarwal 2009).

Pathways, unlike protein–protein interactions, are by definition subjective. A biologically meaningful protein–protein interaction either exists or it does not; it is the goal of interaction detection experiments to determine which interactions truly exist, and which do not. Pathways on the other hand, are a way of synthesising and representing expert knowledge about cellular biology. They generally consist of well-studied interactions, and are recorded in manually curated databases such as KEGG (Kanehisa and Goto 2000) and Reactome (Croft et al. 2013). Since interaction data contained in pathways is considered to be of very high quality, pathways are often used for benchmarking interaction detection efforts or data processing procedures. For example, benchmarking against KEGG is used extensively in calculating STRING channel scores (Szklarczyk, Gable, et al. 2018).

1.3.3 Biological networks

Pathway data and functional annotations can provide valuable insight into protein function, but they do not account for all detected and inferred PPIs. Protein

interaction networks (PINs) are a way of representing a large set of PPIs and analysing them collectively (Newman 2018b).

In PINs, proteins are represented by nodes and the interactions between them are represented by edges (see Figure 1.4). Network construction depends on the research question and may contain, for example, a group of proteins of interest and their one- and two-step neighbours (e.g. J. Wang et al. 2006), a set of proteins known or suspected to be involved in the same pathway (e.g. Hughes et al. 2008), or the entire proteome of a particular organism (e.g. Uetz et al. 2000). PINs can contain both physical and functional interactions, with the choice of data depending on the particular application. Interactions are typically modelled by undirected edges, although directed edges are sometimes used for functional interactions, for example in gene regulatory networks (Abraham et al. 2016). Regardless of the application and

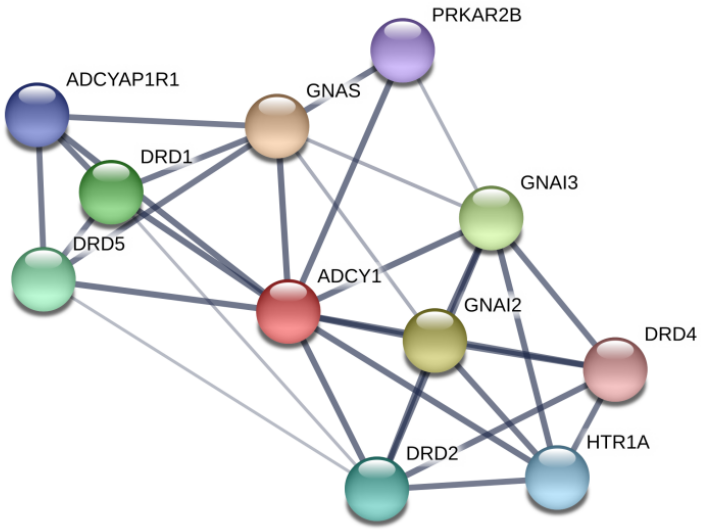

Figure 1.4: Example protein interaction network. The network depicts adenylyl cyclase (ADCY1, red) and some of its interacting partners. GNAI3, GNAI2 and GNAS are all G_α subunits of G proteins. DRD1, DRD2, DRD4 and DRD5 are dopamine receptors, ADCYAP1R1 is a peptide receptor, and HTR1A is a serotonin receptor. Finally, PRKAR2B is a subunit of PKA. Edge width corresponds to STRING confidence scores. This figure was generated using the STRING web service (Szklarczyk, Gable, et al. 2018).

the data used, PINs tend to be treated as deterministic networks. Edges are generally not weighted and hence the networks are binary (De Las Rivas and Fontanillo 2010). Existing edges are assumed to denote meaningful interactions. Implicitly, non-edges

are therefore assumed to denote a lack of a meaningful interaction. In the case where scored data, such as data from STRING or HitPredict, is available a threshold is applied and only interactions scoring above the threshold are included in the network (see Szklarczyk, Gable, et al. 2018 for a discussion on thresholding, and Krogan et al. 2006 for an example). Choosing a threshold introduces a trade-off between false positives and false negatives in the network, and it is not necessarily clear how to make this choice. Furthermore, our analysis (see Chapter 2) shows that different thresholds may yield topologically very different networks when applied to the same dataset. It is difficult to argue which of these networks is the "correct" one to analyse.

Gene co-expression data can also be represented by networks. These are networks built exclusively, or predominantly, from gene expression data rather than protein–protein interaction data. Unlike PINs, they can be either weighted or unweighted (Langfelder and Horvath 2008). In addition to the choice of experimental data to use for network construction, and the noise or uncertainty associated with it, the structure of gene expression networks can depend on how the data is pre-processed (Abbas-Aghababazadeh et al. 2018; T. Park et al. 2003), whether confounding factors are taken into account (Parsana et al. 2019) and how co-expression is measured (Gonzalez-Valbuena and Treviño 2017).

1.3.4 Applications

The analysis of protein–protein interaction networks and gene co-expression networks can provide a systematic view of the interactome of an organism and can help address a range of research questions (e.g. Barabasi and Oltvai 2004; Vidal 2009; Vidal et al. 2011). In addition to elucidating protein function, biological networks are of significant interest in the field of drug discovery and systems medicine (e.g. Hopkins 2008; González-Couto 2011).

Networks are used to predict protein function (e.g. Hishigaki et al. 2001; Sharan et al. 2007; Q. Wu et al. 2014) and disease relevance (e.g. Ideker and Sharan 2008; Hase et al. 2009; Taylor and Wrana 2012). This is typically done through "guilt-by-association" approaches: if a significant portion of a protein's interacting

partners are annotated with a particular biological process, then that protein is more likely to be associated with the same process. Both annotations and protein interactions suffer from research bias, which may influence the results of such analysis (Luecken et al. 2017).

Networks are also be used to identify possible drug targets, especially in the case of multi-target drug discovery (Hopkins 2008). A common aim of PIN analysis is the identification of key actors in the network for the purposes of drug target choice (Navlakha and Kingsford 2010; Abraham et al. 2016; Han et al. 2017). There exist many tools and strategies for predicting protein function, as well as for identifying protein or gene modules of interest (Shannon et al. 2003). Comparing the protein interaction networks of different species can be used to make inference about evolution (Zitnik et al. 2019). In the rest of this chapter we describe some network analysis tools, which can be applied to protein interaction data, or to relational data more generally.

1.4 Network analysis

Thus far we have discussed the relationships between different genes and their protein products. We introduced protein interaction networks as a way to model cellular architecture. In this section, we will abstract away from the biological context, and discuss some frequently used tools for network analysis.

1.4.1 Definitions and notation

Throughout this thesis we will assume that a network $G = (V, E)$ is a collection of nodes or vertices V, pairs of which are connected by edges $E \subset V \times V$. The nodes of a network are labelled and uniquely identifiable. They will usually represent genes or proteins. While node attributes (e.g. function annotation) are sometimes studied, we will generally not take these into account.

Each edge $e = (u, v) \in E$ connects two nodes u and v in the network. In the context of protein interaction networks, an edge could denote either a physical interaction or a functional association between two proteins. Networks can be directed

or undirected, weighted or unweighted, signed or unsigned, depending on edge properties. For example, in gene regulatory networks, edges are directed to denote which gene regulates which (Karlebach and Shamir 2008). In gene co-expression networks, edges can have weight, which measures the strength of co-expression (e.g. Langfelder and Horvath 2008) and occasionally sign, which shows whether the two genes are positively or negatively co-expressed (e.g. Mason et al. 2009).

A common edge attribute we will use throughout Chapters 2 and 3 is a *confidence score*, which measures the amount of confidence (or certainty) we have about the existence and biological relevance of a protein interaction. We can use confidence scores directly from databases (e.g. Szklarczyk, Gable, et al. 2018), or calculate them independently, which we do in Chapter 3. Generally, we will assume scores are non-negative and that a score of zero is equivalent to the edge not being present in the network.

We will also, unless specified otherwise, assume that edges are undirected, so $e = (u, v) = (v, u) \in E$ means that the nodes u and v are connected. We will further assume that an edge cannot connect a node to itself, i.e. that there are no self-loops in the network. Such a network is called *simple and undirected*. In a simple, undirected network on $N = |V|$ nodes, there can be at most $\binom{N}{2}$ edges. The *network density* is the proportion of pairs of nodes in the network which are connected by edges:

$$p = \frac{|E|}{\binom{N}{2}}. \tag{1.1}$$

Given a network, we say two nodes $u, v \in V$ are *neighbours* if they are connected by an edge $(u, v) \in E$. An edge $e \in V$ is *incident* to a node $u \in V$ if it connects u to some other node $w \in V$. We can also talk about the *distance* $l(u, v)$ between two nodes u and v as the length of a shortest path traversing edges of the network between u and v, where each edge has length one. While in the case of undirected networks, the distance is symmetric $l(u, v) = l(v, u)$, this is not necessarily the case in directed networks (Figure 1.5). The distance is poorly defined for disconnected nodes.

A network can be described by its node and edge sets, or equivalently by its *adjacency matrix*. The adjacency matrix A of the graph $G = (V, E)$ is a $N \times N$

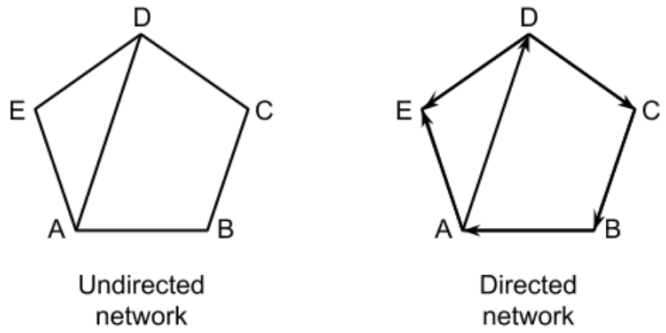

Figure 1.5: Undirected and directed networks. The network on the left is undirected. The distance between A and D is one, since an edge connects the two nodes. However, in the directed network on the right, the distance from A to D is one, and the distance from D to A is three (traversing the edges from DC, CB and BA).

square matrix, in which both rows and columns correspond to nodes in the network. Matrix entries are equal to one for edges and zero for non-edges:

$$A_{u,v} = \begin{cases} 1 & \text{if } (u,v) \in E, \\ 0 & \text{otherwise.} \end{cases} \tag{1.2}$$

The adjacency matrix of the undirected network in Figure 1.5 is therefore

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix} \tag{1.3}$$

where nodes are indexed alphabetically, so $A_{1,2} = 1$ corresponds to the edge between nodes A and B.

The adjacency matrix of a simple, undirected network is symmetric, since $A_{u,v} = 1 \iff (u,v) \in E \iff (v,u) \in E \iff A_{v,u} = 1$. Further, its diagonal entries are zero, since $(u,u) \notin E$. For a weighted network with weight function $w : E \to \mathbb{R}^+$, the *weighted adjacency matrix* is:

$$W_{u,v} = \begin{cases} w(u,v) & \text{if } (u,v) \in E, \\ 0 & \text{otherwise.} \end{cases} \tag{1.4}$$

One of the basic summary statistics for a node v in a network is its *degree* $deg(v)$, i.e. the number of neighbours it has. The degree can be calculated from

the adjacency matrix as

$$deg(v) = \sum_{u \in V} A_{u,v}. \tag{1.5}$$

The *degree sequence* of a graph is the ordered set of degrees observed in the graph $\{deg(v)\}_{v \in V}$. It can be described by *the degree matrix*—a diagonal $N \times N$ matrix where every diagonal entry corresponds to the respective node degree:

$$D_{u,v} = \begin{cases} deg(u) & \text{if } u = v, \\ 0 & \text{if } u \neq v. \end{cases} \tag{1.6}$$

The degree matrix for the undirected network in Figure 1.5 is:

$$D = \begin{pmatrix} 3 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix}. \tag{1.7}$$

The notion of degree and the corresponding degree matrix can also be generalised to the weighted case. The *weighted degree* or *strength* of a node $v \in V$ is calculated by summing over the weights of the edges incident to v, rather than counting these edges. This is equivalent to replacing the adjacency matrix A with the weighted adjacency matrix W in Equation 1.5.

Finally, the graph Laplacian combines the degree and the adjacency matrix as

$$L = D - A. \tag{1.8}$$

The graph Laplacian, which can be redefined and normalised in a number of ways, is related to random walks performed on graphs and spectral analysis (e.g. Spielman 2007). There are many other ways to represent or summarise a network. In the next section we outline some global network summaries, each of which provides a quantitative overview of some aspect of network structure.

1.4.2 Global network summaries

A global one-dimensional network summary is a function $f : G \to \mathbb{R}$, which can be used to describe some aspect of network structure. The number of nodes, number

of edges, and edge density of a network discussed above are all global network summaries. Below we outline four others, which we use throughout Chapters 2–4. Unless otherwise stated, the networks these metrics are applied to are simple, undirected, and unweighted.

Number of connected components

The undirected network in Figure 1.5 is connected, i.e. there exists a path from every node to every other node in it. However, this is not the case for all networks—in a world-wide rail network connecting cities through direct rail links, Europe would be disconnected from Australia, say. Figure 1.6 shows another disconnected network. By a maximal connected component or simply a *connected component*

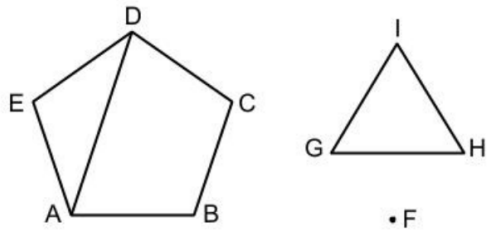

Figure 1.6: A disconnected network. This network on $N = 9$ nodes contains three connected components: $\{A, B, C, D, E\}$, $\{F\}$, and $\{G, H, I\}$. There exist no edges across the components.

we understand a group of nodes, which are connected to each other, but which are not connected to any of the remaining nodes in the network. The network in Figure 1.6 contains three such components: $\{A, B, C, D, E\}$, $\{F\}$, and $\{G, H, I\}$. One of these, $\{F\}$, is a single node, which we call an *isolated node*. Since every node belongs to a single connected component, the components of a network define a partition of the node set. We call the number of these components, or equivalently the size of the partition, the *number of connected components*. The number of connected components of a network is equal to the multiplicity of the eigenvalue zero of the graph Laplacian (Newman 2018b).

Size of largest connected component

Many real-world networks are connected, or nearly connected. When they are not, their different components can often be thought of as independent entities.

The phenomenon of a *giant connected component* arises in networks where one connected component accounts for the vast majority of network edges. Giant connected components are often present in real-world networks, including in protein interaction networks (e.g. Reka Albert 2005). The size of the largest connected component, i.e. the number of nodes in that component, is one way of measuring how close a network is to being fully connected. This can be done, for example, by calculating what proportion of the nodes in a network belong to its largest connected component.

Global clustering coefficient

While the number of edges and the number and size of the components of a network describe in very general terms how connected it is, none of these metrics describe the presence or absence of patterns in the network. Such patterns are known as *graphlets* or *motifs*: small subgraphs which occur frequently, perhaps more often than one would expect at random, in large networks (Milo et al. 2002). Graphlet distributions can be used to not only describe network structure, but also to compare networks (Pržulj 2007; Wegner et al. 2018).

Perhaps the most frequently studied graphlet is the triangle, i.e. a set of three nodes, which are all connected by pairwise edges. A triangle implies a transitive relationship between the three nodes (*"If u connects to v, and v connects to w, then u connects to w."*). The *global clustering coefficient*, also known as the transitivity, is a measure of how often edges in a network form triangles. It is calculated as the proportion of connected triplets, which also form triangles:

$$C_{global} = \frac{3 \times \#\ of\ triangles}{\#\ connected\ triplets}. \tag{1.9}$$

The factor of three takes into account symmetry—for every triangle (u, v, w) there are three distinct connected triplets: $u - v - w$, $v - w - u$, and $w - u - v$. The

global clustering coefficient ranges from zero (no triangles in the network), to one (all connected triplets are triangles). All connected components of a network with $C_{global} = 1$ must be complete graphs.

Natural connectivity

Triangles are redundant in the sense that even if one of the edges was removed, the three nodes would remain connected to each other via the remaining two edges. More generally, the same is true for any closed path or cycle. Therefore, the more closed paths there are in a network, the more robust we may expect it to be under perturbation. This type of robustness is of importance for the viability of biological and other networks (e.g. Azevedo and Moreira-Filho 2015).

One way of measuring this robustness is through the *natural connectivity* of a network (Jun et al. 2010). Let n_k be the number of closed paths of length k in a network. Then, the Estrada index of the network is:

$$S = \sum_{k=0}^{\infty} \frac{n_k}{k!} = \sum_{u=1}^{N} e^{\lambda_u}, \tag{1.10}$$

where λ_u are the eigenvalues of the adjacency matrix of the network (Estrada 2000). The Estrada index was originally developed to measure the "compactness" of molecules by representing them as graphs. Shorter cycles, or closed paths, are more indicative of compactness or robustness than longer ones, which justifies the scaling by $k!$ in Equation 1.10. The natural connectivity is a transformation of the Estrada index:

$$NC(G) = log\Big(\frac{S}{N}\Big) = log\Big(\frac{1}{N}\sum_{u=1}^{N} e^{\lambda_u}\Big). \tag{1.11}$$

1.4.3 Ego networks and communities

Rather than describing the network as a whole, and calculating various properties of it, it is often useful to consider different parts of the network, or *subnetworks*. A pathway or a set of related pathways may, for example, form a subnetwork within the protein interaction network of a particular organism. Subnetworks of

interest can be identified through annotation data, but they can also be built directly from the protein interaction data.

One type of subnetworks frequently studied in the social sciences are *ego networks*. Ego networks are built from one seed node (the ego), and all of its neighbours (the alters), together with all edges connecting the ego to the alters, and often alters to other alters (Killworth et al. 1990). This approach to network building is often employed in PINs, when a seed set of proteins and all of their interacting partners may be studied together (e.g. Han et al. 2017). In this thesis, we consider two types of ego networks. A one-step ego network of a node $v \in V$ is the network built from that node and its immediate neighbours $\Gamma(v)$, as well as any links between these nodes. Letting $\Gamma_1(v) = \{v \cup \Gamma(v)\} = \{u : l(u, v) \leq 1\}$, we define the one-step ego network as:

$$ego_1(v) = (V, E \cap (\Gamma_1(v) \times \Gamma_1(v))). \tag{1.12}$$

We also consider two-step ego networks, which in addition include all nodes of distance two to the ego. Setting $\Gamma_2(v) = \{u : l(u, v) \leq 2\}$, the step-two ego network of v is

$$ego_2(v) = (V, E \cap (\Gamma_2(v) \times \Gamma_2(v))). \tag{1.13}$$

We construct both one-step and two-step ego networks by preserving all present edges between the selected nodes.

Ego networks describe what a network looks around a particular node of interest. However, sometimes it is of interest how a group of nodes connect to each other. For example, in the context of protein interaction data, proteins are known to perform their function in groups, or modules (Hartwell et al. 1999). Therefore, the identification of such modules without complete *a priori* annotation knowledge can aid function prediction (Pereira-Leal et al. 2004).

This problem can be tackled through community detection. Community detection algorithms aim to identify groups of nodes in a network which are densely connected with each other, but are relatively sparsely connected to the rest of the network. Communities are often assumed to be non-overlapping. Two well-known

algorithms for detecting non-overlapping communities are Girvan–Newman (Girvan and Newman 2002) and Louvain (Blondel et al. 2008). Overlapping communities can be identified using algorithms like BigClam (J. Yang and Leskovec 2013) and Link clustering (Ahn et al. 2010). Enrichment analysis of functional annotation within communities is frequently employed to benchmark different algorithms (Subramanian et al. 2005).

1.4.4 Local network summaries

While global network summaries provide a macroscopic view of network structure, and subgraphs and communities can be used to investigate a group of nodes, local network summaries (often also called centralities) quantify the role of a particular node with respect to the rest of the network. By a local summary we understand a function $f_G : V \to \mathbb{R}$, which assigns a value to each node in the network. Throughout this thesis we will refer to local network summaries as *node metrics*. The degree of a node mentioned earlier in this chapter is one such metric.

Local clustering coefficient

While the global clustering coefficient measures how frequently connected triplets in a network form triangles, the local clustering coefficient is an analogous metric, defined for a particular node:

$$C_{local}(v) = \frac{\#\ of\ pairs\ of\ neighbours\ of\ v\ that\ are\ connected}{\#\ of\ pairs\ of\ neighbours\ of\ v}. \tag{1.14}$$

As the number of neighbours of $v \in V$ is $deg(v)$, the denominator above is equal to $\binom{deg(v)}{2}$. The *average local clustering coefficient*,

$$\bar{C} = \frac{1}{N} \sum_{v \in V} C_{local}(v) \tag{1.15}$$

is a global network summary similar to the global clustering coefficient. Higher degree nodes, which account for more connected triplets, will have a bigger impact on the global clustering coefficient than on the average local clustering. This means that the two measures, while similar in nature, in general result in different values.

Redundancy

Like the global clustering coefficient, local clustering is related to robustness or redundancy in a network. In the context of social science, we may expect that if a person has two friends, then those friends are also very likely to be friends themselves (i.e. we may expect high local clustering); the reverse is indicative of a structural hole in the network (Burt 2009). A related concept is that of *redundancy*. The redundancy of a node $v \in V$ is the average number of edges from a neighbour of v to other neighbours of v. This can be simplified to

$$R(v) = C_{local}(v) \times (deg(v) - 1). \tag{1.16}$$

Closeness centrality

The local clustering coefficient describes the immediate neighbourhood of a node. It is also possible to quantify a node's placement within a network with respect to its distance to other nodes. One way to do this is through the closeness centrality (Freeman 1978). It is calculated as:

$$closeness\ (v) = \frac{1}{\sum_{u \neq v} l(u, v)}, \tag{1.17}$$

where $l(u, v)$ is the distance between u and v. If the two nodes u and v belong to different connected components of the network, their distance is not well defined. Therefore, closeness is ill-defined for disconnected networks. This can be resolved by setting $l(u, v) = |V| = N$ for disconnected pairs of nodes u and v (Csardi, Nepusz, et al. 2006).

Harmonic centrality

A related measure is that of *harmonic centrality* (Marchiori and Latora 2000). Harmonic centrality also measures how close a node is to all other nodes in the network, but is calculated differently from closeness:

$$H(v) = \sum_{u \neq v} \frac{1}{l(u, v)}. \tag{1.18}$$

It is defined unambiguously for networks with multiple connected components by taking $1/l(u, v) = 0$ for disconnected pairs of nodes.

Betweenness centrality

Both closeness and harmonic centrality assess how well-connected, or central, a node is to the rest of the network. Information from a node with high closeness may reach the rest of the network quickly, since the node is only a short distance from all other nodes. This may be a desirable property in some contexts, such as when information is shared, and an undesirable property in others, such as disease spread (Salathé and J. H. Jones 2010).

A related but conceptually different property is that of node *betweenness* (Anthonisse 1971; Freeman 1977). The betweenness of a node measures how important the node is for connecting different parts of the network. A node is important for information spread if many shortest paths between other pairs of nodes in the network run through it. Conversely, it is not essential if removing it does not affect shortest paths significantly. To measure betweenness, for a triplet of connected nodes $u, v, w \in V$ we let:

- σ_{uv} = the number of distinct shortest paths between u and v,
- σ_{uv}^{w} = the number of distinct shortest paths between u and v which pass through w.

Then the betweenness of the node w is defined as

$$\textit{betweenness}\ (w) = \sum_{u,v \neq w} \frac{\sigma_{uv}^{w}}{\sigma_{uv}}, \tag{1.19}$$

where the sum is over node pairs (u, v) in the same component as w.

PageRank

The degree of a node measures the size of the immediate reach of a node $v \in V$, and closeness centrality measures how easy it is to reach any other node from v. From the perspective of v, all other nodes of the network are in a sense of equal importance. For example, each neighbour w of v contributes 1 to its degree, regardless of the degree (or other properties) of w. However, in social networks a

different type of centrality, or "importance" exists: a person can be "important" if they know a large number of people, or if they know other "important" people.

This property can be captured by a range of node metrics, the simplest of which is the *eigenvector centrality* (Bonacich 1987). The eigenvector centrality of a node has the property that it is proportional to the sum of the eigenvector centralities of that node's neighbours. Its name reflects the fact that the vector of eigenvector centralities $x = (x_1, \ldots, x_N)^T$ satisfies

$$Ax = \lambda_{(1)} x, \tag{1.20}$$

where $\lambda_{(1)}$ is the leading, i.e. largest, eigenvalue of the adjacency matrix A.

A related measure is the *Katz centrality* (Katz 1953). Like the eigenvector centrality, the Katz centrality of a node depends linearly on the Katz centralities of its neighbours. The difference is that in addition to the linear dependence, each node is also given a constant centrality term. In matrix form this corresponds to

$$x = \alpha A x + \beta, \tag{1.21}$$

where α is the linear dependence factor, and β is the constant. Usually this is simplified by setting $\beta = 1$, in which case the Katz centrality can be calculated as

$$x = (I - \alpha A)^{-1}. \tag{1.22}$$

An undesirable feature of both the eigenvalue and the Katz centralities is that a node of high centrality will boost the centrality of all of its neighbours, regardless of how many of them there are. In the context of the World Wide Web, a frequently linked-to website, such as a news website, would increase the eigenvector centrality of the thousands other web pages it links to. This effect can be damped by scaling the contribution of each node to its neighbours by the node's degree. The PageRank centrality, historically employed by Google's search engine, does this (Brin and Page 1998). In matrix form, the degree scaling can be formalised as

$$x = \alpha A D^{-1} x + \beta, \tag{1.23}$$

where D is the degree matrix. Setting $\beta = 1$ gives

$$x = D(D - \alpha A)^{-1}. \tag{1.24}$$

The constant α is known as a *damping factor* and is usually set around $\alpha = 0.85$. For details on the existence and uniqueness of non-negative centrality vectors x solving Equations 1.20–1.24 above, refer to Newman 2018b.

1.4.5 Random graph models

So far we have discussed how to calculate different properties of a single, isolated network. However, networks are often studied by comparison to each other. We may wish to know whether a network is particularly dense, or particularly modular, or whether it is more or less robust to perturbations, for example. In order to benchmark algorithms on networks, or study their properties, randomly generated networks are often employed. Numerous different models exist—see Newman 2018b for an extensive review of commonly used models. Here we describe three models which are employed in Chapters 2–4.

The Erdős–Rényi random graph model

Perhaps the simplest model for generating random graphs is the *Erdős–Rényi random graph* model, also known as the $\mathcal{G}(N, p)$ model, or as the Bernoulli random graph model (Solomonoff and Rapoport 1951; Erdős and Rényi 1960; Erdős and Rényi 1961). The $\mathcal{G}(N, p)$ model generates networks on N nodes such that no multiple edges or self-loops exist and each of the $\binom{N}{2}$ possible edges exists independently of every other edge with probability p. Thus all simple networks on N nodes with $e(G)$ edges are equally likely and occur with probability

$$\mathbb{P}(G) = p^{e(G)}(1 - p)^{\binom{N}{2} - e(G)}. \tag{1.25}$$

Bernoulli random graphs are often used as a benchmark to compare against. Further, many properties of Bernoulli random graphs have been analytically studied—for example, it is possible to derive formal results about their edge density and degree distributions, as well as about the sizes of their connected components and about motif counts (e.g. Ross et al. 2011).

The stochastic block model

The stochastic block model (SBM), also known as the planted partition model, is an extension of the Bernoulli random graph such that community structure can be introduced to the graph (Holland et al. 1983). The nodes are first partitioned into r communities $C_1, C_2, \ldots, C_r$, and then edges are generated independently between every pair of nodes $v \in C_i, u \in C_j$ with probability P_{ij}. Thus, rather then generating all edges with probability p, under the SBM, an $r \times r$ symmetric probability matrix P is provided. Often intra-community edges are assumed to occur with higher probability than inter-community edges, i.e. for $i \neq j$, generally $P_{ii} > P_{ij}$. Two variants of the model are frequently employed—where community allocation is fixed and part of model input, and where the community structure is also randomly generated.

SBMs are often employed to benchmark community detection algorithms (e.g. Martin et al. 2016). They can also be used to perform community detection—e.g. Martin et al. 2016 employ a belief propagation algorithm to fit an SBM to uncertain data, in order to recover community structure.

The Bernoulli random graph is an extreme case of the SBM, where there is a single community, or alternatively, where $P_{ij} = p$ for all i, j. The other extreme case—where every node belongs to a separate community, or where every pair of nodes (u, v) is associated with its own probability p_{uv}, is employed in Chapter 3.

The configuration model

An alternative way of prescribing a stricter structure to random graph models than provided by the $\mathcal{G}(N, p)$ model is to specify a degree sequence and then generate a graph uniformly at random from all graphs with that degree sequence. This is known as a *configuration model* (Molloy and Reed 1995). Several variants exist, using different implementations. Unlike the Bernoulli random graph and the SBM, configuration models are generally allowed to have multi-edges and self-loops. In this thesis, we generate networks uniformly at random from all simple, undirected,

unweighted networks given a degree sequence $\{d_1, d_2, \ldots, d_N\}$. We employ this configuration model in order to construct synthetic PIN-like networks in Chapter 3.

Not all non-negative integer sequences can be used as input to the configuration model. The exact degree sequence provided must be graphical (Choudum 1986). However, the degree distribution condition can also be relaxed—e.g. the Chung–Lu model generates networks with the correct expected degree distribution (F. Chung and Lu 2002a; F. Chung and Lu 2002b).

Other notable models

Many other random graph models exist. Some popular models are preferential attachment, Watts–Strogatz, exponential random graph models, and duplication–divergence models.

Preferential attachment models aim to capture a "rich-get-richer" dynamic in network growth (Barabási and Réka Albert 1999). Networks are typically built from a small seed of potentially connected nodes. When a new node is introduced into the network, it tends to forge connections to nodes that already have high degree. This results in a power-law degree distribution. Power-law networks are widely studied, as it is frequently observed that real-world networks obey a power-law (Mitzenmacher 2004). However, this claim is often not formally tested and has been subject to debate (Broido and Clauset 2019).

Another frequently observed property in real-world networks is the small-world phenomenon, also known as "six degrees of separation" (Milgram 1967). It dictates that even in very large networks—such as the human social network encompassing the whole world—a relatively small number of friendships separates any two randomly chosen individuals. A random graph model frequently used to illustrate the combination of low network density and a small distance between nodes is the Watts–Strogatz model (Watts and Strogatz 1998).

It is often desirable to fit observed network properties, such as a power-law degree distribution or a low average distance, to a random graph model. A rich family of models which allows this are exponential random graph models or ERGMs

(Robins et al. 2007). Given a set of summary statistics $s(\cdot)$, a network G is generated with probability

$$\pi(G) \propto e^{\theta^T s(G)}, \tag{1.26}$$

where the parameters θ are fitted to the data of interest. The summary statistics encoded by $s(\cdot)$ can be arbitrary, making the model extremely flexible. ERGMs are routinely employed in neuroscience (Simpson et al. 2011). However, estimating θ can be computationally intensive and intractable for larger networks, such as protein interaction networks.

A way to model protein interaction networks is the duplication–divergence model (Ispolatov et al. 2005). Networks generated from the duplication–divergence model are grown from a seed, like preferential attachment networks. New nodes are introduced by duplicating an existing node and its connections, akin to a gene duplication event, and by rewiring some of its edges, to mimic divergence due to random mutation. The duplication–divergence model has been shown to fit some, but far from all, protein interaction networks (Ospina-Forero et al. 2018).

1.4.6 Uncertainty and errors on networks

In Section 1.3.1 we discussed some of the methods used to detect or infer protein–protein interactions and the experimental errors associated with them, as well as estimates on how many interactions are yet to be detected. These errors imply that protein interaction networks built from state-of-the-art data are not perfect representations of cellular biology. False negatives, or pairwise interactions which have simply not been tested, can be thought of as missing edges in PINs, while false positives are erroneous observed edges.

Predicting missing network edges, as well as identifying spurious ones, is a subject of interest both in theoretical study and in a range of applications. It is often carried out based on the community structure of a network: if two communities are known or suspected to exist, we may predict additional intra-community edges, and identify some inter-community edges as spurious. This can be done by fitting

the data to a model, such as SBM (Guimerà and Sales-Pardo 2009) or a hierarchical random graph model (Clauset et al. 2008). Validation of such approaches is typically performed by taking either a fixed real-world network, or generating a random one, and introducing some artificial noise process on it, such as random edge deletion. However, given the complexity, biases, and incompleteness in protein interaction data, it is hard to argue that such validation is sufficient to show that link prediction tools are a reliable way of "correcting" the available data.

A more data-driven approach to validation may be to treat interaction detection as a time-evolving process, and predict changes in the network at time $t = t_1$ based on previously observed data at time $t = t_0 < t_1$. This has been done in the context of social interaction networks (Liben-Nowell and Kleinberg 2007). However, social interactions can be presumed to evolve at a steady rate over time, whereas significant changes in protein interaction data are often due to single high-throughput experiments (e.g. Rolland et al. 2014; Luck et al. 2019), making them hard to predict rare events.

Rather than trying to predict missing edges in PINs or to identify spurious ones, it may be beneficial to incorporate uncertainty directly into the network and carry out downstream network inference with respect to that uncertainty. Ahnert et al. 2007 assume that instead of fixed edges, we observe the probability that each edge exists, and derive formal results about the expectation of network metrics such as degree distributions, clustering coefficients, and network diameter. Martin et al. 2016 use the same framework to perform uncertainty-aware community detection on networks. However, in both cases a number of simplifying assumptions are made, the key one of which is that edges in the network behave like independent random variables. Batch effects in experimental assays, research bias, and the different modes of applicability of different interaction detection experiments imply that this would be far from the case in protein interaction data.

More generally, a Bayesian framework for network data can allow the incorporation of uncertainty, or noise, in observed networks. Under the assumption that a single, "true" protein interaction network exists, one can think of protein

interaction data as observations conditional on the true network state (Newman 2018a). The benefit of such an approach is that a network prior allows incorporating known or expected properties of the network, while a flexible data model implies that arbitrary, heterogeneous interaction data can be used. However, to our knowledge only the independent edge case for the network has been studied thus far (Newman 2018a; Peixoto 2018).

Any Bayesian framework, be it aimed at network analysis or not, allows the combination of *a priori* knowledge with observational data. The more data is observed, the stronger its influence is on any *a posteriori* model predictions. If sufficient PPI data existed, the flawed assumption that edges, or interactions, occur independently of each other in the network, would have little impact on the PIN posterior. However, as discussed in Section 1.3.1 this is far from the case—many protein pairs have not been screened for interaction, or have not been screened multiple times or using different techniques. Therefore, we may expect the network prior to be very influential on the model posterior.

A way of finding a good prior, or any good PIN model, would be to test a number of models, and either choose the best one or study how and where they agree (Vallès-Català et al. 2018). However, such ensemble approaches are not guaranteed to improve predictive power (Stumpf 2019). Due to the complexity of the available data, validation remains difficult. From the perspective of network analysis, handling uncertainty in protein interaction data continues to be an open problem.

1.5 Thesis outline

The range of data sources and the different approaches to data curation mean that many different protein interactions can be built to model the same organism (or, more generally, the same set of biological processes). These networks can then be analysed using a number of tools in order to hopefully gain biological insight. In order for such analysis to be reliable, we may require that the same—or at least similar—results must be obtained across different networks. In Chapter 2 we discuss how such an approach can be employed in the context of scored protein

interaction data, where a threshold is applied so only data of sufficiently high quality is included in the network. The work contained in this chapter appears in the following publication:
Bozhilova, L. V., Whitmore, A. V., Wray, J., Reinert, G., & Deane, C. M. (2019). Measuring rank robustness in scored protein interaction networks. BMC Bioinformatics, 20(1), 446.

We follow Chapter 2 with a discussion on how experimental uncertainty can be formally incorporated directly into network construction and network analysis. In Chapter 3 we discuss how uncertain networks can be employed for this purpose. This approach relates to work done by Ahnert et al. 2007 and Martin et al. 2016. However, rather than discussing results obtained from a single simplifying assumption, we focus on how different underlying network models can affect network structure and therefore argue that given state-of-the-art data, such modelling is of limited applicability.

After discussing both yeast two-hybrid and gene co-expression data in Chapter 3, in Chapter 4 we focus explicitly on gene expression data. We note that unlike physical interactions, genetic interactions as described by gene co-expression are not necessarily associated with an unobserved but fixed "true" interaction. We show that despite this, a notion of uncertainty persists in the context of co-expression data. We have developed a software package, COGENT, which quantifies such uncertainty. COGENT can be used as an annotation-free tool for network validation. In particular, it has been developed in order to help select between different network construction methodologies. The work contained in Chapter 4 is also described in a separate manuscript, to be submitted for publication.

We conclude by discussing possible directions for future work.

Why, sometimes I've believed as many as six impossible things before breakfast.

— Lewis Carroll, *Alice in Wonderland*

2

Measuring rank robustness in scored protein interaction networks

Contents

This chapter is based on work described in the following publication:

Bozhilova, L. V., Whitmore, A. V., Wray, J., Reinert, G., & Deane, C. M. (2019). Measuring rank robustness in scored protein interaction networks. BMC Bioinformatics, 20(1), 446.

2.1 Introduction

Protein interaction networks (PINs) are models of cellular architecture in which proteins are represented by nodes and the biologically meaningful interactions between them are represented by edges. As discussed in Chapter 1, PIN analysis has a wide range of applications, including drug target identification (Hopkins 2008). A frequent aim of such analysis is to identify key actors in the network, for example for the purposes of drug target choice (Navlakha and Kingsford 2010). This can be done by calculating a node metric of interest, such as degree or betweenness centrality, and selecting the nodes in the network with the highest metric values.

As with all PIN analyses, the results will depend on the data used to construct the network. As described in Chapter 1, there exist a number of publicly available protein–protein interaction databases. Some of these, such as STRING (Szklarczyk, Franceschini, et al. 2014) and HitPredict (López et al. 2015), quantify the strength of the supporting evidence for each reported interaction by assigning a confidence score to it. While useful, these scores are not readily interpretable, and, as we illustrate, tend not to be comparable across databases. They are designed to provide a comparison between different interactions (an interaction with confidence score of 0.40 is supported by weaker evidence than an interaction with score of 0.90), so researchers can control strength of evidence by imposing a score threshold. STRING, for example, suggests thresholds of 0.15 (low confidence), 0.40 (medium confidence), 0.70 (high confidence) or 0.90 (highest confidence), whereas HitPredict identifies all interactions scoring below 0.28 as medium-high confidence, and interactions scoring above as high confidence.

The wide range of available databases and quality assessment tools for protein interaction data mean that for many systems of interest multiple networks can be built. For example, the STRING or HitPredict databases can be used with different thresholds. Given interaction detection error rates, and the incomplete coverage of interaction detection experiments (e.g. Hart et al. 2006; Rolland et al. 2014), it is extremely unlikely that any one of these PINs is a perfect representation of the underlying biological processes it aims to model, regardless of how it is

constructed. PIN analysis should therefore ideally be performed with awareness of the uncertainty observed in the data.

Rather than considering a single network, the observation of which is subject to a difficult to model noise process, assessing the robustness of a PIN analysis pipeline can be done by repeating the analysis across different networks representing the same part of the interactome. One way to do this would be to consider building different networks from the same scored interaction database by varying the confidence score threshold. We postulate that network features which are persistent across different thresholds are more likely to be informative of the biological state of interest than features which are only present at isolated thresholds. This hypothesis is in line with network research in neuroscience, where a similar heuristic is employed to identify which parts of a brain network are active across different observations (e.g. Curto 2017).

In this chapter we provide a framework for assessing the robustness of node metrics to threshold choice. Our framework is based on a measure of rank similarity described by Trajanovski et al. 2013 and on an extension of it we introduce. We introduce three robustness measures—rank continuity, rank identifiability, and rank instability—which can be used to quantify how consistent a node metric is across different thresholds. Our methodology is particularly relevant to cases where a node metric is used to identify highly ranking nodes, which may correspond to key proteins in a particular process, for example for the purposes of drug target identification (e.g. Abraham et al. 2016).

By analysing the effects of threshold change on a set of twenty-five node metrics across four scored PINs we show that some node metrics tend to be more robust—and are therefore possibly more relevant to biological research—than others. We identify the number of edges in the one-step ego network, and LOUD (leave-one-out-difference) average redundancy, LOUD number of edges in the one-step ego network, and LOUD natural connectivity, as significantly more robust to threshold choice than more commonly used metrics, such as local clustering coefficient, betweenness centrality, and in some cases even degree.

Promisingly, our results show good agreement between networks from different organisms and databases. Complemented with analysis of synthetic data, we further show that robustness depends both on network topology and on score allocation across network edges.

2.2 Materials and methods

2.2.1 Protein interaction and synthetic networks

In order to assess the rank robustness of different network metrics, four scored protein interaction networks were used. The networks ranged across two databases and three organisms. A confidence score quantifying the reliability of available interaction evidence was available for each detected edge across all four networks.

Three organism networks—*Plasmodium vivax* (retrieved March 2018), *Escherichia coli*, and *Saccharomyces cerevisiae* (both retrieved February 2018)—were obtained from STRING (Szklarczyk, Franceschini, et al. 2014). In the text below, these are referred to as PVX, ECOLI, and YEAST, respectively. STRING contains both physical and functional association data, collected across a range of experimental and *in silico* interaction detection techniques. The organisms were chosen because they are model organisms with higher-than-average coverage of protein–protein interaction screens, while also having relatively small proteomes, thus reducing the computational cost of the analysis.

In order to allow for a comparison between databases, the interaction network for *S. cerevisiae* from HitPredict (López et al. 2015) was also downloaded. Unlike STRING, HitPredict is a curated database containing only medium and high-confidence physical interactions. We refer to this network as HPRED.

Filters were applied to remove duplicate interaction records, self-interactions, and interactions to proteins of other organisms. Only combined interaction scores were considered in all four cases, ignoring any available sub-scores. Confidence score distributions for the four networks can be seen in Figure 2.1.

Two synthetic scored networks were also analysed. One, SYN-GNP, was generated using a Bernoulli random graph model on $N = 500$ nodes and with

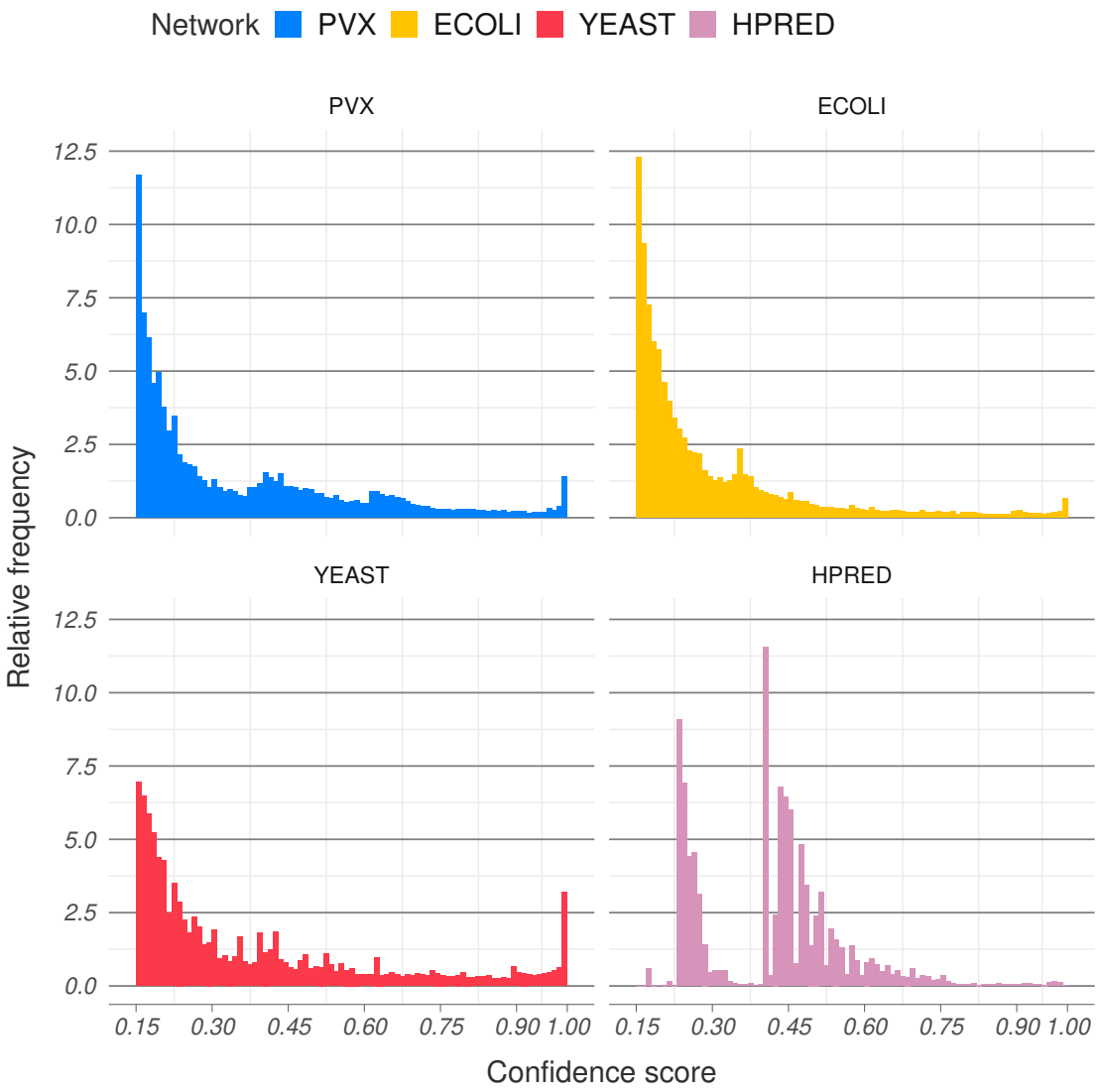

Figure 2.1: Confidence score distributions in each of the four studied PINs. Bin width in all four cases has been set to 0.01. Scores from the HitPredict network (bottom right) follow a different distribution and cannot necessarily be interpreted in the same way as STRING scores.

edges sampled independently at random with probability $p = 0.06$. The value of p was chosen to be close to network density in the STRING PINs before thresholding. Edge scores were sampled with replacement from the *P. vivax* confidence score distribution. The second network, SYN-PVX, was an induced random subgraph of the *P. vivax* network on $N = 1000$ nodes. The organism network to sample from was chosen arbitrarily from the three STRING networks. The available edge scores for the subgraph were randomly rearranged and placed over the fixed edges.

This resulted in a network which is PIN-like in topology, but which contains no local dependency between edge scores. A summary of all six analysed networks can be found in Table 2.1.

Name	Network	Number of nodes	Number of edges	Edge density
PVX	*P. vivax*, STRING	3255	344691	∼0.065
ECOLI	*E. coli*, STRING	4144	583440	∼0.068
YEAST	*S. cerevisiae*, STRING	6418	939998	∼0.046
HPRED	*S. cerevisiae*, HitPredict	5673	113001	∼0.007
SYN-GNP	Synthetic, Bernoulli	500	7459	∼0.060
SYN-PVX	Synthetic, randomised *P. vivax*	1000	30516	∼0.061

Table 2.1: Summary statistics for the six analysed networks. The left-most column corresponds to the names the networks are referred as later in the text. The number of edges and edge density refer to the all scored edges before any threshold is applied to the network.

2.2.2 Thresholding

Applying a threshold θ to a scored PIN means discarding all edges in the network with scores strictly lower than θ and creating a simple, unweighted network from the remaining edges. The node set is not altered. Figure 2.2 provides a schematic of how thresholds are applied.

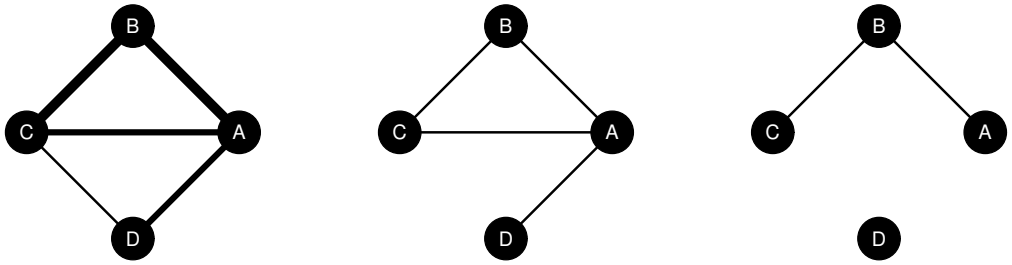

Figure 2.2: Thresholding scored networks. A scored network, with edge widths corresponding to confidence scores (left). At a low threshold, only the lowest scoring edge CD is removed (middle). At a higher threshold, only the highest scoring edges AB and BC remain in the network (right). Edge scores are otherwise ignored in the thresholded networks.

All reported confidence scores in the analysed PINs were between 0.15 and 1.00. Thresholds were applied from 0.15 to 0.99 inclusive at 0.01 intervals, resulting in a set

of 85 distinct thresholded networks for each of the scored PINs. The same node set was preserved across all 85 thresholded networks, even when thresholding resulted in isolating nodes from the rest of the network. An overly stringent threshold may remove so many interactions that network structure is destroyed. However, even at the highest thresholds we consider, a giant connected component accounts for most of the nodes in each protein interaction network (91% of nodes in PVX, 79% in ECOLI, 93% in YEAST, and all but two nodes in HPRED).

The majority of interactions in STRING are re-scored across different database releases (see Chapter 1 Section 1.3.2) which indicates that the scores themselves should be treated with some error tolerance. In order to take this into account while retaining score interpretability, a wide medium-high confidence region was set between 0.60 and 0.90. Medium-high confidence scores in STRING occur at similar, low frequencies across the three organisms (Figure 2.1). HitPredict generally contains higher quality data and employs a different, more stringent scoring procedure—while an interaction scoring 0.40 in STRING would be considered medium confidence, the same score in HitPredict would indicate high confidence. Since HitPredict scores above 0.28 are considered to be high-confidence (López et al. 2015), the truncated medium-high confidence region for the HPRED network was set between 0.15 and 0.28.

2.2.3 Metric extraction and ranking

The rank robustness of twenty-five node metrics was studied. These included twelve node centralities, and thirteen leave-one-out difference (LOUD) global network summaries.

Commonly used metrics, such as degree, local clustering coefficient, betweenness and closeness centralities were included in the node metric set. In addition, metrics based on the size and density of the one-step and two-step ego networks for each node were calculated.

The LOUD metrics were used as a way to assess the effect of perturbing the network by isolating each node in turn. LOUD metrics for a node v are based on

Name	Details
Degree	Number of neighbours of v
Local clustering	Proportion of pairs of neighbours of v which are also connected
Redundancy	(Local clustering) $\times$ (Degree - 1) (Borgatti 1997)
PageRank	Calculated with the default damping factor $d = 0.85$ (Brin and Page 1998)
Closeness	Reciprocal to the sum over all u of node-to-node distances $d(u, v)$ (Freeman 1978)
Harmonic centrality	The sum over all u of $1/d(u, v)$ (Marchiori and Latora 2000)
Betweenness	Measures how many shortest paths a node v contributes to (Freeman 1978)
$e_{one}(v)$	Number of edges in the one-step ego-network of v
$n_{two}(v)$	Number of nodes in the two-step ego-network of v
$n_{diff}(v)$	Number of nodes that have exactly distance two to v
$n_{sqdiff}(v)$	A measure of relative local density calculated as $n_{two}(v) - degree(v)^2$
$n_{ratio}(v)$	The ratio of one-step to two-step neighbourhood sizes for v
LOUD Average local clustering	$f(G)$ is the average local clustering coefficient
LOUD Global clustering	$f(G)$ is the global clustering coefficient
LOUD Average redundancy	$f(G)$ is the average redundancy
LOUD Average closeness	$f(G)$ is the average closeness
LOUD Average path length	$f(G)$ is the average path length
LOUD Number of connected pairs	$f(G)$ is the number of pairs of nodes, which are in the same connected component
LOUD Average betweenness	$f(G)$ is the average betweenness
LOUD Natural connectivity	$f(G)$ is the natural connectivity (Jun et al. 2010)
LOUD Average $e_{one}(v)$	$f(G)$ is the average $e_{one}(v)$
LOUD Average $n_{two}(v)$	$f(G)$ is the average $n_{two}(v)$
LOUD Average $n_{diff}(v)$	$f(G)$ is the average $n_{diff}(v)$
LOUD Average $n_{sqdiff}(v)$	$f(G)$ is the average $n_{sqdiff}(v)$
LOUD Average $n_{ratio}(v)$	$f(G)$ is the average $n_{ratio}(v)$

Table 2.2: The complete set of twenty-five standard and LOUD metrics, calculated at each node v. Standard metrics are above the line break. LOUD metrics are below the line break.

global metrics $f(\cdot)$ calculated both for each thresholded network $G = (V, E)$, and for a modified network, where node v has been isolated from its neighbours $G_v = (V, E \setminus \{(u, v)|u \in V\})$. The difference between the two metrics is denoted by $f_{LOUD}(v) = f(G) - f(G_v)$. The global metrics studied included, wherever applicable, local metrics averaged over the entire network (e.g. average local clustering coefficient), as well as metrics which are by definition global (e.g. global clustering coefficient). The set also included natural connectivity, a spectral metric designed to measure the overall robustness of a network (Jun et al. 2010). Due to the associated computational costs, LOUD metrics were only calculated for nodes with degree at least two. Metric values for isolated nodes and nodes of degree one were set to NA (not available).

The nodes in each thresholded network were ranked by each of the metrics, with high ranks corresponding to high metric values. Node ranks for nodes for which LOUD metrics were not evaluated were set to first, i.e. smallest. Ties were resolved at random, independently across different metrics and different thresholds. A full list of node metrics can be found in Table 2.2. Further details on their computation

are given in Chapter 1 Sections 1.4.2 and 1.4.4.

2.2.4 Evaluation of rank robustness

As the threshold applied to the scored networks increases, the networks become lower in edge density and node metric values will be affected—for example, node degrees will decrease. In order to assess the robustness of each metric to changes in the threshold, the node rankings induced by the metric at different thresholds were compared instead of the calculated raw values.

Rank similarity is typically measured by a rank correlation coefficient such as Spearman or Kendall. These coefficients are used to compare whole rank vectors. In the context of bioinformatics applications, node metrics are often used to identify the key actors in a particular process, and therefore it is natural to focus on the highest ranking nodes only. In order to do this, robustness analysis was based on the rank similarity measure M_k proposed by Trajanovski et al. 2013. Briefly, rank k-similarity measures whether two rank vectors identify similar sets of highest ranking nodes. If both rankings identify nearly the same sets of nodes as high-ranking, their k-similarity will be close to one. Conversely, if they do not agree on which nodes are among the highest ranking, their k-similarity will be close to zero. Formally, the measure is defined as follows.

A ranking A is a vector $A = \{A(v) : v \in \{1, \ldots, N\}\}$ of ranks assigned to the network nodes $v \in \{1, \ldots, N\}$, e.g. by considering their degree at a particular threshold. Ties are resolved at random, so each ranking is a permutation of $\{1, \ldots, N\}$. The k-similarity of two rankings A_θ and A_μ, obtained at thresholds θ and μ respectively, is the scaled overlap between their $100k\%$ highest ranking nodes, where $k \in (0; 1]$,

$$M_k(A_\theta, A_\mu | k) = \frac{|\{v : \min(A_\theta(v), A_\mu(v)) > (N(1-k))\}|}{Nk}. \tag{2.1}$$

Rank k-similarity is based on a simple set overlap between the highest ranking nodes in A_θ and A_μ. One could, in theory, design an alternative similarity measure based on a correlation coefficient between these highest ranking nodes. However,

such a measure would be significantly more stringent—it would capture not only whether the highest ranking nodes in A_θ and A_μ are similar, but also whether they are ordered in a similar way. Such a measure would not be able to successfully distinguish between cases where the highest ranking nodes are completely different, and cases where there is good overlap, in combination with significant reordering.

The measure of rank k-similarity is symmetric, and is therefore useful for cases where both rankings carry the same meaning, e.g. when they are obtained at consecutive thresholds. However, it is too restrictive when the rankings being compared are interpreted differently. The *α-relaxed k-similarity* of a ranking A to some ranking B is the proportion of the top $100k\%$ highest ranking nodes in B which are also within the set of $100k\alpha\%$ highest ranking nodes in A, $k \in (0; 1]$, and $\alpha \in (0; \frac{1}{k}]$:

$$M_k^\alpha(A, B|k, \alpha) = \frac{|\{v : A(v) > N(1 - k\alpha) \text{ and } B(v) > N(1 - k)\}|}{Nk}. \tag{2.2}$$

The α-relaxed k-similarity allows for more user control compared to k-similarity when the rankings compared are not interpreted in the same way, and need therefore not be treated in the same way. For example, if A is obtained from a single threshold $A = A_\theta$, and B is some overall ranking, α-relaxed k-similarity may be used to identify whether the top 10 nodes overall, i.e. in B, are among the top 20 for the particular observed threshold, i.e. in A_θ.

Rank continuity

We introduce *rank continuity* of each metric in each network to assess the similarity between rankings induced at consecutive thresholds. In all cases a set of values for the proportion of nodes k considered to be the top ranking were used, ranging from 0.001 to 0.05 at 0.001 intervals. An overall continuity measure was calculated based on how often the observed similarity was high (0.90 or above), as follows.

Suppose that a metric $f : V \to \mathbb{R}$ induces node rankings $A_\mu, A_{\mu+0.01}, \ldots, A_\nu$ at each threshold within the medium-high confidence region $[\mu, \nu]$. Then we define the *rank continuity* of $f(\cdot)$ as the fraction of cases where the k-similarity (Equation 2.1)

between consecutive A_θ and $A_{\theta+0.01}$ is at least 0.90 for $\theta \in \{\mu, \mu + 0.01, \ldots, \nu\}$ and $k \in \{0.001, 0.002, \ldots, 0.05\}$:

$$\textit{rank continuity } (f|\mu,\nu) = \frac{|\{(\theta,k) : M_k(A_\theta, A_{\theta+0.01}|k) \geq 0.90\}|}{5000(\nu-\mu)}. \quad (2.3)$$

The scaling constant is the product of the number of different k considered (50) and the number of thresholds ($100(\nu - \mu)$). It guarantees that rank continuity is between zero and one, with higher values corresponding to better overlap. Since a range of different values of k up to 0.05 was considered in calculating a single continuity measure, higher ranking nodes contribute more to the overall rank continuity of a metric.

Rank identifiability

Further, for each metric an overall ranking was calculated for the truncated threshold region. The overall ranks for all nodes were calculated by first averaging over node ranks at all relevant thresholds, and then ranking the resulting values. Ties were resolved at random.

For our definition of rank identifiability, for each metric the α-relaxed k-similarity between overall ranks B and threshold ranks A_θ for each threshold θ in the medium-high confidence region was calculated. We define the *rank identifiability* score for each metric $f(\cdot)$ as the minimum observed α-relaxed k-similarity (Equation 2.2) between overall and threshold ranks:

$$\textit{rank identifiability } (f|\mu,\nu) = min_{\theta\in[\mu,\nu]}\{M_k^\alpha(A_\theta, B|k = 100/N, \alpha = 1.5)\}. \quad (2.4)$$

In all cases apart from the Bernoulli network, which had the smallest number of nodes, the ability to recover the top $n = 100$ nodes overall (i.e. $k = 100/N$) among the top 150 (i.e. $\alpha = 1.5$) at any threshold was tested. The parameter α can be thought of as a tolerance parameter: if the overall top k fraction of nodes are of interest, we allow these to be observed at slightly lower ranks at individual thresholds. The tolerance α should therefore be relatively small, $k\alpha \ll 1$. However, it should also be large enough for the relaxed similarity measure $M_k^\alpha(\cdot)$ to capture

information genuinely different from $M_k(\cdot)$. Therefore, we propose that generally α should be set as $\alpha = O(1)$. The value $\alpha = 1.5$ was chosen arbitrarily. Further, in the Bernoulli network, n was lowered to $n = 20$ to account for the significantly lower number of nodes in the network, and the parameter α remained fixed at $\alpha = 1.5$.

Rank instability

Another way to assess rank robustness is through the *instability* of node ranks attained by different thresholds.

For each metric $f(\cdot)$, the top 1% overall top ranking nodes $U = \{v|B(v) > 99\%N\}$ were identified. Then the rank ranges attained by each of these nodes $v \in U$ over the medium-high confidence region were calculated as:

$$range(v|\mu, \nu) = max_{\theta\in[\mu,\nu]}(A_\theta(v)) - min_{\theta\in[\mu,\nu]}(A_\theta(v)), \tag{2.5}$$

where $\{A_\theta\}_{\theta\in[\mu,\nu]}$ are the ranks obtained from the same metric $f(\cdot)$ over the medium-high confidence region. For example, ubiquitin, which is always the highest degree node in the YEAST network, would have a rank range of zero. We define the rank instability as the average scaled rank range for the nodes in U:

$$\textit{rank instability } (f) = \frac{1}{N|U|}\sum_{v\in U} range(v). \tag{2.6}$$

If ranks remain very stable under $f(\cdot)$, then rank ranges would be low and rank instability would be close to zero. Conversely, if the ranks were relatively unstable, or near-random, rank ranges would be high, and rank instability would be closer to one.

Choices for all parameters discussed above were made so that rank robustness measures capture information about the highest ranking nodes. Overly stringent parameter choices would make the different measures reward only perfect or near-perfect rank agreement and ignore persistent trends of good rank overlap. Conversely, lenient parameters would reward bad as well as good rank agreement. Provided either extreme is avoided, parameters can be set in a number of ways. In test cases, we find that perturbations from the values used above do not lead to qualitatively different results. While we provide these as recommended values, our methodology is fully flexible and allows users to explore different options.

2.3 Results

2.3.1 Thresholding effects

The reliability of a detected interaction between two proteins is often quantified by a confidence score, with lower scores corresponding to weaker interaction evidence. When PINs are constructed from such data, a threshold on the confidence scores is usually applied in an attempt to filter out spurious interactions. While it is possible to incorporate the confidence scores as edge weights in the network, these scores represent neither interaction strength, nor distance, making classical weighted network analysis techniques difficult to interpret. Moreover, confidence scores vary across databases, both in their derivation and interpretation, as well as in their values.

Confidence scores are designed specifically to allow researchers a degree of control over data quality, usually through thresholding. Threshold choice over the confidence scores introduces a trade-off between the numbers of false positive and false negative interactions. A low threshold may introduce many interactions which have been detected experimentally, but which have no biological relevance, while a high one will reduce the number of such false positives, but may also lead to more genuine interactions being excluded from the network.

Imposing different thresholds will affect PIN structure, and may affect PIN analysis in complex, and potentially difficult to predict, ways. Some metrics, such as edge density, node degree, and natural connectivity, will decrease monotonically with threshold increase. Other network metrics, such as clustering coefficients and betweenness centrality, do not necessarily behave monotonically and it is unclear how to predict their rate of change (or even its direction) between thresholds.

To examine the effect of threshold choice we considered three full organism networks obtained from the STRING database—*P. vivax* (PVX), *E. coli* (ECOLI), and *S. cerevisiae* (YEAST). STRING suggests using one of four thresholds as a default—low (0.15), medium (0.40), high (0.70), and highest (0.90). For each threshold, an unweighted network between the proteins is constructed which includes

only those edges for which the score is at least as high as the threshold. The average degree in each of the three STRING networks analysed decreased with increasing thresholds, from over 200 at low confidence, to under 25 at highest confidence (Figure 2.3A). In the PVX network, the average local clustering decreased monotonically from 0.50 to 0.20 across the four suggested thresholds. However, in the ECOLI network, the average local clustering increased from 0.24 (low confidence) to 0.40 (high confidence), before decreasing to 0.35 (highest confidence). In the YEAST network, the average local clustering remained stable around 0.27 between low and medium confidence, and then steadily increased to 0.36 at the highest confidence threshold before dropping off again (Figure 2.3B). Unlike average degree, the average local clustering is non-linear, and is heavily influenced by low-degree nodes in sparse networks. A small number of edge deletions can dramatically change the local clustering coefficients of such nodes, making it difficult to predict *a priori* how the average local clustering will change with the threshold.

Figure 2.3 illustrates two aspects of scored PINs: firstly that node metrics can vary significantly in their raw values across thresholds, and secondly, that this variation is qualitatively different for different metrics. The incomplete coverage and experimental error of interaction detection techniques imply that it is most unlikely that any particular thresholded network describes perfectly all biologically relevant interactions and only them. Therefore, any robust, biologically informative PIN analysis pipeline should ideally show agreement in results obtained across at least a narrow range of different thresholds. In the context of using node metrics to identify key proteins in a network, such an agreement may translate to identifying the same set of highest ranking nodes.

We analysed 25 node metrics, of which 12 were node centralities, and 13 were global network summaries which we redefined as leave-one-out difference metrics. Four scored PINs were considered, spanning three organisms and two databases—the three PINs from STRING, illustrated in Figure 2.3 and the *S. cerevisiae* network obtained from HitPredict (HPRED). For each PIN, these metrics were calculated for all nodes in a set of 85 thresholded networks, obtained at equidistant

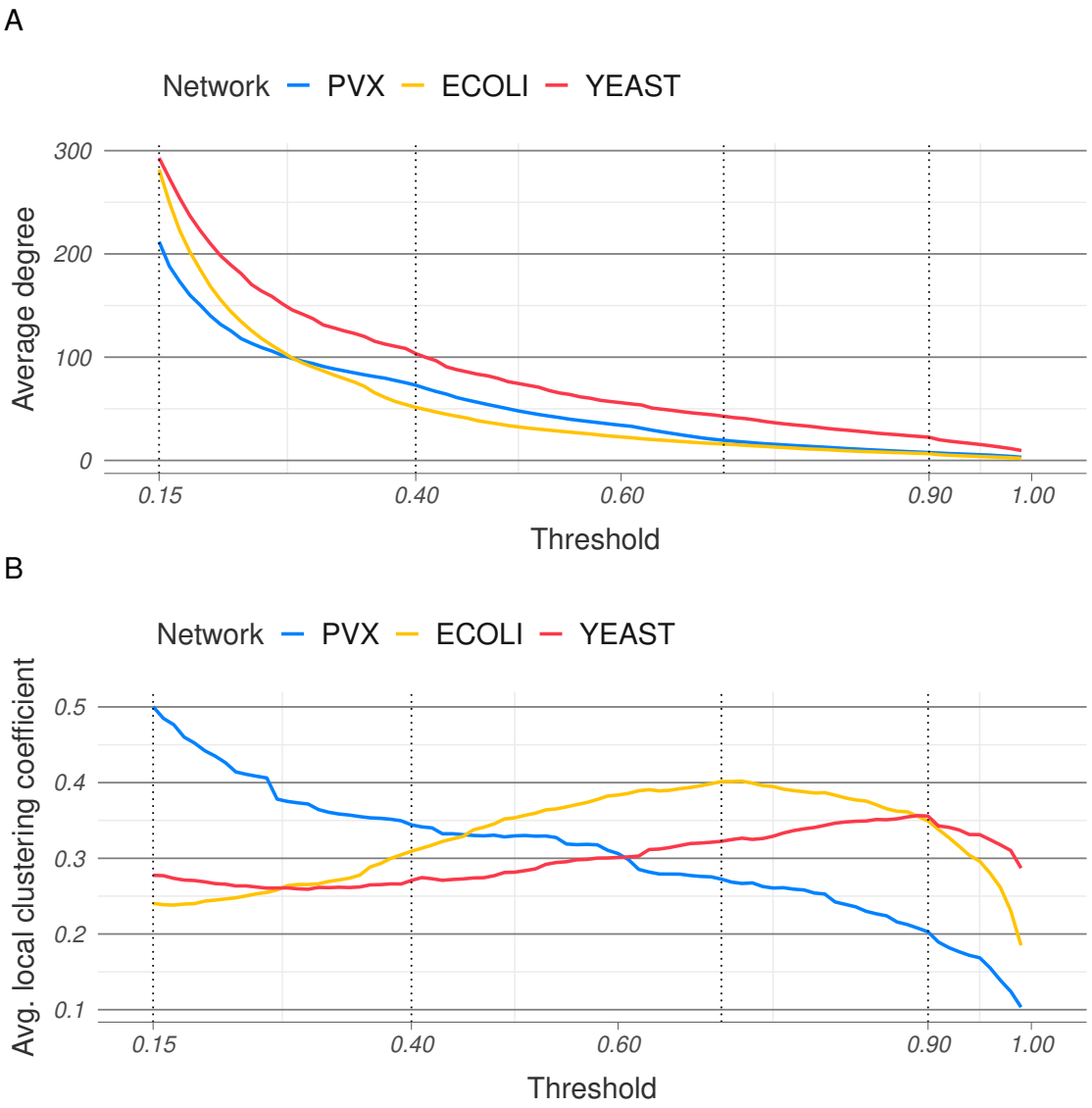

Figure 2.3: Thresholding effects in STRING networks. Average degree (A) and average local clustering coefficient (B) as functions of the threshold in the three STRING networks. The dotted vertical lines correspond to the four default STRING threshold values.

thresholds from 0.15 (the lowest recorded) to 0.99 at 0.01 intervals. In addition we considered two synthetic scored networks—SYN-GNP based on a Bernoulli random graph, and SYN-PVX, based on a re-scored subset of the PVX network. The node rankings induced by a metric at each of the thresholded networks were used to assess the rank robustness of a metric.

2.3.2 Rank continuity

PIN analysis often aims to identify key proteins in a particular biological process or context, for example by studying which nodes in the network attain high values across different node metrics. This problem relates to the node rankings induced by the metrics (*"Which are the nodes of highest degree?"*), rather than to exact metric values (*"What is the degree of these nodes?"*).

Exact metric values can be difficult to interpret and will vary both between PINs and with the PIN confidence score threshold. For example, ubiquitin (YLL039C) has degree between 262 and 4254 across different thresholds of the YEAST network (values obtained at score thresholds 0.99 and 0.15 respectively). These values are vastly different, and not easily interpretable or comparable outside the context of the particular thresholded networks they are obtained from. In contrast, the fact that ubiquitin is the single highest degree node across all thresholds for the scored YEAST network demonstrates its biological role more clearly.

We propose that for a node metric to be reliably indicative of the biological state described by a scored PIN, it should identify similar sets of key, i.e. highest ranking, proteins at a range of thresholds. In particular, rankings obtained at consecutive thresholds (e.g. at 0.40, the proposed medium confidence threshold in STRING, and at the slightly higher 0.41) should be in good agreement. Large differences could imply that (a) the metric is too influenced by pre-processing decisions to be informative, or (b) that the confidence score distribution is highly concentrated between these two thresholds and moving from one to the other significantly changes network topology.

For each analysed node metric, rank similarity was measured using Trajanovski's k-similarity (see Section 2.2.4) between each two consecutive thresholds, across all scored networks—the three STRING networks, the *S. cerevisiae* network from HitPredict, and two synthetic networks (Figures 2.4 and A.1–A.4). In the analysed PINs, three different modes of behaviour were observed: some metrics exhibited consistently high similarity, some consistently low, and for others k-similarity steadily decreased with threshold increase (Figures 2.4A, A.1 and A.2).

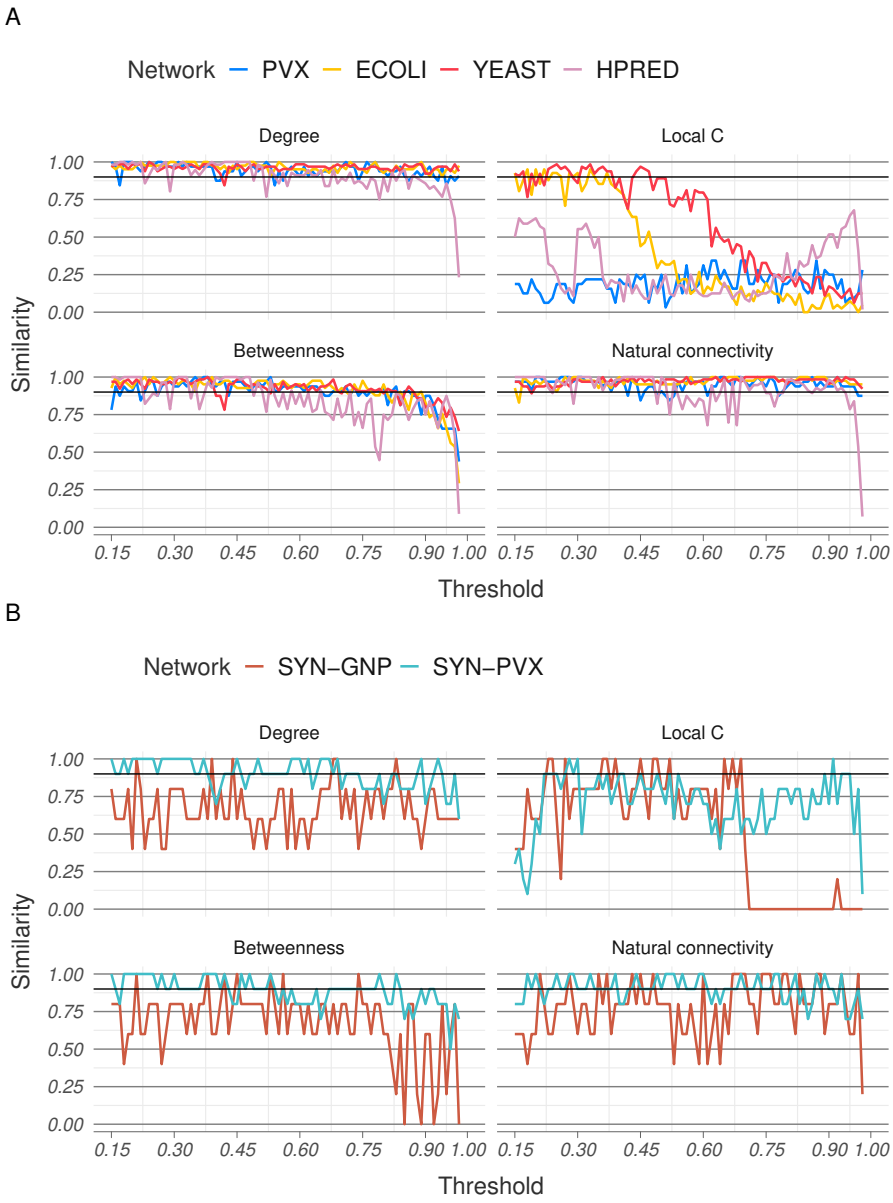

Figure 2.4: Metric rank similarity between consecutive thresholds. Rank k-similarity has been plotted for $k = 0.01$. (A) In the four PINs, metrics were either consistently stable (e.g. degree and LOUD natural connectivity), consistently unstable (e.g. local clustering coefficient), or showed decreasing stability (e.g. betweenness). (B) The synthetic network based on a randomly rescored subset of the PVX network, SYN-PVX, and the network based on a Bernoulli random graph, SYN-GNP, exhibited different behaviour, with metrics showing the least similarity across thresholds in the SYN-GNP network.

We propose a rank continuity measure based on how often k-similarity between consecutive thresholds reaches 0.90 (details in Section 2.2.4). Continuity was measured for our set of twenty-five metrics (Table 2.2) across the six scored networks

(Tables A.1 and A.2). Between 7 and 16 metrics were found to have continuity measures over 0.90 for the medium-high confidence region in each of the scored PINs. The value of 0.90 was chosen since we believe robust metrics should produce nearly identical sets of high-ranking nodes across most pairs of consecutive thresholds. Seven metrics were found to have continuity measures over 0.90 in all four networks, and an additional four metrics had high continuity in three out of the four networks (all but the PVX network). Eleven of the twenty-five metrics—degree, redundancy, PageRank, harmonic closeness, LOUD natural connectivity, LOUD global clustering, LOUD average redundancy, and four of the ego-network based metrics—had an average score across the four PINs above 0.90. Nine metrics, including the commonly used local clustering coefficient and betweenness centrality, did not achieve a high continuity measure in any of the four PINs.

Spearman rank correlations of the continuity measures (Table 2.3) showed extremely good agreement between the three STRING networks (all correlations were above 0.95), and very good agreement between the STRING networks and the HPRED network (correlations between 0.89 and 0.92). These correlations were higher than correlations between the STRING networks and either of the synthetic networks (between 0.30 and 0.68), suggesting that how edge scores are placed over the network (biased as opposed to random) plays an important role in metric rank continuity. Finally, continuity in the synthetic Bernoulli network, SYN-GNP, was considerably lower across all metrics (Figure 2.4B, Table A.2). This implies that the metric continuity is sensitive to network structure—the reported continuity values will not necessarily hold for other types of networks (e.g. social, transport, etc.), where other node metrics may be more stable.

Our continuity analysis suggests that nearly half of the tested node metrics are robust to small threshold perturbation in PINs. However, incremental changes in the set of overlapping nodes between consecutive thresholds may result in a high continuity measure but low similarity between rankings at more distant thresholds. This can be undesirable, since often confidence scores are not readily interpretable

	PVX	ECOLI	YEAST	HPRED	SYN-GNP	SYN-PVX
PVX	1.00	0.95	0.96	0.92	0.46	0.48
ECOLI	0.95	1.00	0.96	0.92	0.58	0.54
YEAST	0.96	0.96	1.00	0.89	0.56	0.57
HPRED	0.92	0.92	0.89	1.00	0.42	0.30
SYN-GNP	0.46	0.58	0.56	0.42	1.00	0.79
SYN-PVX	0.48	0.54	0.57	0.30	0.79	1.00

Table 2.3: Spearman correlations between rank continuity measures across the different networks. High correlations between the protein interaction networks, and lower correlations with the synthetic networks suggest that metric rank continuity depends both on network topology and on score placement.

and there may exist a wide range of permissible thresholds (e.g. anywhere between 0.15 and 0.90 in STRING).

2.3.3 Rank identifiability

In order to assess the robustness of a node metric across a range of medium-high confidence score thresholds, overall ranks were calculated and compared to ranks induced at single thresholds. These overall ranks were designed to represent the relative position (i.e. importance) of each node across a range of medium-high confidence thresholds, and were calculated as rank averages across the threshold region. For example, consider two proteins X and Y. Suppose that for thresholds 0.60 to 0.70, X is the highest degree node in the network, and Y is the second highest, and vice versa for thresholds 0.71 to 0.90. Since it is more often the case that Y has higher degree than X, Y would be the overall highest ranking node with respect to degree. We define a rank identifiability measure which quantifies the ability to recover the set of overall highest ranking nodes by considering any single threshold in the region. Our identifiability measure is based on an asymmetric version of k-similarity, which we introduce and call α-relaxed k-similarity (defined in Section 2.2.4). Intuitively, a rank identifiability measure of 0.90 implies that at least 90 of the 100 overall highest ranking nodes are also among the top 150 at any given threshold. So if only a single threshold was considered, it would still contain the majority of nodes which rank highly across the entire region.

In each of the three STRING networks, 5 (for PVX) or 6 (for YEAST and ECOLI) node metrics were found to have rank identifiability measures above 0.90 (Table A.3). In the HPRED network, where the medium-high confidence region is shorter than in STRING, identifiability measures were higher and 16 metrics attained a score above 0.90. Of these, four metrics—redundancy, number of edges in the one-step ego network, LOUD natural connectivity, and LOUD average number of edges in the one-step ego network—had identifiability measures above 0.90 across all four PINs.

The similarities of rankings induced at thresholds outside the medium-high confidence region to the overall ranks were also calculated, although these similarities did not contribute to the rank identifiability scores. Since the overall ranks were calculated over the medium-high confidence region for each network, it is natural to expect α-relaxed k-similarities to be higher within the region than outside it (Figures 2.5, A.5–A.8). This trend is observed even in the case of the truly randomised SYN-GNP network (Figure 2.5B). In the HPRED network, for example, some metrics showed high rank similarity for thresholds as high as 0.45. This indicates that the exact boundaries of the region do not necessarily heavily influence identifiability results.

Like rank continuity, rank identifiability is a context-dependent property of network metrics. The three STRING networks closely agreed in identifiability measures (Spearman correlations between metric identifiability scores were all 0.94 and above), and were more similar to measures obtained from analysing the HPRED network (correlations above 0.81) than any of the synthetic networks (Table 2.4).

	PVX	ECOLI	YEAST	HPRED	SYN-GNP	SYN-PVX
PVX	1.00	0.96	0.94	0.85	0.42	0.62
ECOLI	0.96	1.00	0.97	0.84	0.34	0.54
YEAST	0.94	0.97	1.00	0.81	0.37	0.57
HPRED	0.85	0.84	0.81	1.00	0.46	0.60
SYN-GNP	0.42	0.34	0.37	0.46	1.00	0.78
SYN-PVX	0.62	0.54	0.57	0.60	0.78	1.00

Table 2.4: Spearman correlations between rank identifiability measures across the different networks. Similar to rank continuity (Table 2.3), correlations are highest between the scored PINs. The SYN-PVX network correlates to the scored PINs better than the SYN-GNP network does.

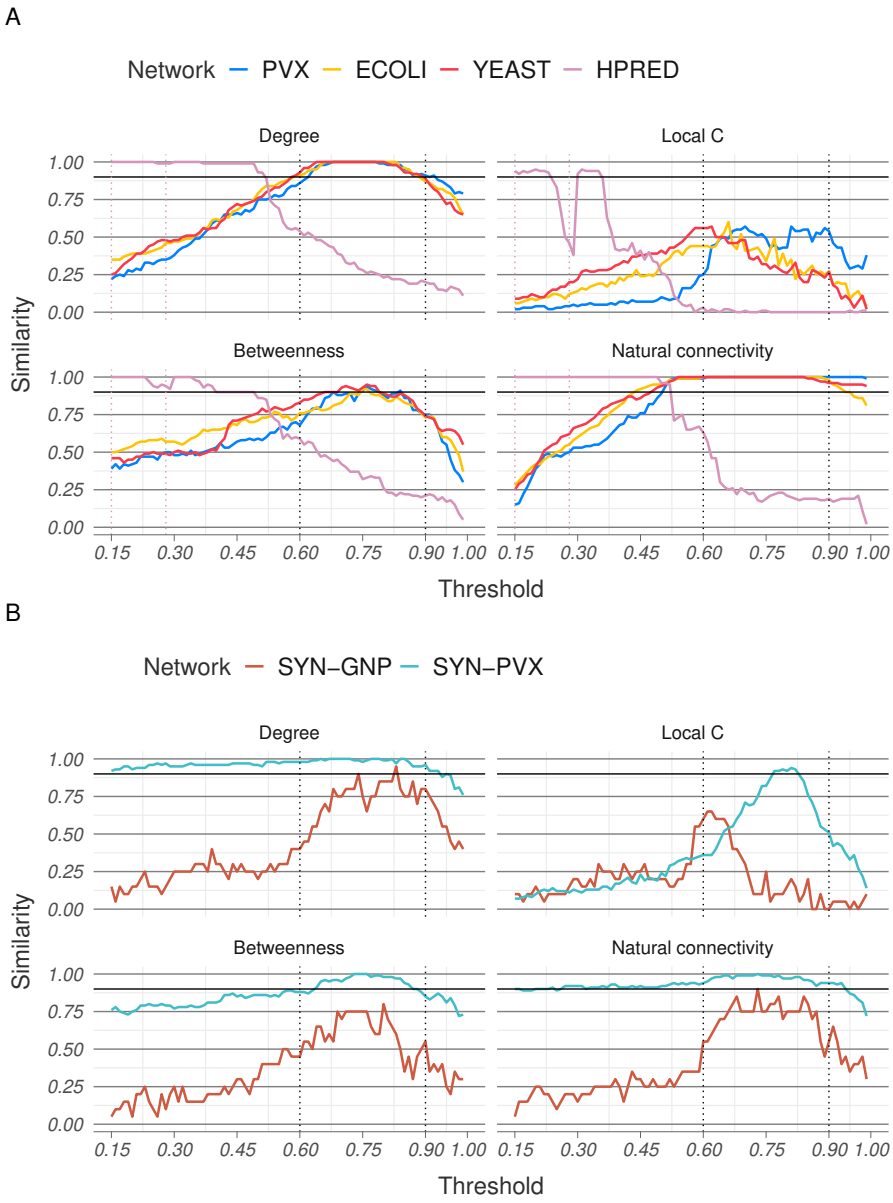

Figure 2.5: Relaxed similarity between overall and threshold ranks in the scored PINs. (A) The overall ranks have been calculated over the medium-high confidence regions—0.60 to 0.90 for the three STRING networks (black dotted lines) and 0.15 to 0.28 for the HitPredict network (pink dotted lines). (B) The STRING medium-high confidence interval was also used for the synthetic networks. The SYN-GNP network, where both structure and score allocation are uniform, exhibits lower relaxed similarity. The SYN-PVX network has inherently heterogeneous network structure, on which scores are assigned randomly. Score thresholding introduces the same rate of change in different parts of the network, so that the relative node degree, for example, may remain largely unchanged across a series of thresholds. In contrast, the heterogeneous score allocation in PINs makes rank reorderings more likely, and identifiability may be expected to be lower.

While rank identifiability can be used to quantify rank conservation across

many thresholds, it does not account for all types of rank variability. Intuitively, a metric which preserves the exact same ranking in the set of top n nodes at every threshold is more robust than one in which the top node set is preserved, but re-ranked. However, since our rank continuity and identifiability measures are both based on set overlap only, in both the preserved and the re-ranked case the metrics would achieve a perfect score of 1. To take this difference into account we introduce a measure for rank instability.

2.3.4 Rank instability

A different way of assessing how well top-ranking nodes (or nodes in general) preserve their ranks across different thresholds is to calculate the ranges of ranks they attain. A robust metric should result in relatively narrow rank ranges. In particular, overall top ranking nodes should have relatively narrow rank ranges.

In order to quantify this behaviour we define rank instability as the scaled average rank range of the overall top 1% ranking nodes (details in Section 2.2.4). Unlike the rank continuity and identifiability measures, where values close to one represent robustness, instability values close to zero correspond to narrower rank ranges, and therefore more consistent node metric behaviour. The instability measure was lower in the HPRED network, where the medium-high confidence interval is shorter, and similar across the three STRING PINs (Figure 2.6A and Table A.4). Only four metrics had instability measures below 0.01 in all PINs—number of edges in the one-step ego network, LOUD natural connectivity, LOUD average redundancy and LOUD average number of edges in the one-step ego network.

Rank instability measures in the synthetic networks were generally higher than in the scored PINs (Figure 2.6B and Table A.4). Spearman correlations of metric rank instability were higher across the scored PINs than they were between PINs and either of the synthetic networks (Table 2.5), suggesting once again that PINs exhibit behaviour distinct from that of other networks.

Overall, the three measures for rank robustness of node metrics described here—rank continuity, identifiability, and instability—agree across the four studied protein

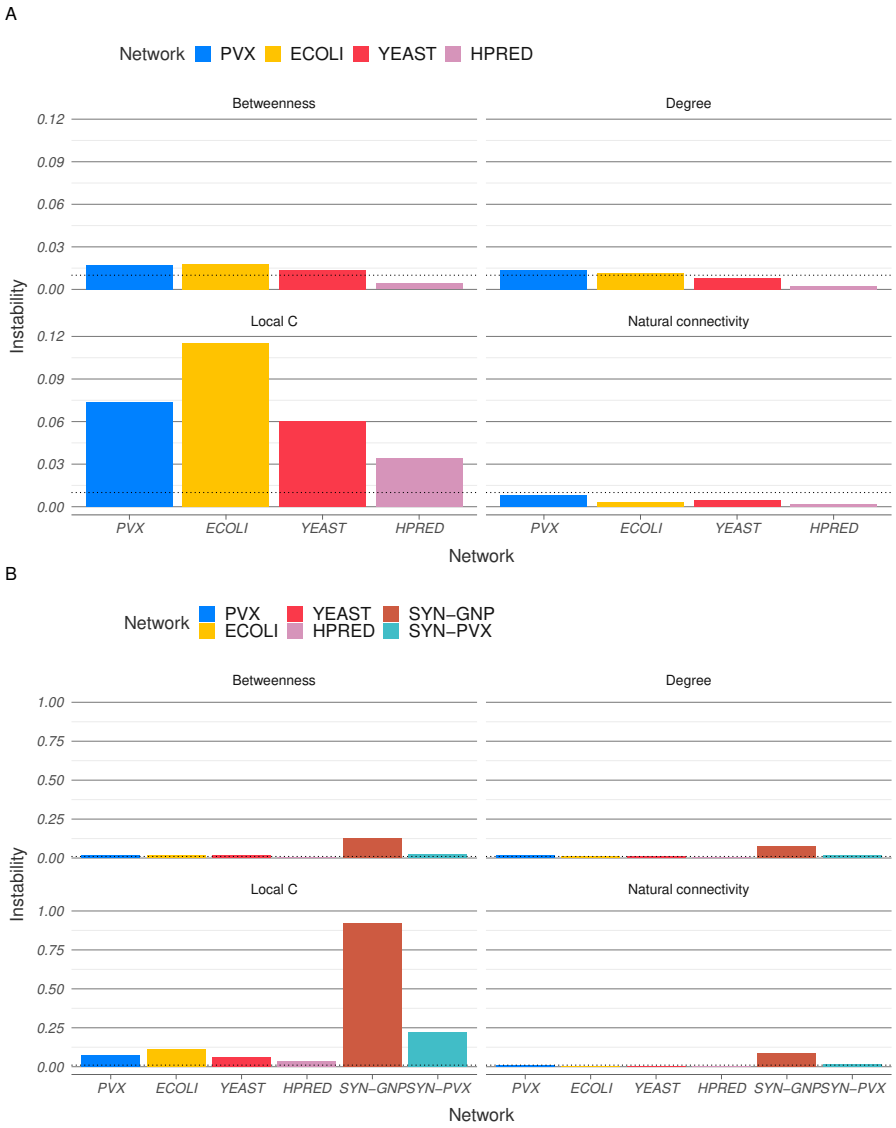

Figure 2.6: Rank instability of metrics in the scored networks. (A) Rank instability in the four PINs. The dotted lines correspond to 0.01. Instability measures in the HPRED network were generally narrower. (B) Rank instability in the synthetic networks. Instability measures in PINs were generally lower and have been plotted for comparison. Note the different scales between plots in A and in B.

interaction networks and identify four node metrics to be robust to thresholding: number of edges in the one-step ego network, LOUD natural connectivity, LOUD average redundancy and LOUD average number of edges in the one-step ego network. Measures of rank continuity, identifiability, and instability for all 25 analysed metrics averaged over the four PINs can be found in Table A.1.

	PVX	ECOLI	YEAST	HPRED	SYN-GNP	SYN-PVX
PVX	1.00	0.95	0.94	0.91	0.40	0.64
ECOLI	0.95	1.00	0.97	0.92	0.37	0.62
YEAST	0.94	0.97	1.00	0.92	0.39	0.61
HPRED	0.91	0.92	0.92	1.00	0.26	0.61
SYN-GNP	0.40	0.37	0.39	0.26	1.00	0.83
SYN-PVX	0.64	0.62	0.66	0.61	0.83	1.00

Table 2.5: Spearman correlations between rank instability measures across the different networks. Correlations are again highest between the scored PINs. Like rank continuity and rank identifiability, (Tables 2.3 and 2.4), rank instability measures are more highly correlated between the two synthetic networks than they are between the SYN-GNP network and the scored PINs.

2.4 Discussion

Protein interaction network analysis typically starts with network construction—some interaction data of interest is obtained, either experimentally or from publicly available databases, and is then pre-processed before being used to create a network. Different types of protein interactions, experimental systems for detecting them, and data clean-up choices mean that multiple networks can be built to represent the same underlying biological process. Since typically only one of these networks will be analysed, it is not immediately clear how much these network models of cellular biology and any conclusions drawn from them might differ. In this chapter we have demonstrated that varying the confidence score threshold for interactions can result in structurally different networks, both in terms of density and in terms of properties like average local clustering coefficient. We propose that if PIN analysis aims to provide reliable, reproducible biological insight, it should show some agreement across alternative network models. This property is desirable in a range of contexts: any network analysis pipeline which relies on thresholding in cases where an optimal threshold cannot be identified with high confidence should produce similar results across different thresholds. Such analysis may have different goals, such as protein function prediction or community detection, and may employ a range of tools, such as explicit node metrics and machine learning tools.

In this chapter, we have focused on one frequent goal of PIN analysis—identifying key proteins (nodes) in a particular piece of biological architecture (network). This can be done by calculating node metrics, such as degree or betweenness centrality, and then identifying the set of nodes which rank highest based on metric performance. There are many different metrics one can use in this context, and it is not always clear which are the most suitable. We argue that one desirable feature a suitable metric should posses is *rank robustness*, or the ability to identify the same or at least largely similar sets of top ranking proteins when the network construction process is altered. Here we have considered robustness to variation in the confidence score threshold required to include an interaction in the network.

Networks constructed at lower thresholds are denser and potentially include more spurious interactions than networks constructed at higher thresholds. Increasing the stringency of data quality, however, may result in potentially important but understudied interactions to be omitted. Different scoring procedures across databases, and even within different versions of the same database (see Chapter 1 Section 1.3.2), make optimal threshold choice difficult. Therefore, a level of rank robustness across different thresholds is desirable for node metrics. We have proposed three measures with which to assess such robustness—rank continuity, identifiability, and instability. The relevance of each measure will depend on the research question at hand and on the reliability of the confidence scoring procedure.

Rank continuity captures the effect of small threshold perturbations. This measure may be of particular interest when a narrow band of permissible thresholds has been identified. For example, if researchers are only interested in high confidence STRING interactions (threshold 0.75), they may wish to explore whether any metrics they use have high rank continuity scores, and in particular whether results obtained at thresholds 0.74 to 0.76 are similar.

Rank identifiability compares ranks at given thresholds against overall ranks. It may be particularly informative when there are no known optimal threshold values. For example, if researchers are uncertain of their threshold choice, they

may wish to use metrics with high rank identifiability, since these metrics show good agreement across a wider range of thresholds.

Finally, rank instability assesses the variation in threshold ranks for the overall top 1% of nodes. It is our most stringent measure, requiring not only that highly ranked proteins remain highly ranked but also that they retain their ordering across thresholds. It should be used when absolute rank is important. For example, if the researchers are not only interested in what the key (i.e. top ranking) nodes in the network are, but also how they are ordered, they should focus on metrics with low rank instability. If the top three nodes by degree were A, B and C at every threshold, but these were differently ordered—say A–B–C at threshold 0.75 and C–B–A at threshold 0.76—then rank continuity and rank identifiability would not penalise the reordering, while rank instability would.

We calculated the rank continuity, identifiability, and instability of 25 node metrics in each of six scored networks, four of which were PINs and two of which were synthetically generated. We limited ourselves to networks with fewer than 7000 nodes because of the computational cost associated with metric extraction, and in particular with calculating LOUD natural connectivity. Calculating natural connectivity for a single node at a single threshold for our largest network, the STRING network for yeast, takes approximately 88 seconds on a standard desktop computer. Although often PINs from higher eukaryotes with larger proteomes (e.g. human) are of interest, often only subnetworks with fewer nodes are studied. We therefore believe our methodology will be useful for a wide range of applications.

Our rank continuity measure, which is based on a rank similarity measure originally proposed by Trajanovski et al. 2013, quantifies the agreement between node rankings obtained at consecutive thresholds. If networks obtained at two close thresholds, say 0.50 and 0.51, yield considerably different rankings, this may suggest that threshold choice plays an overwhelmingly important role in network construction, and may obscure any underlying biological signal that could otherwise be detected. Conversely, high continuity measures correspond to rankings which are unlikely to significantly change with small threshold perturbations.

Confidence scores are not always readily interpretable, which might make threshold choice difficult. Therefore, agreement over a wider threshold region might also be desirable. Small differences in consecutive thresholds might be responsible for large discrepancies between more distant thresholds (say 0.50 and 0.70), while still preserving a high rank continuity measure. In order to take this effect into account, we introduce rank identifiability to measure the agreement between different threshold rankings and a single overall ranking.

Finally, our rank instability measure provides an alternative way of analysing rank robustness which is not based on rank overlap but instead focuses on the different ranks a particular node attains at different thresholds. High instability corresponds to the overall top ranking nodes attaining a wide range of individual threshold ranks.

Our analysis identified four node metrics—number of edges in the one-step ego network, LOUD average redundancy, LOUD average number of edges in the one-step ego network, and LOUD natural connectivity—which induce robust ranks across all four analysed PINs. More commonly used metrics such as degree, local clustering coefficient, betweenness, and closeness did not perform as well. For example, when comparing the top 100 nodes obtained at the start and the end of the medium-high confidence region for the YEAST network (thresholds set at 0.60 and 0.90 respectively), node sets obtained using LOUD natural connectivity showed a three quarter overlap (75 out of 100). In contrast, the overlap of the top ranking sets identified by betweenness was less than half (41 out of 100), and the overlap between sets identified using local clustering coefficient was less than 10% (9 out of 100).

Spearman rank correlations between robustness measures across the four different PINs were consistently high (0.81 and above). In particular, the two yeast networks—YEAST, obtained from STRING, and HPRED, obtained from HitPredict—were in good agreement, in that the same metrics appeared as robust across both networks, despite the different types of data and scoring procedures used (Spearman correlation coefficients for metric robustness across the two networks are 0.89 for rank continuity, 0.81 for rank identifiability, 0.92 for rank instability, Tables 2.3–2.5). The analysed PINs varied in organism, database, confidence scoring methodology, and even

interaction type, yet the rank robustness results across them were very similar. This finding suggests that the presented robustness results may be readily applicable to other PINs. In contrast, the lower correlations observed when scored PINs were compared with the two synthetic networks (Spearman correlations below 0.64, Tables 2.3–2.5) indicate that rank robustness is context-specific. The differences observed in the SYN-GNP network imply that network topology (i.e. how the edges are placed across nodes) plays a role in metric robustness. Meanwhile, the differences observed in the SYN-PVX network show that even if network topology resembles that of a PIN, how scores are allocated to network edges also plays a role in rank robustness.

In this chapter we argued that one way of analysing protein interaction data with an awareness of the uncertainty and noise present in it would be to construct multiple networks from the data and require consensus among them. Our methodology was deterministic: the analysed networks were all simple and unweighted, and we made no assumptions about the role that uncertainty plays on network structure. An alternative strategy is to model protein interaction data as the realisation of some random process, such as a combination of PPI detection experiments, carried out on an unobserved "true" network, which captures the biological state of interest. We explore this approach in the next chapter.

I'm not lost for I know where I am. But however, where I am may be lost.

— A.A. Milne, *Winnie-the-Pooh*

3

Generative models based on uncertain protein interaction networks

Contents

3.1 Introduction

As described in Chapter 1 different types of experimental data and methods can be used for protein interaction network construction (Berggård et al. 2007), and as discussed in Chapter 2, such networks are usually built after some data pre-processing, and are subsequently treated as fixed, fully deterministic objects. The purpose of the data pre-processing step is to reduce the amount of experimental error or noise, so that the resulting networks best capture biologically relevant interactions. However, since protein–protein interaction detection and prediction are imperfect (e.g. H. Huang and J. S. Bader 2009), there remains uncertainty associated with the resulting networks.

In Chapter 2 we discussed one approach to handling such uncertainty. Different networks can be built from the same underlying interaction data, for example by choosing different confidence score thresholds. We argued that most likely none of these networks will accurately represent the underlying biological state, and postulated that reliable, biologically relevant findings from a network analysis pipeline should ideally be reproduced across a range of such PINs. An alternative approach would be to incorporate uncertainty directly into a single network and approach network analysis as a stochastic problem. In this chapter we investigate how uncertain networks (Ahnert et al. 2007, Martin et al. 2016) may be employed to do this.

Under the uncertain network paradigm, we assume that for a set of proteins V there exists a single, unobserved *true* network $G_T = (V, E_T)$ containing all biologically relevant pairwise interactions between the proteins. The aim of uncertain protein interaction network analysis is to estimate properties of G_T. Rather than observing the true network directly, we observe an *uncertain network* $G_U = (V, E_U)$ with $E_T \subseteq E_U$. Each observed uncertain edge $e \in E_U$ has a score $s(e)$ associated with it, such that $s(e) = \mathbb{P}(e \in E_T)$. In the context of protein interaction networks, these scores could be derived from the experimental interaction evidence. The more supporting evidence there is for a particular interaction, the higher the likelihood that the interaction is indeed biologically relevant, which would in turn result in a

higher score. Since scores represent a marginal likelihood of interaction existence, we would expect approximately half of the interactions with scores of 0.50 to be true, as well as approximately three quarters of the interactions with scores of 0.75, etc. Uncertain network analysis involves using the scored uncertain network G_U to estimate properties of the true network G_T.

Confidence scores in STRING (Szklarczyk, Franceschini, et al. 2014) can be interpreted in the manner above, and have previously been used to build and analyse uncertain networks (Martin et al. 2016). However, as discussed in Chapter 1 Section 1.3.2, STRING does not report any scores below 0.15. Moreover, there exist significant differences in both interactions and their scores across STRING versions. The combination of missing data and variable, difficult to reproduce scores means that STRING is not necessarily an appropriate data source for the construction of uncertain PINs.

We begin this chapter by illustrating how different data sources can be combined to calculate edge scores for uncertain PINs. We use yeast (*Saccharomyces cerevisiae*) as a model organism, and focus on two types of data. The yeast two-hybrid (Y2H) data obtained from BioGRID (Chatr-Aryamontri et al. 2017) is binary, meaning interactions are either reported as observed or not. We complement the two-hybrid data with gene co-expression data from COXPRESdb (Obayashi, Kagaya, et al. 2018). Co-expression in this instance is calculated as a Pearson correlation coefficient and is a continuous measurement between -1 and 1. We construct a positive reference set (PRS) of high-scoring STRING interactions and use it to develop a scoring procedure for protein interactions based on these two types of data. This allows us to construct an uncertain protein interaction network for yeast.

Once such a network G_U is constructed, properties of the underlying true network G_T can be estimated from it. For example, Martin et al. 2016 fit a stochastic block model to uncertain networks in order to perform community detection. They treat the true network G_T as a realisation of a random graph model with edge probabilities given by the scores in the uncertain network G_U. The reason to consider such a framework is that if we assume $G_T \sim M(G_U)$ for some model $M(\cdot)$, then we

could estimate properties of G_T by studying the distribution of $M(G_U)$. Martin et al. 2016 treat edges as independent random variables, so every edge of G_T is treated as an independent random realisation of a Bernoulli random variable with probability equal to the score of the corresponding edge in G_U. However, models which incorporate different types of edge dependence also exist.

In this chapter, we identify a family of such models $M(\cdot)$. We choose three models to investigate in detail—the model assuming independent edges, as well as two further models incorporating different types of local edge dependence. We generate random realisations from each of the three models and study the distributions of a set of frequently used global network metrics. If the true yeast protein interaction network G_T can be treated as a realisation of one of the described random graph models, then its properties could be estimated through sampling. However, it is not known which—if any—of the models may be appropriate. Without prior knowledge of a suitable model, such an approach to PIN analysis could be informative if the samples obtained across different models generally agree, i.e. if the edge scores in G_U, rather than the model $M(\cdot)$, largely determine network structure.

However, we find that the opposite is true. Different models based on the same uncertain PIN result in networks of significantly different structure. This is true for the yeast network we construct, as well as for a set of synthetic networks. Results discussed in Chapter 2 already show that synthetic "PIN-like" networks can exhibit significantly different behaviour to that observed in real data. Therefore, we conclude that without extensive research on an appropriate generative model, sampling based on uncertain networks may not be a suitable approach to studying PINs.

3.2 Data

Two main data sources were used in order to build an uncertain network for yeast. Microarray gene expression data for yeast was downloaded from COXPRESdb and gene co-expression was calculated from it. In addition, interactions identified through yeast two-hybrid assays were extracted from BioGRID. Finally, a positive reference set was built from high scoring STRING interactions.

The data describes both genetic association (gene co-expression) and physical protein interactions (Y2H). To avoid ambiguity, all data points were mapped to Entrez gene identifiers, which are the primary identifier used in both COXPRESdb and BioGRID. For simplicity, we generally refer to these as "proteins", and to any interactions between them as "protein interactions". For further details on the connection between genes and proteins in this context, refer to Chapter 1 Section 1.2.

3.2.1 Co-expression data from COXPRESdb

Microarray expression data was downloaded from COXPRESdb v.7. The data contained expression profiles of 4461 genes across 3593 samples obtained using the A-AFFY-47 platform. COXPRESdb data is already pre-processed and normalised, so no further data normalisation was carried out.

The dataset contained no missing values, and no expression values set to zero. Co-expression was unambiguously calculated by taking pairwise Pearson correlation coefficients between expression profiles. The resulting values had mean 0.062 and standard deviation 0.237 (Figure 3.1).

3.2.2 Yeast two-hybrid data from BioGRID

The co-expression data was complemented by protein interactions from Y2H assays. To build the Y2H dataset, interaction records were extracted from BioGRID v.3.5.173. Filters were applied to reduce the data to records from Y2H screens, where both interactors were yeast proteins. This resulted in a set of 16579 interaction records.

The set was further filtered to those interactors, for which gene expression data was available from COXPRESdb. This resulted in a set of 11805 records covering interactions between 2904 unique proteins. Removing duplicate records, as well as 608 self-interactions reduced the set to 8422 unique interacting pairs across 2867 proteins.

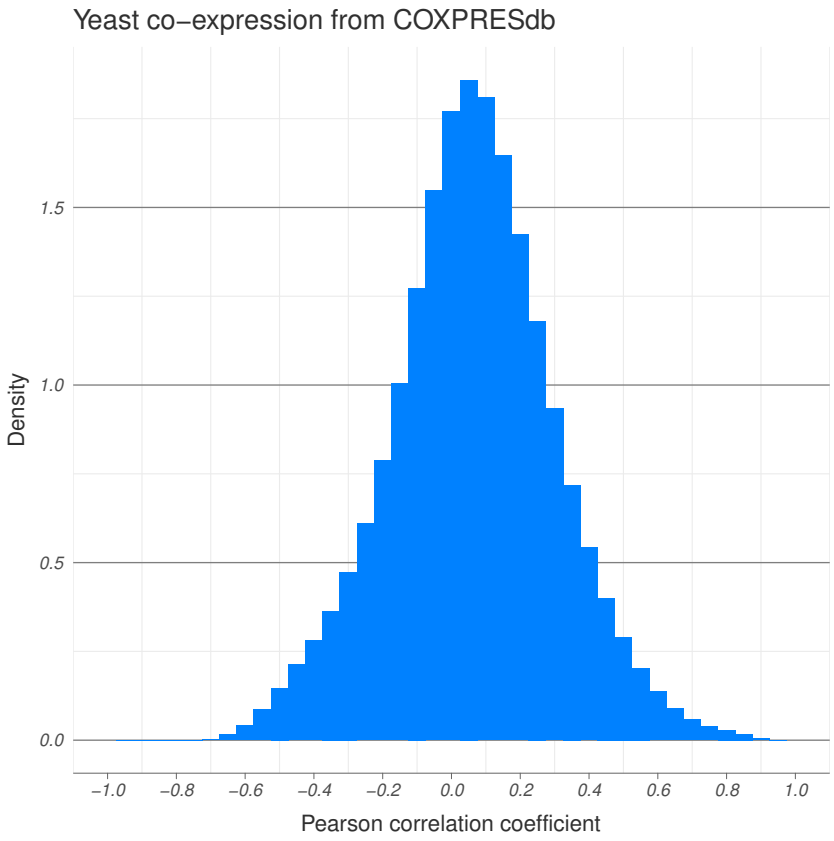

Figure 3.1: Gene co-expression for yeast. Distribution of Pearson correlation coefficients calculated from microarray data from COXPRESdb. Bin width has been set to 0.05.

3.2.3 Positive reference interaction dataset from STRING

The COXPRESdb and BioGRID data were used as the primary data sources for building an uncertain network for yeast. A positive reference set (PRS) of interactions was used for score development (see Section 3.3.1). The PRS was built from STRING.

STRING v.11.0 contains 667947 interactions across 4419 of the 4461 proteins with associated COXPRESdb gene expression data. Scores across different data source channels are available for each interaction, as well as a combined score quantifying the overall confidence in the interaction evidence. Both channel subscores and the combined score can be interpreted as interaction likelihoods. The channels correspond to different types of interaction evidence and are titled *neighbourhood*, *fusion*, *cooccurence*, *coexpression*, *experimental* and *database* (see

Chapter 1 Section 1.3.2 for details).

In order to create a PRS which is orthogonal to the co-expression data, the combined STRING scores were recalculated, omitting the *coexpression* channel. This was done according to STRING methodology. First the fixed prior $p = 0.041$ was removed from each non-zero score $s_i(u, v)$ for every channel i to obtain a prior-free score $s_i^*(u, v)$ for the interaction between u and $v \in V$:

$$s_i^*(u, v) = \frac{s_i(u, v) - p}{1 - p}. \tag{3.1}$$

Then, the prior-free scores for each interaction were combined to compute a prior-free combined score. The *coexpression* channel score $s_{coex}^*(u, v)$ was not included in this calculation:

$$s_{comb}^*(u, v) = 1 - \prod_{i \neq coex} (1 - s_i^*(u, v)). \tag{3.2}$$

Finally, the prior was added to the combined score:

$$s_{comb}(u, v) = s_{comb}^*(u, v) + p(1 - s_{comb}^*(u, v)). \tag{3.3}$$

This rescoring procedure was applied to all interaction records which had a non-zero score for at least one channel other than *coexpression*. Since 55285 interactions were only supported by co-expression evidence, this reduced the set of interactions to 612662 (see Figure 3.2).

A similar correction for Y2H evidence was not performed, since STRING does not have a separate Y2H channel. The *experimental* channel contains high-throughput Y2H data, as well as data from other experiments.

The PRS was built from the 9002 interactions with recomputed scores equal to or above 0.995. These interactions spanned across 2101 proteins from the COXPRESdb set. Interactions within the PRS were supported from a range of different sources, and are therefore unlikely to be strongly biased in favour of Y2H evidence. Approximately 85% of the interactions had a high *database* score (above 0.90), 75% had a high *experimental* score, and 38% had a high *textmining* score. While co-expression did not contribute to the recomputed scores, 42% of the PRS interactions were strongly supported through the *coexpression* channel as well. See Table 3.1 for a summary of all data used.

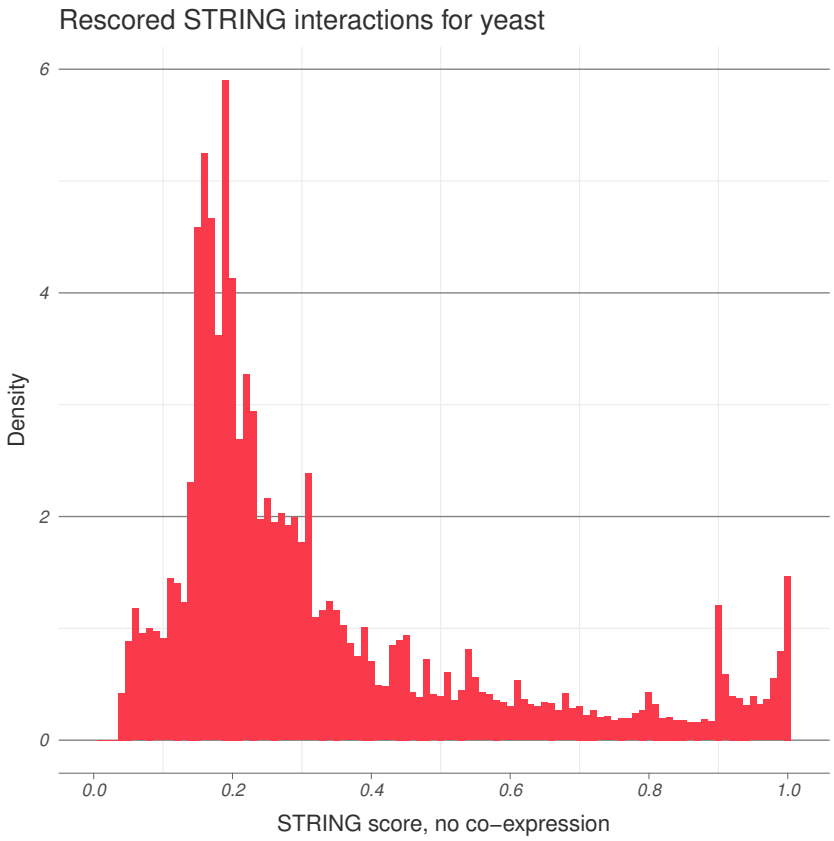

Figure 3.2: Recalculated STRING scores for yeast. Binwidth has been set to 0.01. The distribution resembles STRING score distributions (e.g. Figure 2.1). The majority of interactions have low scores, and a small number of interactions have scores close to one. There is an additional peak around 0.90. Rescoring has further resulted in interactions below 0.15. These interactions were originally primarily supported by co-expression data. Weak support from other evidence channels has resulted in a low score.

Data	Number of proteins	Number of interactions
Co-expression	4461	9948030
Y2H	2867	8422
PRS	2101	9002

Table 3.1: Gene co-expression and protein interaction data. Co-expression data was available for all pairs of proteins. The PRS and Y2H sets overlapped on 988 interactions across 1012 proteins.

3.3 Uncertain network construction

An uncertain network for yeast was constructed based on the data described in Section 3.2. Nodes in this network correspond to proteins, and edges between pairs of nodes are scored. The score $s(u, v)$ for a pair of nodes u, v corresponds to the likelihood that the proteins u and v interact, given the available co-expression and

Y2H data. Interactions here are interpreted to be functional associations between protein pairs, rather than physical binding. See Chapter 1 Section 1.2.2 for a discussion on the difference between the two.

We developed a scoring procedure $s(u, v)$ using estimates for the co-expression distributions, as well as estimates for the probabilities of a positive Y2H screen between interacting and non-interacting protein pairs. The former were estimated using the PRS, and the latter were computed from published estimates (H. Huang and J. S. Bader 2009). We constructed an uncertain network for yeast from the computed scores.

Since protein interactions within yeast are not known with absolute certainty, the true network underlying this uncertain network is unknown. We therefore constructed an additional "yeast-like" synthetic uncertain network, for which the underlying true state was fixed, and co-expression and Y2H evidence was randomly generated and then scored. Finally, we constructed two further uncertain networks, for which scores were generated randomly according to *Beta* and $Uniform$ distributions.

3.3.1 Co-expression and Y2H scoring procedure

The scoring procedure for yeast is derived directly from Bayes' rule. Suppose the true yeast network is $G_T = (V, E_T)$ and let $u, v \in V$ be two distinct, randomly chosen proteins. Let $I_{uv} = \mathbb{1}\{(u, v) \in E_T\}$ be the indicator variable that u and v interact, so $I_{uv} = 1$ if and only if there exists a real, biologically relevant protein–protein interaction between u and v. Then $I_{uv} \sim Bernoulli(p)$, where p is the edge density of G_T. We estimate p by the STRING prior $\hat{p} = 0.041$. Note that there does not exist a single widely accepted strategy for estimating PIN density, and different values can be used here.

Yeast two-hybrid

Now let $Y_{uv} \in \{0, 1\}$ be the outcome of a Y2H screen on u and v, without specifying which protein is used as bait and which is used as prey. Then, conditional on I_{uv},

$Y_{uv} \sim Bernoulli(q(I_{uv}))$, where for any pair $u \neq v$,

$$q(I_{uv}) = \begin{cases} q_0 & \text{if } I_{uv} = 0, \\ q_1 & \text{if } I_{uv} = 1. \end{cases} \tag{3.4}$$

In the above q_0 is the probability of a false positive Y2H screen, and q_1 is the probability of a true positive. To estimate these we referred to H. Huang and J. S. Bader 2009, who determine the false discovery rate of the Y2H system to be $FDR = 0.099$ and the false negative rate to be $FNR = 0.51$ for *S. cerevisiae*. These rates are based on physical binding only, and need to be adjusted since functional interactions can exist between proteins which do not bind.

We approximated the proportion of physical interactions for yeast as the proportion of STRING interactions for yeast which have been tagged as "binding". These are all STRING interactions with some evidence of protein binding. We scaled each interaction by its score, so if π is the proportion of binding interactions in the set of true interactions, we estimated

$$\hat{\pi} = \frac{\sum_{binding} s_{STRING}(u, v)}{\sum_{all} s_{STRING}(u, v)} = 0.213. \tag{3.5}$$

The sum is carried out over all interactions for yeast recorded in STRING, with the original STRING scores (rather than the recalculated scores used to form the PRS). Since the sum is over STRING scores, the numerator in Equation 3.5 is the expected number of physical binding interactions, and the denominator is the expected number of total interactions in yeast.

To calculate the estimates for q_0 and q_1, we used $\hat{p}$, $\hat{\pi}$ and the definitions of FDR and FNR for physical binding. Note that a positive Y2H screen on a functional interaction contributes to the FDR as calculated by H. Huang and J. S. Bader 2009, but also contributes to the *true positive* probability q_1 in our model. We assumed a non-binding interaction has the same probability of being detected by Y2H as a genuine non-interaction, i.e. q_0. We further assumed that physical binding interactions have a probability of being detected z. This makes the overall probability of detecting a true interaction (physical or not) $q_1 = \pi z + (1 - \pi)q_0$. Further, the FNR as defined in H. Huang and J. S. Bader 2009, i.e. the *physical FNR*, is

$$FNR = 1 - z \tag{3.6}$$

and the *physical FDR* is

$$FDR = \frac{(1 - p\pi)q_0}{(1 - p\pi)q_0 + p\pi z}. \tag{3.7}$$

Rearranging the FDR equation and plugging in the estimates we obtained

$$\hat{q}_0 = \frac{FDR}{(1 - FDR)} \frac{\hat{\pi}\hat{p}}{(1 - \hat{\pi}\hat{p})}(1 - FNR) = 0.0005 \tag{3.8}$$

and

$$\hat{q}_1 = \hat{\pi}(1 - FNR) + (1 - \hat{\pi})\hat{q}_0 = 0.104. \tag{3.9}$$

Note that these estimates do not take into account more recent developments in Y2H methodology, which will have increased screening accuracy. Further, they are based on raw Y2H data, and not on the curated datasets deposited in BioGRID.

Co-expression

Further, let $C_{uv} \in [-1, 1]$ be the gene co-expression between u and v. We assume that C_{uv} and Y_{uv} are conditionally independent given I_{uv}, and that gene co-expression follows a mixture model with $C_{uv} \sim f_1(\cdot)$ for true interactions, and $C_{uv} \sim f_0(\cdot)$ for non-interacting protein pairs. Intuitively, we expect interacting pairs to have higher co-expression, since two proteins need to be in the cell at the same time in order to interact. A similar framework is used by GeneNet (Schäfer and Strimmer 2004) to separate interactions from non-interactions.

Co-expression density estimates $\hat{f}_0(c)$ and $\hat{f}_1(c)$ were calculated by applying kernel density estimators to the data. Since only a small proportion $\hat{p} = 0.041$ of pairs are assumed to interact, the distribution $\hat{f}_0(c)$ was estimated from the full set of co-expression values. The co-expression density for interacting pairs $\hat{f}_1(c)$ was estimated from co-expressions values for protein pairs in the PRS. In both cases an Epanechnikov kernel (Epanechnikov 1969) was applied with Silverman's rule of thumb bandwidth (Silverman 1986). Due to the relatively large number of data points, alternative density estimators did not produce significantly different results. See Figure 3.3 for the two density estimates.

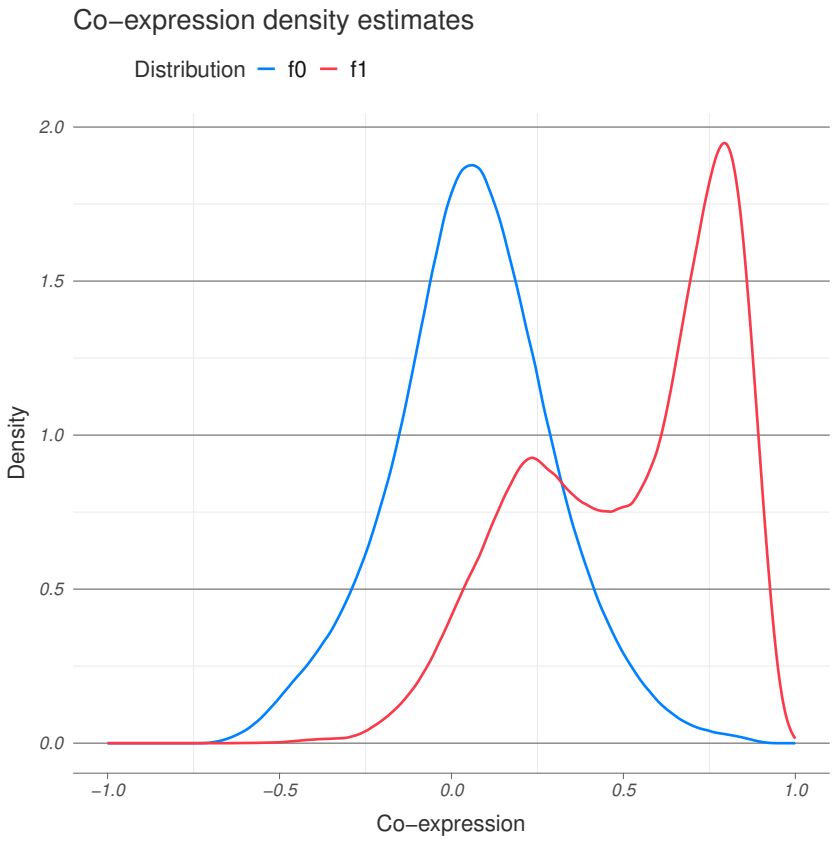

Figure 3.3: Co-expression density estimates for interacting and non-interacting pairs. The co-expression density for non-interacting pairs $f_0(\cdot)$ was estimated using the full dataset (blue). The co-expression for interacting pairs $f_1(\cdot)$ was estimated using the PRS obtained from STRING (red). Interacting pairs have significantly higher co-expression.

Score

Under the conditional independence assumption and using Bayes' Rule, two proteins u and v with yeast two-hybrid outcome $y \in \{0, 1\}$ and co-expression value $c \in [-1, 1]$ interact with probability

$$\begin{aligned}\mathbb{P}(I_{uv} = 1 | Y_{uv} = y, C_{uv} = c) = \\ = \frac{\mathbb{P}(Y_{uv} = y | I_{uv} = 1) f(C_{uv} = c | I_{uv} = 1) \mathbb{P}(I_{uv} = 1)}{f_{Y,C}(Y_{uv} = y, C_{uv} = c)},\end{aligned} \tag{3.10}$$

where $f_{Y,C}(y, c)$ is the joint distribution of Y_{uv} and C_{uv}. Since $I_{uv} \sim Bernoulli(p)$, $Y_{uv} \sim Bernoulli(q(I_{uv}))$, and $C \sim f_{I_{uv}}(\cdot)$ as described above,

$$\begin{aligned}\mathbb{P}(I_{uv} = 1 | Y_{uv} = y, C_{uv} = c) = \\ = \frac{q_1^y (1 - q_1)^{1-y} f_1(c) p}{q_1^y (1 - q_1)^{1-y} f_1(c) p + q_0^y (1 - q_0)^{1-y} f_0(c)(1 - p)}.\end{aligned} \tag{3.11}$$

In order to score yeast interactions, we applied the estimates described earlier in the section:

$$s(uv|Y_{uv} = y, C_{uv} = c) = \frac{\hat{q}_1^y(1-\hat{q}_1)^{1-y}\hat{f}_1(c)\hat{p}}{\hat{q}_1^y(1-\hat{q}_1)^{1-y}\hat{f}_1(c)\hat{p} + \hat{q}_0^y(1-\hat{q}_0)^{1-y}\hat{f}_0(c)(1-\hat{p})}. \quad (3.12)$$

This procedure allows us to construct uncertain PINs based on experimental interaction data. As already discussed in Chapter 1, as well as above, many estimates for error rates and true and false positive co-expression distributions are potentially inaccurate. However, this simple model can be easily modified to accommodate for changes in our understanding of the processes involved—e.g. improvements in Y2H screening accuracy and new estimates for PIN edge density.

3.3.2 The yeast uncertain network

The Y2H and co-expression data were integrated and scored using the procedure described above, in order to obtain an uncertain protein interaction network for yeast. We refer to this as the YEAST network (note this is different from the network of the same name in Chapter 2). The network was constructed on the 4461 proteins for which gene expression was available. A score was calculated for every pairwise interaction, using Equation 3.12 and the available data. The resulting score distribution had mean 0.026, standard deviation 0.066 and was heavily skewed, with 76% of scores falling below the mean. See Figure 3.4 for the score distribution of the yeast uncertain network.

This uncertain network can be thresholded, like the scored networks from Chapter 2. It can also be used as the input of a random graph model, the properties of which can then be analysed. However, since the underlying true state of the PIN is not known, it cannot be easily used to assess different analysis techniques. For this purpose, a synthetic "yeast-like" network was created.

3.3.3 Synthetic "yeast-like" network

The synthetic "yeast-like" network was generated by first generating a "true" network, and then associating to every pair of nodes a simulated gene co-expression

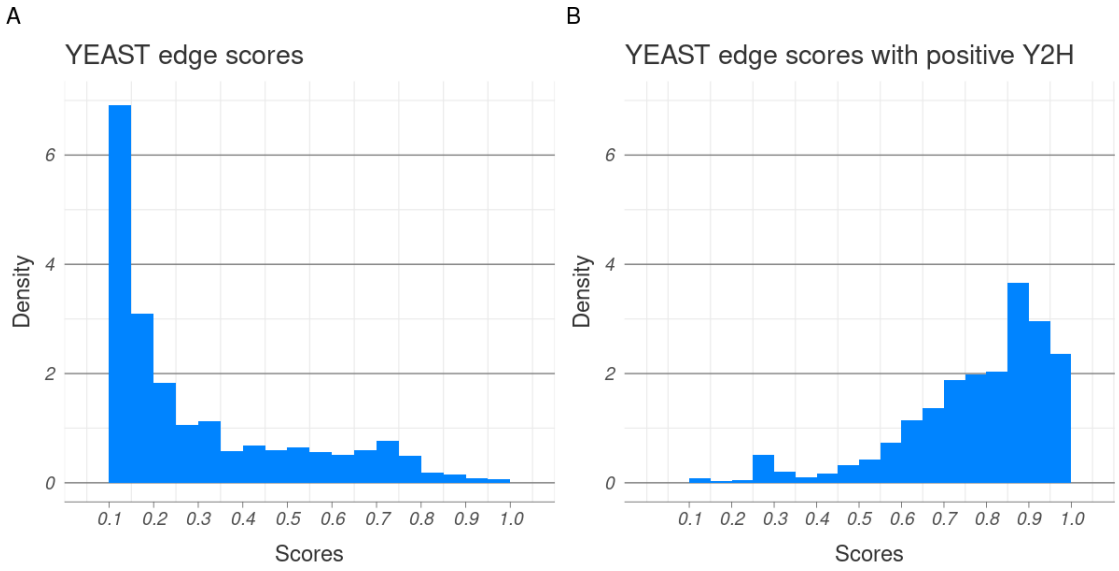

Figure 3.4: Uncertainty scores for the YEAST network. Only scores above 0.10 have been plotted. Binwidth has been set to 0.05. (A) The majority of scores in the network were low. (B) However, edges with associated positive Y2H screen had higher scores.

and Y2H outcome. Afterwards, the simulated data was scored using the scoring procedure described in Equation 3.12. The result was a pair of networks—an observed true network G_T, and an uncertain scored network G_U. We refer to the uncertain network as SYN. The network was generated so that the true network has PIN-like structure, and the associated synthetic experimental data follows the framework described in Section 3.3.1.

The true network was generated by thresholding STRING data. In order to achieve true network density $p = 0.041$ as in our model, the threshold was set at $\theta = 0.226$. Only proteins which occur both in STRING and in COXPRESdb were used for this. This resulted in a true network $G_T = (V, E_T)$ with 4419 nodes and 401980 edges. Three nodes were isolated (had no neighbours) in the network.

Each pair of proteins (u, v) was assigned a randomly generated Bernoulli Y2H outcome Y_{uv} with probability $\hat{q}_1$ for $uv \in E_T$ and $\hat{q}_0$ otherwise. The Y2H outcomes were generated independently of each other, so the only dependency present would be the one imposed by the network connectivity. This resulted in 46408 positive Y2H screens in total, of which 41863 were true positives and 4545 were false positives.

The co-expression score for each protein pair was generated using $C_{uv} \sim \hat{f}_1(\cdot)$ for edges and using $C_{uv} \sim \hat{f}_0(\cdot)$ for non-edges. Co-expression values were generated independently of each other and independently of Y2H outcomes.

The Y2H and co-expression values were then scored using the same scoring procedure as for the yeast network. The resulting score distribution had mean 0.041 and standard deviation 0.120 (see Figure 3.5). The mean matched the true network density p by construction.

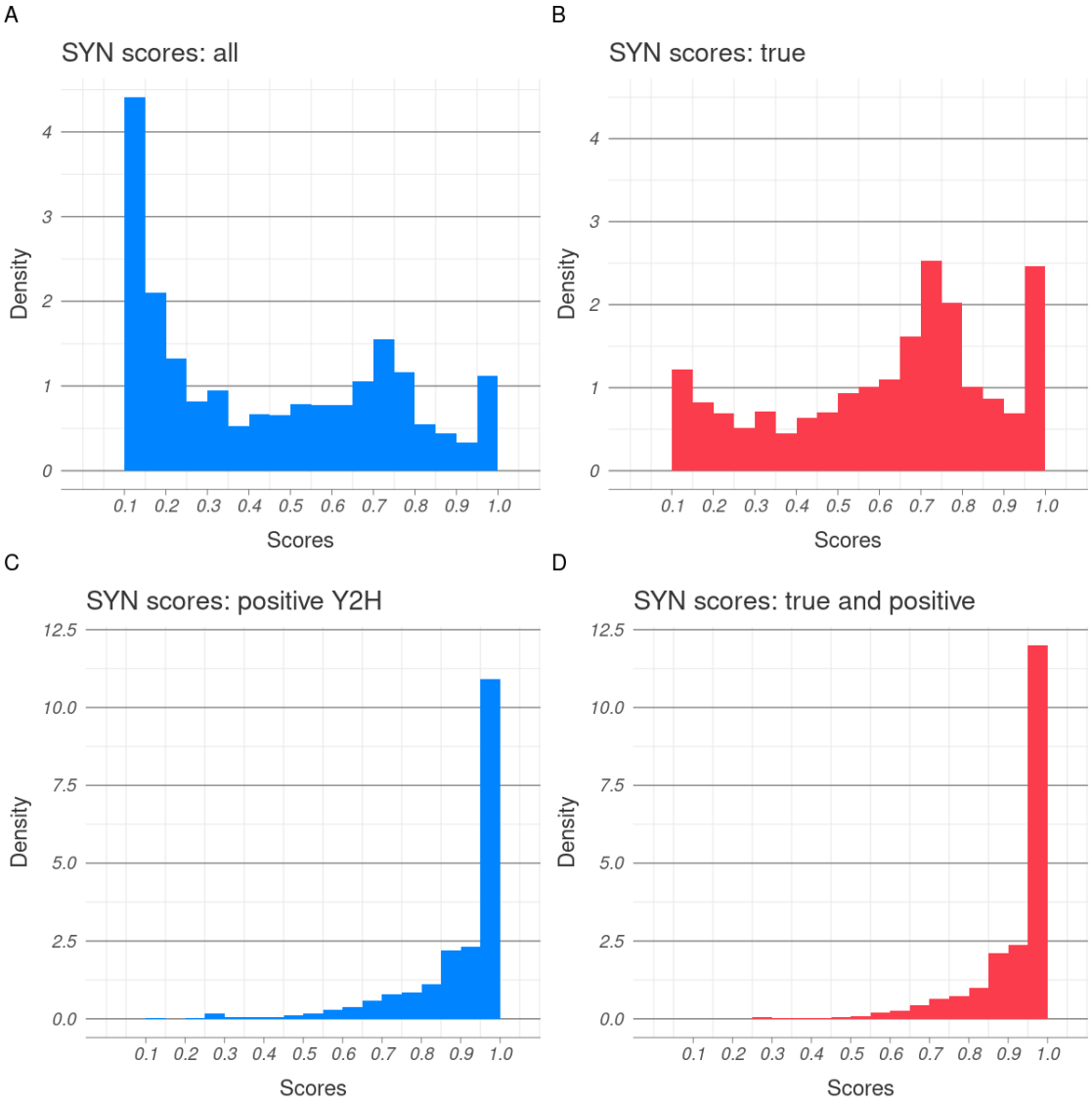

Figure 3.5: Uncertainty scores for the SYN network. Only scores above 0.10 have been plotted. Binwidth has been set to 0.05. The majority of scores in the network were low (A), with true edges more likely to be assigned high scores (B). A positive Y2H screen is much more indicative of high score than a high co-expression value (C and D).

3.3.4 Synthetic Beta and Uniform networks

Two further synthetic networks were generated in order to study how thresholding and random network generation perform when applied to other types of data. Both networks were built on 1000 nodes. In the first network, a *Beta* $(0.01, 0.49)$ distribution was used to generate independent, identically distributed edge scores between every pair of nodes. The parameters for the *Beta* distribution were chosen so that the underlying true network described is expected to be sparse (expected density 0.02). We refer to this network as the BETA network. In the second network, independent and identically distributed scores were generated using a *Uniform* $(0, 1)$ distribution. We refer to this network as the UNI network.

A summary of all four networks can be found in Table 3.2. These uncertain networks were analysed by thresholding and repeated random network generation in order to establish what connectivity patterns (or lack thereof) can be inferred from them.

Name	Score distribution	Number of nodes	Expected density
YEAST	Scored, real data	4461	0.026
SYN	Scored, synthetic data	4419	0.041
BETA	Beta $(0.01, 0.49)$	1000	0.020
UNI	Uniform $(0, 1)$	1000	0.500

Table 3.2: Summary of the four uncertain networks. The left-most column corresponds to the names the networks are referred as later in the text. The expected density refers to the expected density of the underlying true network.

3.4 Generative models based on uncertain networks

As they have been described so far, the scores in uncertain networks only contain information about the marginal probability that edges exist, without capturing any dependencies between edges. However, such dependencies are likely to exist and may influence any inference on the structure of the underlying true network. The same scored uncertain network G_U can describe random graph models with the same marginal but different joint edge probabilities.

For example, consider a complete uncertain network on 10 nodes, where each edge has confidence score 1/3. An Erdős–Rényi random graph $\mathcal{G}(n,p)$ with $n = 10$, $p = 1/3$ is characterised by such an uncertain network with the added assumption that the edges in the uncertain network behave independently. This independence assumption is common in the literature (e.g. Martin et al. 2016; Newman 2017; Ahnert et al. 2007). However, the same uncertain network also accurately describes the marginal edge probabilities of a random 3-regular graph on 10 nodes. Therefore, even basic properties of the underlying true network cannot necessarily be inferred from the uncertain network without any additional assumptions. In this section we outline three different algorithms for generating random graphs which are compatible with a given uncertain network.

The easiest way to generate a random network $G_R = (V, E_R)$ that satisfies the marginal edge probabilities defined by an uncertain network $G_U = (V, E_U)$ is to treat each edge $e \in E_U$ as an independent $Bernoulli(s(e))$ random variable with probability of success equal to its score $s(e)$. This is formalised in Algorithm 3.1.

Algorithm 3.1 The independent edges algorithm (IE).

Require: $G_U = (V, E_U)$
  Initialise $E_R \leftarrow \emptyset$
  for $e \in E_U$ **do**
    Generate $X_e \sim Bernoulli(s(e))$
    if $X_e = 1$ **then**
      $E_R \leftarrow E_R \cup \{e\}$
    end if
  end for
  return $G_R = (V, E_R)$

The random graph model defined by this algorithm is an extension of the $\mathcal{G}(n,p)$ model, in which different probabilities may be associated with different edges. Ahnert et al. 2007 have derived some theoretical results about the expected values of network metrics under this model.

Often real-life networks are characterised by some form of local edge dependence. In the context of biological networks, if protein u interacts with two proteins v and w, which are known to participate in the same process, we would expect

that v and w are more likely to interact with each other than a random protein pair. One way to introduce local dependence in random network generation while preserving the marginal edge probabilities is to sample edges jointly. This can be achieved, for example, by partitioning the uncertain edge set E_U into non-overlapping subsets $E_1, E_2, \ldots$ and then introducing dependence between the edges of the same subset, but not between edges across different subsets. A simple way of introducing such dependence is by inverse transform sampling using the same realisation of a $Uniform(0,1)$ random variable for all edges within a partition. We call this the fixed edge partition (FEP) algorithm (see Algorithm 3.2).

Algorithm 3.2 The fixed edge partition algorithm (FEP).

Require: $G_U = (V, E_U)$, $\{E_i\}_{1 \leq i \leq k}$—a fixed partition of E_U
  Initialise $E_R \leftarrow \emptyset$
  Generate i.i.d $\{U_i\}_{1 \leq i \leq k} \sim Uniform(0; 1)$
  for $i \in \{1, \ldots, k\}$ **do**
    for $e \in E_i$ **do**
      if $s(e) \geq U_i$ **then**
        $E_R \leftarrow E_R \cup \{e\}$
      end if
    end for
  end for
  return $G_R = (V, E_R)$

To see why a realisation of Algorithm 3.2, $G_R = (V, E_R)$, satisfies the marginal probabilities given by $s(e)$, consider any $e \in E_U$. Note that $e \in E_i$ for some unique E_i in the partition of E_U. Then

$$\mathbb{P}(e \in E_R) = \mathbb{P}(e \in E_i) = \mathbb{P}(U_i \leq s(e)) = s(e). \tag{3.13}$$

This can further be extended to random partitions of the edge set, provided that U_i are independent of the choice of edge partition. We call this the random edge partition (REP) algorithm (see Algorithm 3.3). For example, suppose each edge is randomly assigned to one of k partition subsets, and then each subset is assigned a uniform random variable. The partition can be generated according to a multinomial distribution, a Chinese Restaurant process, or in any other way. Whether the edge

appears in the random network will depend on a randomly chosen variable in the set $U_1, \ldots, U_k$, and the probability of it appearing will still be equal to its score.

Algorithm 3.3 The random edge partition algorithm (REP).

Require: $G_U = (V, E_U)$, $f : E_U \to \{E_i\}_{i \geq 1}$—a *random* function which partitions E_U
  Initialise $E_R = \emptyset$
  Generate $\{E_i\}_{1 \leq i \leq k} \sim f(E_U)$
  Generate i.i.d $\{U_i\}_{1 \leq i \leq k} \sim Uniform(0; 1)$
  for $i \in \{1, \ldots, k\}$ **do**
    for $e \in E_i$ **do**
      if $s(e) \geq U_i$ **then**
        $E_R \leftarrow E_R \cup \{e\}$
      end if
    end for
  end for
  return $G_R = (V, E_R)$

To see why the marginal probabilities are satisfied under the REP algorithm, consider an output $G_R = (V, E_R)$ and let $e \in E_U$. Then, by conditioning on the random partition $f(E_U)$ and using the fixed partition argument above,

$$\begin{aligned} \mathbb{P}(e \in E_R) &= \sum_{\{E_1, \ldots\}} \mathbb{P}(e \in E_R | \{E_1, \ldots, E_k\}) \mathbb{P}(f(E_U) = \{E_1, \ldots, E_k\}) \\ &= \sum_{\{E_1, \ldots\}} s(e) \mathbb{P}(f(E_U) = \{E_1, \ldots, E_k\}) \\ &= s(e) \times \sum_{\{E_1, \ldots\}} \mathbb{P}(f(E_U) = \{E_1, \ldots, E_k\}) \\ &= s(e) \times 1 = s(e). \end{aligned} \tag{3.14}$$

These algorithms both introduce dependencies within each member of the edge partition. To see this, consider two edges, e_1 and e_2, with similar confidence scores $s(e_1) \approx s(e_2)$. If the two edges are in different partitions, they will be placed in the random network independently of each other. However, if they are placed within the same partition, they will almost always either both be present, or both be absent in the random network.

Since edge dependence tends to be a local phenomenon in real-life networks, we consider two different ways of partitioning the edge set, so nearby edges more

commonly fall within the same edge partition. One of these is a FEP algorithm, and the other one is a REP algorithm.

Two protein interactions with similar confidence scores may appear close to each other in the network (e.g. the same protein is an interactor in both) because the data for them originated from the same study or experimental sample, and is therefore subject to the same error. Therefore, an algorithm which places edges with similar scores within the same partition might be suitable for studying uncertain PINs. We call this the score-oriented FEP (see Algorithm 3.4). The edge set is first partitioned by binning the edge scores. Further, we split each subset E_i into $E_{i,1}, E_{i,2}, \ldots, E_{n_i}$ so that each $E_{i,j}$ defines a connected component in a graph with edge set E_i. Thus each subset $E_{i,j}$ consists of a connected group of edges, all of which have scores within the same range (x_{i-1}, x_i). Any two different resulting subsets $E_{i,j}$ and $E_{k,l}$ either contain edges with different scores (if $i \neq k$) or contain similar scores, but are separated in space (if $i = k$ and $j \neq l$).

Algorithm 3.4 The score-oriented FEP.

Require: $G_U = (V, E_U)$ and $0 = x_0 \leq x_1 \leq, \ldots, \leq x_{k-1} \leq x_k = 1$
  Initialise $E_1 = \emptyset, \ldots, E_k = \emptyset$.
  for $e \in E_U$ **do**
    Find i such that $s(e) \in (x_{i-1}, x_i)$
    $E_i \leftarrow E_i \cup \{e\}$
  end for
  for $i \in \{1, \ldots, k\}$ **do**
    $G_i = (V, E_i)$
    Find the connected components $E_{i,1}, \ldots, E_{i,n_i}$ of G_i
  end for
  Define the partition $P(E_U) = \{E_{i,j} |\ 1 \leq i \leq k,\ 1 \leq j \leq n_i\}$.
  $G_R \leftarrow FEP(G_U, P(E_U))$ (see Algorithm 3.2)
  return G_R

A way to introduce local dependence with random partitioning is to randomly iterate over the proteins in the network. We define partitions iteratively by starting at a node v_1 chosen uniformly at random from V and taking all edges incident to it to form E_1. Then we pick another node v_2 and take all edges incident to that, aside from $v_1v_2 \in E_U$, which is already in E_1. Then we pick v_3, build a

star around it from the remaining edges to form E_3 and so on. We describe the node-oriented REP algorithm (Algorithm 3.5) by generating the partition explicitly, and feeding the output to the FEP algorithm.

Algorithm 3.5 The node-oriented REP.

Require: $G_U = (V, E_U)$
  Generate a random permutation $\{v_1, \ldots, v_n\}$ of V
  Initialise $E_1 = \emptyset, \ldots, E_{n-1} = \emptyset$
  for $e = (v_i, v_j) \in E_U$ **do**
    if $i \leq j$ **then**
      $E_i \leftarrow E_i \cup \{e\}$
    else
      $E_j \leftarrow E_j \cup \{e\}$
    end if
  end for
  Define the partition $P(E_U) \leftarrow \{E_1, \ldots, E_{n-1}\}$
  $G_R = FEP(G_U, P(E_U))$ (see Algorithm 3.2)
  return G_R

To study the behaviour of random network generation under different algorithms, we apply independent edge sampling, the score-oriented FEP, and the node-oriented REP to each uncertain network described in the previous section (YEAST, SYN, BETA and UNI). For each uncertain network, 500 random network realisations were generated using the independent edges algorithm and the node-oriented algorithm, and 100 realisations were generated using the score-oriented algorithm. The lower number of score-oriented algorithm samples was due to the associated computational costs. Edge partitions corresponding to 1/500 quantiles for the YEAST and SYN networks, and 1/100 quantiles for the toy UNI and BETA networks, were used. In addition, we compare the random networks to networks obtained through thresholding, with 100 equidistant thresholds between 0.00 and 0.99.

As discussed in Chepter 2, thresholding is commonly used for scored protein–protein interaction data. A thresholded network is obtained by removing all edges with scores below the threshold, and preserving the edges with scores at or above the threshold. Since an edge score represents the marginal likelihood that the edge belongs to the true network, it is extremely unlikely that any thresholded

network captures the true state accurately. However, if the set of thresholded networks do not vary considerably, or if a threshold which matches the true state closely exists and can be easily identified, then thresholding may be a quick and efficient way of inferring properties of the true network. Including this approach, the final network dataset consisted of 1600 simple, binary networks for each of the scored uncertain networks—500 for each of the generative models, and 100 obtained through thresholding.

3.5 Results

In order to investigate how the random networks generating algorithms and thresholding behave, we studied the simple networks generated from the YEAST, SYN, BETA and UNI uncertain networks. First, we examined the agreement between confidence scores and frequency of edge occurrence in the randomly generated datasets. In theory, an edge uv with confidence score $s(uv)$ should appear in approximately $s(uv)$ of all network realisations for each of the random network algorithms. We studied whether the correct frequency of edge occurrence is observed in relatively small sample sizes.

The datasets (both thresholded and randomly generated) were then compared with respect to a set of network metrics: number of edges per network, size of the largest connected component, number of connected components, global clustering coefficient, and average local clustering coefficient. We chose this set of metrics because they are well-known, easy to compute, and of interest in the context of PINs. The overall number of edges has important implications in determining whether part of a network is relatively dense and may therefore be of interest for further study (G. D. Bader and Hogue 2003). Separate connected components may correspond to largely independent pathways and processes, whereas subnetworks with high clustering (i.e. a large number of triangles) can help identify protein complexes and functional units within the cell (Ravasz et al. 2002).

3.5.1 Frequency of edge occurrence

As discussed in Section 3.4, both the node-oriented REP and score-oriented FEP should produce networks with marginal edge probabilities equal to the scores of the uncertain networks used as input. However, the edge dependencies introduced in both algorithms make it difficult to evaluate the covariance in edge occurrence and therefore the overall variability in the produced networks.

We compared the frequency of edge occurrence and confidence scores for the three random network algorithms (independent edges, score-oriented FEP and node (or vertex) oriented REP) across all four uncertain networks. The independent edges and node-oriented algorithms showed consistently stable behaviour, i.e. the observed frequencies of edge occurrence closely matched their respective confidence scores. The score-oriented algorithm showed correct behaviour on average but larger deviation from the confidence scores in all networks (Figures 3.6, B.1–B.3).

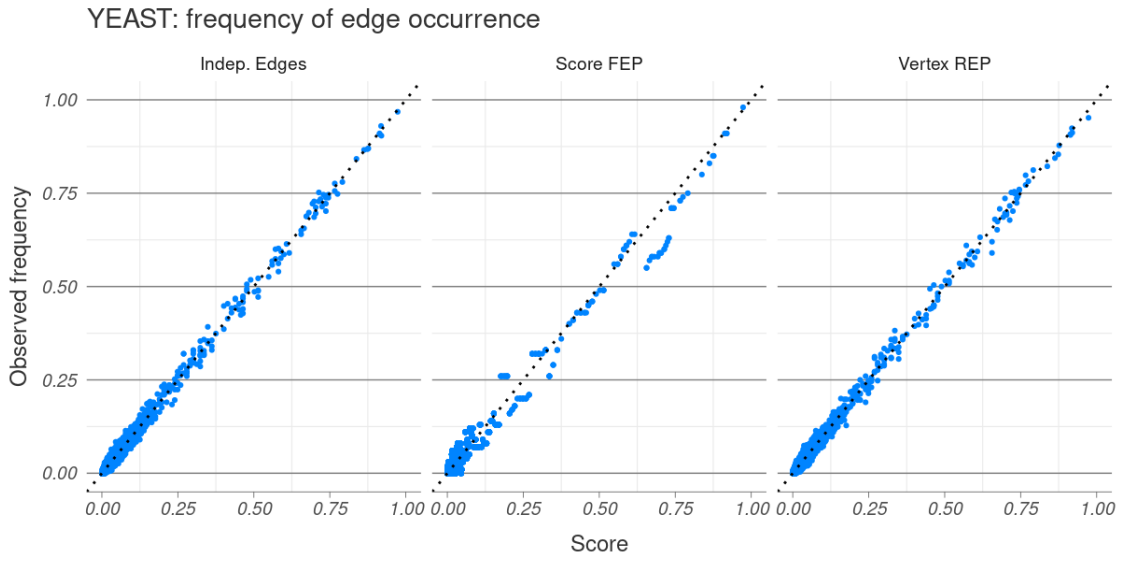

Figure 3.6: Frequency of edge occurrence for the YEAST network. Ten thousand arbitrarily chosen edges are plotted. The dotted line corresponds to the identity. All three methods on average produce networks with the correct edge frequency. Largest deviations are observed in the score-oriented FEP (middle), which also has a smaller sample size (100 as opposed to 500 networks).

3.5.2 Number of edges

Since all algorithms produce networks with the correct marginal edge probabilities, the edge counts in the datasets generated using them agreed on average with the sum of all confidence scores in the corresponding uncertain networks. This is the case as the expected number of edges in a randomly generated network only depends on the marginal probability of edge occurrence.

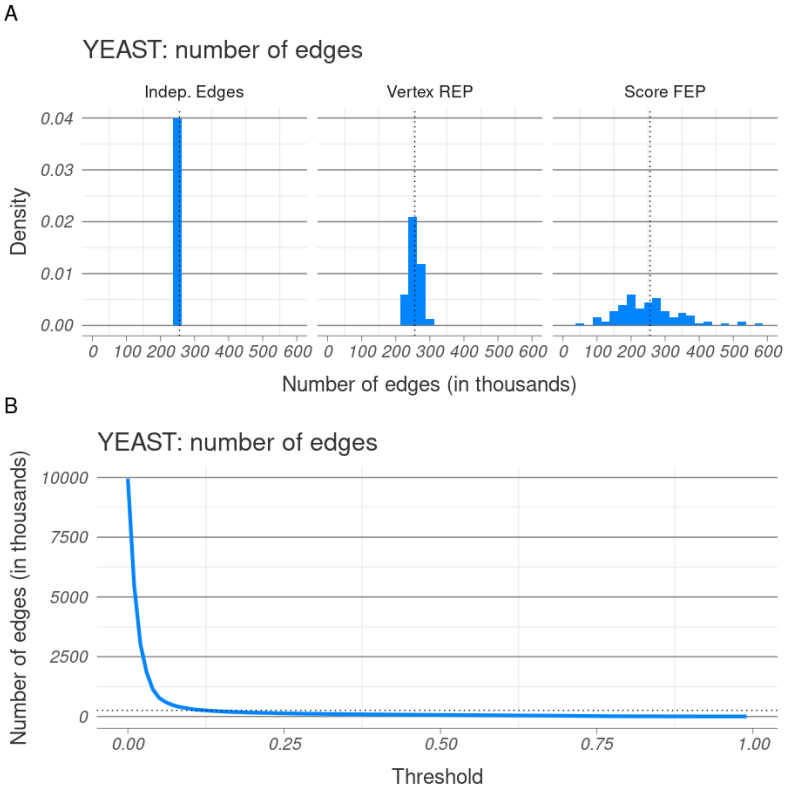

Figure 3.7: Number of edges for the YEAST network. (A) Histograms of edge counts, in thousands, obtained across the different algorithms. (B) Edge counts, in thousands, obtained by thresholding. The dotted lines correspond to the expected number of edges in the true network.

The independent edge algorithm consistently showed the smallest variance in total number of edges, and the score-oriented algorithm showed the largest out of the random generators (see Figures 3.7, B.4–B.6). In all cases the networks produced through thresholding showed a much wider range of values for the total number of edges than any of the random algorithms—from the complete set of $\binom{N}{2}$ edges across N nodes for a threshold of 0, down to close to zero edges for a threshold

of 0.99. The threshold which most closely matched the expected number of edges in the true network differed for the four uncertain networks. It was 0.13 for YEAST, 0.42 for BETA and 0.51 for UNI. For the SYN network, where the true state G_T is known, the true number of edges was most closely matched by thresholding at 0.17.

3.5.3 Size of largest connected component and number of connected components

Protein interaction networks are generally presumed to consist of dense communities which are sparsely connected to each other (Ravasz et al. 2002). One way of examining how well these sparse connections may be captured by random network generation or thresholding is by studying the size of the largest connected component and the overall number of connected components in the generated datasets. With limited and uncertain data it is easy to see how these sparse connections may fail to be captured by random network generation or by thresholding, possibly resulting in multiple connected components.

All random network algorithms generated connected or nearly connected networks, across all four uncertain networks (Figures 3.8, B.7–B.11). This is likely due to sampling behaviour—in large networks, a single edge is enough to connect a node to the giant component and every node is incident to a large number of potential edges. Thresholding, in contrast, reduces the size of the largest connected component and increases the total number of connected components. This effect is more strongly pronounced in YEAST (Figures 3.8 and B.9) than in SYN (Figures B.7 and B.10), where disconnect occurs at higher thresholds. This may be due to the way the SYN network was generated—evidence (including gene co-expression) was assigned independently to different node pairs. In the YEAST network, in contrast, we would expect that if the protein pair (u, v) and the protein pair (v, w) both have high co-expression scores, then the pair (u, w) would also be characterised with higher co-expression. This would lead to aggregation of high scores which would not be present in the case where co-expression scores are assigned at random.

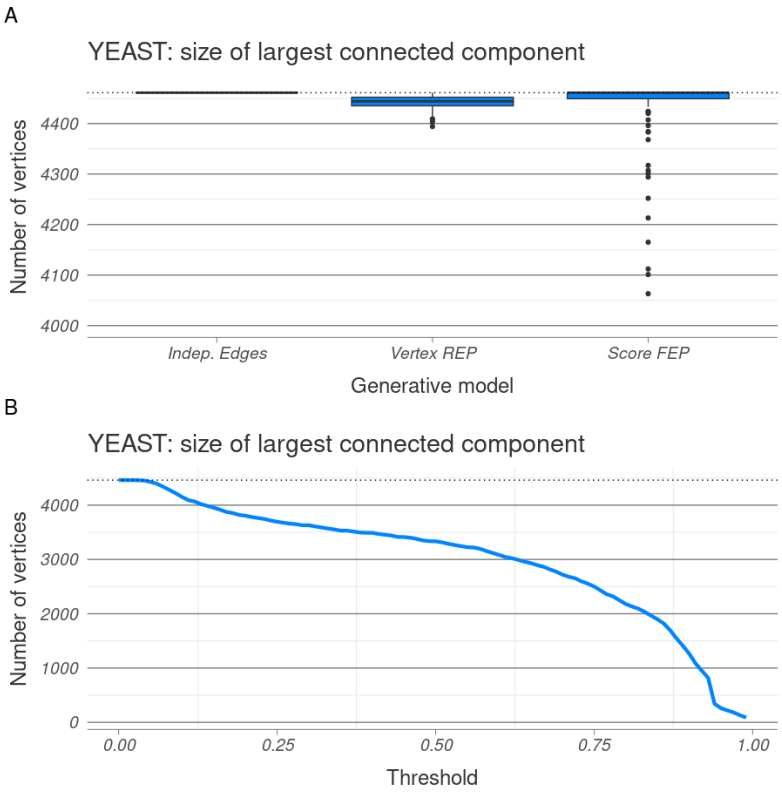

Figure 3.8: Largest connected component size for the YEAST network. (A) Box plot of the size of the largest connected component across different algorithms. (B) Size of the largest connected component across different thresholds. The dotted line corresponds to the number of nodes in the network.

Out of the random network generating algorithms,the score-oriented FEP most commonly produced disconnected networks when applied to both YEAST and SYN. However, when applied to BETA, the node-oriented REP was most likely to produce disconnected networks (Figures B.8 and B.11). Therefore, algorithm behaviour can vary significantly both quantitatively and qualitatively with the uncertain network input.

All random network algorithms generated only connected networks from UNI. Thresholding also produced fully connected networks for thresholds up to and including 0.99. This result is not surprising: for an uncertain network to be connected after thresholding at 0.99, it must have at least one spanning tree consisting of edges with scores at or above 0.99. By Cayley's formula, there exist 1000^{998} different labelled trees on 1000 nodes. For such a tree T, $\mathbb{P}(s(e) \geq 0.99 \ \forall e \in T) = 0.01^{999}$. Therefore, the uniform toy network is expected to have $1000^{998} \times 0.01^{999} = 10^{969}$

labelled spanning trees with scores above 0.99.

3.5.4 Global clustering coefficient

Another commonly used network metric is the global clustering coefficient, or the overall proportion of connected triplets in the network which are also triangles. Like the size of the largest connected component, the global clustering coefficient demonstrates further the difference between the YEAST and SYN uncertain networks.

The YEAST network is characterised by a higher global clustering coefficient (Figure 3.9), which further shows that in this network, high scores are more likely to be close together. Both thresholding and random network realisations produced significantly lower global clustering in the SYN network (Figure B.12). None of the algorithms, nor thresholding reached the global clustering of the underlying true network for SYN. The low global clustering obtained from the random network algorithms suggests that an algorithm with a stronger dependency between edges is needed.

In both the SYN and BETA networks (Figures B.12 and B.13, respectively), the three random network algorithms produced networks with comparable global clustering coefficients. However, the YEAST and UNI uncertain networks produced three significantly different distributions across the three algorithms (Figures 3.9 and B.14 respectively). In all cases, a larger variance was observed for the score-oriented FEP, which tended to be close to or agree on average with the independent edge algorithm.

The node-oriented REP behaved differently with respect to the other two algorithms across different uncertain networks. It generally produced networks with lower global clustering from the YEAST network, with comparable clustering from the SYN and BETA networks, and with higher clustering from the UNI network.

3.5.5 Average local clustering coefficient

A metric closely related to the global clustering coefficient is the average local clustering coefficient. It is a measure of the proportion of triangles around each node, averaged over the entire network. While the two metrics are similar, they are

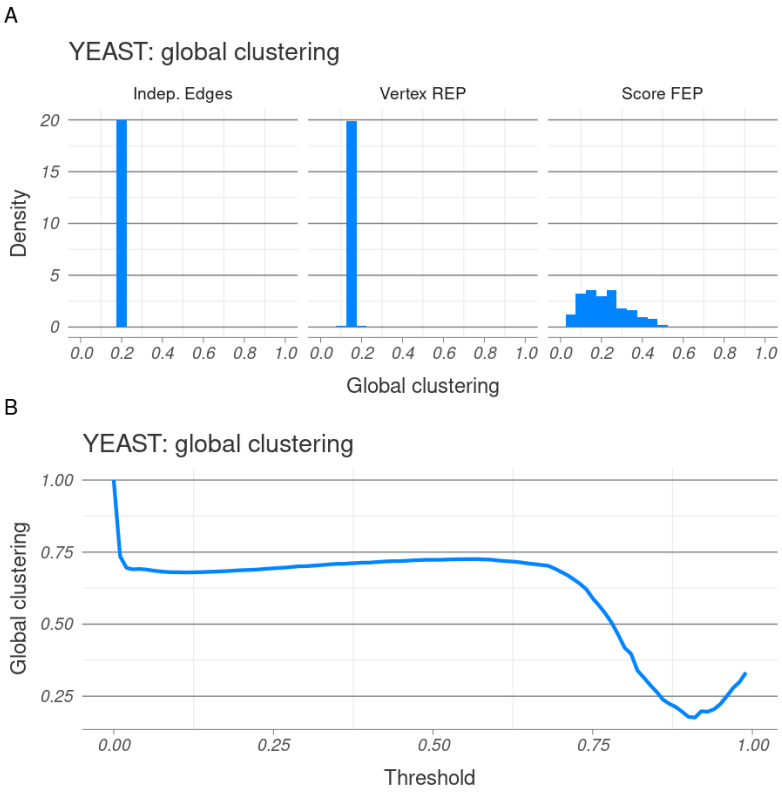

Figure 3.9: Global clustering coefficient for the YEAST network. (A) Histograms of the global clustering coefficient across different algorithms. (B) Global clustering coefficient across different thresholds.

not necessarily close in value—e.g. compare how the two change with the threshold in the case of the YEAST network (Figures 3.9B and 3.10B).

In our datasets, the independent edge algorithm and score-oriented FEP produced networks with similar mean average local clustering across all four uncertain networks. This is in agreement with the behaviour of global clustering. Average local clustering was also consistently higher in networks constructed using the node-oriented REP (Figures 3.10 and B.17). The node-oriented REP was the only algorithm which produced any values above 0.50 when applied to the YEAST network (Figure 3.10), as well as values close to the average local clustering in the true network for SYN (Figure B.15).

Overall, the analysis so far has shown that thresholding confidence scores can result in very different networks, and that there exists no single threshold which can be used to reliably estimate a number of different metrics (e.g. thresholding at

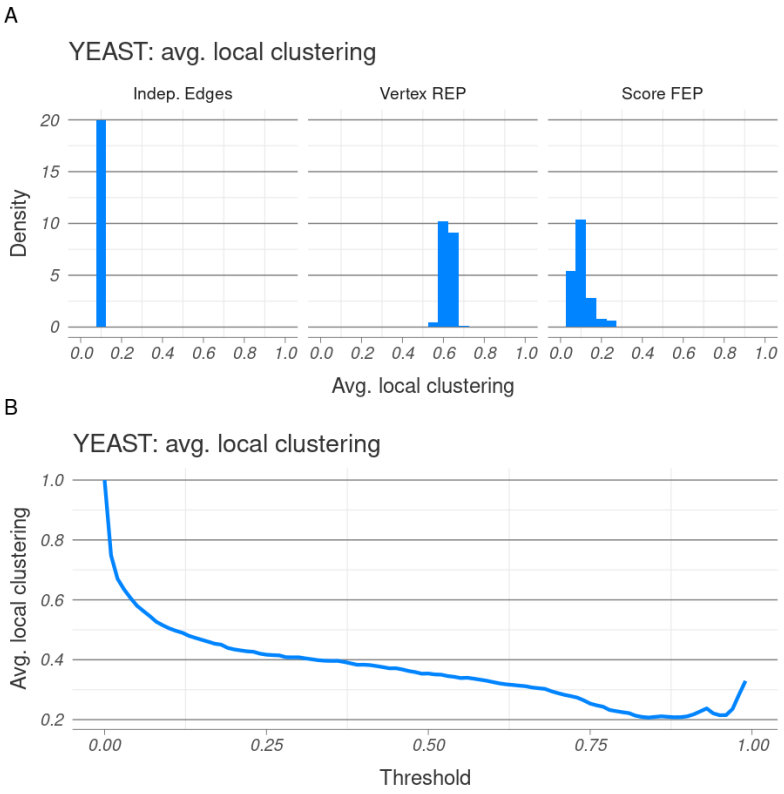

Figure 3.10: Average local clustering coefficient for the YEAST network. (A) Histograms of the average local clustering coefficient across different algorithms. (B) Average local clustering coefficient across different thresholds.

0.17 will match the true number of edges in the synthetic yeast network but will underestimate the global clustering coefficient). Further, the three random network generating algorithms we have described can be used to produce significantly different networks. These differences depend on the type of uncertain network the algorithms are applied to.

We have also shown that a significant difference exists between the YEAST network, which was constructed using real experimental data, and the SYN network, which was randomly generated. In order to be able to identify whether any algorithm can be used to generate plausibly PIN-like networks, it will be necessary to develop better synthetic networks to serve as a benchmark for performance.

3.6 Discussion

Uncertainty in experimental data is an issue in most, if not all, branches of biomedical science. In the context of protein interaction networks, classically this is tackled during data pre-processing, for example by assigning confidence scores to interactions and applying a threshold. Our analysis so far has shown that this thresholding procedure has some significant shortcomings—the choice of a single threshold value is not necessarily obvious, and different thresholds can result in significantly different networks. A more robust methodology for analysing uncertainty in protein interaction networks is clearly needed.

In Chapter 2 we discussed robustness across thresholds as a possible approach. Following on work carried out by Martin et al. 2016, in this chapter we investigated uncertain networks as a possible way of incorporating data uncertainty directly into PINs. Uncertain networks have scores associated with each edge. The score of an edge corresponds the likelihood that the edge is truly present (i.e. the protein interaction exists and is biologically meaningful) given the available evidence. It is then of interest to estimate the properties of the unobserved underlying true network of biologically meaningful interactions based on the observed uncertain network.

We constructed four such uncertain networks—one based on Y2H and co-expression data for yeast (*S. cerevisiae*), one synthetic "yeast-like" network, and two toy networks with confidence scores generated from $Beta$ and $Uniform$ distributions. We analysed these networks through thresholding at values between zero and one. We further investigated how random network generation could be employed to estimate true network properties. By definition uncertain networks impose constraints on the marginal probability of edge existence but not on any joint probabilities. Therefore, different random network generating algorithms could result in marginal edge probabilities consistent with the same uncertain network. We generated networks from three such algorithms, applied to each of the four uncertain networks, resulting in twelve distinct sample sets. The first algorithm, which to our knowledge is the only one to have been studied in this context before (Ahnert et al. 2007; Martin et al. 2016; Newman 2017), assumes that edges are independent of each other. The

node-oriented REP introduces dependency between edges which are close to each other in the network, i.e. which share a node. Finally, the score-oriented FEP introduces dependency between edges with similar confidence scores.

Our simulations show that these three algorithms result in significantly different networks, as measured by global metrics such as the number of edges, size of largest connected component, and clustering coefficients of the resulting samples. Further, the observed differences depend on the type of uncertain network the algorithms are applied to—for example, the node-oriented REP resulted in comparatively high clustering for the UNI network, but comparatively low clustering for the YEAST network. Therefore, only taking into account experimental error on the interaction level may not be enough to successfully analyse uncertain PINs. Instead, we need further knowledge about the dependencies between interactions and how to best model these.

In our analysis, the different generative models aimed to capture these dependencies. Alternatively, a hierarchical model could be used to capture both the structure of the underlying true network and its relationship to the observed interaction data (Newman 2018a; Peixoto 2018).

Under such a model, the true network would no longer be treated as a realisation of a generative model based on the interaction data. Instead, it would be assumed to come from a separate random graph model $G_T \sim \psi(\cdot)$. In the simplest case, $\psi(\cdot)$ could be a Bernoulli random graph $\mathcal{G}(N, p)$. A generalisation using graphons would allow for different values of p_{uv} for different pairs of nodes $(u, v) \in V^2$ (Olhede and P. J. Wolfe 2014). Other choices of $\psi(\cdot)$ are also possible. For example, a duplication–divergence model, which is designed to describe PINs, can be used (Ispolatov et al. 2005). Alternatively, an ERGM parametrised using specific network properties of interest can be employed (Robins et al. 2007).

Under the hierarchical model, the unobserved or latent G_T would give rise to the observed interaction data (Y, C). This could be done in an analogous way to Section 3.3.1: Y2H outcomes could be modelled as a Bernoulli random variables $Y_{uv} \sim Bernoulli(q(I_{uv}))$, and gene co-expression values could be modelled

as continuous random variables on $C_{u,v} \sim f_{I_{uv}}(\cdot)$, conditional on the presence or absence of edges in the true network $I_{uv} = \mathbb{1}\{(u,v) \in E_T\}$.

Under this framework, the network model $\psi(\cdot)$ can be thought of as a prior distribution for the true network G_T. Meanwhile, the data model provides a sampling distribution $f_{Y,C}(Y, C|G_T)$ linking the data (Y, C) and the unobserved network G_T. A posterior distribution for the true network can therefore be calculated $\hat{\psi}(G_T|Y, C) \propto \psi(G_T) f_{Y,C}(Y, C|G_T))$.

From this distribution, a maximum *a posteriori* estimate of the true network can be calculated. Alternatively, the distribution of various network properties of interest may be studied. However, the feasibility of such an approach will depend on the computational tractability of $\hat{\psi}(\cdot)$. Further, we may expect that the posterior distribution $\hat{\psi}(\cdot)$ will depend strongly on the choice of prior $\psi(\cdot)$, much like the results obtained in this chapter depend on the choice of generative model.

It is unrealistic to assume that we can identify which (if any) priors $\psi(\cdot)$ or generative models are the most suitable for the study of PINs through trial and error. Therefore, we may need to develop a data- or knowledge-based approach to model choice. In the next chapter we tackle the problem of choosing network construction methods in the context of gene co-expression data.

Faith had always told herself that she was not like other ladies. But neither, it seemed, were other ladies.

— Frances Hardinge, *The Lie Tree*

4

COGENT: evaluating the consistency of gene co-expression networks

Contents

A manuscript describing the work presented in this chapter is under preparation.

The density adjustment using random samples from the configuration model described in Section 4.3.2 was developed by Javier Pardo-Diaz.

4.1 Introduction

As discussed in Chapter 1, gene expression is a powerful resource for understanding genetic function under different experimental conditions (Petryszak et al. 2015). A common way of exploring these data are gene co-expression networks (Lee et al. 2004). In these networks genes are represented by nodes and highly co-expressed gene pairs are connected by edges. In Chapter 3 we used gene expression data in order to construct uncertain protein interaction networks, reasoning that in order for two proteins to interact, they need to co-occur in the cell. Gene co-expression networks have been used in many other ways, including for gene function prediction and the identification of disease- or tissue-relevant gene modules (van Dam et al. 2017).

Gene expression data typically takes the form of a matrix, in which rows correspond to genes and columns correspond to samples. Samples may be obtained under the same or different experimental conditions. Expression values can be either relative or absolute expression counts, and will depend on the experimental protocol. Gene co-expression network construction from this data usually consists of three steps—the data is pre-processed, a measure of co-expression is calculated for every pair of genes, and a score cut-off is applied, so that only highly co-expressed genes are connected by edges in the final network. Different approaches to data pre-processing and normalisation exist, both for microarray and for RNA-Seq data (T. Park et al. 2003, Abbas-Aghababazadeh et al. 2018). Further, after normalisation co-expression can be measured in a number of different ways—including correlation coefficient, mutual information or Euclidean distance. More sophisticated approaches involving model fitting have also been employed (e.g. Saha et al. 2017). Then either a score cut-off is imposed, resulting in a simple, binary network, or a network with edge weights corresponding to co-expression is produced (Langfelder and Horvath 2008). These choices mean that there exist numerous available methods for network construction which can be applied to a given data set. These different methods can lead to different networks from the same data set.

It is often not clear which of the available network construction techniques is most appropriate for a given data set. In some cases the network produced

can be validated through enrichment analysis or by comparing to orthogonal data, such as protein interaction data. However, external data is not always available in the quality or quantity necessary for validation, e.g. in the case of tissue-specific data (Lonsdale et al. 2013). It is therefore not always clear which of the available network construction methods should be prioritised in different cases (De Smet and Marchal 2010).

In Chapter 3 we treated gene co-expression as a noisy observation of an underlying true network of biologically relevant protein-protein interactions. However, such interpretation is not always necessary, or assumed. Co-expression networks can be thought of simply as a way of representing gene expression data. Edges in co-expression networks do not need to approximate, or represent, a clearly defined binary interaction such as physical binding between proteins for the networks to be useful in bioinformatics analysis.

Calculated co-expression values, and the networks which they result in, can be thought of as a combination of two factors: genuine gene product co-occurrence in the cell (signal) and experimental fluctuations (noise). A good method for constructing co-expression networks should prioritise the former over the latter. It is natural to assume that signal results in consistent co-expression patterns across samples and noise is random and does not. Therefore, a good network construction method should produce similar networks when applied to two subsets of the available samples. This line of thinking is similar to the argument underpinning Chapter 2, where we reasoned that reliable biologically relevant findings should be replicated across different thresholded networks.

In this chapter, we introduce COGENT (COnsistency of Gene Expression NeTworks), an R package designed to aid the choice of a network construction pipeline without the use of any external data or annotation. COGENT works by repeatedly splitting gene expression samples in two sets, constructing a co-expression network from each one, and measuring the agreement across the two networks. It can be used to choose between competing gene expression similarity measures—e.g. between Pearson and Kendall correlation coefficients. It can also

be used to inform score cut-off choice, when simple unweighted networks are required. While designed for gene expression data, COGENT can be applied to other cases where network construction relies on similarity profiling, including microbiome or synthetic lethality data.

We begin with brief implementation details and a description of the protocol employed by COGENT. The key step is network comparison, which can be performed in two different ways. Firstly, we compare networks by studying their intersection or edge overlap. We describe a set of network similarity measures used for this purpose, as well as similarity measures based on node metric agreement. We then illustrate how COGENT can be employed to choose between measures of co-expression and for threshold determination.

4.2 Software description

4.2.1 Implementation details

COGENT was built in R v.6.1.1 (R Core Team 2019) using devtools (Wickham, Hester, et al. 2019) and roxygen2 (Wickham, Danenberg, et al. 2018). It imports the core R package parallel (v.3.6.1 and higher), as well as the packages Matrix (v.1.2.17 and higher) and igraph (v.1.2.4.1 and higher), all of which are available on CRAN. A list of functions implemented in COGENT can be found in Table C.1. Appendix C contains further implementation details.

The main COGENT functions take as input gene expression data and a network construction function and return a set of consistency metrics. These consistency metrics aim to capture the suitability of the network construction function when applied to the data. The higher the consistency, the better the function.

4.2.2 Workflow

If two gene products consistently appear in the cell at the same time or under the same conditions, they are likely to be functionally related (Stuart et al. 2003). Gene product abundance, and hence also co-occurrence, is a continuous-time phenomenon, which is experimentally observed at discrete time points or samples.

The construction of a gene co-expression network can therefore be thought of as an estimation problem—we aim to infer general co-expression patterns from a limited set of data points. One way of investigating the success of such a procedure is through resampling. Networks constructed from a subset of all available expression samples are likely to be noisier than the network constructed built from all available data. However, they should still resemble it, as well as each other: if subsetting the data results in networks with little to no overlap, then either the network construction procedure is too sensitive to noise in the data, or the data itself is not of high enough quality.

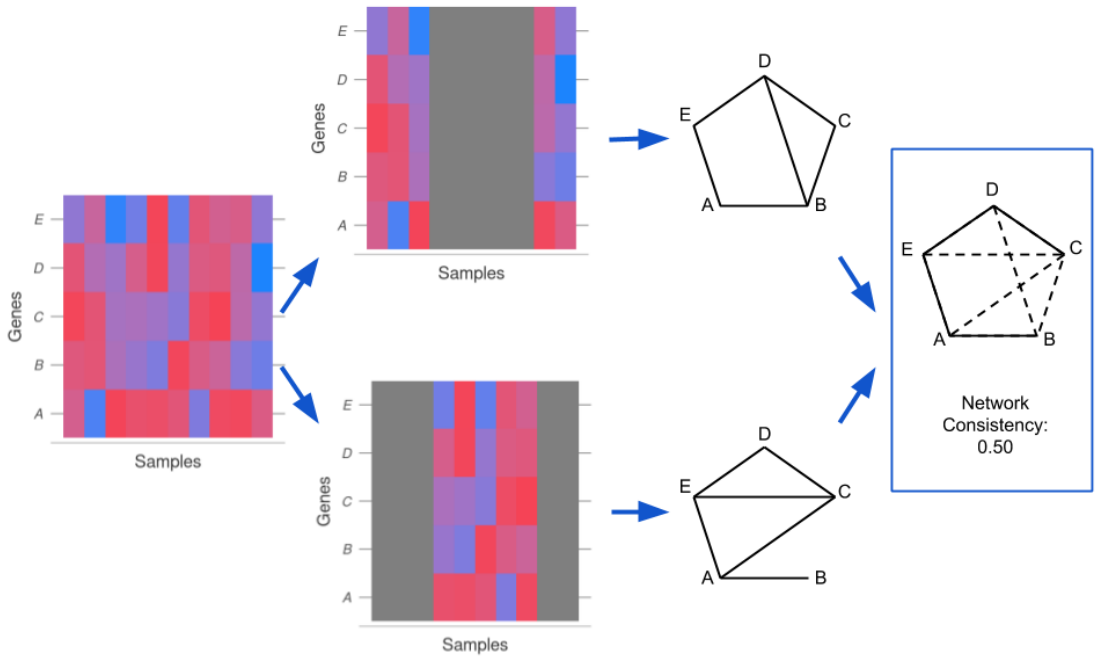

Figure 4.1: COGENT workflow schematic. The input data is a gene expression matrix with rows $A, B, \ldots$, corresponding to genes and columns corresponding to samples (far left). First, the columns are randomly split into two possibly overlapping groups of equal size (left). Then, a network is constructed from each of the sample groups (right). Finally, the two resulting networks are compared and the consistency between them is calculated (far right). In this example, the two networks have six edges each, and overlap at four of these edges. One measure of their consistency is the Jaccard index between their edge sets (see Section 4.3.1), which in this case is 0.50.

COGENT evaluates network construction methods through iterative resampling. At each step, the gene expression samples are split into two possibly overlapping sets of equal size. The overlap amount can be controlled by the user, and the sets

are generated at random. In cases where a large number of samples are available, the subsets can be left disjoint. However, most single-experiment data sets only have a small number of samples. In such cases, the sets can be constructed so that they overlap in the majority of their samples.

After the two random gene expression subsets are generated, the same network construction method is applied to both in order to obtain two gene co-expression networks. Since the data is split by samples and not by genes, the two resulting networks should have identical or significantly overlapping node sets. COGENT then calculates several measures of consistency between these two networks. This workflow is illustrated in Figure 4.1.

Algorithm 4.1 COGENT workflow.

Require: Gene expression data M, network construction method $\psi(\cdot)$, number of repetitions n, [optional] node metric $f(\cdot)$
  for $i \in \{1, 2, \ldots, n\}$ **do**
    Randomly split M into M_1, M_2
    Create networks $G_1 = \psi(M_1)$, $G_2 = \psi(M_2)$
    Calculate the edge set consistency EC_i between G_1 and G_2
    if node metric $f(\cdot)$ is provided **then**
      Calculate the node metric consistency NC_i between metric value vectors $\overrightarrow{f(V(G_1))}$ and $\overrightarrow{f(V(G_2)}$
    end if
  end for
  return $\overrightarrow{EC} = \{EC_1, \ldots, EC_n\}$ and [optional] $\overrightarrow{NC} = \{NC_1, \ldots, NC_n\}$

The entire process is repeated multiple times in order to obtain robust results (see Algorithm 4.1). Consistency can be calculated in a number of different ways, chosen and parametrised by the user. Network construction methods which result in highly similar pairs of networks are considered to be consistent. When two or more competing methods are considered, the method exhibiting higher internal consistency should be preferred.

4.3 Network consistency

Consistency in COGENT is measured through a network comparison step at each iteration. Suppose we are interested in studying some gene expression data set

M_0, and wish to construct a co-expression network G_0 from it by applying some function $G_0 = \psi(M_0)$ to the data. For example, $\psi(\cdot)$ may involve first calculating Pearson correlation coefficients between all pairs of gene expression profiles, i.e. all pairs of rows of M_0, and then selecting the top 10% of these as edges. This will result in a co-expression network, where the 10% most strongly correlated pairs of genes are linked by edges.

The first step of COGENT is to randomly split the data samples M_0 in two groups of equal size, M_1 and M_2. Note that the split is performed only over the samples, so both groups will contain data for all genes in M_0. By default, M_1 and M_2 do not overlap. However, a level of overlap can be set by the user if the overall number of samples in M_0 is low. The network construction function $\psi(\cdot)$ is then applied to M_1 and M_2 in order to produce two networks, $G_1 = \psi(M_1)$ and $G_2 = \psi(M_2)$.

Consistency is measured by comparing G_1 and G_2. We first use edge overlap as a measure of consistency. We calculate a global edge overlap across the networks, and local overlap, which measures how well each gene neighbourhood is preserved across G_1 and G_2 (Kao and Porter 2018). Both of these can be applied to unweighted, as well as weighted networks. Since edge overlap is expected to be higher for denser networks, we introduce a density adjustment for the global edge overlap measure. The density adjusted consistency measure is only suitable for unweighted networks and can be used to inform threshold choice. Finally, we introduce an optional measure of node metric consistency. If, for example, genes of high betweenness in the co-expression network are of interest, then G_1 and G_2 can be compared by the betweenness centrality of their nodes, rather than by the overlap of their edges.

4.3.1 Edge set consistency

Suppose that a single iteration of COGENT applied to some some data M_0 and network construction method $\psi(\cdot)$ results in two networks $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ with a shared node set V and edge sets E_1 and E_2. The two networks G_1 and G_2 can be considered to be similar if $E_1 \approx E_2$. If such similarity is replicated across COGENT iterations, $\psi(\cdot)$ is a consistent method for constructing

networks from the data M_0. We measure agreement between E_1 and E_2 using a Jaccard index if the networks are binary (i.e. unweighted), and a weighted Jaccard index if they are weighted.

Unweighted and weighted Jaccard index

Let A and B be two finite non-empty sets. These could be, for example, the edge sets of two unweighted networks. The *Jaccard index* (Jaccard 1908) between them is calculated as

$$Jacc(A, B) = \frac{|A \cap B|}{|A \cup B|}. \tag{4.1}$$

This is a generic metric of set overlap, and is not restricted to network comparison that is also known as Tanimoto similarity (Tanimoto 1957; Bajusz et al. 2015). It is always in the range $[0, 1]$, with zero corresponding to no set overlap $A \cap B = \emptyset$ and one corresponding to perfect overlap $A \equiv B$.

Applied over edge sets, the Jaccard index can be used to provide a measure of both global and local network similarity, for example for community detection in multilayer networks (Vörös and Snijders 2017; Kao and Porter 2018). In COGENT, we generalise these to the weighted case and use them to perform network comparison between G_1 and G_2 at every iteration of our algorithm.

The Jaccard index can be used to evaluate the similarity of two edge sets, but does not take into account edge weights. A measure of set overlap which penalises weight difference is the *weighted Jaccard index*. Suppose that for some finite index set $\mathcal{I}$, we have two non-negative weight vectors $A = \{a_e\}_{e \in \mathcal{I}}$ and $B = \{b_e\}_{e \in \mathcal{I}}$. In the context of weighted networks, a_e could be the edge weight of $e \in V^2$ in the network G_1 and b_e could be the weight of e in G_2. Non-edges are assumed to have zero weight. The weighted Jaccard index between the weight vectors A and B is

$$Jacc(A, B) = \frac{\sum_{e \in \mathcal{I}} min(a_e, b_e)}{\sum_{e \in \mathcal{I}} max(a_e, b_e)}. \tag{4.2}$$

Like the Jaccard index (Equation 4.1), the weighted Jaccard index is in the range $[0, 1]$, with higher values corresponding to better agreement between A and

B. Further, there is a correspondence between the two indices. If the weights are binary, $\forall e \in \mathcal{I}\ a_e \in \{0;1\}$ and $b_e \in \{0;1\}$, then the weighted Jaccard index $Jacc(A,B)$ reduces to the unweighted $Jacc(A^*,B^*)$ between $A^* = \{e \in \mathcal{I} : a_e = 1\}$ and $B^* = \{e \in \mathcal{I} : b_e = 1\}$.

Global similarity

Suppose $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ are binary, i.e. unweighted, and non-empty. The *global similarity* between them is defined as the Jaccard index between their edge sets:

$$\textit{global similarity}\ (G_1, G_2) = \frac{|E_1 \cap E_2|}{|E_1 \cup E_2|}. \tag{4.3}$$

It can be interpreted as the ratio between the number of edges of the intersection of G_1 and G_2 and the number of edges of their union. If the two networks are similar, then their intersection should be relatively large and their union should be relatively small, resulting in a global similarity close to one.

This measure does not take into account edge weights. It would be unsuitable for comparing dense, possibly complete co-expression networks, in which non-negative edge weights correspond to gene co-expression. Suppose G_1 and G_2 are weighted, and that for $e \in V^2$, $w_1(e)$ is the weight of e in G_1, and w_2 is the weight of e in G_2. We treat non-edges as equivalent to zero-weight edges, i.e. $w_i(e) = 0 \iff e \notin E_i$.

Intuitively, the two weighted networks G_1 and G_2 are similar if corresponding edges have similar weights, i.e. if for most $e \in V^2$, $w_1(e) \approx w_2(e)$. To measure similarity between the networks, we use the weighted Jaccard index over the edge weights:

$$\textit{global similarity}\ (G_1, G_2) = \frac{\sum_{e \in V^2} \min(w_1(e), w_2(e))}{\sum_{e \in V^2} \max(w_1(e), w_2(e))}. \tag{4.4}$$

The unweighted global similarity is equivalent to the weighted one, where edges are all assigned equal weight one. More generally, in the unweighted case, a possible edge $e \in V^2$ contributes to the numerator of the fraction, i.e. the graph intersection, if it is present in both networks, and to the the denominator of the fraction, i.e. the graph union, if it is present in at least one of the networks. In the weighted

case, the smaller weight contributes to the numerator or the "intersection"—in both networks, the edge appears with weight at least $\min(w_1(e), w_2(e))$—and the larger weight contributes to the denominator or the "union"—in at least one of the networks, a weight of $\max(w_1(e), w_2(e))$ is observed.

Local similarity

Rather than considering the complete edge sets, we can calculate a *local similarity* specific to the neighbourhood of every node in the network. Such a measure is of interest when heterogeneous noise is present. This can be the case in gene expression data, where noise is sometimes assumed to have gene-specific variance (e.g. Gao et al. 2016). If a gene v is strongly affected by experimental noise, we would expect that its neighbourhood in a gene co-expression network to be more strongly dependent on the choice of samples used for network construction. In this case, in a single iteration of COGENT, the neighbourhoods $\Gamma_{(1)}(v)$ of v in G_1 and $\Gamma_{(2)}(v)$ of v in G_2 may differ more than the neighbourhoods of other genes. In the unweighted case (Kao and Porter 2018), this is quantified by:

$$\textit{local similarity } (v; G_1, G_2) = \frac{|\Gamma_{(1)}(v) \cap \Gamma_{(2)}(v)|}{|\Gamma_{(1)}(v) \cup \Gamma_{(2)}(v)|}. \tag{4.5}$$

This is equivalent to the global similarity (Equation 4.3), where instead of the complete edge sets, only the neighbourhoods of v are considered. For the weighted case, the measure can be defined analogously to Equation 4.4.

$$\textit{local similarity } (v; G_1, G_2) = \frac{\sum_{u \in V, u \neq v} \min(w_1(uv), w_2(uv))}{\sum_{u \in V, u \neq v} \max(w_1(uv), w_2(uv))}. \tag{4.6}$$

Like global similarity, local similarity for a node is in $[0, 1]$, with higher values corresponding to better preserved networks. Local similarity can be used to identify particularly consistent—or inconsistent—parts of the network, but it can also be used to provide a global network summary by averaging over all nodes (Kao and Porter 2018). The average local similarity and the global similarity are related but not necessarily identical. Their relationship is similar to that between the average local clustering coefficient and the global clustering coefficient (see Chapter 1

Section 1.4), in that the former treats all nodes equally, and the latter is more strongly influenced by high-degree nodes.

Edge overlap and network density

Both global and local similarities are measured by scaling the edge overlap between two networks. However, edge overlap is affected by network density. Denser networks are more likely to share edges by chance than sparser ones. Consider, for example, two random graphs on 4 nodes, each of which has precisely one edge. Since there are $\binom{4}{2} = 6$ ways of placing the edge, the probability that the networks share an edge is 1/6, and most of the time the graphs would have a global similarity of zero. If, however, the two graphs instead had five randomly placed edges each, they would always share at least four edges, meaning their global similarity would be at least 2/3. However, in both cases the overlap is entirely random—so in a sense neither pair should exhibit better "consistency".

Since COGENT works by repeatedly applying a network construction function $\psi(\cdot)$ to random subsets of the data M of equal size, we would expect that the pairs of graphs $(G_1, G_2)_j$ obtained across different iterations $j \in \{1, 2, \dots\}$ to have similar density. Hence, global and local similarity can be meaningfully compared across iterations. If we consider an alternative network construction function $\xi(\cdot)$ which produces networks (H_1, H_2) of similar density to the ones produced by $\psi(\cdot)$, then we can also meaningfully compare $\psi(\cdot)$ to $\xi(\cdot)$. For example, if $\psi(\cdot)$ builds edges from the top 10% of highest Pearson correlations between gene expression profiles, and $\xi(\cdot)$ builds edges from the top 10% of highest Kendall correlations, then global similarity would be a good way of identifying which method is more consistent.

If, however, $\chi(\cdot)$ uses the top 20% of Pearson correlation coefficients instead, comparing $\psi(\cdot)$ and $\chi(\cdot)$ would be less straightforward. As $\chi(\cdot)$ produces denser networks, we would expect the global and local similarities at COGENT iterations for it to be higher. If $\chi(\cdot)$ outperformed $\psi(\cdot)$, it would not be clear whether this was due to the method being genuinely more consistent. Higher similarity scores for

$\chi(\cdot)$ compared to $\psi(\cdot)$ may just arise from the increase in network density. In order to account for this, we introduce a density adjusted edge set consistency metric.

4.3.2 Density adjusted edge set consistency

Density adjustment is required in order to compare between network construction methods $\psi(\cdot)$ and $\chi(\cdot)$ which result in networks of considerably different edge densities. This is particularly relevant when COGENT is used to determine a threshold or score cut-off in the construction of binary networks—e.g. whether to take the top 10% or top 20% of most highly co-expressed gene pairs to build a network from. The density adjustment implemented in COGENT relies on a comparison to randomly generated networks. First we explain the fully random density adjustment developed by Javier Pardo-Diaz and study its dependency on network density in fully randomised networks. We then define a semi-random adjustment, and show that both adjustments agree when applied to synthetic data. Both procedures are available within COGENT, but we note that the semi-random procedure is more computationally efficient and should generally be preferred. Density adjustment is designed specifically for unweighted networks. In the case of weighted networks, a density adjustment can be carried out by transforming network weights to quantiles, and calculating weighted global and local similarities on the transformed values.

Fully random density adjustment

Consider a single COGENT iteration, resulting in two binary networks $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$. Let d_1 and d_2 be the degree sequences of G_1 and G_2 respectively, so $d_1(v)$ is the degree of v in G_1, and $d_2(v)$ is the degree of the same node v in G_2.

Intuitively, we would like to know how much of the overlap between E_1 and E_2 is "genuine", and how much of it we might expect from any two networks of similar degree sequences. Density adjustment in COGENT corrects for the effect of random overlap by generating two networks, $G_1^* = (V, R_1)$ and $G_2^* = (V, R_2)$ from the configuration model (see Chaper 1 Section 1.4.5), with random permutations d_1^* and d_2^* of the degree sequences d_1 and d_2 respectively. We define a correction

term based on the pairwise comparisons between each co-expression network G_i and the randomly generated network G^*_{3-i} with the other degree sequence:

$$\alpha(G_1, G_2|G_1^*, G_2^*) = \frac{1}{2}(|E_1 \cap R_2| + |E_2 \cap R_1|). \tag{4.7}$$

Heuristically, the value of α corresponds to a level of intersection between G_1 and G_2 we may expect at random. Each of the terms in the sum is the overlap between one "real" co-expression network and a random network which resembles its counterpart. To define the *adjusted consistency* between G_1 and G_2 we subtract $\alpha(G_1, G_2|G_1^*, G_2^*)$ from the graph intersection and add it to the graph union of G_1 and G_2 in the unweighted global similarity (Equation 4.3), to give what we call the *fully random adjusted consistency*:

$$\alpha\text{-}\textit{adjusted consistency}\ (G_1, G_2|G_1^*, G_2^*) = \frac{|E_1 \cap E_2| - \alpha(G_1, G_2|G_1^*, G_2^*)}{|E_1 \cup E_2| + \alpha(G_1, G_2|G_1^*, G_2^*)}. \tag{4.8}$$

Unlike global and local similarity, the density adjusted consistency measure can be negative. The lowest possible adjusted consistency is obtained when G_1 and G_2 share no edges, but $G_1^* \equiv G_2$ and $G_2^* \equiv G_1$. In this case, the adjusted consistency is $-1/3$. Perfectly overlapping G_1 and G_2 will have adjusted consistency close to 1, with higher consistency for lower density networks. If network overlap is random, then $|E_1 \cap E_2| \approx \alpha(G_1, G_2|G_1^*, G_2^*)$ and the adjusted consistency would be close to 0.

We illustrate that the fully random consistency in Equation 4.8 does not increase with density like the global and local similarities with a simulation study. We generated pairs networks G_1 and G_2 from the $\mathcal{G}(N, p)$ model on $N = 100$ nodes and with edge probability $p \in \{0.01, 0.02, \ldots, 0.50\}$. For each value of p we generated 100 pairs (G_1, G_2) and calculated the global similarity and the fully random adjusted consistency of each one. This resulted in a total of 5000 pairs of networks. Since the networks are random, we observed adjusted consistencies close to zero at all values of p. Further, there was no relationship between the adjusted consistencies and p (Pearson correlation coefficient 0.016). In contrast, the global similarity increased with p (Pearson correlation coefficient 0.994). See Figure 4.2 for details.

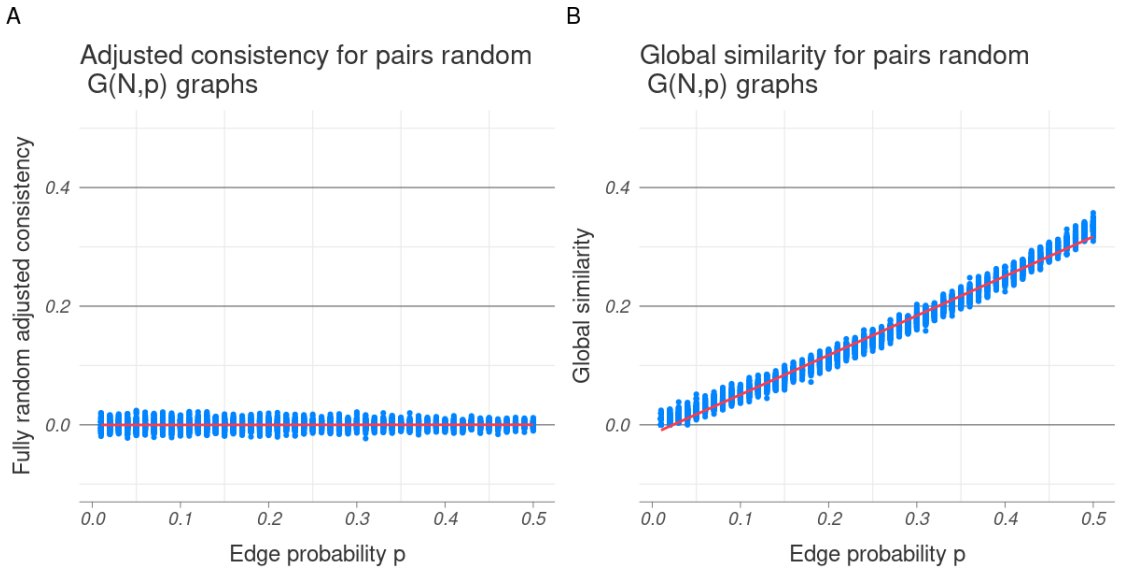

Figure 4.2: Fully random adjusted consistency of $\mathcal{G}(N,p)$ pairs. For different values of p, one hundred pairs of networks were generated from $\mathcal{G}(100,p)$ and the fully random adjusted consistency (A) and global similarity (B) were calculated for each pair. The dots (blue) correspond to the obtained values and the lines (red) were fitted using linear models. Consistency remained close to zero far all network densities, while global similarity increased with p.

Generating the pair of random networks G_1^* and G_2^* can be computationally expensive when the number of nodes in the network is large. This is due to the random graph generation algorithm: first, "stubs" are attached to every node in the network, and then pairs of stubs are randomly connected to build edges. However, every time the procedure creates a self-edge (two stubs of the same node are connected), or a multi edge (two stubs are connected between the pair of nodes which already share an edge), the process is restarted. In order to speed up the comparison, we introduce a semi-random density adjustment, based on approximate expectations of edge occurrence rather than on randomly generated networks.

Semi-random density adjustment

In Equation 4.7 we introduced a density adjustment term by randomising the degree sequences of $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ and creating two random networks $G_1^* = (V, R_1)$ and $G_1^* = (V, R_2)$ with the same degree distributions as G_1 and G_2 respectively. We then calculated the graph intersection between the fixed G_1 and the random G_2^*, as well as between G_2 and G_1^*. These intersections

are both also random graphs by construction. Based on the intersections, we calculated a correction term α.

Rather than generating the random graph intersections, we can calculate a correction term β analogous to α by approximating their expected size. Without loss of generality, consider the intersection between G_1 and G_2^*. First, note that if an edge $e = (u, v)$ is not present in G_1, i.e. $e \notin E_1$, then it clearly cannot be present in the intersection either. Therefore, with G_1 fixed,

$$\mathbb{P}(e \in E_1 \cap R_2) = \mathbb{1}\{e \in E_1\}\mathbb{P}(e \in R_2). \tag{4.9}$$

The graph G_2^* is generated from the configuration model, after discarding any networks with self-loops and multi-edges. However, for sparse, large networks, the density of self-loops and multi-edges in the configuration model is low (Newman 2018b). Therefore, we can approximate the probability of an edge $e = (u, v)$ existing in G_2^* with the probability of edge occurrence in the standard configuration model, which is proportional to $d_2^*(u)d_2^*(v)$. We normalise so that the expected number of edges in the network is $|R_2| = |E_2| = \frac{1}{2}\sum_u d_2^*(u)$ to get

$$\mathbb{P}(e = (u, v) \in R_2) \approx \frac{d(u)d(v)}{K}, \tag{4.10}$$

where the normalising constant K is

$$K = \frac{1}{|E_2|}\sum_{u>v} d(u)d(v). \tag{4.11}$$

Given a random degree permutation $\{d_2^*(u)\}_{u\in V}$, the probability in Equation 4.9 can be approximated by

$$\mathbb{P}(e = (u, v) \in E_1 \cap R_2) \approx \mathbb{1}\{e \in E_1\}\frac{|E_2|d_2^*(u)d_2^*(v)}{\sum_{u>v} d_2^*(u)d_2^*(v)} =: p_e. \tag{4.12}$$

This allows us to calculate an approximation to the expected adjacency matrix of the intersection between G_1 and G_2^*, given the degree permutation used for generating G_2^*. This adjacency matrix can be thought of as an uncertainty matrix like the ones discussed in Chapter 3, or, more simply, as a weighted network, where

edge weights p_{uv} correspond to probability of edge occurrence. The expected number of edges in the overlap is then approximately the sum of all p_{uv}:

$$O_{12} = \sum_e p_e \propto \sum_{e=(u,v)\in E_1} d_2^*(u) d_2^*(v). \tag{4.13}$$

We can approximate the overlap O_{21} between G_2 and G_1^* analogously, replacing the degree sequence d_2 with a random permutation d_1 of the degree sequence of G_1. Using both, we define a new correction term β:

$$\beta(G_1, G_2|d_1^*, d_2^*) = \frac{1}{2}(O_{12} + O_{21}). \tag{4.14}$$

The new correction term $\beta(G_1, G_2|d_1^*, d_2^*)$ approximates the expectation of $\alpha(G_1, G_2|G_1^*, G_2^*)$ given that the degree sequence permutations d_1^* and d_2^* were used to generate G_1^* and G_2^* respectively. We call this correction term "semi-random" because while it does not require the generation of the graphs G_1^* and G_2^*, it still requires the random permuting of the degree sequences d_1 and d_2. Finally, we use β to define a *semi-random adjusted consistency* measure, similar to the fully random one in Equation 4.8:

$$\textit{β-adjusted consistency } (G_1, G_2|d_1^*, d_2^*) = \frac{|E_1 \cap E_2| - \beta(G_1, G_2|d_1^*, d_2^*)}{|E_1 \cup E_2| + \beta(G_1, G_2|d_1^*, d_2^*)}. \tag{4.15}$$

In its implementation, the semi-random adjusted consistency relies entirely on matrix algebra and array permutations, rather than on network generation. This makes it more computationally efficient than the fully random one.

To evaluate its performance, semi-random adjusted consistency was measured for the set of $\mathcal{G}(N, p)$ pairs of networks discussed in the previous section. Like the fully random adjusted consistency, its values were close to zero and did not increase with network density. Further, they correlated well with the fully random values (Pearson correlation coefficient 0.814). Therefore, the approximations made are appropriate. See Figure 4.3 for details.

Overall, we have demonstrated that both the fully random and the semi-random density adjusted consistency measures can be used to measure the agreement between two binary networks, controlling for their densities. This makes them

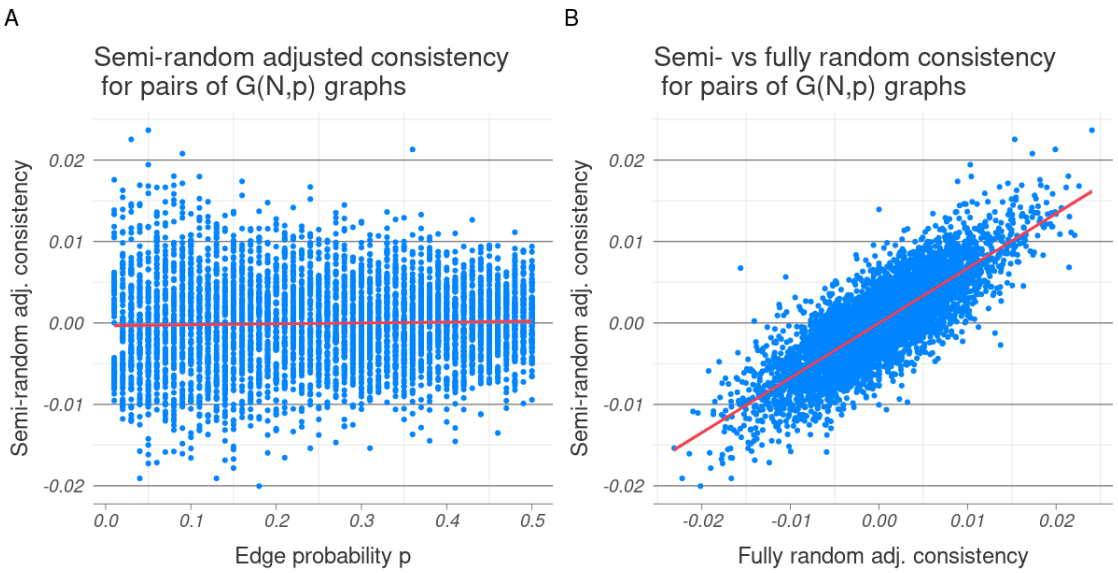

Figure 4.3: Semi-random adjusted consistency of $\mathcal{G}(N,p)$ pairs. (A) Like the fully random adjusted consistency (Figure 4.2A), the semi-random measure remained close to zero as the edge probability p increased. (B) The fully and semi-random measures were well-correlated. The dots (blue) correspond to the obtained values and the lines (red) were fitted using linear models.

suitable for comparing network construction algorithms resulting in binary networks of significantly different densities. Like global and local similarity, density adjusted consistency is a measure of edge overlap. Within COGENT, we introduce a further way of evaluating network consistency using node metric agreement.

4.3.3 Node metric consistency

Edge overlap is not the only way to think about network similarity. For example, consider two transport networks of Britain, where nodes correspond to villages, towns and cities, and edges are either roads or direct railway connections. We may argue that the railway network is in a sense functionally similar to the road network, since both identify the same large cities as major transport hubs. This similarity is between the node properties of the two networks, and is not based on edge overlap. It also depends on the node metric of choice—while the high-degree nodes might be the same across the two networks, the high betweenness nodes may be different.

Both degree and betweenness centrality as well as other node metrics are used in co-expression network analysis to identify key functionally relevant genes (see

e.g. van Dam et al. 2017). It therefore may be desirable to study the consistency of a network construction method with respect to these metrics. That is, we may ask how much we can trust node metric values obtained from the co-expression network, rather than how reliable the network itself is. Node metric consistency can be calculated in three different ways within COGENT.

Suppose again that a COGENT iteration results in a pair of networks $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$, and let $f : V(G) \to \mathbb{R}$ be a node metric of interest. A network construction method may be considered consistent *with respect to the node metric* f if $f(v|G_1) \approx f(v|G_2)$ for all $v \in V$. COGENT can measure the agreement between $f(v|G_2)$ and $f(v|G_2)$ in three different ways:

- through Euclidean distance,
- through a correlation coefficient, and
- through rank k-similarity (Trajanovski et al. 2013, Bozhilova et al. 2019).

The Euclidean distance measures the distance between the vectors $f(V(G_1))$ and $f(V(G_2))$ in $\mathbb{R}^V$ space. The obtained value is non-negative, with *lower* values corresponding to better network consistency. Optionally, a scaled Euclidean distance can be calculated, for which metric values are first scaled to the range $[0, 1]$ using

$$f^*(u) = \frac{f(u) - \min_{v \in V}\{f(v)\}}{\max_{v \in V}\{f(v)\} - \min_{v \in V}\{f(v)\}} \tag{4.16}$$

and then the Euclidean distance between $f(V(G_1))^*$ and $f(V(G_2))^*$ is itself scaled between zero and one using

$$d(f^*(V(G_1)), f^*(V(G_2))) = \frac{||f^*(V(G_1)) - f^*(V(G_1))||}{\sqrt{|V|}}. \tag{4.17}$$

This results in a measure between zero and one, where again *lower* values correspond to higher network consistency.

The correlation between the node metric vectors $f(V(G_1))$ and $f(V(G_2))$ can also be calculated. By default, Pearson correlation coefficients are calculated, and missing data is handled by pairwise deletions. That is, only nodes where a metric

value was obtained in both G_1 and G_2 are used. However, in our implementation options are directly passed to the R-base correlation function $cor(\cdot)$, which allows Pearson, Spearman and/or Kendall correlation coefficients to be calculated, and missing data to be handled in different ways.

In Chapter 2, we discussed how rank k-similarity can be used to identify robust metrics for scored protein interaction networks. It can also be used to evaluate network consistency within COGENT. Briefly, rank k-similarity measures how well the sets of highest ranking nodes with respect to the metric $f(\cdot)$ in G_1 and G_2 agree. For example, it can be used to say what proportion of the top 50 highest degree nodes in G_1 are also among the highest degree nodes in G_2. For more details refer to Chapter 2, Section 2.2.4. The default value of k in COGENT is 10% of the number of genes $|V|$.

Note that, since k-similarity is rank-based, it is also not dependent on the network density. More generally, if the node metric $f(\cdot)$ used or the consistency measure applied to it is not dependent on network density, node metric consistency can be used instead of density adjusted edge set consistency to choose between network construction methods of different densities.

Together, the different edge set and node metric consistency measures provide a profile of the network similarity between G_1 and G_2 at every COGENT iteration. By aggregating across multiple iterations, COGENT outputs a consistency profile of a network construction method $\psi(\cdot)$ when applied to a gene expression data set M. In the next section we illustrate how COGENT can be used to inform network construction in different cases. We focus on unweighted networks; however, applications with weighted networks are analogous.

4.4 Applications

The network comparison measures described above can be used to evaluate the consistency of a network construction method $\psi(\cdot)$ as applied to a gene expression data set M. Comparing the consistency of different methods $\psi(\cdot)$ and $\xi(\cdot)$ allows the user to prioritise a construction method for further analysis. This prioritisation can

happen with respect to overall network consistency, or with respect to a particular application. For example, if genes of high betweenness centrality are of interest, network consistency can be measured regarding betweenness centrality. Our method allows the user to assess network quality without the use of any additional data, such as protein interaction or functional annotation data.

In this section we illustrate how COGENT can be used in two different scenarios—to choose between measures of gene co-expression for network construction and to set a co-expression score cut-off. In both applications we focus on a gene expression dataset obtained from the Expression Atlas (Papatheodorou et al. 2017).

4.4.1 Gene expression data

RNA-seq data of *Saccharomyces cerevisiae* (yeast) expressing pathways designed to increase ATP or GTP consumption was obtained from the Expression Atlas (accession number E-MTAB-5174). The original experiment as deposited in ArrayExpress (Athar et al. 2018) contains 156 samples, sequenced using Illumina HiSeq 2500. The curated data obtained from the Expression Atlas was reduced to expression profiles for 3103 genes across 26 samples. We further removed any genes where expression data was missing for over 25% of all samples, resulting in a dataset of 1920 gene expression profiles. Finally, we took the 500 genes with highest variation in gene expression as measured by the mean absolute deviation. This resulted in a final dataset M of 500 gene expression profiles across 26 samples. Expression values for 20 randomly selected genes are shown in Figure 4.4.

4.4.2 Choosing between measures of co-expression

We consider two simple ways of building gene co-expression networks from the data described above. One is by calculating Pearson correlation coefficients between expression profiles and building edges between the gene pairs with highest co-expression. This is sometimes known as hard thresholding (Langfelder and Horvath 2008). In the first instance, we take the top 5% of the co-expression values between pairs of genes, which results in a gene co-expression network on 500 nodes and 6000

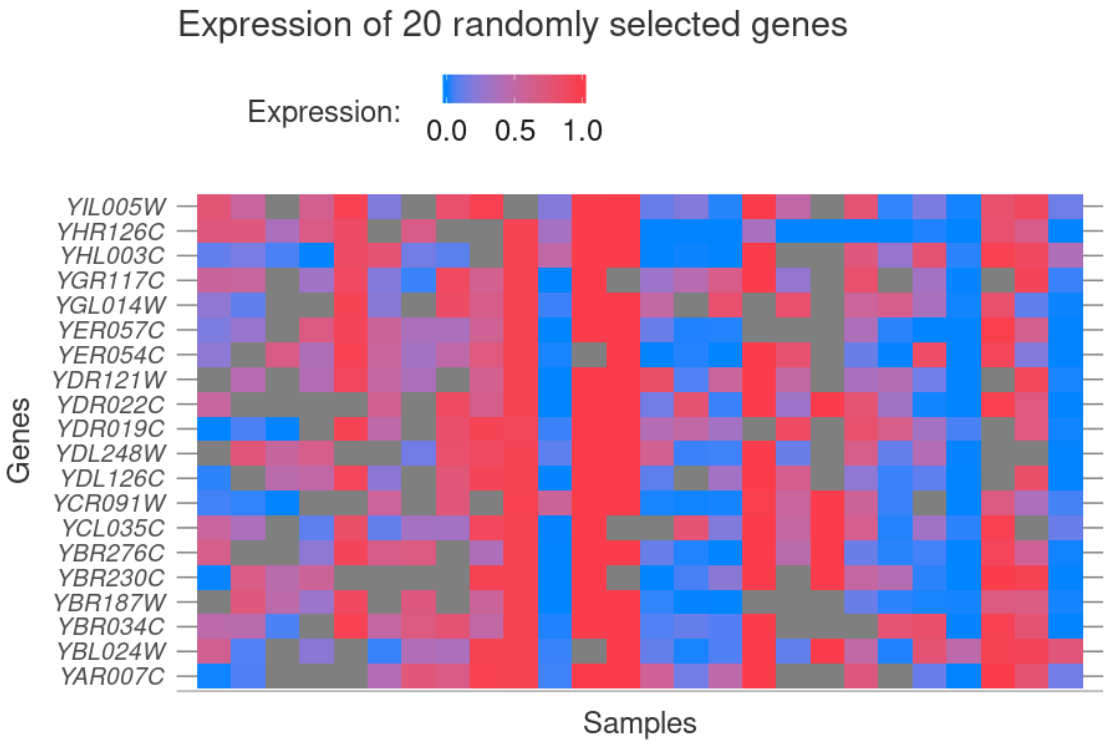

Figure 4.4: Sample gene expression data for yeast. Rows correspond to genes and columns correspond to samples. Expression values have are scaled between zero and one, with low expression shown in blue and high expression shown in red. Missing values are shown in grey. Rows with similar colour patterns indicate high gene co-expression.

edges. There are $\binom{500}{2} = 124750$ gene pairs, 5% of which is 6237.5 expected edges in such a network. The lower number of actual edges is due to co-expression ties and numerical approximations. A total of 42 nodes in the network are isolated.

The second construction method uses Kendall correlation coefficients instead of Pearson correlation coefficients. Again, we take the top 5% of obtained values to correspond to edges. This results in a network on 500 nodes and 6190 edges. Of the 40 isolated edges in the Kendall network, 25 are also isolated in the Pearson network.

We can use COGENT both to examine the consistency of a network construction method and to compare the Pearson and Kendall networks. The global similarity between the two networks is 0.403, corresponding to an overlap of 3502 edges, or nearly 60% of the edges in each network. Edge differences are not spread uniformly across the networks—higher degree nodes generally have higher local similarity and nodes of extremely low degree in either network account for peaks

at zero and one (see Figure 4.5).

The Pearson and Kendall networks, while sharing a significant number of edges, are different. We use COGENT to determine which of the two network correlation calculations is more consistent with respect to data resampling and which network should therefore be prioritised for further analysis. Since the networks are of the same density 0.05 by construction, density adjustment is not required. For illustration purposes, we evaluate edge set consistency through global similarity and node metric consistency for the degree through rank k-similarity.

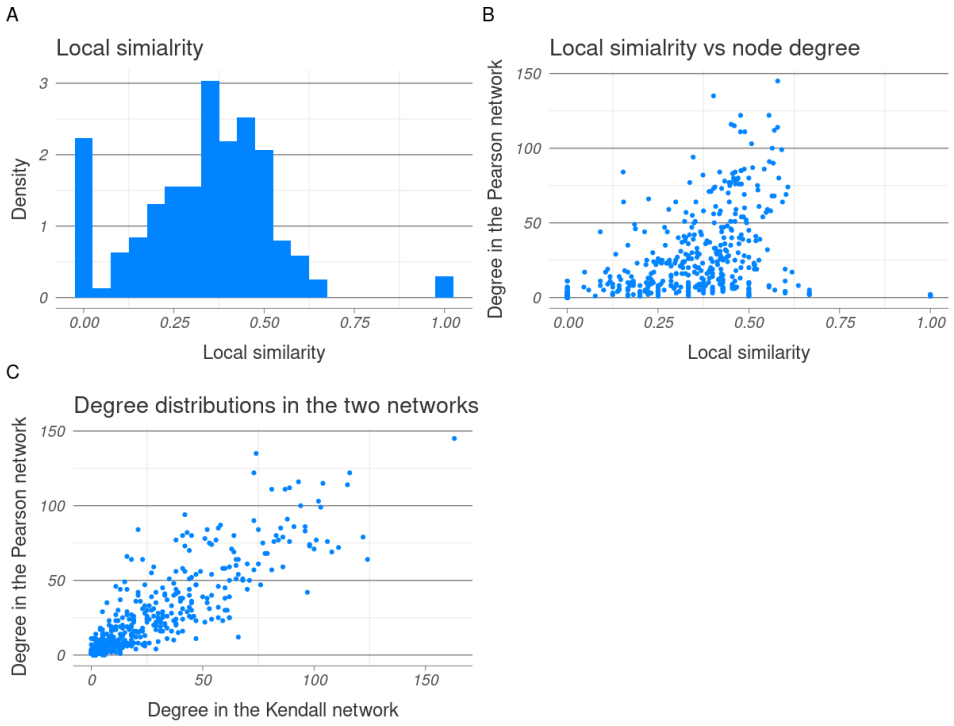

Figure 4.5: Similarities between the Pearson and Kendall co-expression networks. (A) Local similarity for each node (mean 0.33, st. dev. 0.19). Most values are symmetric around the mean, with additional peaks around 0 and 1. Binwidth is set to 0.05. (B) Local similarity plotted against node degree in the Pearson network. The peaks at zero and one correspond to nodes of low degree. However, higher degree nodes tend to exhibit higher similarity. (C) The two degree distributions are positively correlated (Pearson correlation coefficient 0.86).

Since the dataset contains only 26 samples, we run COGENT for both methods with a sample overlap of 50% of the data at each iteration. Thus, when the data M is resampled and split into subsets M_1 and M_2, 13 samples are randomly selected and included in both sets, and the remaining samples are randomly allocated to

only one of the sets. One hundred COGENT iterations are used to evaluate the stability of each method. The computational complexity of the COGENT pipeline is linear in the complexity of the network construction method (and the node metric calculation). Thus, medium-to-large Pearson correlation networks can be analysed without any parallelisation on a standard desktop computer. COGENT has an additional inbuilt parallel mode, which is employed for the Kendall networks, where computing correlation matrices is slower.

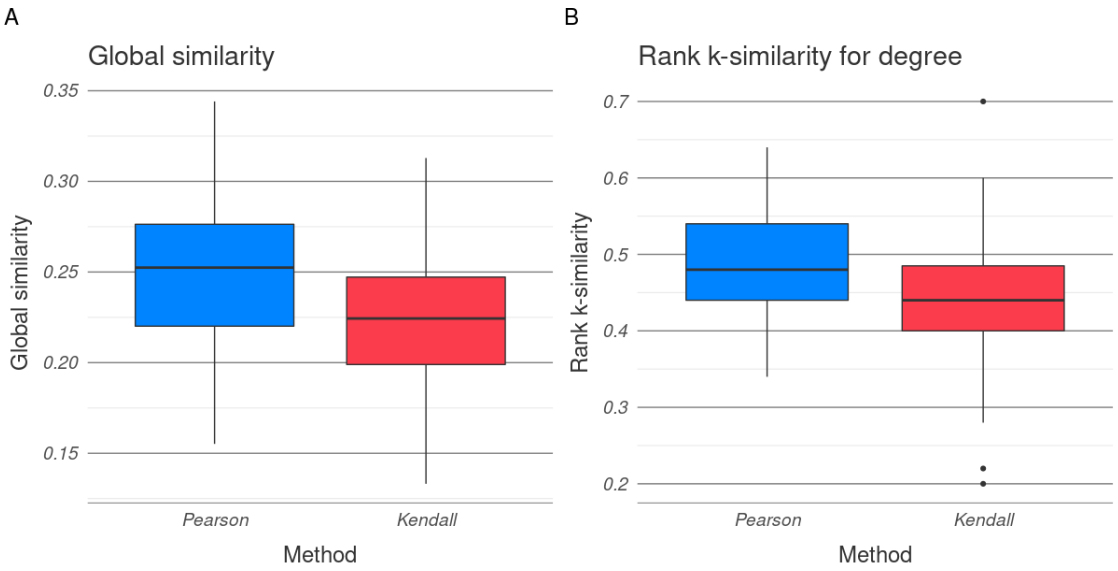

Figure 4.6: COGENT analysis for Pearson and Kendall co-expression networks. (A) Global similarity is visibly higher for the Pearson network construction method (mean 0.25, st. dev. 0.04) than for the Kendall method (mean 0.22, st. dev. 0.04). (B) While Pearson (mean 0.48, st. dev. 0.07) still outperforms Kendall (mean 0.44, st. dev. 0.08) in rank k-similarity for the degree, the difference is slightly less pronounced. This may be expected, since the two networks are in better agreement at higher degree nodes (Figures 4.5B and 4.5C). Note the difference in scales on the y-axis.

Edge set consistency as measured by global similarity is higher for the Pearson than the Kendall method (Figure 4.6). Pearson also outperforms Kendall in rank k-similarity for the degree with the default settings. While the difference is less pronounced for k-similarity than it is for global similarity, both are statistically significant. A two-sided Wilcoxon rank sum test for global similarity has p-value $p \approx 2.6 \times 10^{-6}$, and the same test for rank k-similarity has p-value $p \approx 6.3 \times 10^{-5}$.

The comparison between network consistency for both Pearson and Kendall correlation networks therefore indicates that the Pearson network should be preferred.

However, this analysis was performed only on networks thresholded to ensure a 5% network density. In the next section, we use COGENT to study threshold choice.

4.4.3 Imposing a co-expression score cut-off

In the previous section, we used COGENT to compare the consistency of two methods which both produced networks of 5% edge density. However, the value of 5% was arbitrary. We can also use COGENT in order to determine whether a different value may result in more consistent networks. We illustrate this by exploring different thresholds for a Pearson correlation network constructed using the same gene expression data.

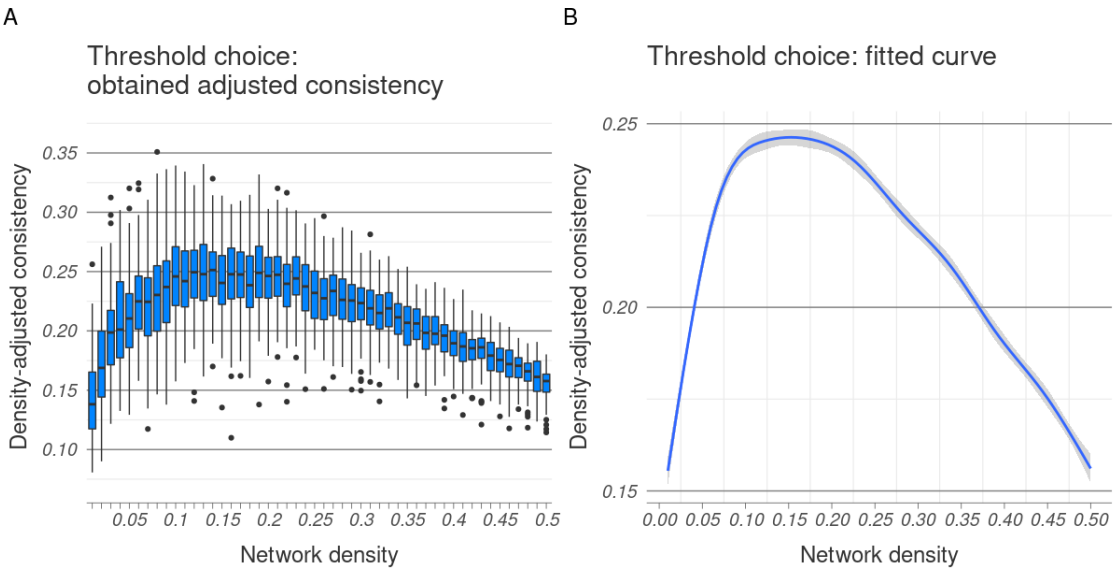

Figure 4.7: COGENT analysis for Pearson thresholds. (A) Obtained density adjusted edge set consistencies at different network densities. (B) Curve fitted to the data using a generalised additive model. The curve reaches its maximum around density 0.15, suggesting taking the top 15% highest correlation values may be optimal. Note the difference in scales on the y-axis.

As discussed in Figure 4.2, global similarity cannot be used to distinguish between methods with different network densities, because denser networks are more likely to have higher edge overlap by chance. In order to correct this, we use the semi-random density-adjusted edge set consistency measure in COGENT. We consider 50 density thresholds between 0.01 and 0.50, i.e. from taking only the top 1% of highest pairwise gene expression correlations down to taking the top half of these. At each

threshold, we run the COGENT pipeline for 100 iterations, and split the data the same sample overlap as before (13 out of 26 shared samples in each sample subset).

The density adjusted edge set consistency for the Pearson network at the 5% density threshold has mean 0.21 and st. dev. 0.04. As expected due to the semi-random correction term β, this is slightly lower than the global similarity calculated using the same parameters (mean 0.25, st. dev. 0.04). As the network density increases, the adjusted similarity also increases until it plateaus for densities around 15%. For higher densities, the adjusted consistency then decreases (Figure 4.7). Fixing the network density around 15% may therefore be most preferable.

The highest observed adjusted consistency is at network density 13% (mean 0.25 and std. dev. 0.03). This is a significant improvement over the value at 5% density (Wilcoxon test p-value $p \approx 2.3 \times 10^{-11}$). However, differences in the plateau region are not necessarily statistically significant—e.g. values obtained at 13% density are not significantly different from values obtained at 15% density (p-value $p \approx 0.11$). Therefore, while 13% density results in networks of the highest consistency, COGENT identified a range of optimal thresholds.

We have illustrated the two main expected uses of COGENT with respect to gene co-expression data: to choose between different measures of co-expression, and to inform threshold choice for binary networks. Both of these problems arise when networks are constructed from other types of occurrence data.

4.5 Discussion

Gene expression—the level of a particular genetic product produced in the cell—changes under different experimental conditions. This change can be associated with genetic function. For example, heat shock proteins (HSPs) are produced at higher rates by cells exposed to heat or other stress. Heat can cause proteins to unfold, and some HSPs act as chaperones, aiding protein folding and preventing partially unfolded proteins from aggregating (Welch 1992). Analogously to how gene expression helps us understand gene or protein function in the context of cellular response, gene co-expression can be used to infer genetic interactions. Subunits of

protein complexes will be co-expressed, since they need to be present in the cell at the same time in order to form a functioning complex. Similarly, genes within the same functional pathway need to be co-expressed for the pathway to be activated. Identifying gene co-expression patterns can be a valuable tool for inferring genetic function, e.g. through guilt-by-association approaches (C. J. Wolfe et al. 2005).

Co-expression is calculated as a similarity score between gene expression profiles, e.g. using a correlation coefficient or mutual information. Once calculated, it can be treated as a continuous measure of how strongly two genes are related. However, it is often discretised by imposing a score cut-off. Thus, gene co-expression networks can be constructed. In such networks, nodes represent genes, and edges connect highly co-expressed pairs of genes. There exist many different ways of normalising the data and calculating co-expression meaning that there are many different ways to construct a network from the same expression data. Often, a single approach is taken and it is not clear whether other choices would have resulted in more suitable networks.

The validation of co-expression networks is difficult. While physical protein binding can be experimentally verified, gene co-expression is an abstract measure of relatedness between genes, without a clearly defined ground truth. Network validation often relies on external functional annotation (Gillis and Pavlidis 2011)—intuitively, better networks are more likely to connect genes we already know are functionally related. However, this validation technique is subject to biases in the annotation data and cannot be applied to cases where such data is absent. In this chapter, we have introduced an alternative tool for network validation called COGENT.

COGENT works on the assumption that two factors contribute to computed gene co-expression—a genuine relationship in expression patterns, which is observed across samples, and experimental noise, which is sample-specific. A good network should prioritise the former over the latter. Therefore, if we were to create two networks from two subsets of all available expression samples using a "good" method, the two networks should be similar. Conversely, if the overlap between them is low,

then the network construction method may be picking up noise, rather than genuine co-expression patterns. We therefore introduce a concept of *network consistency*, capturing the extent to which a network construction method results in similar networks under resampling. We evaluate network consistency in two groups of ways—with respect to edge overlap, and with respect to node metric agreement. COGENT provides several different ways of calculating each one.

COGENT can be used for two types of tasks. The first one is to select between different measures of co-expression. We illustrate this by analysing co-expression networks calculated using Pearson and Kendall correlation coefficients. The second is in selecting a score cut-off for creating binary co-expression networks, which we apply to a Pearson correlation network.

While COGENT was designed for use with gene expression data, it could be applied to any other type of data where co-occurrence is indicative of a functional relationship. Examples of other possible use cases include microbiome data (Faust and Raes 2012) and synthetic lethality data (Barido-Sottani et al. 2019).

COGENT is an annotation-free tool for the validation of gene co-expression and other co-occurrence networks. It is implemented as an R package, and employs generic R classes and tools, making it transferable and easy to use.

"I'll stay till the wind changes," she said shortly, and she blew out her candle and got into bed.

— P.L. Travers, *Mary Poppins*

5 Conclusions and future directions

Contents

Protein–protein interaction (PPI) data is rich, heterogeneous, and noisy. One way of representing such data is through protein interaction networks (PINs). In Chapter 1 we discussed how these networks can be built and analysed, and what kinds of research questions can be tackled through such analysis. While PINs are a valuable tool in bioinformatics research, they often simplify PPI data and in particular do not explicitly address the uncertainty associated with it. In this thesis we have explored some of the ways in which data uncertainty can impact network construction and network analysis, and proposed novel approaches for studying the effects of uncertainty in different applications.

5.1 Measuring rank robustness in scored protein interaction networks

In order to quantify the reliability of PPI data, a number of databases provide a confidence score associated with each recorded interaction. Researchers can then choose a threshold, or a minimum score cut-off, in order to build networks from the data. While guidelines exist, these thresholds are often chosen in an *ad hoc* fashion. By varying the threshold, different networks can be built to represent the same biological state of interest, and it is not clear where and how these networks will differ. This has consequences with respect to the reliability and interpretability of downstream network analysis. If two networks obtained at different thresholds result in different conclusions about the underlying cellular architecture, it is not necessarily clear which of the two conclusions to trust.

Identifying key actors in a network is a common aim of PIN analysis. One way of doing so is to first calculate a node metric, such as degree or betweenness, for all the nodes in the network, and to then identify the highest ranking nodes as the key actors. In Chapter 2 we studied the effects of threshold choice on node metrics. We argued that if a node metric is to produce reliable and biologically-relevant results, it should identify the same, or at least similar, highest ranking nodes at different thresholds. We called this property *rank robustness*, and introduced three measures in order to quantify it—rank continuity, rank identifiability, and rank instability.

We studied a set of twenty-five node metrics by applying our methodology to four scored PINs, spanning three different organisms and two different databases. We further considered two synthetic networks. Our experiments showed that the same metrics appear to be consistently robust across PINs, regardless of organism or database. In contrast, our synthetic networks produced notably different robustness profiles. Our results point to a similarity between PINs, and we believe that once a metric has been identified as robust in one scored PIN, it can be applied with confidence to others. However, our synthetic network analysis shows that rank robustness results do not immediately translate to other network data. Overall, our research clearly illustrates the need to take data quality into account when

choosing how to analyse PINs. It also shows that conclusions drawn from synthetic networks cannot be assumed to hold for biological data.

Our methodology for evaluating rank robustness can be applied to any case where scored PPI data is available. We illustrated it with data from model organisms and a wide range of node metrics. However, in order to ensure its applicability and relevance with respect to biomedical research, it may be informative to carry out similar analysis on specific data and metrics used in published articles. We hope that in the future our methodology may be used to guide network analysis in applications.

Identifying key proteins is only one goal of PIN analysis. Many other exists, such as identifying functionally related modules, predicting protein function, or performing network comparison. A range of techniques can be used to address each of these problems, and we expect the effect of threshold choice will be different for each one. We believe robustness with respect to thresholding is important for a range of applications. While we have proposed a methodology for studying the robustness of node metrics, the same remains to be done for other types of network summaries and analysis techniques.

5.2 Generative models based on uncertain protein interaction networks

Threshold variation is one way of studying the effects of uncertainty on PINs. A different approach would be to incorporate uncertainty explicitly into the network model, and treat network inference as a stochastic task rather than a deterministic one. Such an approach is attractive, since it addresses the issue of noise directly, instead of focussing on the effects of data pre-processing. Uncertain networks are one way of incorporating scored relational data into a stochastic network framework (Ahnert et al. 2007; Martin et al. 2016). In Chapter 3 we investigated whether they may be suitable for representing and studying PPI data.

An uncertain network is a scored network, which is interpreted as a noisy observation of an underlying true network of interest. In the context of PINs, an uncertain network may be built from scored detected or predicted interactions, and

the underlying true network would be the set of genuine, biologically relevant PPIs. The aim of uncertain network analysis is to make inference about the true state based on the available scored data. The score associated with each edge in an uncertain network corresponds to the likelihood that the edge is also present in the true network. These scores only represent marginal likelihoods and do not take into account any dependence between edges. In this chapter we employed different generative network models to study the effects of edge dependence on true network inference.

We created an uncertain network for yeast based on Y2H and gene co-expression data. We then treated the unobserved "true" yeast PIN as a realisation of a random graph model with marginal edge probabilities given by the uncertain network. We identified a family of such models, and chose three models, corresponding to three different types of edge dependence, for further study. We argued that if sampling from different models produced similar networks, then we could infer properties of the true yeast PIN based on these random samples. However, we found the different models resulted in networks of significantly different structure.

Coupled with results obtained from synthetic data, our analysis showed that both the uncertainty scores and the generative model play an important role in determining network structure. In particular, we showed that the commonly made simplifying assumption that edges behave independently of each other fails to recover properties of the underlying true network in a synthetic example. Therefore, a good model for the underlying true network is required if uncertain networks are to be useful in the context of PPI data.

Our methodology treats the unobserved true state as random, and the observed data as fixed. More recently, a related Bayesian approach to network inference has been proposed (Newman 2018a; Peixoto 2018). It relies on prior assumptions about the true network structure, as well as a model of how measurements (e.g. gene co-expression) are made on it. By combining the two, a posterior distribution for the true network can be derived, and a maximum *a posteriori* estimate can be made. The posterior distribution will depend on both the network prior and on the observed data, in a similar way to how our network samples depended on

both the generative model and on the uncertain network. The work we carried out could be rephrased in such a Bayesian framework, but we anticipate that similar issues will arise. Despite the wealth of PPI data, we expect there is still not enough data to outweigh the effect of the network prior on the posterior. An important direction for further study is therefore identifying better models of "true" PINs. We further expect better data—especially consistent reporting of negative outcomes in screening assays, as well as detailed information on repeat screens—will also improve the performance and applicability of such models.

5.3 COGENT: evaluating the consistency of gene co-expression networks

In Chapter 3 we treated continuous measurements such as gene co-expression as a noisy observation made on a fixed underlying "true" network. However, such an interpretation is not required for the construction and analysis of gene co-expression networks. These networks aim to capture the functional relationships between genes based on the co-occurrence of their products in the cell. If two proteins tend to appear in the cell at the same time, they are more likely to be functionally related. However, a notion of "true" co-expression does not exist in the same way that a notion of true interaction does. Co-expression can be measured in different ways, resulting in different network construction methods. The validation of these networks often relies on external data, such as functional annotation data, which is not always available. In Chapter 4 we described a software package, COGENT (COnsistency of Gene Expression NeTworks), which can be used to validate and prioritise gene co-expression networks in the absence of external data.

We argued that, however calculated, gene co-expression is a combination of two factors: genuine co-occurrence of genetic products in the cell, and random fluctuations, i.e. noise, in the gene expression data. A good network construction algorithm should prioritise the former, which captures gene (or equivalently protein) function, over the latter. We assumed genuine co-expression would be more consistently observed across samples than noise is. Therefore, we proposed that a

suitable network construction algorithm should produce similar networks when only a subset of all available gene expression samples are used. We called this property *consistency*, and described different ways of measuring it, both for unweighted and for weighted networks. In particular, we developed a way of comparing the consistency of networks of significantly different density. Our density-adjusted consistency measure can be used to inform threshold choice for the construction of simple, unweighted networks from continuous co-expression values.

The complete COGENT pipeline involves randomly splitting the expression data in two sets, constructing a network from each set, and measuring the consistency between the two networks. Given an input network construction method and expression data, this results in a consistency profile. When two or more competing methods are analysed, the one exhibiting better consistency should be prioritised for further study. Thus, COGENT can be used to choose between different ways of measuring co-expression, as well as between different score cut-offs.

At every COGENT iteration, we measure consistency as a function of edge overlap (*"Do the two networks have the same edges?"*) or as node metric similarity (*"Do the nodes in the two networks have the same properties?"*). Gene co-expression networks are often employed to identify modules of functionally related genes. It may therefore be beneficial to introduce a new set of consistency measures, which capture how well modules are preserved (*"Do the two networks group the same nodes together?"*). Other measures, such as different ways of measuring density-adjusted consistency, could be implemented.

While further additions to our methodology are possible, at this stage the most immediate direction of future work would be extensive testing. Work carried out by Javier Pardo-Diaz already suggests that networks prioritised by COGENT recover more known PPIs than networks with lower consistency profiles (results not published). In addition to comparisons with PPI data, validation can also be carried out by checking whether higher consistency networks are also more likely to identify gene modules of known shared function.

5.4 Closing remarks

Uncertainty and experimental noise play an important, albeit rarely discussed, role in the study of biological networks. A single, unifying theory of how uncertainty affects PINs appears to be no easier to develop than a single, universally applicable PPI detection screen.

As seen in applications, PIN analysis is the product of a number of *ad hoc* decisions—what raw data to use, how to curate it, how to represent it as a network, and what tools to use to analyse this network. There is no single widely-accepted policy on how to make these decisions, and we do not expect such a policy to emerge in the near future. However, we believe that at every step in the process, decisions can and should be carried out with an awareness of the properties and in particular the limitations of the data.

In this thesis we have explored some ways in which this can be done. We have proposed novel approaches which can aid researchers in the choice of network analysis techniques (Chapter 2), as well as in the choice of network construction methodology (Chapter 4). We have also highlighted some issues, which can arise when simplifying assumptions are made in the absence of sufficient data (Chapter 3). However, it is beyond the scope of this work to explore all the ways in which uncertainty enters PIN analysis, and how these can be addressed. We hope that in the future we will see more network analysis performed in an uncertainty-aware way, and that more novel methodology for doing so will be developed—both in the context of PPI data and for other applications.

Appendices

A

Rank robustness in scored PINs: additional figures and tables

Contents

A.1 Results across all four networks

Metric	Continuity	Identifiability	Instability
Degree	**0.96**	0.90	0.01
Local clustering	0.21	0.28	0.07
Redundancy	0.92	**0.97**	0.01
PageRank	0.94	0.89	0.01
Closeness	0.86	0.83	0.01
Harmonic centrality	0.93	0.84	0.01
Betweenness	0.64	0.77	0.01
$\boldsymbol{e_{one}(v)}$	**0.96**	**0.96**	**<0.005**
$n_{two}(v)$	0.84	0.79	0.02
$n_{diff}(v)$	0.71	0.74	0.02
$n_{sqdiff}(v)$	**0.96**	0.90	0.01
$n_{ratio}(v)$	0.17	0.12	0.23
LOUD Average local clustering	0.31	0.45	0.07
LOUD Global clustering	0.94	0.94	0.01
LOUD Average redundancy	**0.96**	**0.95**	**<0.005**
LOUD Average closeness	0.32	0.66	0.03
LOUD Average path length	0.32	0.28	0.09
LOUD Number of connected pairs	0.05	0.34	0.35
LOUD Average betweenness	0.31	0.21	0.11
LOUD Natural connectivity	**0.96**	**0.98**	**<0.005**
LOUD Average $\boldsymbol{e_{one}(v)}$	**0.97**	**0.96**	**<0.005**
LOUD Average $n_{two}(v)$	0.87	0.82	0.01
LOUD Average $n_{diff}(v)$	0.79	0.78	0.01
LOUD Average $n_{sqdiff}(v)$	**0.96**	0.91	0.01
LOUD Average $n_{ratio}(v)$	0.50	0.44	0.05

Table A.1: Continuity, identifiability and instability measures for all metrics, averaged across the four PINs. Metrics for which average continuity and identifiability were above 0.95 and average instability was below 0.005 are in bold.

A.2 Rank continuity: additional results

Metric	PVX	ECOLI	YEAST	HPRED	SYN-GNP	SYN-PVX
Degree	**0.91**	**0.96**	**0.99**	**0.97**	0.10	0.59
Local clustering	0.14	∼0	0.25	0.43	0.07	0.17
Redundancy	0.83	**0.94**	**0.95**	**0.98**	0.08	0.11
PageRank	0.87	**0.95**	**0.99**	**0.95**	0.31	0.67
Closeness	0.70	0.85	**0.96**	**0.93**	0.12	0.49
Harmonic centrality	0.84	**0.93**	**0.98**	**0.97**	0.11	0.57
Betweenness	0.54	0.55	0.69	0.77	0.09	0.54
$e_{one}(v)$	**0.92**	**0.96**	**0.99**	**0.99**	0.10	0.52
$n_{two}(v)$	0.64	**0.90**	**0.95**	0.85	0.16	0.56
$n_{diff}(v)$	0.38	0.89	**0.93**	0.63	0.12	0.54
$n_{sqdiff}(v)$	**0.91**	**0.96**	**0.99**	**0.97**	0.31	0.69
$n_{ratio}(v)$	∼0	0.02	0.04	0.62	0.04	∼0
LOUD Average local clustering	0.16	0.05	0.34	0.69	0.09	0.22
LOUD Global clustering	0.83	0.95	**0.98**	**0.99**	0.13	0.12
LOUD Average redundancy	**0.92**	**0.96**	**0.99**	**0.99**	0.07	0.15
LOUD Average closeness	0.15	0.14	0.32	0.66	0.10	0.36
LOUD Average path length	0.22	0.02	0.43	0.63	0.10	0.34
LOUD Number of connected pairs	0.05	0.01	0.04	0.08	0.06	0.08
LOUD Average betweenness	0.18	0.02	0.40	0.62	0.09	0.26
LOUD Natural connectivity	**0.92**	**0.96**	**0.99**	**0.99**	0.25	0.52
LOUD Average $e_{one}(v)$	**0.92**	**0.97**	**0.99**	**0.98**	0.12	0.54
LOUD Average $n_{two}(v)$	0.74	**0.94**	**0.97**	0.82	0.26	0.63
LOUD Average $n_{diff}(v)$	0.49	**0.91**	**0.98**	0.78	0.18	0.62
LOUD Average $n_{sqdiff}(v)$	**0.91**	**0.96**	**0.99**	**0.98**	0.26	0.65
LOUD Average $n_{ratio}(v)$	0.39	0.34	0.65	0.63	0.12	0.25

Table A.2: Rank continuity for the 25 metrics across each of the six scored networks. Values above 0.90 have been highlighted. The PINs show good general agreement, and the SYN-GNP network consistency exhibits lower rank continuity.

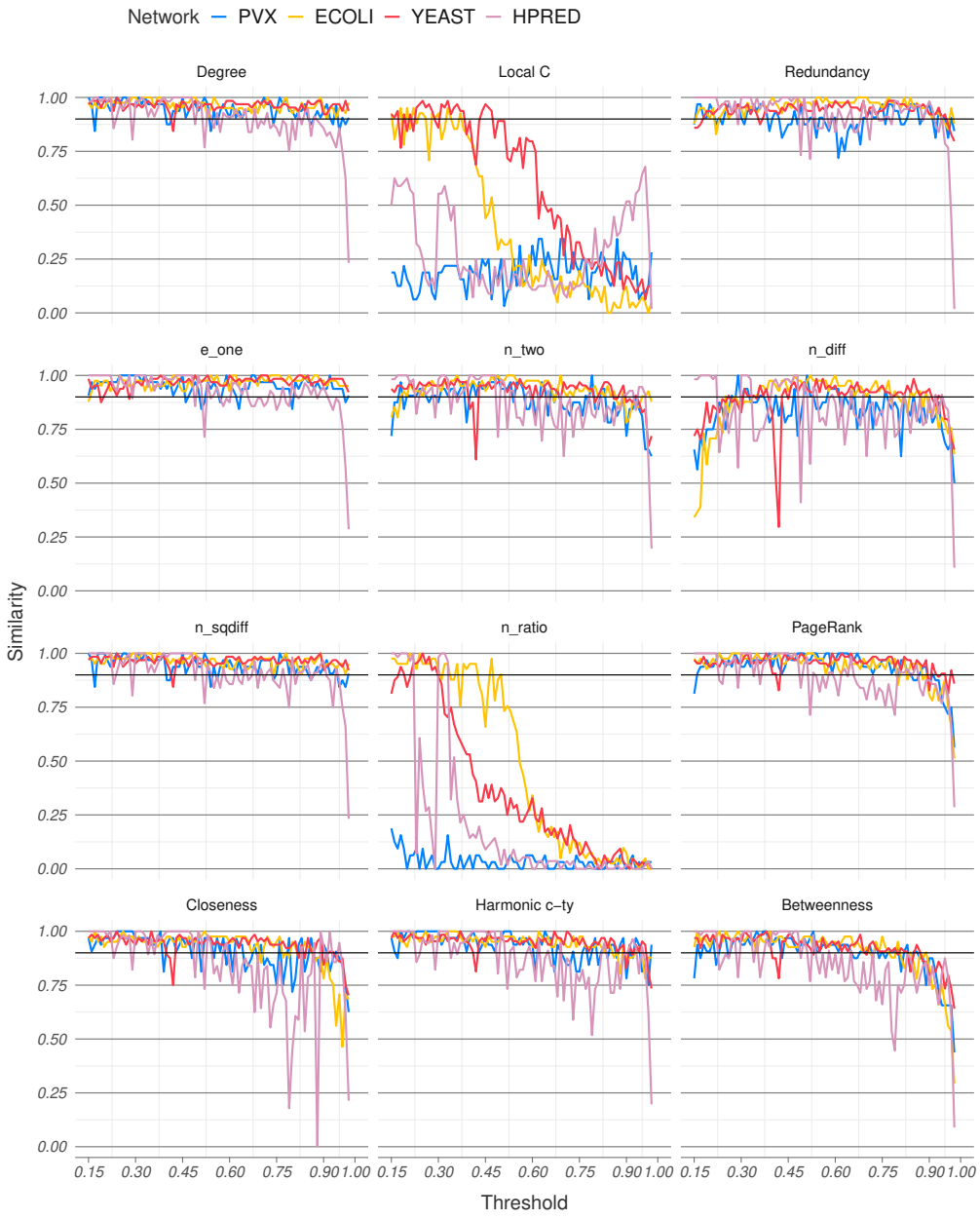

Figure A.1: Standard metric rank similarity between consecutive thresholds for the four PINs. PINs across different species and databases showed generally good agreement. Local clustering coefficient (Local C) and the ratio between step-1 and step-2 ego networks (n_ratio) perform noticeably worse than other metrics.

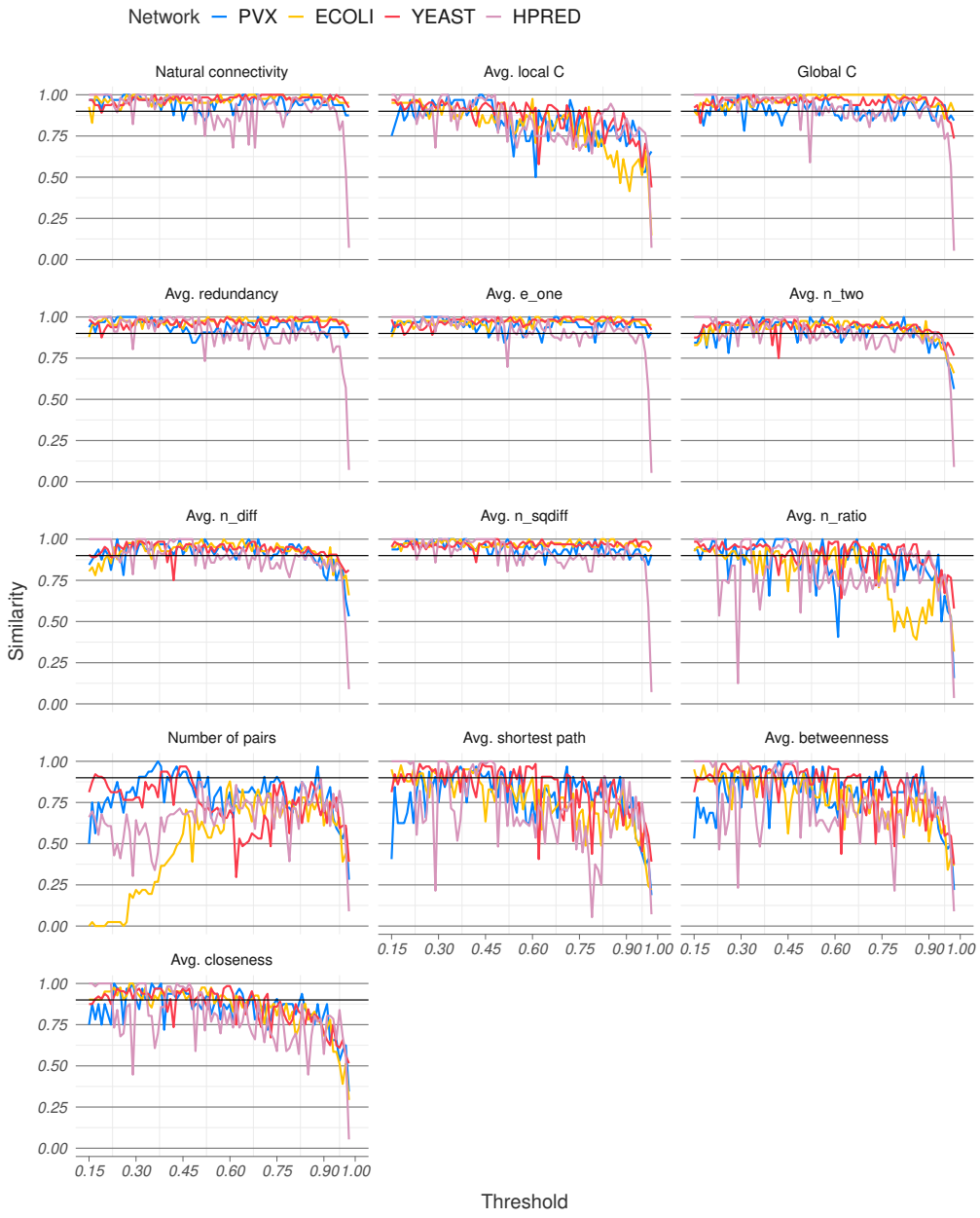

Figure A.2: LOUD metric rank similarity between consecutive thresholds for the four PINs. Protein interaction networks across different species and databases showed generally good agreement. Average local clustering coefficient, average shortest path, average betweenness, and average closeness all exhibit a similar pattern of decreasing k-similarity as the threshold increases.

Figure A.3: Standard metric rank similarity between consecutive thresholds for the synthetic networks. The Bernoulli synthetic network, SYN-GNP, exhibits consistently lower similarity across all node metrics aside from the ratio between the step-1 and step-2 ego networks (n_ratio).

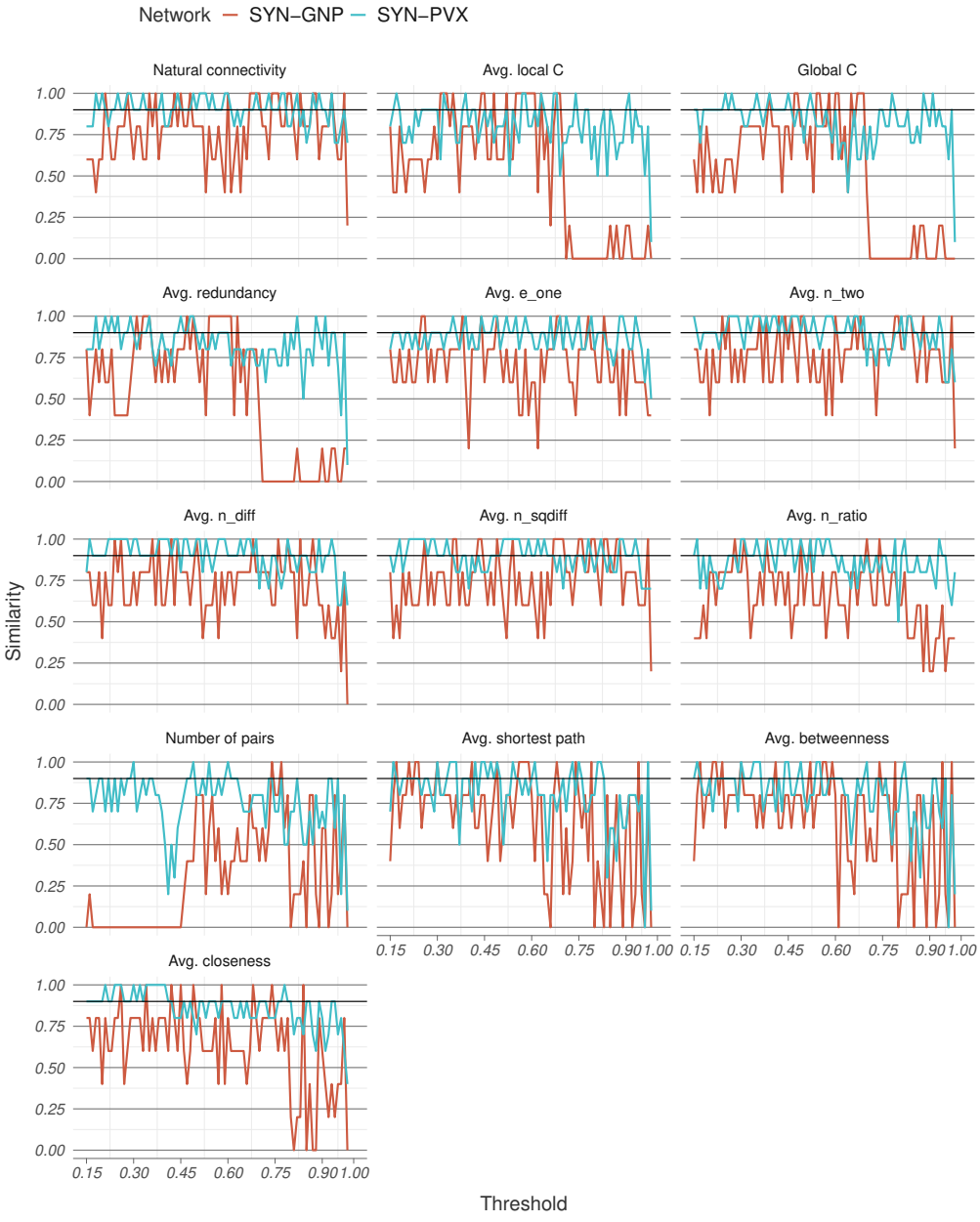

Figure A.4: LOUD metric rank similarity between consecutive thresholds for the synthetic networks. The SYN-GNP network exhibits lower k-similarity across most metrics and thresholds, as well as generally a wider variability in k-similarity scores.

A.3 Rank identifiability: additional results

Metric	PVX	ECOLI	YEAST	HPRED	SYN-GNP	SYN-PVX
Degree	0.86	0.87	0.89	**0.99**	0.40	**0.95**
Local clustering	0.25	0.22	0.20	0.46	∼0	0.36
Redundancy	**0.94**	**0.99**	**∼1**	**0.95**	∼0	0.54
PageRank	0.83	0.83	0.89	**∼1**	0.45	**0.91**
Closeness	0.74	0.77	0.82	**∼1**	0.25	0.89
Harmonic centrality	0.82	0.77	0.81	**0.97**	0.35	**0.91**
Betweenness	0.68	0.73	0.74	**0.93**	0.30	0.85
$e_{one}(v)$	**0.97**	**0.92**	**0.96**	**0.98**	0.50	**0.93**
$n_{two}(v)$	0.68	0.74	0.79	**0.96**	0.45	**0.92**
$n_{diff}(v)$	0.59	0.73	0.78	0.86	0.40	**0.92**
$n_{sqdiff}(v)$	0.86	0.88	0.88	**0.99**	0.45	**0.91**
$n_{ratio}(v)$	0.09	0.14	0.12	0.13	0.05	0.37
LOUD Average local clustering	0.45	0.30	0.38	0.69	0.10	0.54
LOUD Global clustering	0.89	**0.93**	**0.98**	**0.98**	0.05	0.53
LOUD Average redundancy	0.89	**0.96**	**0.96**	**0.99**	∼0	0.56
LOUD Average closeness	0.63	0.55	0.58	0.88	0.20	0.86
LOUD Average path length	0.22	0.02	0.43	0.63	∼0	0.50
LOUD Number of connected pairs	0.41	0.23	0.25	0.45	0.15	0.44
LOUD Average betweenness	0.22	0.07	0.21	0.35	0.05	0.51
LOUD Natural connectivity	**0.99**	**0.98**	**0.96**	**∼1**	0.45	**0.92**
LOUD Average $e_{one}(v)$	**0.97**	**0.92**	**0.96**	**0.98**	0.40	**0.94**
LOUD Average $n_{two}(v)$ nodes	0.72	0.74	0.87	**0.95**	0.45	**0.93**
LOUD Average $n_{diff}(v)$	0.62	0.74	0.85	**0.92**	0.45	**0.94**
LOUD Average $n_{sqdiff}(v)$	**0.94**	0.86	0.87	**0.97**	0.40	**0.94**
LOUD Average $n_{ratio}(v)$	0.44	0.42	0.51	0.39	0.35	0.40

Table A.3: Identifiability for the 25 metrics across each of the six scored networks. Values over 0.90 have been highlighted. The SYN-GNP network exhibits generally lower rank identifiability across node metrics compared to the scored PINs. The often higher SYN-PVX identifiability may be due to the random allocation of scores among network edges. This results in edges being deleted homogeneously across the network as the threshold is increased, meaning higher degree nodes are more likely to remain at relatively high degree at different thresholds.

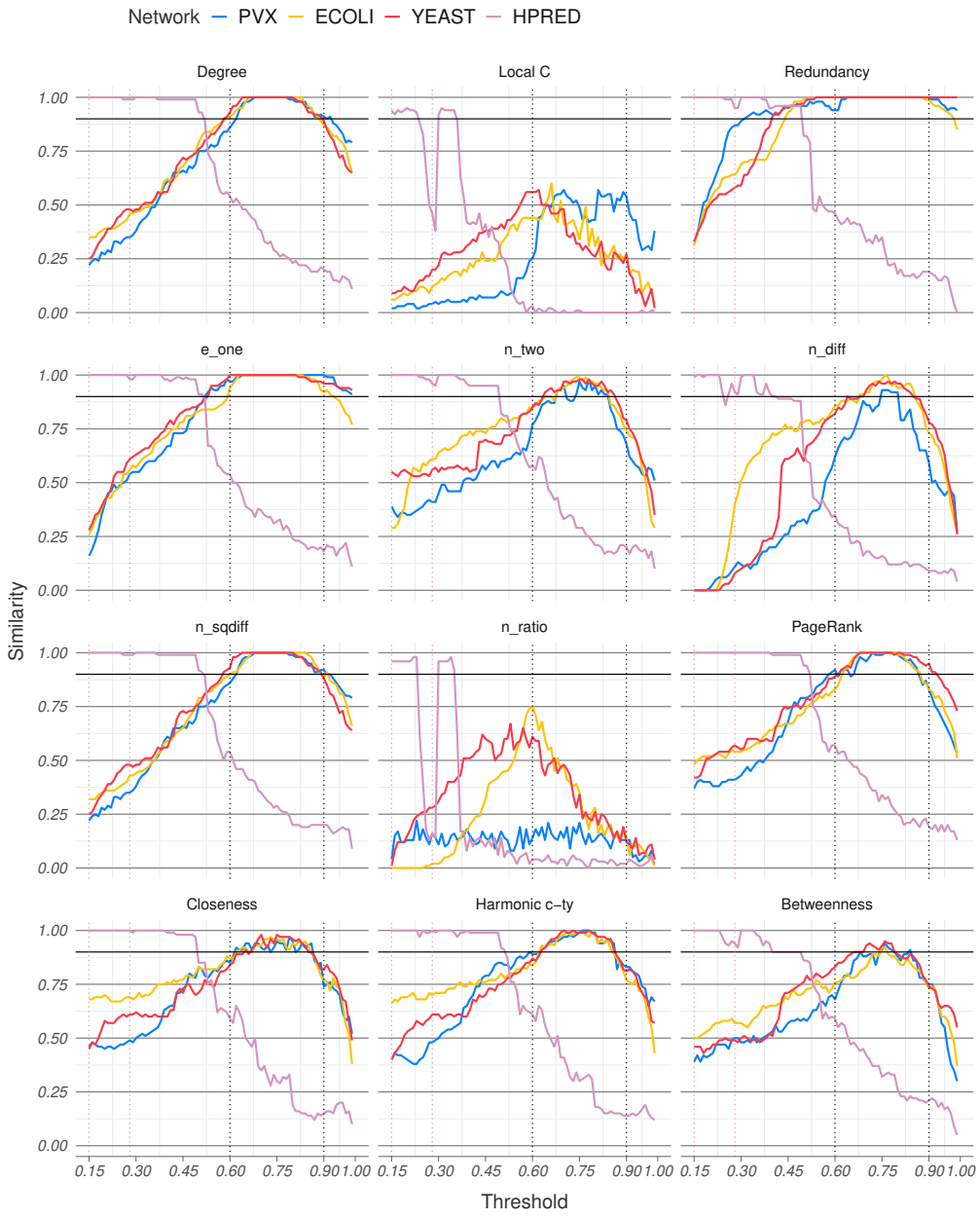

Figure A.5: Standard metric relaxed similarity between thresholded and overall ranks for the four PINs. The three STRING networks show generally good agreement. The HPRED network, which has been optimised over a different medium-high threshold region (0.15 to 0.28 as opposed to 0.60 to 0.90) behaves significantly differently as the threshold changes.

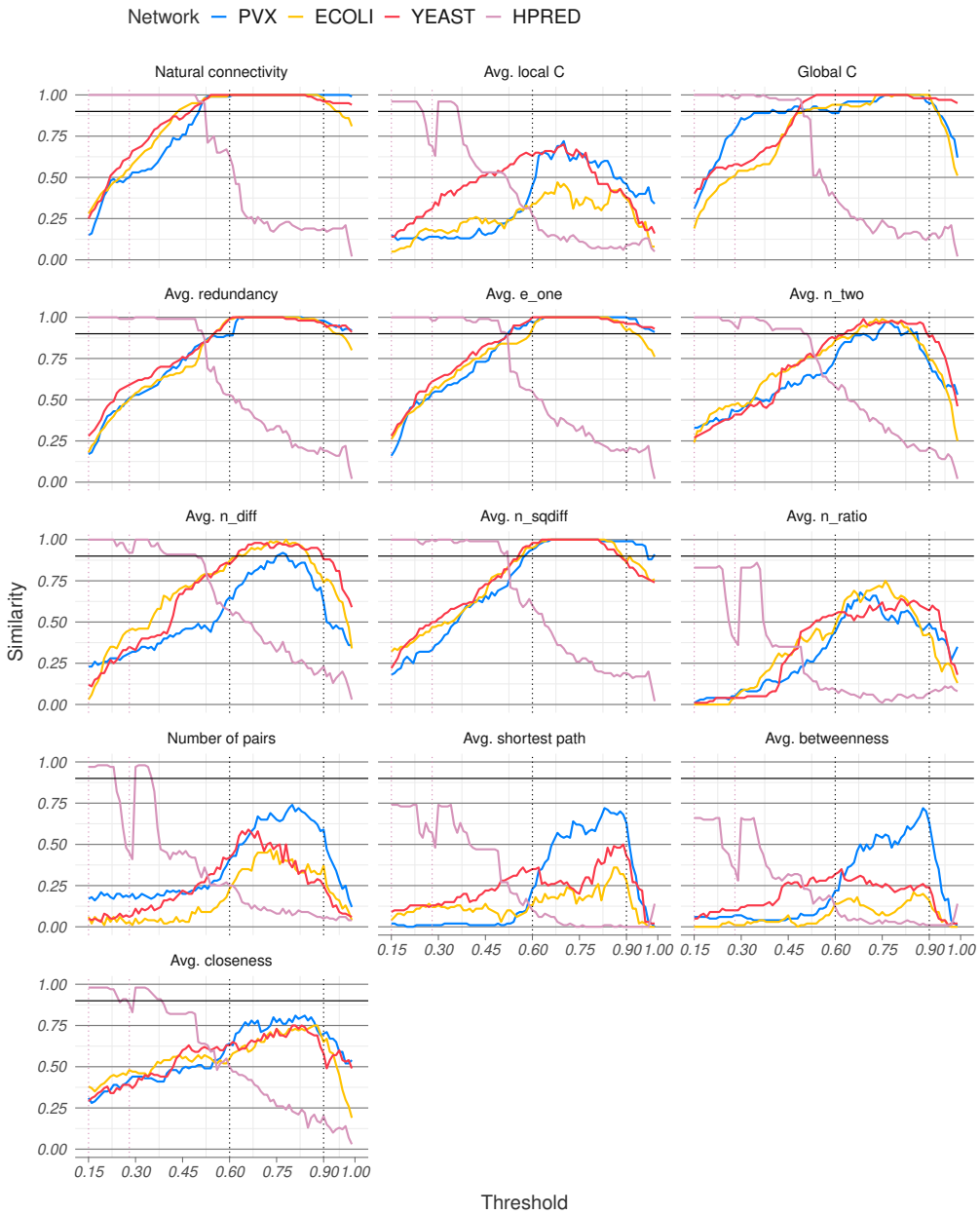

Figure A.6: LOUD metric relaxed rank similarity between between thresholded and overall ranks for the four PINs. As with the standard metrics (Figure A.5), relaxed similarity in the HPRED network behaves differently as a function of the threshold.

Figure A.7: Standard metric relaxed similarity between thresholded and overall ranks for the synthetic networks. The SYN-PVX network shows better relaxed rank similarity than the three STRING PINs (Figure A.5) overall, while the SYN-GNP network almost never reaches relaxed similarity of 0.90.

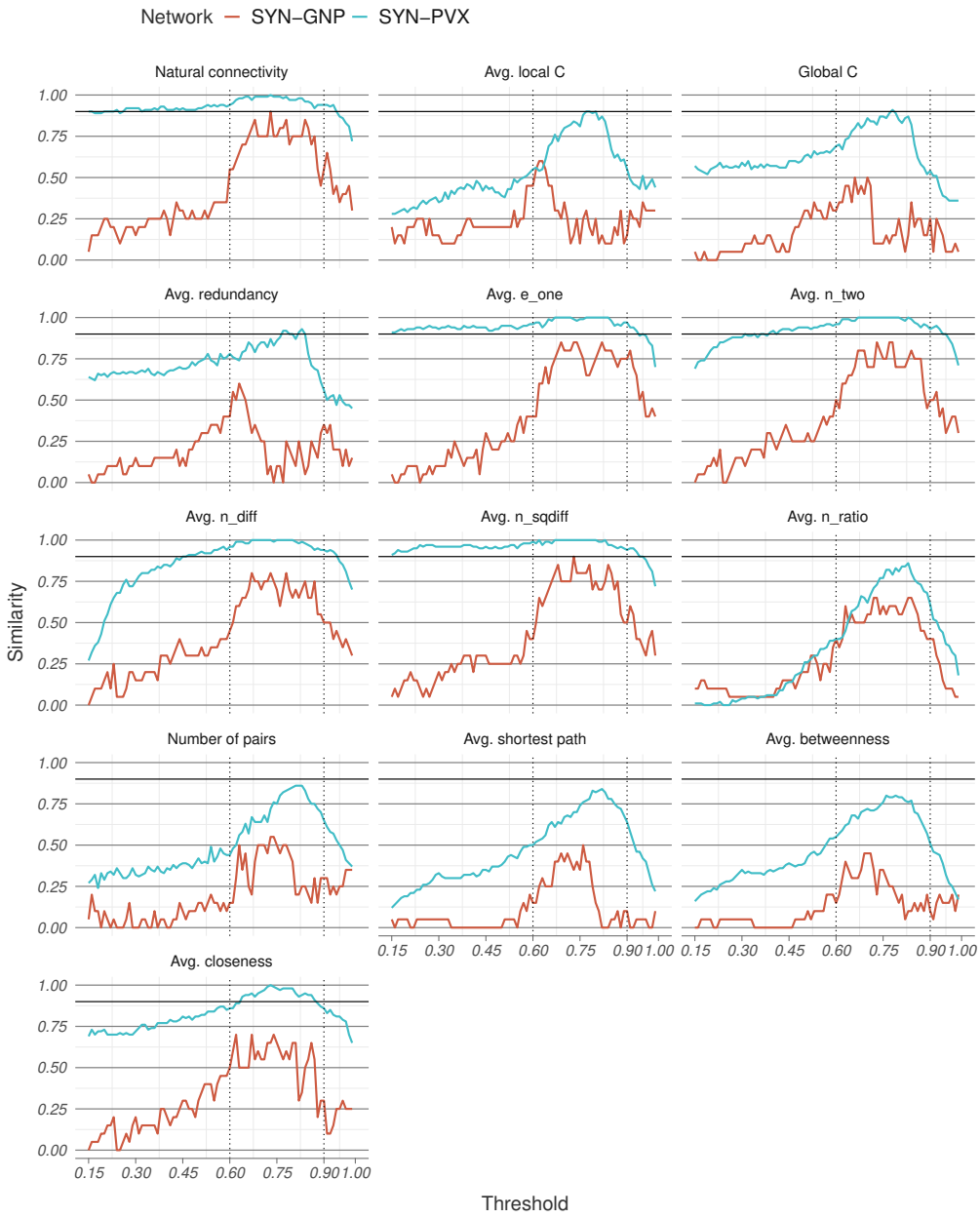

Figure A.8: LOUD metric relaxed rank similarity between between thresholded and overall ranks for the synthetic networks. The SYN-PVX network shows better relaxed rank similarity than the three STRING PINs (Figure A.6) overall, while the SYN-GNP network almost never reaches relaxed similarity of 0.90.

A.4 Rank instability: additional results

Metric	PVX	ECOLI	YEAST	HPRED	SYN-GNP	SYN-PVX
Degree	0.01	0.01	**0.01**	**∼0**	0.07	0.01
Local clustering	0.07	0.12	0.06	0.03	0.92	0.22
Redundancy	0.01	**0.01**	**0.01**	**∼0**	0.96	0.07
PageRank	0.02	0.01	**0.01**	**∼0**	0.07	0.01
Closeness	0.03	0.02	0.01	**∼0**	0.21	0.02
Harmonic centrality	0.02	0.02	0.01	**∼0**	0.12	0.01
Betweenness	0.02	0.02	0.01	**∼0**	0.12	0.02
$e_{one}(v)$	**0.01**	**∼0**	**∼0**	**∼0**	0.09	0.01
$n_{two}(v)$	0.04	0.02	0.01	**∼0**	0.09	0.01
$n_{diff}(v)$	0.05	0.02	0.01	0.01	0.12	0.02
$n_{sqdiff}(v)$	0.01	0.01	**0.01**	**∼0**	0.08	0.01
$n_{ratio}(v)$	0.46	0.27	0.15	0.05	0.38	0.31
LOUD Average local clustering	0.08	0.13	0.05	**0.01**	0.55	0.16
LOUD Global clustering	0.01	**∼0**	**0.01**	**∼0**	0.52	0.07
LOUD Average redundancy	**0.01**	**∼0**	**∼0**	**∼0**	0.60	0.08
LOUD Average closeness	0.04	0.03	0.03	**0.01**	0.35	0.05
LOUD Average path length	0.08	0.17	0.10	0.02	0.41	0.09
LOUD Number of connected pairs	0.25	0.53	0.54	0.09	0.55	0.11
LOUD Average betweenness	0.09	0.22	0.12	0.02	0.44	0.07
LOUD Natural connectivity	**0.01**	**∼0**	**∼0**	**∼0**	0.09	0.02
LOUD Average $e_{one}(v)$	**0.01**	**∼0**	**∼0**	**∼0**	0.11	0.01
LOUD Average $n_{two}(v)$	0.02	0.01	**0.01**	**∼0**	0.08	0.02
LOUD Average $n_{diff}(v)$	0.03	0.01	**0.01**	**∼0**	0.09	0.02
LOUD Average $n_{sqdiff}(v)$	0.01	**0.01**	**0.01**	**∼0**	0.08	0.01
LOUD Average $n_{ratio}(v)$	0.08	0.04	0.04	0.02	0.21	0.08

Table A.4: Instability for the 25 metrics across each of the six scored networks. Values under 0.01 have been highlighted. Values rounded down to 0.01 have not been highlighted. Both synthetic networks tend to have higher instability than the scored PINs across all node metrics.

B

Generative models based on uncertain PINs: additional figures

Contents

B.1 Frequency of edge occurrence

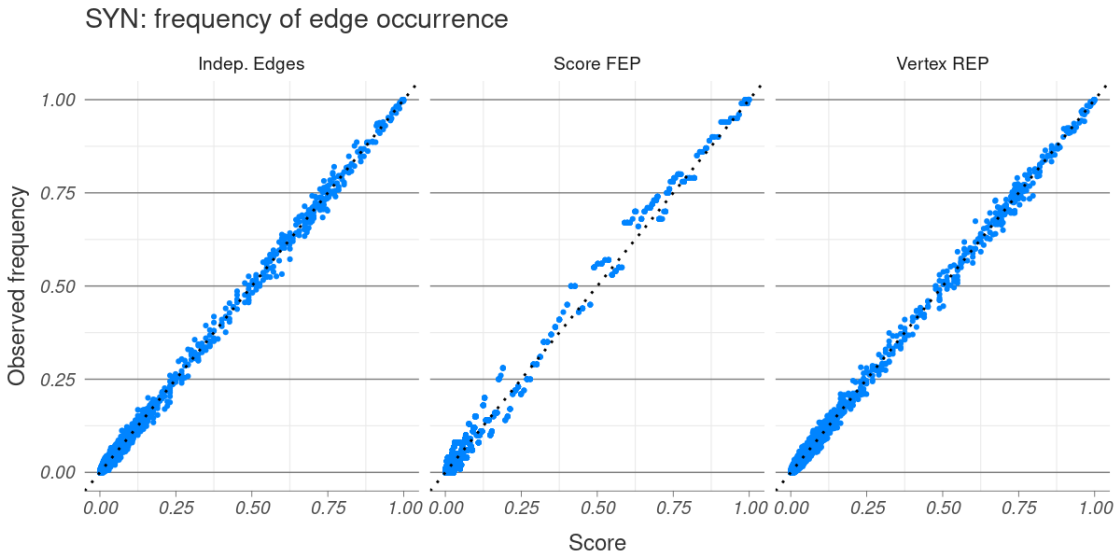

Figure B.1: Frequency of edge occurrence for the SYN network. Ten thousand arbitrarily chosen edges are plotted. The dotted line corresponds to the identity. All three methods on average produce networks with the correct edge frequency.

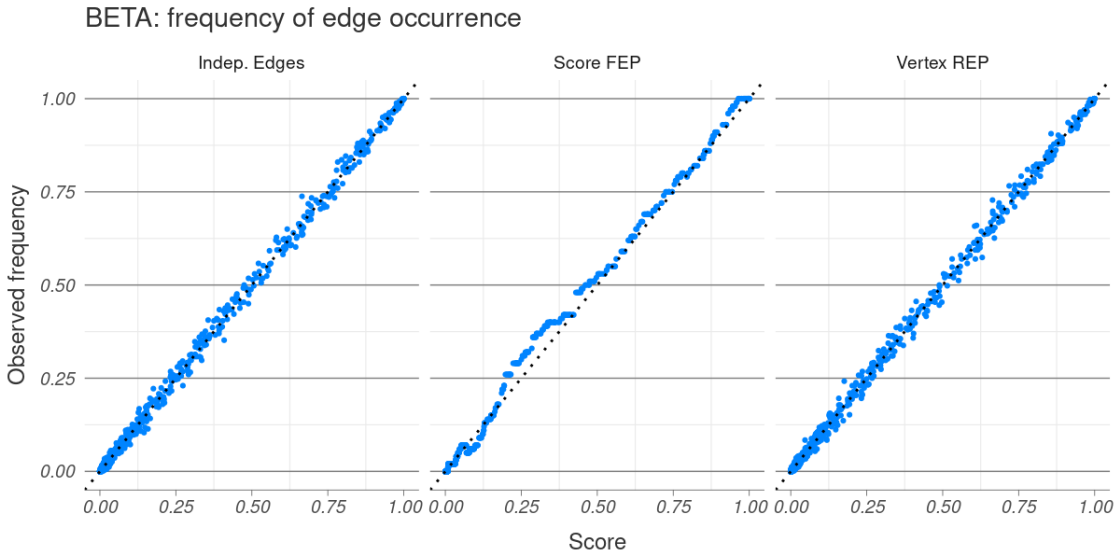

Figure B.2: Frequency of edge occurrence for the BETA network. Ten thousand arbitrarily chosen edges are plotted. The dotted line corresponds to the identity. All three methods on average produce networks with the correct edge frequency.

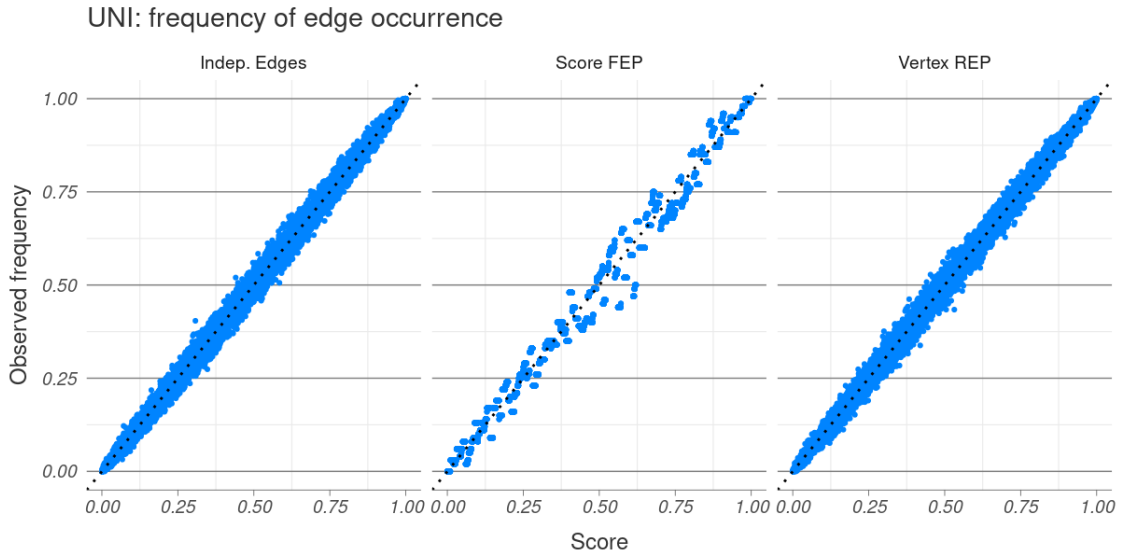

Figure B.3: Frequency of edge occurrence for the UNI network. Ten thousand arbitrarily chosen edges are plotted. The dotted line corresponds to the identity. All three methods on average produce networks with the correct edge frequency.

B.2 Number of edges

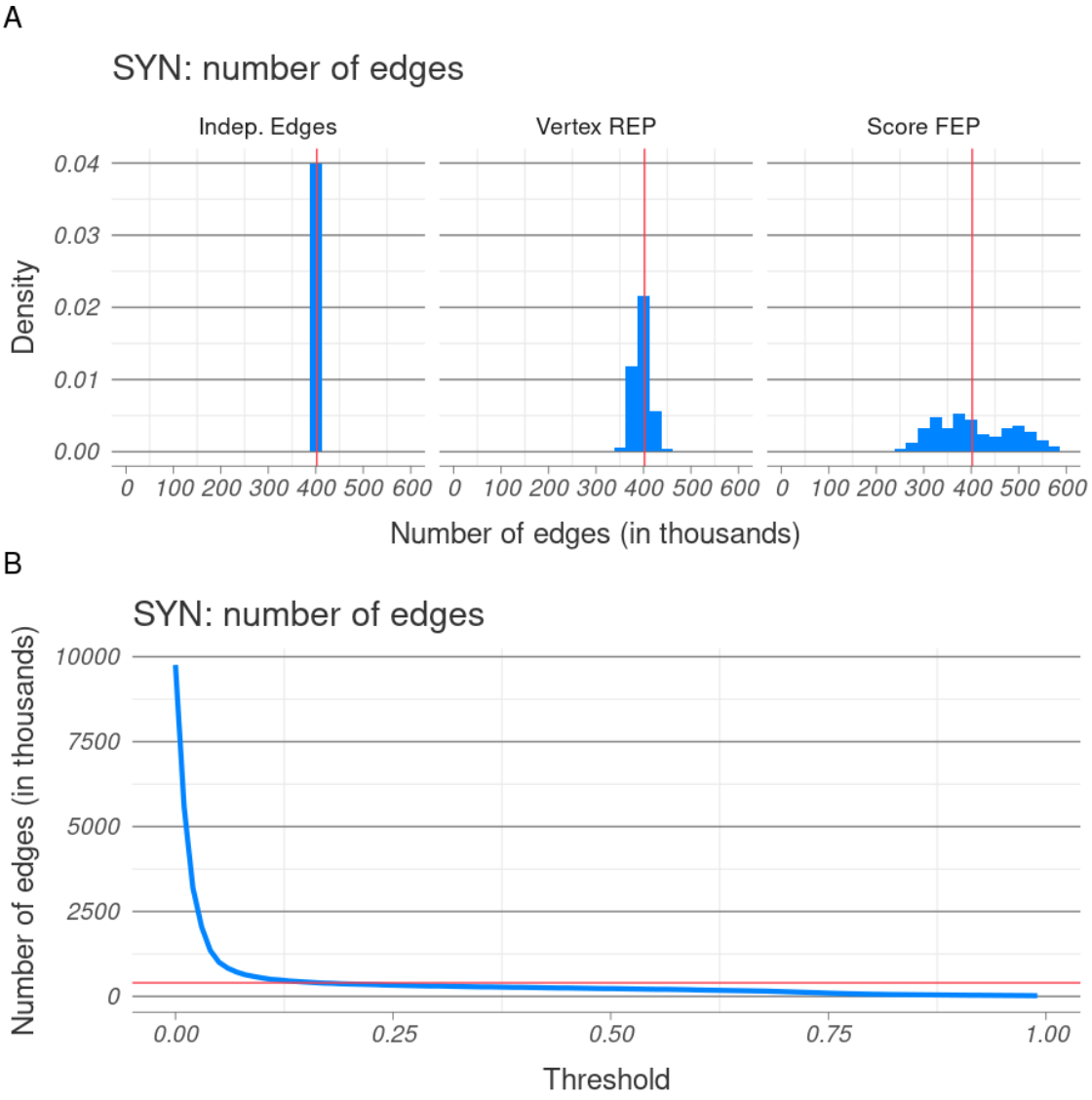

Figure B.4: Number of edges for the SYN network. (A) Histograms of edge counts, in thousands, obtained across the different algorithms. (B) Edge counts, in thousands, obtained by thresholding. The red lines correspond to the number of edges in the true network.

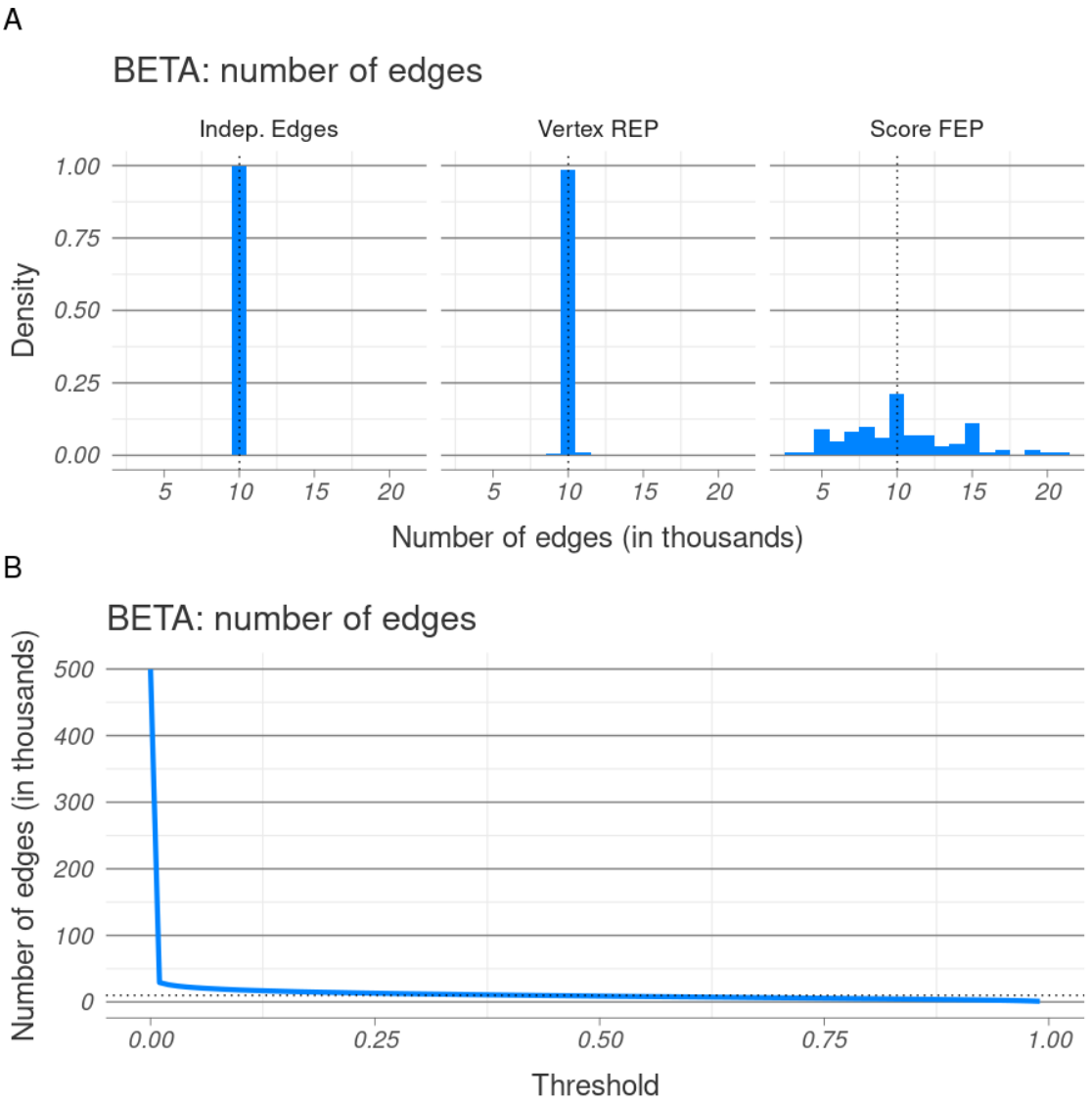

Figure B.5: Number of edges for the BETA network. (A) Histograms of edge counts, in thousands, obtained across the different algorithms. The dotted line corresponds to the expected number of edges in the true network. (B) Edge counts, in thousands, obtained by thresholding. The dotted lines correspond to the expected number of edges in the true network.

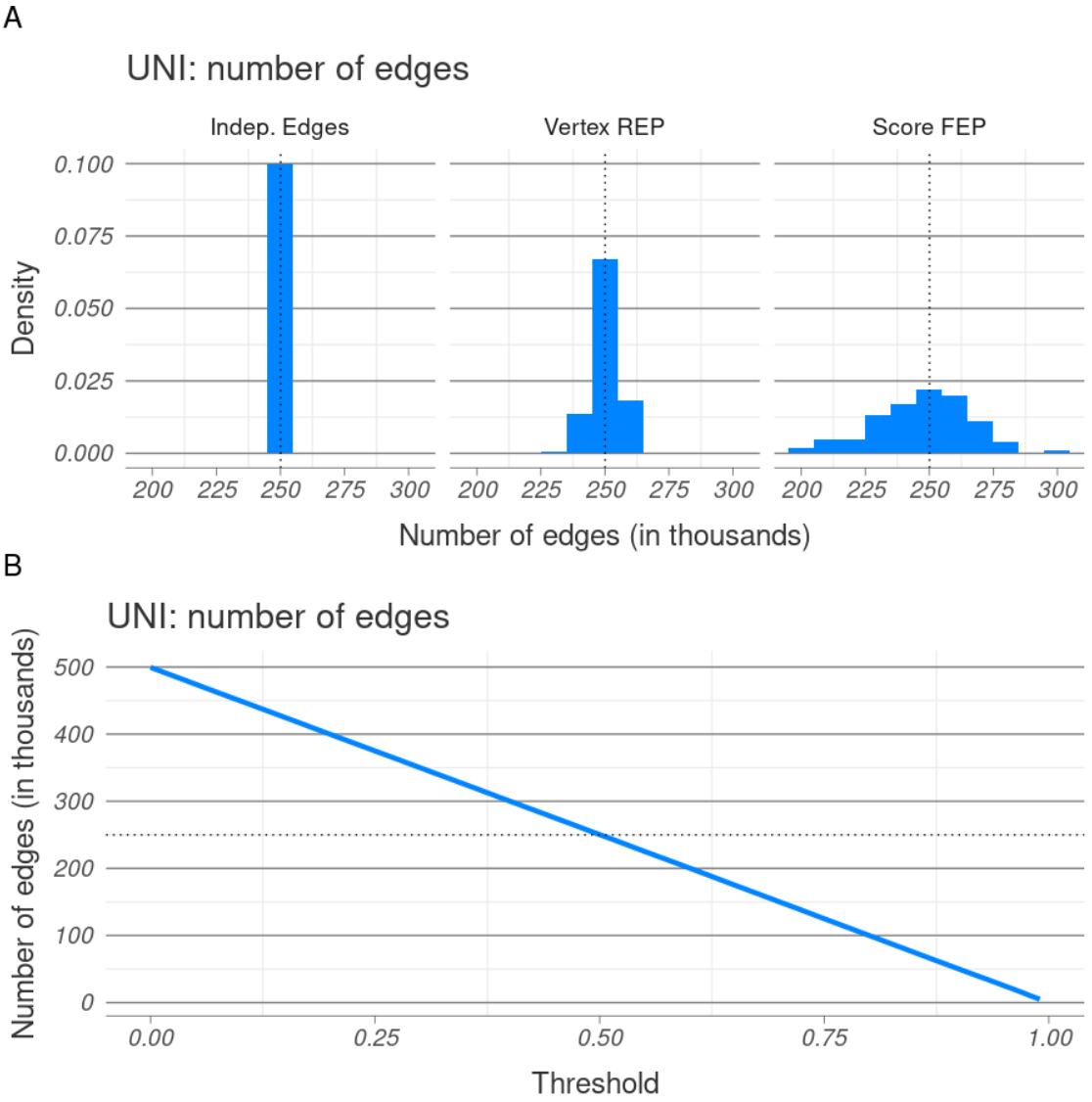

Figure B.6: Number of edges for the UNI network. (A) Histograms of edge counts, in thousands, obtained across the different algorithms. The dotted line corresponds to the expected number of edges in the true network. (B) Edge counts, in thousands, obtained by thresholding. The dotted lines correspond to the expected number of edges in the true network.

B.3 Largest connected component

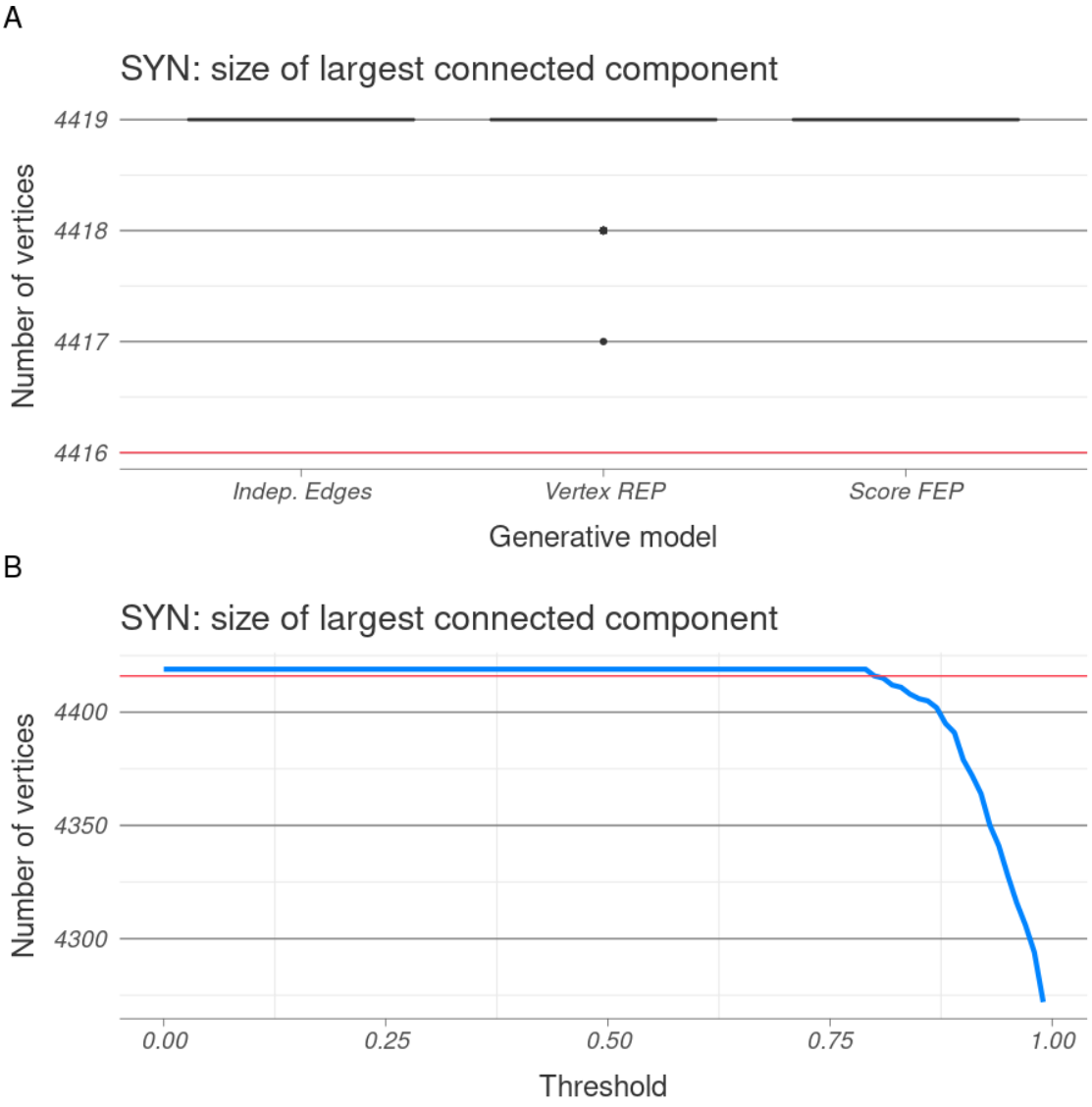

Figure B.7: Largest connected component size for the SYN network. (A) Box plot of the size of the largest connected component across different algorithms. (B) Size of the largest connected component across different thresholds. The red lines correspond to the number of connected components in the true network.

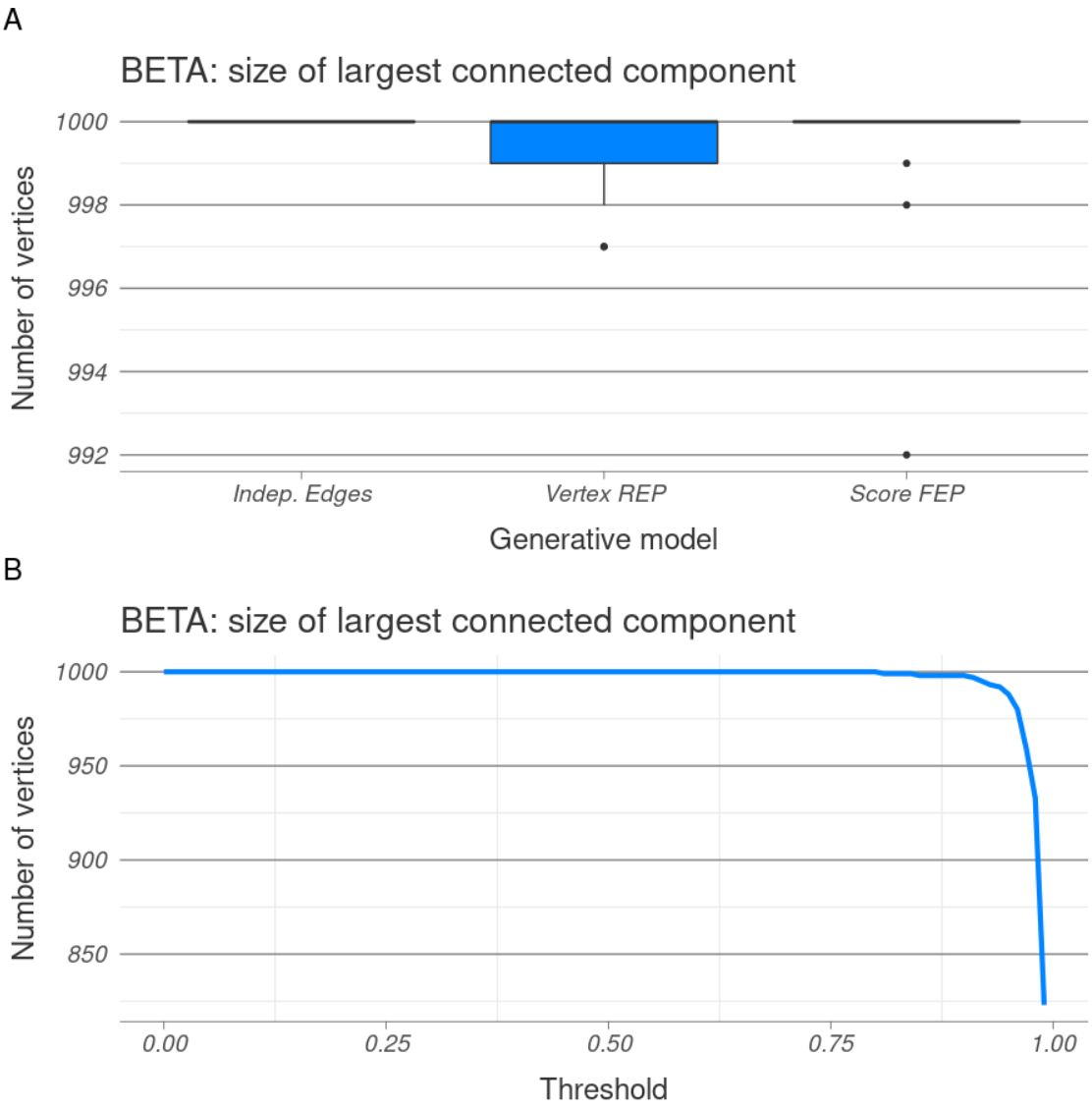

Figure B.8: Largest connected component size for the BETA network. (A) Box plot of the size of the largest connected component across different algorithms. (B) Size of the largest connected component across different thresholds.

B.4 Number of connected components

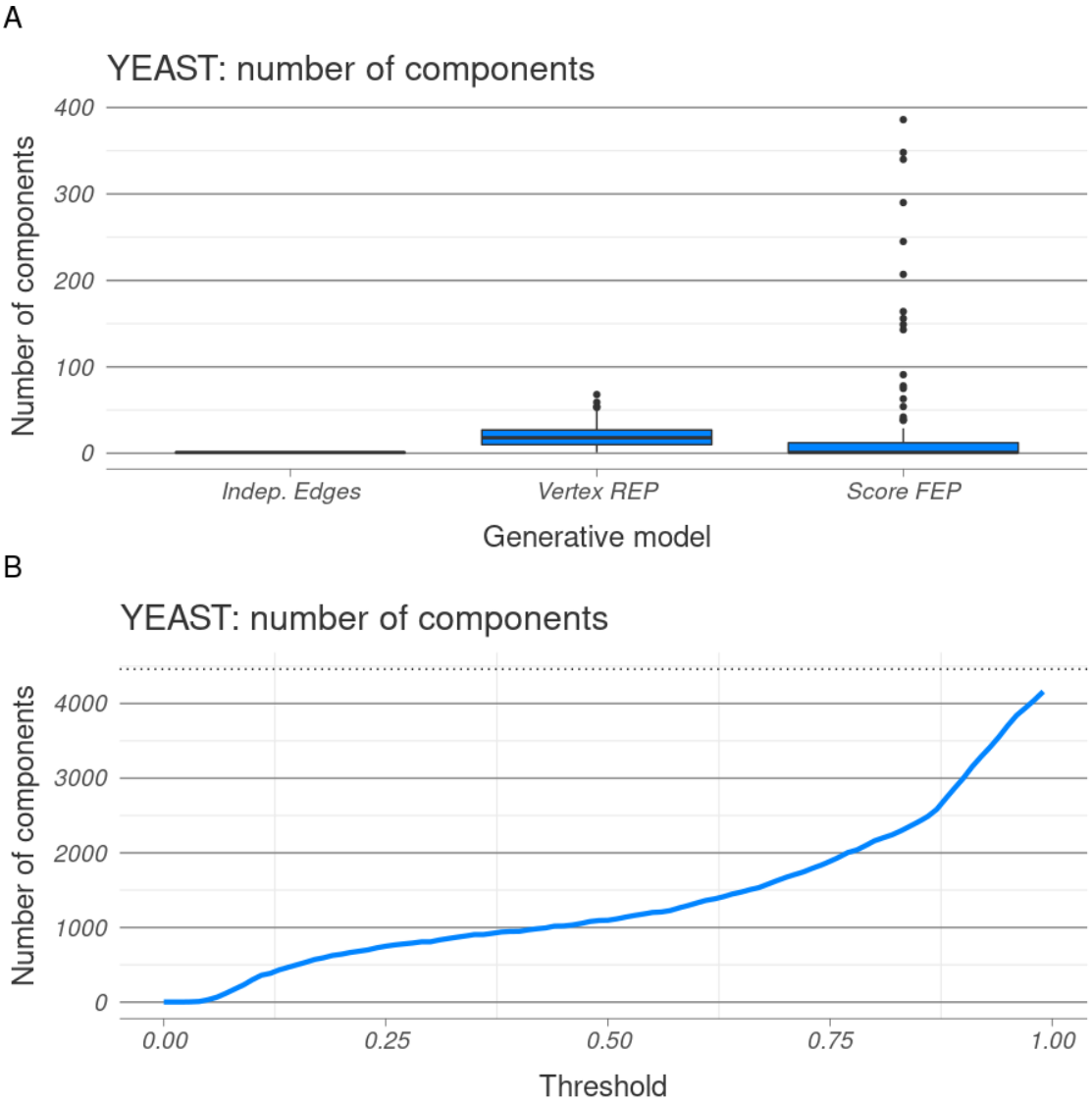

Figure B.9: Number of connected components in the YEAST network. (A) Box plot of the number of connected components across different algorithms. (B) Number of connected components across different thresholds.

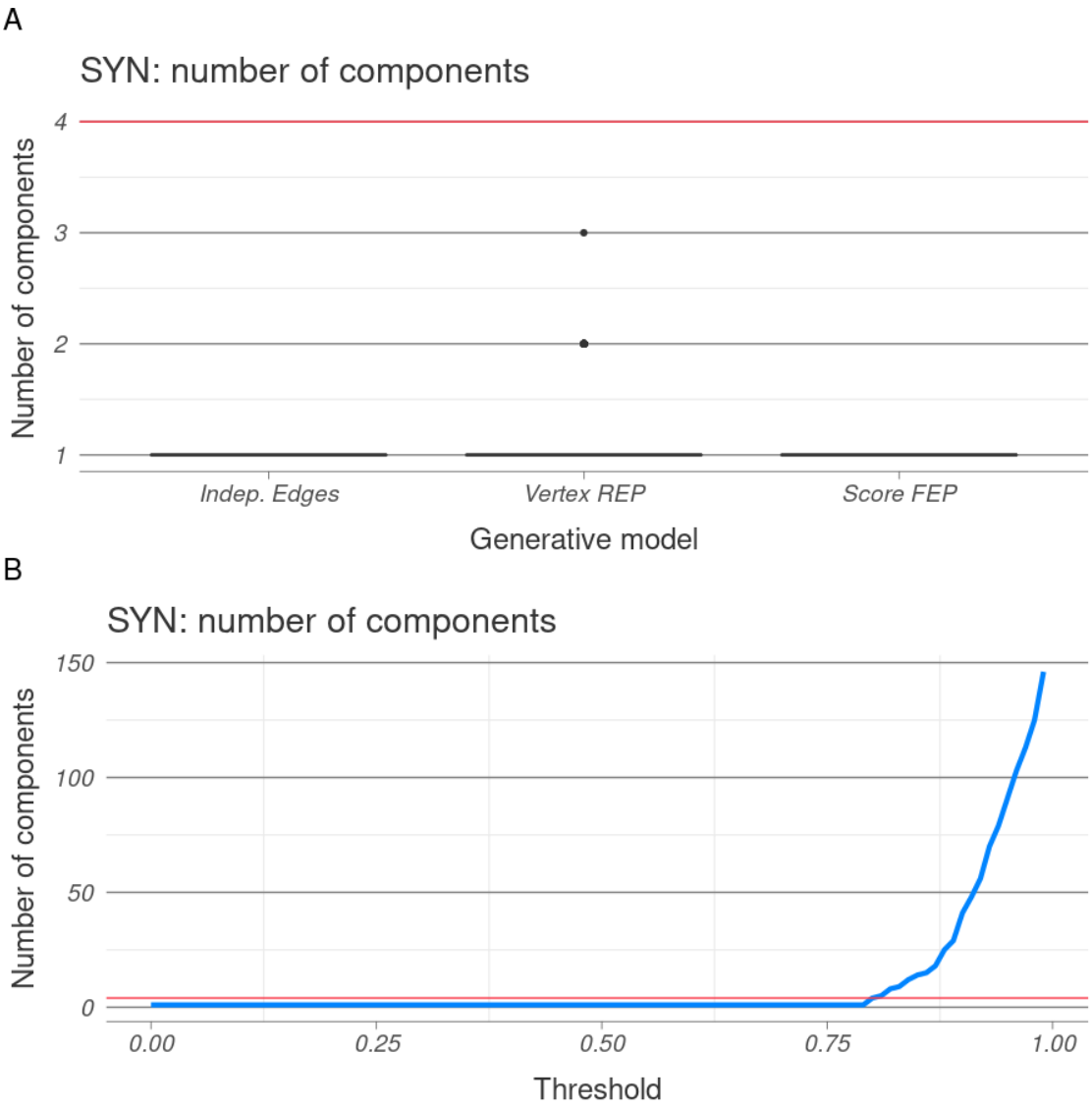

Figure B.10: Number of connected components in the SYN network. (A) Box plot of the number of connected components across different algorithms. (B) Number of connected components across different thresholds. The red lines correspond to the true number of connected components.

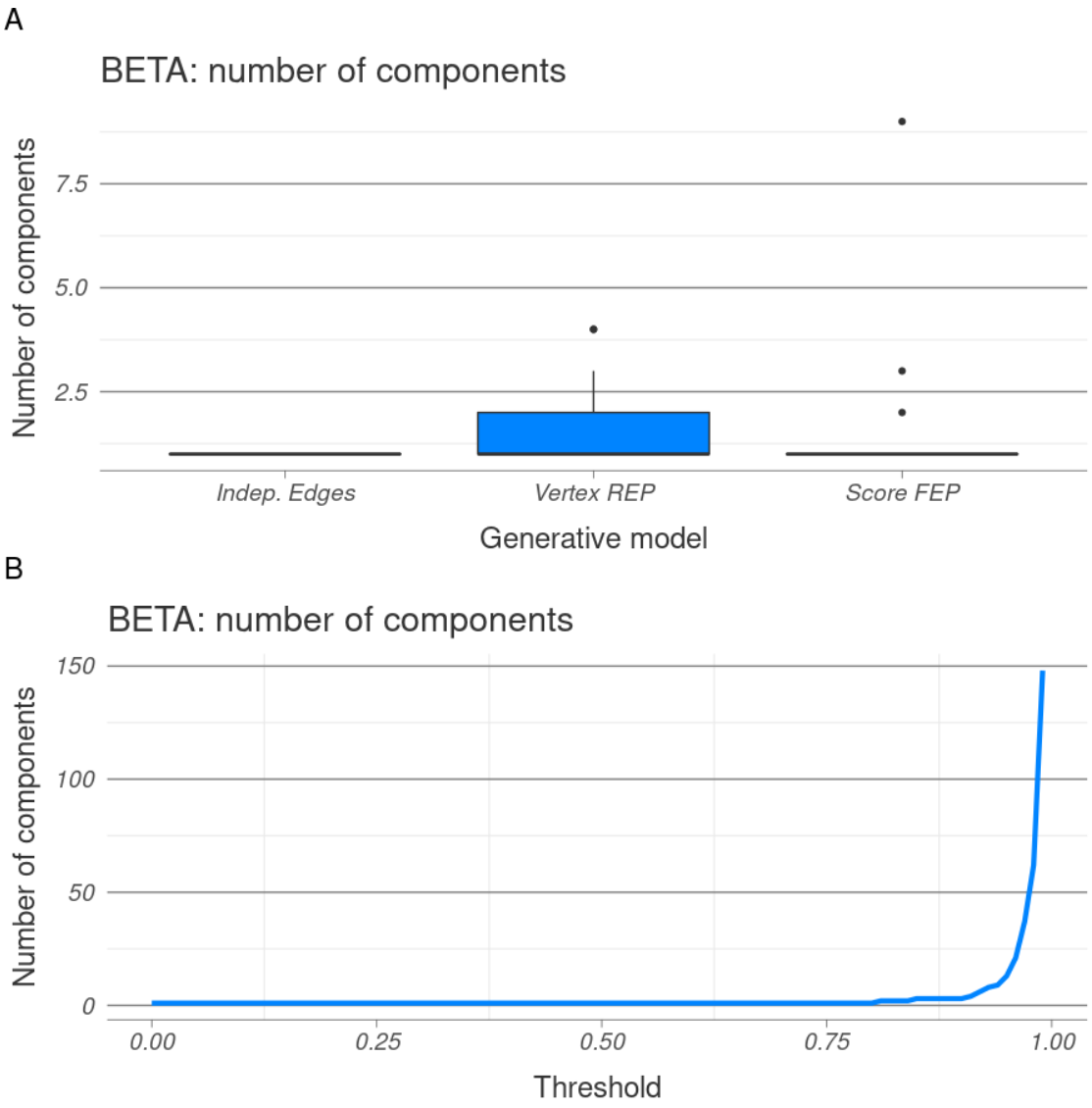

Figure B.11: Number of connected components in the BETA network. (A) Box plot of the number of connected components across different algorithms. (B) Number of connected components across different thresholds.

B.5 Global clustering coefficient

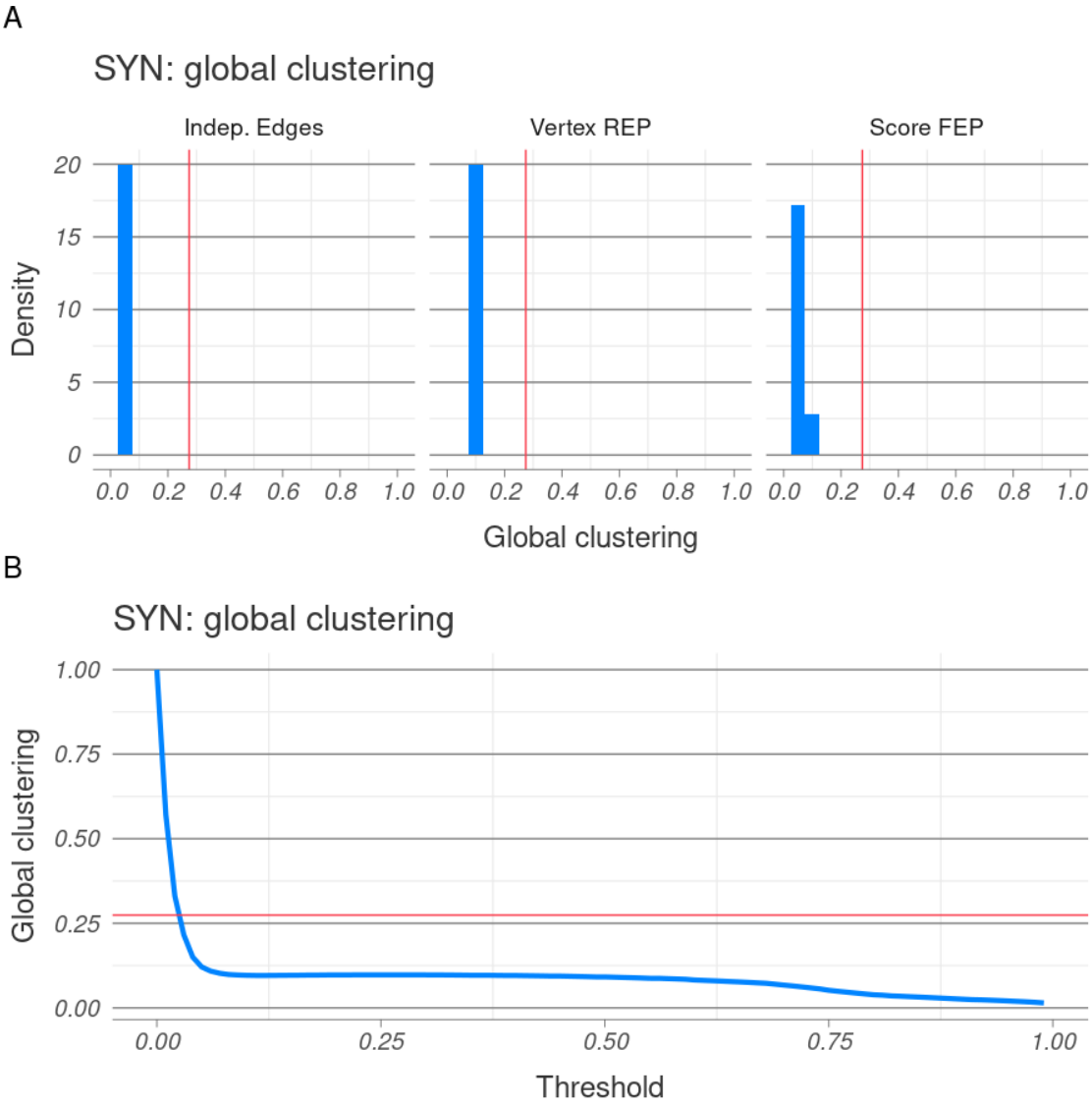

Figure B.12: Global clustering coefficient in the SYN network. (A) Histograms of the global clustering coefficient across different algorithms. (B) Global clustering coefficient across different thresholds. The red lines correspond to global clustering in the true network.

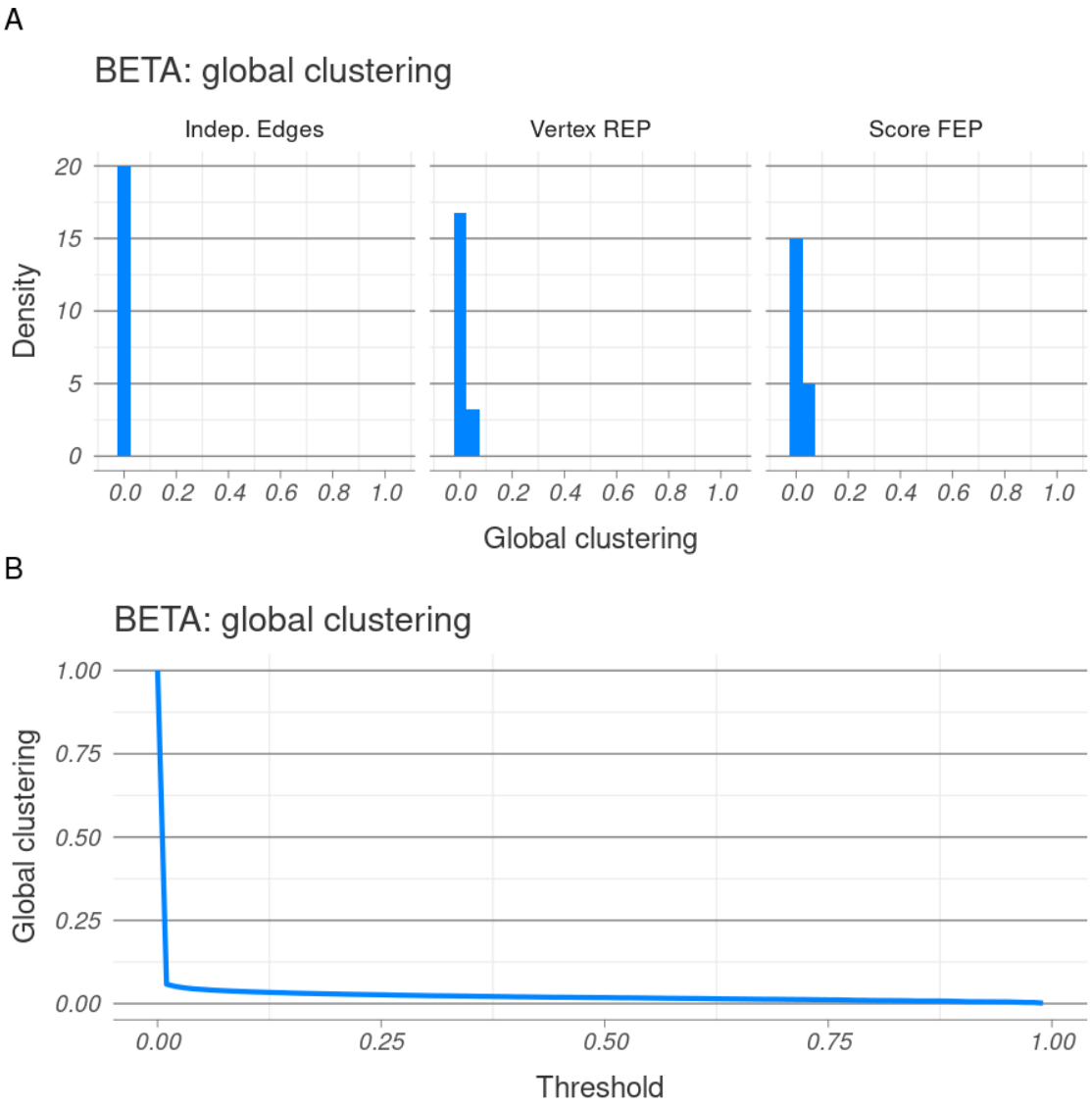

Figure B.13: Global clustering coefficient in the BETA network. (A) Histograms of the global clustering coefficient across different algorithms. (B) Global clustering coefficient across different thresholds.

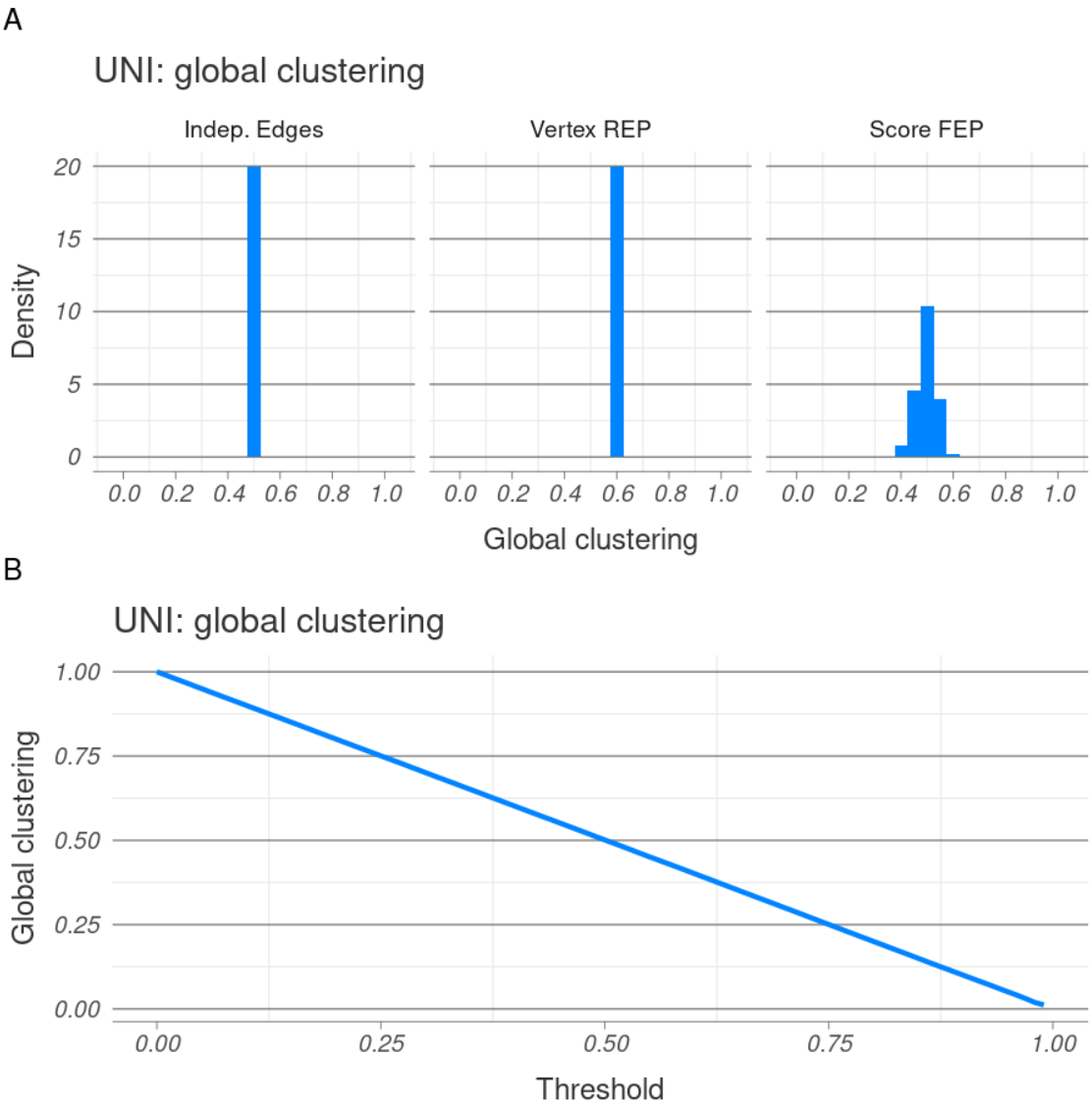

Figure B.14: Global clustering coefficient in the UNI network. (A) Histograms of the global clustering coefficient across different algorithms. (B) Global clustering coefficient across different thresholds.

B.6 Average local clustering coefficient

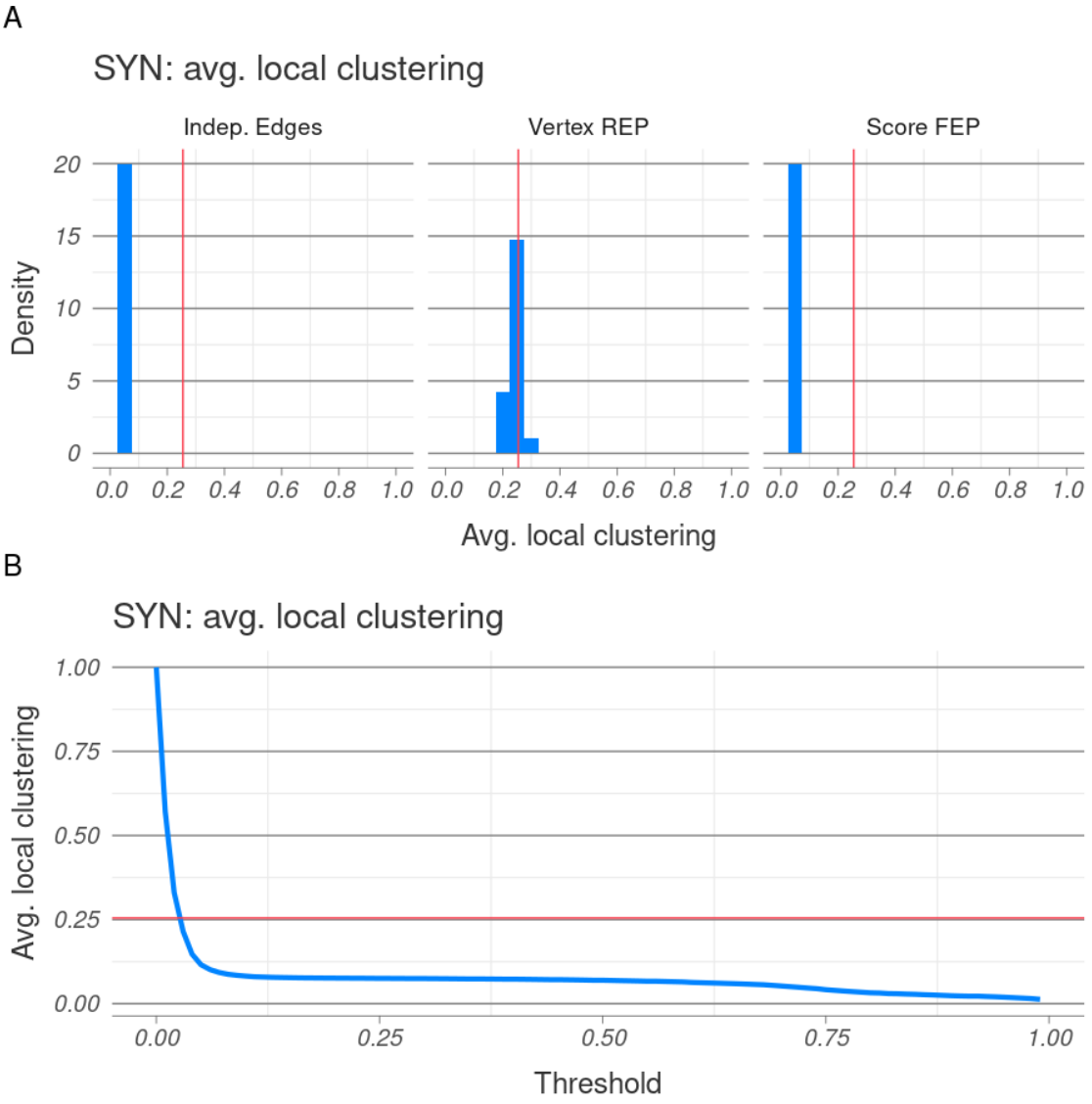

Figure B.15: Average local clustering coefficient in the SYN network. (A) Histograms of the average local clustering coefficient across different algorithms. (B) Average local clustering coefficient across different thresholds. The red lines correspond to average local clustering in the true network.

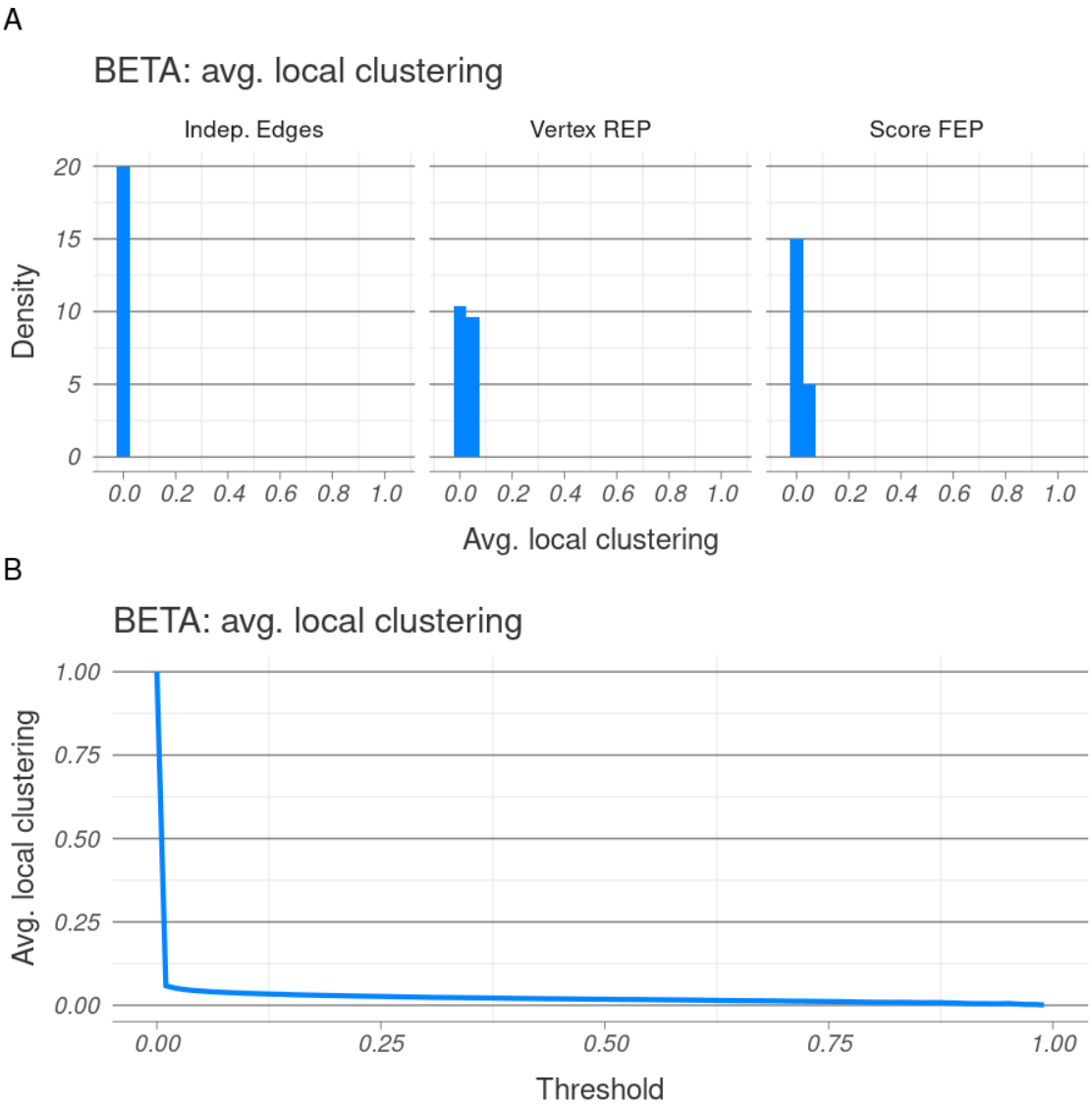

Figure B.16: Average local clustering coefficient in the BETA network. (A) Histograms of the average local clustering coefficient across different algorithms. (B) Average local clustering coefficient across different thresholds.

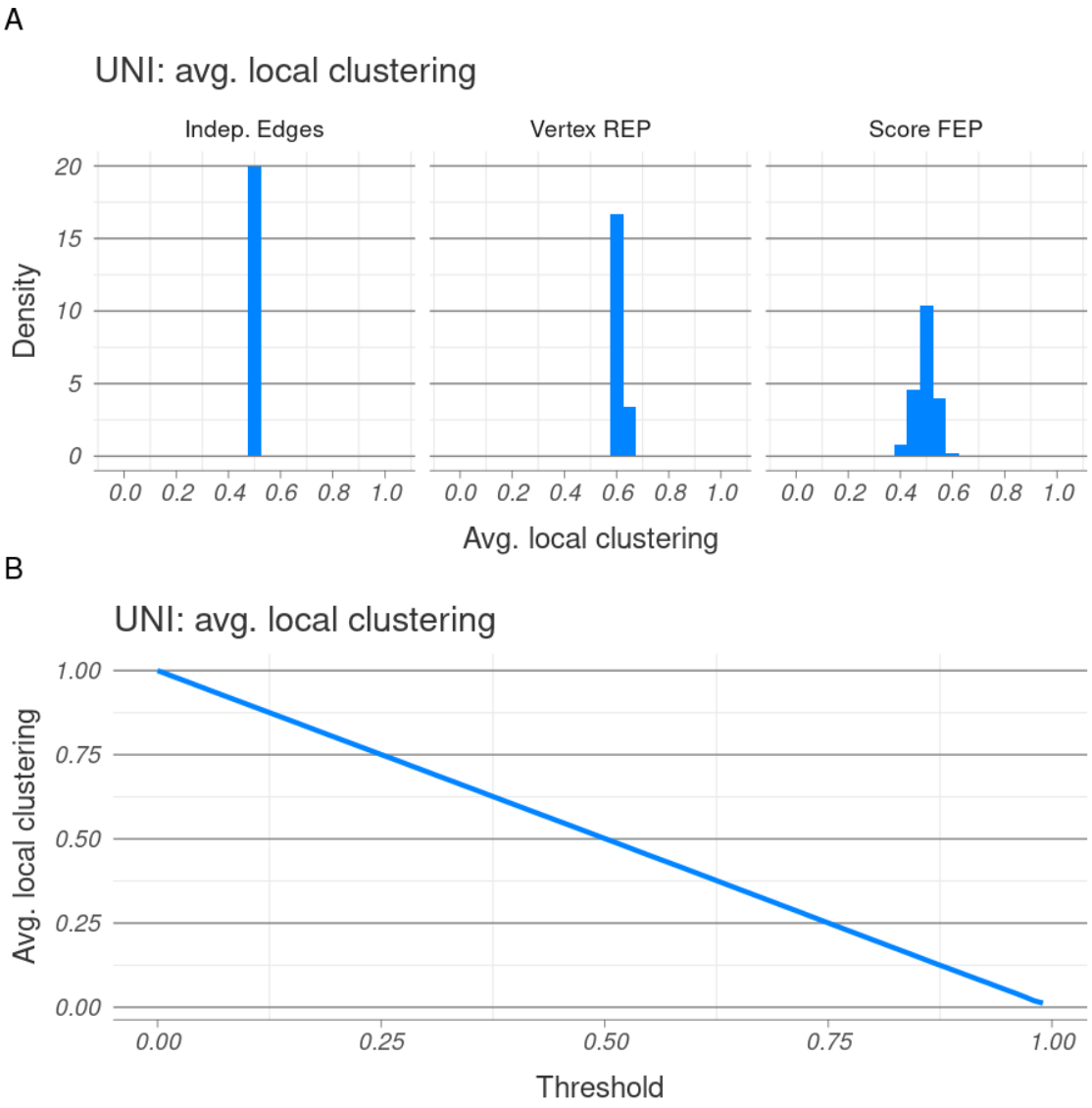

Figure B.17: Average local clustering coefficient in the UNI network. (A) Histograms of the average local clustering coefficient across different algorithms. (B) Average local clustering coefficient across different thresholds.

C

COGENT: implementation details

Contents

C.1 Functions available in COGENT

The R package COGENT includes 15 functions, listed in Table C.1. They are split into four categories— input checks, internal functions, network similarity (i.e. consistency) functions, and main COGENT functions.

C.1.1 Input checks

The input check functions are `checkArrayList`, `checkExpressionDF`, `checkFun` and `checkMatrixList`. They are used to check whether the input (and output) of

Function	Description
alignArrays	Align node metric arrays
alignMatrices	Align adjacency matrices
calculateEuclideanDistance	Euclidean distance
calculateKSimilarity	Rank k-similarity
checkArrayList	Check node metric list
checkExpressionDF	Check a gene expression data frame
checkFun	Check function
checkMatrixList	Check adjacency matrix list
cogentLinear	Multiple COGENT calls, executed on a single thread
cogentParallel	Multiple COGENT calls, executed in parallel
cogentSingle	Single COGENT call
getEdgeSimilarity	Get the edge similarity for two networks
getEdgeSimilarityCorrected	Get the edge similarity with configuration model correction
getNodeSimilarity	Get the node similarity of two networks
splitExpressionData	Randomly subset expression samples

Table C.1: Functions implemented in COGENT.

intermediate steps of the COGENT pipeline adhere to the required format.

Different types of common errors are clearly documented, so that input checks can be used for debugging before any computationally demanding analysis is performed.

C.1.2 Internal functions

The internal functions include alignment functions (`alignArrays` and `alignMatrices`), similarity and distance functions (`calculateEuclideanDistance` and `calculateKSimilarity`) and the random data split function (`splitExpressionData`).

Network comparison in COGENT relies on a one-to-one mapping of nodes in the pair of networks generated at each iteration. If the network construction method does not respect the order genes are labelled in the expression data set, e.g. by removing isolated nodes, an additional alignment step is needed. Similarly, if the node metric function does not respect a fixed node ordering, alignment may be required before the node metric comparison step. Alignment is not performed by default, and appropriate error messages are produced if a mismatch is detected.

The similarity and distance functions are the parametrised rank k-similarity and Euclidean distance with optional rescaling to $[0, 1]$. For rank k-similarity, the

top proportion of ranks to be compared can be given as a quantile (e.g. the top 10% of nodes) or as an integer (e.g. the top 100 nodes).

C.1.3 Network similarity

Network similarity functions include the edge set consistency functions calculated by `getEdgeSimilarity`, the density adjusted edge set consistency in `getEdgeSimilarityCorrected`, and node metric consistency functions calculated by `getNodeSimilarity`. Note that global and local similarity are calculated simultaneously by `getEdgeSimilarity`. Density adjustment in `getEdgeSimilarityCorrected` can be carried either with the fully random correction term, or with semi-random correction. Similarly, `getNodeSimilarity` calculates node metric consistency in any of the three available ways (via a correlation, rank k-similarity, or Euclidean distance).

C.1.4 Main functions

The main COGENT functions `cogentSingle`, `cogentLinear` and `cogentParallel` execute the complete COGENT pipeline. The principal input of these are a gene expression data set, stored in a data frame, and a network construction method, which transforms the data frame to a network adjacency matrix.

The function `cogentSingle` carries out a single iteration of COGENT. It splits the data in two random subset, constructs a network from each dataset, and returns a number of consistency metrics (e.g. global similarity, average local similarity, ad rank k-similarity for some user-defined node metric). Optional parameters, e.g. whether to do any alignment or what proportion of the data is shared during the data split step, are passed to the internal and network similarity functions described above.

The functions `cogentLinear` and `cogentParallel` are wrappers for `cogentSingle` and are used for carrying out multiple COGENT iterations. The former executes COGENT linearly, and is suitable for cases where network construction is fast or slow, but parallelised. The latter executes COGENT in parallel on

multiple threads. It is designed for cases where network construction is slow and not parallelised.

C.2 Selected documentation

C.2.1 getEdgeSimilarity(): Get the edge similarity for two networks

Usage

```
getEdgeSimilarity(A, align = FALSE)
```

Aguments

A A list of two square (weighted) adjacency matrices.

align Logical; Whether to align the two adjacency matrices. Only set to `TRUE` if this is not done automatically.

Value

The result is a list of similarity measures of the two networks with adjacency matrices in `A`. This includes:

nodeCount The number of non-isolated genes across the two networks.

globalSimilarity The (weighted) Jaccard index of the edge sets of the two networks.

localSimilarity The (weighted) Jaccard index for each gene neighbourhood.

Description

`getEdgeSimilarity()` calculates the weighted or unweighted Jaccard index between the edge sets of two networks. The calculation is performed both for the full edge set and for each node neighbourhood.

Examples

```
# Generate two adjacency matrices
A1 <- matrix(0, ncol=10, nrow=10); A2 <- matrix(0, ncol=10, nrow=10)
A1[upper.tri(A1)] <- rbinom(45, 1, .2); A1 <- A1+t(A1)
A2[upper.tri(A1)] <- rbinom(45, 1, .4); A2 <- A2+t(A2)
colnames(A1) <- rownames(A1) <- LETTERS[1:10]
colnames(A2) <- rownames(A2) <- LETTERS[6:15]
# Calculate similarity
getEdgeSimilarity(list(A1, A2), align=TRUE)
```

C.2.2 cogentParallel(): Multiple COGENT calls, executed in parallel

Usage

```
cogentParallel(df, netwkFun, nodeFun = NULL, repCount = 100,
  threadCount = 4, propShared = 0, align = FALSE,
  nodeModes = "all", use = "complete.obs", method = "pearson",
  k.or.p = 0.1, scale = FALSE)
```

Aguments

df A COGENT-compatible data frame with rows corresponding to genes and columns corresponding to samples. A column called Name containing gene names is expected.

netwkFun A function mapping a COGENT-compatible data frame to an adjacency matrix.

nodeFun A function calculating node metrics from an adjacency matrix. Defaults to NULL.

repCount Number of repetitions.

threadCount Number of threads to use.

propShared Proportion of samples (from the input `df`) to be shared across the two random split.

align Logical; Whether to align the two outputs of `netwkFun`. Only set to `TRUE` if `netwkFun()` doesn't preserve gene order.

nodeModes Which modes of comparison to use for node metrics. This should be either `"all"` or a subset of {`"cor"`, `"ksim"`, `"L2"`}. Ignored if `nodeFun=NULL`.

use Further argument for node comparison, see `cor`.

method Further argument for node comparison, see `cor`.

k.or.p Further argument for node comparison, see `calculateKSimilarity`.

scale Further argument for node comparison, see `CalculateEuclideanDistance`.

Value

A data frame summarising results. See `cogentSingle` for details. `localSimilarity` is reported as the average over all nodes.

Description

`cogentParallel` will repeatedly randomly split a given gene expression data frame in two sample groups, construct a network from each group, and measure network similarity. It will do so by calling `cogentSingle` in parallel. To be used when network construction is slow and not parallelised.

Examples

```
# Generate some expression data
df <- as.data.frame(matrix(runif(500), nrow=10, ncol=50))
df <- cbind(Name=LETTERS[1:10], df)
# Construct co-expression networks by thresholding the Pearson
   correlation
# coefficient at 0.20.
foo <- function(df){
 A <- cor(t(df[,colnames(df)!="Name"]))
```

```
 A <- 1*(A>0.20)
 return(A)
}
# Use the degree as a node metric
fooNode <- function(x) rowSums(x)
# Calculate stability:
## Without node metric comparison, on two threads
cogentParallel(df, foo, threadCount=2)
## With all types of node metric comparison
cogentParallel(df, foo, fooNode, threadCount=2)
```

When in doubt, go to the library.

— J.K. Rowling, *Harry Potter and the Chamber of Secrets*

References

Aaij, Cees and Piet Borst (1972). "The gel electrophoresis of DNA". In: *Biochimica et Biophysica Acta (BBA)—Nucleic Acids and Protein Synthesis* 269.2, pp. 192–200.

Abbas-Aghababazadeh, Farnoosh, Qian Li, and Brooke L Fridley (2018). "Comparison of normalization approaches for gene expression studies completed with high-throughput sequencing". In: *PloS one* 13.10, e0206312.

Abraham, Sheela A, Lisa EM Hopcroft, Emma Carrick, Mark E Drotar, Karen Dunn, Andrew JK Williamson, Koorosh Korfi, Pablo Baquero, Laura E Park, Mary T Scott, et al. (2016). "Dual targeting of p53 and c-MYC selectively eliminates leukaemic stem cells". In: *Nature* 534.7607, p. 341.

Acuner Ozbabacan, Saliha Ece, Hatice Billur Engin, Attila Gursoy, and Ozlem Keskin (2011). "Transient protein–protein interactions". In: *Protein Engineering, Design and Selection* 24.9, pp. 635–648.

Ahn, Yong-Yeol, James P Bagrow, and Sune Lehmann (2010). "Link communities reveal multiscale complexity in networks". In: *Nature* 466.7307, p. 761.

Ahnert, Sebastian E, Diego Garlaschelli, Thomas MA Fink, and Guido Caldarelli (2007). "Ensemble approach to the analysis of weighted networks". In: *Physical Review E* 76.1, p. 016101.

Alanis-Lobato, Gregorio, Miguel A Andrade-Navarro, and Martin H Schaefer (2016). "HIPPIE v2.0: enhancing meaningfulness and reliability of protein–protein interaction networks". In: *Nucleic Acids Research*, gkw985.

Albert, Reka (2005). "Scale-free networks in cell biology". In: *Journal of Cell Science* 118.21, pp. 4947–4957.

Alberts, Bruce, Dennis Bray, Karen Hopkin, Alexander D Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter (2013). *Essential cell biology*. Garland Science.

Ammari, Mais G, Cathy R Gresham, Fiona M McCarthy, and Bindu Nanduri (2016). "HPIDB 2.0: a curated database for host–pathogen interactions". In: *Database* 2016.

Anthonisse, Jac M (1971). "The rush in a directed graph". In: *Stichting Mathematisch Centrum. Mathematische Besliskunde* BN 9/71.

Athar, Awais, Anja Füllgrabe, Nancy George, Haider Iqbal, Laura Huerta, Ahmed Ali, Catherine Snow, Nuno A Fonseca, Robert Petryszak, Irene Papatheodorou, et al. (2018). "ArrayExpress update–from bulk to single-cell expression data". In: *Nucleic Acids Research* 47.D1, pp. D711–D715.

Azevedo, Hátylas and Carlos Alberto Moreira-Filho (2015). "Topological robustness analysis of protein interaction networks reveals key targets for overcoming chemotherapy resistance in glioma". In: *Scientific Reports* 5, p. 16830.

Bader, Gary D and Christopher WV Hogue (2003). "An automated method for finding molecular complexes in large protein interaction networks". In: *BMC Bioinformatics* 4.1, p. 2.

Bajusz, Dávid, Anita Rácz, and Károly Héberger (2015). "Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations?" In: *Journal of cheminformatics* 7.1, p. 20.

Ballouz, Sara, Wim Verleyen, and Jesse Gillis (2015). "Guidance for RNA-seq co-expression network construction and analysis: safety in numbers". In: *Bioinformatics* 31.13, pp. 2123–2130.

Barabási, Albert-László and Réka Albert (1999). "Emergence of scaling in random networks". In: *Science* 286.5439, pp. 509–512.

Barabasi, Albert-Laszlo and Zoltan N Oltvai (2004). "Network biology: understanding the cell's functional organization". In: *Nature Reviews Genetics* 5.2, p. 101.

Barido-Sottani, Joëlle, Samuel D Chapman, Evsey Kosman, and Arcady R Mushegian (2019). "Measuring similarity between gene interaction profiles". In: *BMC Bioinformatics* 20.1, pp. 1–13.

Berggård, Tord, Sara Linse, and Peter James (2007). "Methods for the detection and analysis of protein–protein interactions". In: *Proteomics* 7.16, pp. 2833–2842.

Berman, Helen M, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne (Jan. 2000). "The Protein Data Bank". In: *Nucleic Acids Research* 28.1, pp. 235–242. eprint: `http://oup.prod.sis.lan/nar/article-pdf/28/1/235/9895144/280235.pdf`. URL: `https://doi.org/10.1093/nar/28.1.235`.

Blondel, Vincent D, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre (2008). "Fast unfolding of communities in large networks". In: *Journal of Statistical Mechanics: Theory and Experiment* 2008.10, P10008.

Bonacich, Phillip (1987). "Power and centrality: A family of measures". In: *American Journal of Sociology* 92.5, pp. 1170–1182.

Bonifacino, Juan S, Esteban C Dell'Angelica, and Timothy A Springer (1999). "Immunoprecipitation". In: *Current Protocols in Protein Science* 18.1, pp. 9–8.

Borgatti, Stephen P (1997). "Structural holes: Unpacking Burt's redundancy measures". In: *Connections* 20.1, pp. 35–38.

Boyer, Paul D (1997). "The ATP synthase—a splendid molecular machine". In: *Annual Review of Biochemistry* 66.1, pp. 717–749.

Bozhilova, Lyuba V, Alan V Whitmore, Jonny Wray, Gesine Reinert, and Charlotte M Deane (2019). "Measuring rank robustness in scored protein interaction networks". In: *BMC Bioinformatics* 20.1, p. 446.

Brin, Sergey and Lawrence Page (1998). "The anatomy of a large-scale hypertextual web search engine". In: *Computer Networks and ISDN Systems* 30.1-7, pp. 107–117.

Broido, Anna D and Aaron Clauset (2019). "Scale-free networks are rare". In: *Nature Communications* 10.1, p. 1017.

Burt, Ronald S (2009). *Structural holes: The social structure of competition*. Harvard University Press.

Butte, Atul J and Isaac S Kohane (1999). "Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements". In: *Biocomputing 2000*. World Scientific, pp. 418–429.

Caspi, Ron, Hartmut Foerster, Carol A Fulcher, Rebecca Hopkinson, John Ingraham, Pallavi Kaipa, Markus Krummenacker, Suzanne Paley, John Pick, Seung Y Rhee, et al. (2006). "MetaCyc: a multiorganism database of metabolic pathways and enzymes". In: *Nucleic Acids Research* 34.suppl_1, pp. D511–D516.

Chatr-Aryamontri, Andrew, Rose Oughtred, Lorrie Boucher, Jennifer Rust, Christie Chang, Nadine K Kolas, Lara O'Donnell, Sara Oster, Chandra Theesfeld, Adnane Sellam, et al. (2017). "The BioGRID interaction database: 2017 update". In: *Nucleic Acids Research* 45.D1, pp. D369–D379.

Cherry, J Michael, Eurie L Hong, Craig Amundsen, Rama Balakrishnan, Gail Binkley, Esther T Chan, Karen R Christie, Maria C Costanzo, Selina S Dwight, Stacia R Engel, et al. (2011). "Saccharomyces Genome Database: the genomics resource of budding yeast". In: *Nucleic Acids Research* 40.D1, pp. D700–D705.

Choi, Hyungwon, Brett Larsen, Zhen-Yuan Lin, Ashton Breitkreutz, Dattatreya Mellacheruvu, Damian Fermin, Zhaohui S Qin, Mike Tyers, Anne-Claude Gingras, and Alexey I Nesvizhskii (2011). "SAINT: probabilistic scoring of affinity purification–mass spectrometry data". In: *Nature Methods* 8.1, p. 70.

Choudum, Sheshayya A (1986). "A simple proof of the Erdos-Gallai theorem on graph sequences". In: *Bulletin of the Australian Mathematical Society* 33.1, pp. 67–70.

Chung, Fan and Linyuan Lu (2002a). "Connected components in random graphs with given expected degree sequences". In: *Annals of Combinatorics* 6.2, pp. 125–145.

— (2002b). "The average distances in random graphs with given expected degrees". In: *Proceedings of the National Academy of Sciences* 99.25, pp. 15879–15882.

Clauset, Aaron, Cristopher Moore, and Mark EJ Newman (2008). "Hierarchical structure and the prediction of missing links in networks". In: *Nature* 453.7191, p. 98.

Clore, G Marius and Angela M Gronenborn (1998). "Determining the structures of large proteins and protein complexes by NMR". In: *Trends in Biotechnology* 16.1, pp. 22–34.

Cong, Qian, Ivan Anishchenko, Sergey Ovchinnikov, and David Baker (2019). "Protein interaction networks revealed by proteome coevolution". In: *Science* 365.6449, pp. 185–189.

Cosgrove, Elissa J, Timothy S Gardner, and Eric D Kolaczyk (2010). "On the choice and number of microarrays for transcriptional regulatory network inference". In: *BMC Bioinformatics* 11.1, p. 454.

Croft, David, Antonio Fabregat Mundo, Robin Haw, Marija Milacic, Joel Weiser, Guanming Wu, Michael Caudy, Phani Garapati, Marc Gillespie, Maulik R Kamdar, et al. (2013). "The Reactome pathway knowledgebase". In: *Nucleic Acids Research* 42.D1, pp. D472–D477.

Csardi, Gabor, Tamas Nepusz, et al. (2006). "The igraph software package for complex network research". In: *InterJournal, Complex Systems* 1695.5, pp. 1–9.

Curto, Carina (2017). "What can topology tell us about the neural code?" In: *Bulletin of the American Mathematical Society* 54.1, pp. 63–78.

D'haeseleer, Patrik, Shoudan Liang, and Roland Somogyi (2000). "Genetic network inference: from co-expression clustering to reverse engineering". In: *Bioinformatics* 16.8, pp. 707–726.

Dandekar, Thomas, Berend Snel, Martijn Huynen, and Peer Bork (1998). "Conservation of gene order: a fingerprint of proteins that physically interact". In: *Trends in Biochemical Sciences* 23.9, pp. 324–328.

De Las Rivas, Javier and Celia Fontanillo (2010). "Protein–protein interactions essentials: key concepts to building and analyzing interactome networks". In: *PLoS Computational Biology* 6.6, e1000807.

De Smet, Riet and Kathleen Marchal (2010). "Advantages and limitations of current network inference methods". In: *Nature Reviews Microbiology* 8.10, p. 717.

Dunham, Wade H, Michael Mullin, and Anne-Claude Gingras (2012). "Affinity-purification coupled to mass spectrometry: Basic principles and strategies". In: *Proteomics* 12.10, pp. 1576–1590.

Eberwine, James, Jai-Yoon Sul, Tamas Bartfai, and Junhyong Kim (2014). "The promise of single-cell sequencing". In: *Nature Methods* 11.1, p. 25.

Edfors, Fredrik, Frida Danielsson, Björn M Hallström, Lukas Käll, Emma Lundberg, Fredrik Pontén, Björn Forsström, and Mathias Uhlén (2016). "Gene-specific correlation of RNA and protein levels in human cells and tissues". In: *Molecular Systems Biology* 12.10.

Edwards, Aled M, Bart Kus, Ronald Jansen, Dov Greenbaum, Jack Greenblatt, and Mark Gerstein (2002). "Bridging structural biology and genomics: assessing protein interaction data with known complexes". In: *Trends in Genetics* 18.10, pp. 529–536.

Enright, Anton J, Ioannis Iliopoulos, Nikos C Kyrpides, and Christos A Ouzounis (1999). "Protein interaction maps for complete genomes based on gene fusion events". In: *Nature* 402.6757, p. 86.

Epanechnikov, Vassiliy A (1969). "Non-parametric estimation of a multivariate probability density". In: *Theory of Probability & Its Applications* 14.1, pp. 153–158.

Erdős, Paul and Alfréd Rényi (1960). "On the evolution of random graphs". In: *Publications of the Mathematical Institute of the Hungarian Academy of Sciences* 5.1, pp. 17–60.

— (1961). "On the strength of connectedness of a random graph". In: *Acta Mathematica Hungarica* 12.1-2, pp. 261–267.

Estrada, Ernesto (2000). "Characterization of 3D molecular structure". In: *Chemical Physics Letters* 319.5-6, pp. 713–718.

Eubel, Holger, Hans-Peter Braun, and A. Harvey Millar (2005). "Blue-native PAGE in plants: a tool in analysis of protein-protein interactions". In: *Plant Methods* 1.1, p. 11.

Faust, Karoline and Jeroen Raes (2012). "Microbial interactions: from networks to models". In: *Nature Reviews Microbiology* 10.8, p. 538.

Fields, Stanley and Ok-kyu Song (1989). "A novel genetic system to detect protein–protein interactions". In: *Nature* 340.6230, pp. 245–246.

Franz, Max, Harold Rodriguez, Christian Lopes, Khalid Zuberi, Jason Montojo, Gary D Bader, and Quaid Morris (2018). "GeneMANIA update 2018". In: *Nucleic Acids Research* 46.W1, W60–W64.

Freeman, Linton C (1977). "A set of measures of centrality based on betweenness". In: *Sociometry*, pp. 35–41.

— (1978). "Centrality in social networks: Conceptual clarification". In: *Social Networks* 1.3, pp. 215–239.

Galperin, Michael Y and Eugene V Koonin (2000). "Who's your neighbor? New computational approaches for functional genomics". In: *Nature Biotechnology* 18.6, p. 609.

Gao, Chuan, Ian C McDowell, Shiwen Zhao, Christopher D Brown, and Barbara E Engelhardt (2016). "Context specific and differential gene co-expression networks via Bayesian biclustering". In: *PLoS Computational Biology* 12.7, e1004791.

Gene Ontology Consortium (2018). "The gene ontology resource: 20 years and still GOing strong". In: *Nucleic Acids Research* 47.D1, pp. D330–D338.

Gerstein, Mark B, Can Bruce, Joel S Rozowsky, Deyou Zheng, Jiang Du, Jan O Korbel, Olof Emanuelsson, Zhengdong D Zhang, Sherman Weissman, and Michael Snyder

(2007). "What is a gene, post-ENCODE? History and updated definition". In: *Genome Research* 17.6, pp. 669–681.

Gething, Mary-Jane and Joseph Sambrook (1992). "Protein folding in the cell". In: *Nature* 355.6355, p. 33.

Giaever, Guri, Angela M Chu, Li Ni, Carla Connelly, Linda Riles, Steeve Veronneau, Sally Dow, Ankuta Lucau-Danila, Keith Anderson, Bruno Andre, et al. (2002). "Functional profiling of the Saccharomyces cerevisiae genome". In: *Nature* 418.6896, p. 387.

Gillis, Jesse and Paul Pavlidis (2011). "The role of indirect connections in gene networks in predicting function". In: *Bioinformatics* 27.13, pp. 1860–1866.

Girvan, Michelle and Mark EJ Newman (2002). "Community structure in social and biological networks". In: *Proceedings of the National Academy of Sciences* 99.12, pp. 7821–7826.

Golub, Todd R, Donna K Slonim, Pablo Tamayo, Christine Huard, Michelle Gaasenbeek, Jill P Mesirov, Hilary Coller, Mignon L Loh, James R Downing, Mark A Caligiuri, et al. (1999). "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring". In: *Science* 286.5439, pp. 531–537.

González-Couto, Eduardo (2011). "Functional and systems biology approaches to Huntington's disease". In: *Briefings in Functional Genomics* 10.3, pp. 109–114.

Gonzalez-Valbuena, Elpidio-Emmanuel and Víctor Treviño (2017). "Metrics to estimate differential co-expression networks". In: *BioData Mining* 10.1, p. 32.

Grishin, Alexander V, Natalia V Lavrova, Alexander M Lyashchuk, Natalia V Strukova, Maria S Generalova, Anna V Ryazanova, Nikita V Shestak, Irina S Boksha, Nikita B Polyakov, Zoya M Galushkina, et al. (2019). "The Influence of Dimerization on the Pharmacokinetics and Activity of an Antibacterial Enzyme Lysostaphin". In: *Molecules* 24.10, p. 1879.

Guimerà, Roger and Marta Sales-Pardo (2009). "Missing and spurious interactions and the reconstruction of complex networks". In: *Proceedings of the National Academy of Sciences* 106.52, pp. 22073–22078.

Guttman, Mitchell, Julie Donaghey, Bryce W Carey, Manuel Garber, Jennifer K Grenier, Glen Munson, Geneva Young, Anne Bergstrom Lucas, Robert Ach, Laurakay Bruhn, et al. (2011). "lincRNAs act in the circuitry controlling pluripotency and differentiation". In: *Nature* 477.7364, p. 295.

Hamer, Rebecca, Qiang Luo, Judith P Armitage, Gesine Reinert, and Charlotte M Deane (2010). "i-Patch: Interprotein contact prediction using local network information". In: *Proteins: Structure, Function, and Bioinformatics* 78.13, pp. 2781–2797.

Han, Lu, Kang Li, Chaozhi Jin, Jian Wang, Qingjun Li, Qiling Zhang, Qiyue Cheng, Jing Yang, Xiaochen Bo, and Shengqi Wang (2017). "Human enterovirus 71 protein interaction network prompts antiviral drug repositioning". In: *Scientific Reports* 7, p. 43143.

Hart, G Traver, Arun K Ramani, and Edward M Marcotte (2006). "How complete are current yeast and human protein-interaction networks?" In: *Genome Biology* 7.11, p. 120.

Hartwell, Leland H, John J Hopfield, Stanislas Leibler, and Andrew W Murray (1999). "From molecular to modular cell biology". In: *Nature* 402.6761supp, p. C47.

Hase, Takeshi, Hiroshi Tanaka, Yasuhiro Suzuki, So Nakagawa, and Hiroaki Kitano (2009). "Structure of protein interaction networks and their implications on drug design". In: *PLoS Computational Biology* 5.10, e1000550.

Hillis, David M, David E Sadava, H Craig Heller, and Mary V Price (2012). *Principles of life*. Macmillan.

Hishigaki, Haretsugu, Kenta Nakai, Toshihide Ono, Akira Tanigami, and Toshihisa Takagi (2001). "Assessment of prediction accuracy of protein function from protein–protein interaction data". In: *Yeast* 18.6, pp. 523–531.

Holland, Paul W, Kathryn Blackmond Laskey, and Samuel Leinhardt (1983). "Stochastic blockmodels: First steps". In: *Social Networks* 5.2, pp. 109–137.

Hopkins, Andrew L (2008). "Network pharmacology: the next paradigm in drug discovery". In: *Nature Chemical Biology* 4.11, p. 682.

Huang, Hailiang and Joel S Bader (2009). "Precision and recall estimates for two-hybrid screens". In: *Bioinformatics* 25.3, pp. 372–378.

Huang, Hailiang, Bruno M Jedynak, and Joel S Bader (2007). "Where have all the interactions gone? Estimating the coverage of two-hybrid protein interaction maps". In: *PLoS Computational Biology* 3.11, e214.

Huang, Justin K, Daniel E Carlin, Michael Ku Yu, Wei Zhang, Jason F Kreisberg, Pablo Tamayo, and Trey Ideker (2018). "Systematic evaluation of molecular networks for discovery of disease genes". In: *Cell Systems* 6.4, pp. 484–495.

Hughes, Julian R, Ana M Meireles, Katherine H Fisher, Angel Garcia, Philip R Antrobus, Alan Wainman, Nicole Zitzmann, Charlotte Deane, Hiroyuki Ohkura, and James G Wakefield (2008). "A microtubule interactome: complexes with roles in cell cycle and mitosis". In: *PLoS Biology* 6.4, e98.

Ideker, Trey and Roded Sharan (2008). "Protein networks in disease". In: *Genome Research* 18.4, pp. 644–652.

Ispolatov, Iaroslav, PL Krapivsky, and Anton Yuryev (2005). "Duplication–divergence model of protein interaction network". In: *Physical Review E* 71.6, p. 061911.

Jaccard, Paul (1908). "Nouvelles recherches sur la distribution florale". In: *Bull. Soc. Vaud. Sci. Nat.* 44, pp. 223–270.

Jones, Susan and Janet M Thornton (1996). "Principles of protein-protein interactions". In: *Proceedings of the National Academy of Sciences* 93.1, pp. 13–20.

Jun, Wu, Mauricio Barahona, Tan Yue-Jin, and Deng Hong-Zhong (2010). "Natural connectivity of complex networks". In: *Chinese Physics Letters* 27.7, p. 078902.

Kaboord, Barbara and Maria Perr (2008). "Isolation of proteins and protein complexes by immunoprecipitation". In: *2D PAGE: sample preparation and fractionation*. Springer, pp. 349–364.

Kaiser, Peter, David Meierhofer, Xiaorong Wang, and Lan Huang (2008). "Tandem affinity purification combined with mass spectrometry to identify components of protein complexes". In: *Genomics Protocols*. Springer, pp. 309–326.

Kanehisa, Minoru and Susumu Goto (2000). "KEGG: Kyoto encyclopedia of genes and genomes". In: *Nucleic Acids Research* 28.1, pp. 27–30.

Kao, Ta-Chu and Mason A Porter (2018). "Layer communities in multiplex networks". In: *Journal of Statistical Physics* 173.3-4, pp. 1286–1302.

Karlebach, Guy and Ron Shamir (2008). "Modelling and analysis of gene regulatory networks". In: *Nature Reviews Molecular Cell Biology* 9.10, p. 770.

Katz, Leo (1953). "A new status index derived from sociometric analysis". In: *Psychometrika* 18.1, pp. 39–43.

Kendrew, John C, G Bodo, Howard M Dintzis, RG Parrish, and Harold Wyckoff (1958). "A three-dimensional model of the myoglobin molecule obtained by x-ray analysis". In: *Nature* 181.4610, pp. 662–666.

Khatter, Heena, Alexander G Myasnikov, S Kundhavai Natchiar, and Bruno P Klaholz (2015). "Structure of the human 80S ribosome". In: *Nature* 520.7549, p. 640.

Killworth, Peter D, Eugene C Johnsen, H Russell Bernard, Gene Ann Shelley, and Christopher McCarty (1990). "Estimating the size of personal networks". In: *Social Networks* 12.4, pp. 289–312.

Klingström, Tomas and Dariusz Plewczynski (2010). "Protein–protein interaction and pathway databases, a graphical review". In: *Briefings in Bioinformatics* 12.6, pp. 702–713.

Krogan, Nevan J, Gerard Cagney, Haiyuan Yu, Gouqing Zhong, Xinghua Guo, Alexandr Ignatchenko, Joyce Li, Shuye Pu, Nira Datta, Aaron P Tikuisis, et al. (2006). "Global landscape of protein complexes in the yeast Saccharomyces cerevisiae". In: *Nature* 440.7084, p. 637.

Kustatscher, Georg, Piotr Grabowski, Tina A Schrader, Josiah B Passmore, Michael Schrader, and Juri Rappsilber (2019). "Co-regulation map of the human proteome enables identification of protein functions". In: *Nature Biotechnology* 37.11, pp. 1361–1371.

Langfelder, Peter and Steve Horvath (2008). "WGCNA: an R package for weighted correlation network analysis". In: *BMC Bioinformatics* 9.1, p. 559.

Lee, Homin K, Amy K Hsu, Jon Sajdak, Jie Qin, and Paul Pavlidis (2004). "Coexpression analysis of human genes across many microarray data sets". In: *Genome Research* 14.6, pp. 1085–1094.

Lengyel, Peter and Dieter Söll (1969). "Mechanism of protein biosynthesis." In: *Bacteriological Reviews* 33.2, p. 264.

Li, Yong and Pankaj Agarwal (2009). "A pathway-based view of human diseases and disease relationships". In: *PloS One* 4.2, e4346.

Liben-Nowell, David and Jon Kleinberg (2007). "The link-prediction problem for social networks". In: *Journal of the American society for Information Science and Technology* 58.7, pp. 1019–1031.

Lodish, Harvey, Arnold Berk, Chris A Kaiser, Monty Krieger, Matthew P Scott, Anthony Bretscher, Hidde Ploegh, Paul Matsudaira, et al. (2008). *Molecular cell biology*. Macmillan.

Lonsdale, John, Jeffrey Thomas, Mike Salvatore, Rebecca Phillips, Edmund Lo, Saboor Shad, Richard Hasz, Gary Walters, Fernando Garcia, Nancy Young, et al. (2013). "The genotype-tissue expression (GTEx) project". In: *Nature Genetics* 45.6, p. 580.

López, Yosvany, Kenta Nakai, and Ashwini Patil (2015). "HitPredict version 4: comprehensive reliability scoring of physical protein–protein interactions from more than 100 species". In: *Database: the Journal of Biological Databases and Curation* 2015.

Lowe, Rohan, Neil Shirley, Mark Bleackley, Stephen Dolan, and Thomas Shafee (2017). "Transcriptomics technologies". In: *PLoS Computational Biology* 13.5, e1005457.

Luck, Katja, Dae Kyum Kim, Luke Lambourne, Kerstin Spirohn, Bridget E Begg, Wenting Bian, Ruth Brignall, Tiziana Cafarelli, Francisco J Campos-Laborie, Benoit Charloteaux, et al. (2019). "A reference map of the human protein interactome". In: *bioRxiv*, p. 605451.

Luecken, Malte D, Matthew JT Page, Andrea J Crosby, Sean Mason, Gesine Reinert, and Charlotte M Deane (2017). "CommWalker: correctly evaluating modules in molecular networks in light of annotation bias". In: *Bioinformatics* 34.6, pp. 994–1000.

Machnicka, Magdalena A, Kaja Milanowska, Okan Osman Oglou, Elzbieta Purta, Malgorzata Kurkowska, Anna Olchowik, Witold Januszewski, Sebastian Kalinowski, Stanislaw Dunin-Horkawicz, Kristian M Rother, et al. (2012). "MODOMICS: a database of RNA modification pathways—2013 update". In: *Nucleic Acids Research* 41.D1, pp. D262–D267.

Magadum, Santoshkumar, Urbi Banerjee, Priyadharshini Murugan, Doddabhimappa Gangapur, and Rajasekar Ravikesavan (2013). "Gene duplication as a major force in evolution". In: *Journal of Genetics* 92.1, pp. 155–161.

Mani, Ramamurthy, Robert P St Onge, John L Hartman, Guri Giaever, and Frederick P Roth (2008). "Defining genetic interaction". In: *Proceedings of the National Academy of Sciences* 105.9, pp. 3461–3466.

Mann, Matthias, Ronald C Hendrickson, and Akhilesh Pandey (2001). "Analysis of proteins and proteomes by mass spectrometry". In: *Annual Review of Biochemistry* 70.1, pp. 437–473.

Mann, Matthias and Ole N Jensen (2003). "Proteomic analysis of post-translational modifications". In: *Nature Biotechnology* 21.3, p. 255.

Marchiori, Massimo and Vito Latora (2000). "Harmony in the small-world". In: *Physica A: Statistical Mechanics and its Applications* 285.3-4, pp. 539–546.

Marcotte, Edward M, Matteo Pellegrini, Ho-Leung Ng, Danny W Rice, Todd O Yeates, and David Eisenberg (1999). "Detecting protein function and protein-protein interactions from genome sequences". In: *Science* 285.5428, pp. 751–753.

Martin, Travis, Brian Ball, and Mark EJ Newman (2016). "Structural inference for uncertain networks". In: *Physical Review E* 93.1, p. 012306.

Mason, Mike J, Guoping Fan, Kathrin Plath, Qing Zhou, and Steve Horvath (2009). "Signed weighted gene co-expression network analysis of transcriptional regulation in murine embryonic stem cells". In: *BMC Genomics* 10.1, p. 327.

McDowall, Mark D, Michelle S Scott, and Geoffrey J Barton (2008). "PIPs: human protein–protein interaction prediction database". In: *Nucleic acids research* 37.suppl_1, pp. D651–D656.

Miernyk, Jan A and Jay J Thelen (2008). "Biochemical approaches for discovering protein–protein interactions". In: *The Plant Journal* 53.4, pp. 597–609.

Milgram, Stanley (1967). "The small world problem". In: *Psychology Today* 2.1, pp. 60–67.

Milo, Ron (2013). "What is the total number of protein molecules per cell volume? A call to rethink some published values". In: *Bioessays* 35.12, pp. 1050–1055.

Milo, Ron, Shai Shen-Orr, Shalev Itzkovitz, Nadav Kashtan, Dmitri Chklovskii, and Uri Alon (2002). "Network motifs: simple building blocks of complex networks". In: *Science* 298.5594, pp. 824–827.

Mitelman, Felix, Bertil Johansson, and Fredrik Mertens (2007). "The impact of translocations and gene fusions on cancer causation". In: *Nature Reviews Cancer* 7.4, p. 233.

Mitzenmacher, Michael (2004). "A brief history of generative models for power law and lognormal distributions". In: *Internet Mathematics* 1.2, pp. 226–251.

Moellering, Raymond E, Melanie Cornejo, Tina N Davis, Cristina Del Bianco, Jon C Aster, Stephen C Blacklow, Andrew L Kung, D Gary Gilliland, Gregory L Verdine, and James E Bradner (2009). "Direct inhibition of the NOTCH transcription factor complex". In: *Nature* 462.7270, p. 182.

Molloy, Michael and Bruce Reed (1995). "A critical point for random graphs with a given degree sequence". In: *Random Structures & Algorithms* 6.2-3, pp. 161–180.

Mosca, Roberto, Arnaud Ceol, Amelie Stein, Roger Olivella, and Patrick Aloy (2013). "3did: a catalog of domain-based interactions of known three-dimensional structure". In: *Nucleic Acids Research* 42.D1, pp. D374–D379.

Navlakha, Saket and Carl Kingsford (2010). "The power of protein interaction networks for associating genes with diseases". In: *Bioinformatics* 26.8, pp. 1057–1063.

Neves, Susana R, Prahlad T Ram, and Ravi Iyengar (2002). "G protein pathways". In: *Science* 296.5573, pp. 1636–1639.

Newman, Mark EJ (2017). "Measurement errors in network data". In: *arXiv preprint arXiv:1703.07376*.

— (2018a). "Network structure from rich but noisy data". In: *Nature Physics* 14.6, p. 542.

— (2018b). *Networks*. Oxford University Press.

Nilsen, Timothy W and Brenton R Graveley (2010). "Expansion of the eukaryotic proteome by alternative splicing". In: *Nature* 463.7280, p. 457.

Nooren, Irene MA and Janet M Thornton (2003). "Diversity of protein–protein interactions". In: *The EMBO journal* 22.14, pp. 3486–3492.

Novick, Peter, Barbara C Osmond, and David Botstein (1989). "Suppressors of yeast actin mutations". In: *Genetics* 121.4, pp. 659–674.

Obayashi, Takeshi, Yuichi Aoki, Shu Tadaka, Yuki Kagaya, and Kengo Kinoshita (2017). "ATTED-II in 2018: a plant coexpression database based on investigation of the statistical property of the mutual rank index". In: *Plant and Cell Physiology* 59.1, e3–e3.

Obayashi, Takeshi, Yuki Kagaya, Yuichi Aoki, Shu Tadaka, and Kengo Kinoshita (2018). "COXPRESdb v7: a gene coexpression database for 11 animal species supported by 23 coexpression platforms for technical evaluation and evolutionary inference". In: *Nucleic Acids Research* 47.D1, pp. D55–D62.

Olhede, Sofia C and Patrick J Wolfe (2014). "Network histograms and universality of blockmodel approximation". In: *Proceedings of the National Academy of Sciences* 111.41, pp. 14722–14727.

Orchard, Sandra, Samuel Kerrien, Sara Abbani, Bruno Aranda, Jignesh Bhate, Shelby Bidwell, Alan Bridge, Leonardo Briganti, Fiona SL Brinkman, Gianni Cesareni, et al. (2012). "Protein interaction data curation: the International Molecular Exchange (IMEx) consortium". In: *Nature Methods* 9.4, p. 345.

Ospina-Forero, Luis, Charlotte M Deane, and Gesine Reinert (2018). "Assessment of model fit via network comparison methods based on subgraph counts". In: *Journal of Complex Networks* 7.2, pp. 226–253.

Oughtred, Rose, Chris Stark, Bobby-Joe Breitkreutz, Jennifer Rust, Lorrie Boucher, Christie Chang, Nadine Kolas, Lara O'Donnell, Genie Leung, Rochelle McAdam, et al. (2018). "The BioGRID interaction database: 2019 update". In: *Nucleic Acids Research* 47.D1, pp. D529–D541.

Overbeek, Ross, Michael Fonstein, Mark D'souza, Gordon D Pusch, and Natalia Maltsev (1999). "The use of gene clusters to infer functional coupling". In: *Proceedings of the National Academy of Sciences* 96.6, pp. 2896–2901.

Palleros, Daniel R, Li Shi, Katherine L Reid, and Anthony L Fink (1994). "HSP70-protein complexes. Complex stability and conformation of bound substrate protein." In: *Journal of Biological Chemistry* 269.18, pp. 13107–13114.

Papatheodorou, Irene, Nuno A Fonseca, Maria Keays, Y Amy Tang, Elisabet Barrera, Wojciech Bazant, Melissa Burke, Anja Füllgrabe, Alfonso Muñoz-Pomer Fuentes, Nancy George, et al. (2017). "Expression Atlas: gene and protein expression across multiple studies and organisms". In: *Nucleic Acids Research* 46.D1, pp. D246–D251.

Park, Taesung, Sung-Gon Yi, Sung-Hyun Kang, SeungYeoun Lee, Yong-Sung Lee, and Richard Simon (2003). "Evaluation of normalization methods for microarray data". In: *BMC Bioinformatics* 4.1, p. 33.

Parsana, Princy, Claire Ruberman, Andrew E Jaffe, Michael C Schatz, Alexis Battle, and Jeffrey T Leek (2019). "Addressing confounding artifacts in reconstruction of gene co-expression networks". In: *Genome Biology* 20.1, p. 94.

Patil, Ashwini and Haruki Nakamura (2005). "HINT: a database of annotated protein-protein interactions and their homologs". In: *Biophysics* 1, pp. 21–24.

Pease, A Caviani, Dennis Solas, Edward J Sullivan, Maureen T Cronin, Christopher P Holmes, and SP Fodor (1994). "Light-generated oligonucleotide arrays for rapid DNA sequence analysis". In: *Proceedings of the National Academy of Sciences* 91.11, pp. 5022–5026.

Peixoto, Tiago P (2018). "Reconstructing networks with unknown and heterogeneous errors". In: *Physical Review X* 8.4, p. 041011.

Pereira-Leal, Jose B, Anton J Enright, and Christos A Ouzounis (2004). "Detection of functional modules from protein interaction networks". In: *Proteins: Structure, Function, and Bioinformatics* 54.1, pp. 49–57.

Perkins, James R, Ilhem Diboun, Benoit H Dessailly, Jon G Lees, and Christine Orengo (2010). "Transient protein-protein interactions: structural, functional, and network properties". In: *Structure* 18.10, pp. 1233–1243.

Petryszak, Robert, Maria Keays, Y Amy Tang, Nuno A Fonseca, Elisabet Barrera, Tony Burdett, Anja Füllgrabe, Alfonso Muñoz-Pomer Fuentes, Simon Jupp, Satu Koskinen, et al. (2015). "Expression Atlas update—an integrated database of gene and protein expression in humans, animals and plants". In: *Nucleic Acids Research* 44.D1, pp. D746–D752.

Pržulj, Nataša (2007). "Biological network comparison using graphlet degree distribution". In: *Bioinformatics* 23.2, e177–e183.

Puig, Oscar, Friederike Caspary, Guillaume Rigaut, Berthold Rutz, Emmanuelle Bouveret, Elisabeth Bragado-Nilsson, Matthias Wilm, and Bertrand Séraphin (2001). "The tandem affinity purification (TAP) method: a general procedure of protein complex purification". In: *Methods* 24.3, pp. 218–229.

Qin, Jun, Olga Vinogradova, and Angela M Gronenborn (2001). "Protein-protein interactions probed by nuclear magnetic resonance spectroscopy." In: *Methods in enzymology* 339, pp. 377–389.

R Core Team (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. URL: `https://www.R-project.org/`.

Rabbitts, Terence H (1994). "Chromosomal translocations in human cancer". In: *Nature* 372.6502, p. 143.

Rao, V Srinivasa, K Srinivas, GN Sujini, and GN Kumar (2014). "Protein–protein interaction detection: methods and analysis". In: *International Journal of Proteomics* 2014.

Rasmussen, Søren GF, Brian T DeVree, Yaozhong Zou, Andrew C Kruse, Ka Young Chung, Tong Sun Kobilka, Foon Sun Thian, Pil Seok Chae, Els Pardon,

Diane Calinski, et al. (2011). "Crystal structure of the β 2 adrenergic receptor–Gs protein complex". In: *Nature* 477.7366, p. 549.

Ravasz, Erzsébet, Anna Lisa Somera, Dale A Mongru, Zoltán N Oltvai, and A-L Barabási (2002). "Hierarchical organization of modularity in metabolic networks". In: *Science* 297.5586, pp. 1551–1555.

Rigaut, Guillaume, Anna Shevchenko, Berthold Rutz, Matthias Wilm, Matthias Mann, and Bertrand Séraphin (1999). "A generic protein purification method for protein complex characterization and proteome exploration". In: *Nature Biotechnology* 17.10, p. 1030.

Robins, Garry, Tom Snijders, Peng Wang, Mark Handcock, and Philippa Pattison (2007). "Recent developments in exponential random graph (p*) models for social networks". In: *Social Networks* 29.2, pp. 192–215.

Robinson, Mark D, Davis J McCarthy, and Gordon K Smyth (2010). "edgeR: a Bioconductor package for differential expression analysis of digital gene expression data". In: *Bioinformatics* 26.1, pp. 139–140.

Rolland, Thomas, Murat Taşan, Benoit Charloteaux, Samuel J Pevzner, Quan Zhong, Nidhi Sahni, Song Yi, Irma Lemmens, Celia Fontanillo, Roberto Mosca, et al. (2014). "A proteome-scale map of the human interactome network". In: *Cell* 159.5, pp. 1212–1226.

Rose, Alexander S, Anthony R Bradley, Yana Valasatava, Jose M Duarte, Andreas Prlić, and Peter W Rose (2018). "NGL viewer: web-based molecular graphics for large complexes". In: *Bioinformatics* 34.21, pp. 3755–3758.

Ross, Nathan et al. (2011). "Fundamentals of Stein's method". In: *Probability Surveys* 8, pp. 210–293.

Saha, Ashis, Yungil Kim, Ariel DH Gewirtz, Brian Jo, Chuan Gao, Ian C McDowell, Barbara E Engelhardt, Alexis Battle, François Aguet, Kristin G Ardlie, et al. (2017). "Co-expression networks reveal the tissue-specific regulation of transcription and splicing". In: *Genome Research* 27.11, pp. 1843–1858.

Salathé, Marcel and James H Jones (2010). "Dynamics and control of diseases in networks with community structure". In: *PLoS Computational Biology* 6.4, e1000736.

Saliba, Antoine-Emmanuel, Alexander J Westermann, Stanislaw A Gorski, and Jörg Vogel (2014). "Single-cell RNA-seq: advances and future challenges". In: *Nucleic Acids Research* 42.14, pp. 8845–8860.

Schäfer, Juliane and Korbinian Strimmer (2004). "An empirical Bayes approach to inferring large-scale gene association networks". In: *Bioinformatics* 21.6, pp. 754–764.

Schägger, Hermann and Gebhard von Jagow (1991). "Blue native electrophoresis for isolation of membrane protein complexes in enzymatically active form". In: *Analytical Biochemistry* 199.2, pp. 223–231.

Schena, Mark, Dari Shalon, Ronald W Davis, and Patrick O Brown (1995). "Quantitative monitoring of gene expression patterns with a complementary DNA microarray". In: *Science* 270.5235, pp. 467–470.

Sciuto, Maria Rita, Uwe Warnken, Martina Schnölzer, Cecilia Valvo, Lidia Brunetto, Alessandra Boe, Mauro Biffoni, Peter H Krammer, Ruggero De Maria, and Tobias L Haas (2018). "Two-step coimmunoprecipitation (TIP) enables efficient and highly selective isolation of native protein complexes". In: *Molecular & Cellular Proteomics* 17.5, pp. 993–1009.

Segal, Eran, Nir Friedman, Daphne Koller, and Aviv Regev (2004). "A module map showing conditional activity of expression modules in cancer". In: *Nature Genetics* 36.10, p. 1090.

Sender, Ron, Shai Fuchs, and Ron Milo (2016). "Revised estimates for the number of human and bacteria cells in the body". In: *PLoS Biology* 14.8, e1002533.

Shannon, Paul, Andrew Markiel, Owen Ozier, Nitin S Baliga, Jonathan T Wang, Daniel Ramage, Nada Amin, Benno Schwikowski, and Trey Ideker (2003). "Cytoscape: a software environment for integrated models of biomolecular interaction networks". In: *Genome Research* 13.11, pp. 2498–2504.

Sharan, Roded, Igor Ulitsky, and Ron Shamir (2007). "Network-based prediction of protein function". In: *Molecular Systems Biology* 3.1, p. 88.

Shaw, A, Peter A Fortes, Charles D Stout, and Victor D Vacquier (1995). "Crystal structure and subunit dynamics of the abalone sperm lysin dimer: egg envelopes dissociate dimers, the monomer is the active species." In: *The Journal of Cell Biology* 130.5, pp. 1117–1125.

Silverman, Bernard W (1986). *Density Estimation for Statistics and Data Analysis*. Vol. 26. CRC Press.

Simonds, William F (1999). "G protein regulation of adenylate cyclase". In: *Trends in Pharmacological Sciences* 20.2, pp. 66–73.

Simpson, Sean L, Satoru Hayasaka, and Paul J Laurienti (2011). "Exponential random graph modeling for complex brain networks". In: *PloS One* 6.5, e20039.

Smith, Graham R and Michael JE Sternberg (2002). "Prediction of protein–protein interactions by docking methods". In: *Current Opinion in Structural Biology* 12.1, pp. 28–35.

Snider, Jamie, Saranya Kittanakom, Dunja Damjanovic, Jasna Curak, Victoria Wong, and Igor Stagljar (2010). "Detecting interactions with membrane proteins using a membrane two-hybrid assay in yeast". In: *Nature Protocols* 5.7, p. 1281.

Solomonoff, Ray and Anatol Rapoport (1951). "Connectivity of random nets". In: *The Bulletin of Mathematical Biophysics* 13.2, pp. 107–117.

Spielman, Daniel A (2007). "Spectral graph theory and its applications". In: *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*. IEEE, pp. 29–38.

Stewart, Adele and Rory A Fisher (2015). "Introduction: G protein-coupled receptors and RGS proteins". In: *Progress in Molecular Biology and Translational Science*. Vol. 133. Elsevier, pp. 1–11.

Stuart, Joshua M, Eran Segal, Daphne Koller, and Stuart K Kim (2003). "A gene-coexpression network for global discovery of conserved genetic modules". In: *science* 302.5643, pp. 249–255.

Stumpf, Michael PH (2019). "Multi-Model and Network Inference Based on Ensemble Estimates: Avoiding the Madness of Crowds". In: *bioRxiv*, p. 858308.

Stumpf, Michael PH, Thomas Thorne, Eric de Silva, Ronald Stewart, Hyeong Jun An, Michael Lappe, and Carsten Wiuf (2008). "Estimating the size of the human interactome". In: *Proceedings of the National Academy of Sciences* 105.19, pp. 6959–6964.

Subramanian, Aravind, Pablo Tamayo, Vamsi K Mootha, Sayan Mukherjee, Benjamin L Ebert, Michael A Gillette, Amanda Paulovich, Scott L Pomeroy, Todd R Golub, Eric S Lander, et al. (2005). "Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles". In: *Proceedings of the National Academy of Sciences* 102.43, pp. 15545–15550.

Szklarczyk, Damian, Andrea Franceschini, Stefan Wyder, Kristoffer Forslund, Davide Heller, Jaime Huerta-Cepas, Milan Simonovic, Alexander Roth, Alberto Santos, Kalliopi P Tsafou, et al. (2014). "STRING v10: protein–protein interaction networks, integrated over the tree of life". In: *Nucleic Acids Research* 43.D1, pp. D447–D452.

Szklarczyk, Damian, Annika L Gable, David Lyon, Alexander Junge, Stefan Wyder, Jaime Huerta-Cepas, Milan Simonovic, Nadezhda T Doncheva, John H Morris, Peer Bork, et al. (2018). "STRING v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets". In: *Nucleic Acids Research* 47.D1, pp. D607–D613.

Tanimoto, Taffee T (1957). "IBM internal report". In: *Nov* 17, p. 1957.

Taylor, Ian W and Jeffrey L Wrana (2012). "Protein interaction networks in medicine and disease". In: *Proteomics* 12.10, pp. 1706–1716.

Tong, Amy Hin Yan, Marie Evangelista, Ainslie B Parsons, Hong Xu, Gary D Bader, Nicholas Pagé, Mark Robinson, Sasan Raghibizadeh, Christopher WV Hogue, Howard Bussey, et al. (2001). "Systematic genetic analysis with ordered arrays of yeast deletion mutants". In: *Science* 294.5550, pp. 2364–2368.

Tong, Amy Hin Yan, Guillaume Lesage, Gary D Bader, Huiming Ding, Hong Xu, Xiaofeng Xin, James Young, Gabriel F Berriz, Renee L Brost, Michael Chang, et al. (2004). "Global mapping of the yeast genetic interaction network". In: *science* 303.5659, pp. 808–813.

Trajanovski, Stojan, Javier Martín-Hernández, Wynand Winterbach, and Piet Van Mieghem (2013). "Robustness envelopes of networks". In: *Journal of Complex Networks* 1.1, pp. 44–62.

Tu, Zhidong, Li Wang, Michelle N Arbeitman, Ting Chen, and Fengzhu Sun (2006). "An integrative approach for causal gene identification and gene regulatory pathway inference". In: *Bioinformatics* 22.14, e489–e496.

Uetz, Peter, Loic Giot, Gerard Cagney, Traci A Mansfield, Richard S Judson, James R Knight, Daniel Lockshon, Vaibhav Narayan, Maithreyan Srinivasan, Pascale Pochart, et al. (2000). "A comprehensive analysis of protein–protein interactions in Saccharomyces cerevisiae". In: *Nature* 403.6770, p. 623.

Vallès-Català, Toni, Tiago P Peixoto, Marta Sales-Pardo, and Roger Guimerà (2018). "Consistencies and inconsistencies between model selection and link prediction in networks". In: *Physical Review E* 97.6, p. 062316.

Van Criekinge, Wim and Rudi Beyaert (1999). "Yeast two-hybrid: state of the art". In: *Biological Procedures Online* 2.1, p. 1.

Van Dam, Sipko, Urmo Vosa, Adriaan van der Graaf, Lude Franke, and Joao Pedro de Magalhaes (2017). "Gene co-expression analysis for functional classification and gene–disease predictions". In: *Briefings in Bioinformatics* 19.4, pp. 575–592.

Vaynberg, Julia, Tomohiko Fukuda, Ka Chen, Olga Vinogradova, Algirdas Velyvis, Yizeng Tu, Lily Ng, Chuanyue Wu, and Jun Qin (2005). "Structure of an ultraweak protein-protein complex and its crucial role in regulation of cell morphology and motility". In: *Molecular cell* 17.4, pp. 513–523.

Vaynberg, Julia and Jun Qin (2006). "Weak protein–protein interactions as probed by NMR spectroscopy". In: *Trends in Biotechnology* 24.1, pp. 22–27.

Vidal, Marc (2009). "A unifying view of 21st century systems biology". In: *FEBS Letters* 583.24, pp. 3891–3894.

Vidal, Marc, Michael E Cusick, and Albert-László Barabási (2011). "Interactome networks and human disease". In: *Cell* 144.6, pp. 986–998.

Villaveces, Jose M, Rafael C Jimenez, Pablo Porras, Noemi del-Toro, Margaret Duesbury, Marine Dumousseau, Sandra Orchard, et al. (2015). "Merging and scoring molecular interactions utilising existing community standards: tools, use-cases and a case study". In: *Database* 2015.

Von Mering, Christian, Lars J Jensen, Berend Snel, Sean D Hooper, Markus Krupp, Mathilde Foglierini, Nelly Jouffre, Martijn A Huynen, and Peer Bork (2005). "STRING: known and predicted protein–protein associations, integrated and transferred across organisms". In: *Nucleic Acids Research* 33.suppl_1, pp. D433–D437.

Vörös, András and Tom AB Snijders (2017). "Cluster analysis of multiplex networks: Defining composite network measures". In: *Social Networks* 49, pp. 93–112.

Wang, Jianlong, Sridhar Rao, Jianlin Chu, Xiaohua Shen, Dana N Levasseur, Thorold W Theunissen, and Stuart H Orkin (2006). "A protein interaction network for pluripotency of embryonic stem cells". In: *Nature* 444.7117, p. 364.

Wang, Zhong, Mark Gerstein, and Michael Snyder (2009). "RNA-Seq: a revolutionary tool for transcriptomics". In: *Nature Reviews Genetics* 10.1, p. 57.

Watson, James D, Francis HC Crick, et al. (1953). "Molecular structure of nucleic acids". In: *Nature* 171.4356, pp. 737–738.

Watts, Duncan J and Steven H Strogatz (1998). "Collective dynamics of 'small-world'networks". In: *Nature* 393.6684, p. 440.

Wegner, Anatol E, Luis Ospina-Forero, Robert E Gaunt, Charlotte M Deane, and Gesine Reinert (2018). "Identifying networks with common organizational principles". In: *Journal of Complex Networks* 6.6, pp. 887–913.

Welch, William J (1992). "Mammalian stress response: cell physiology, structure/function of stress proteins, and implications for medicine and disease". In: *Physiological Reviews* 72.4, pp. 1063–1081.

Wickham, Hadley, Peter Danenberg, and Manuel Eugster (2018). *roxygen2: In-Line Documentation for R*. R package version 6.1.1. URL: `https://CRAN.R-project.org/package=roxygen2`.

Wickham, Hadley, Jim Hester, and Winston Chang (2019). *devtools: Tools to Make Developing R Packages Easier*. R package version 2.2.1. URL: `https://CRAN.R-project.org/package=devtools`.

Winzeler, Elizabeth A, Daniel D Shoemaker, Anna Astromoff, Hong Liang, Keith Anderson, Bruno Andre, Rhonda Bangham, Rocio Benito, Jef D Boeke, Howard Bussey, et al. (1999). "Functional characterization of the S. cerevisiae genome by gene deletion and parallel analysis". In: *Science* 285.5429, pp. 901–906.

Wittig, Ilka, Hans-Peter Braun, and Hermann Schägger (2006). "Blue native PAGE". In: *Nature Protocols* 1.1, p. 418.

Wolfe, Cecily J, Isaac S Kohane, and Atul J Butte (2005). "Systematic survey reveals general applicability of "guilt-by-association" within gene coexpression networks". In: *BMC Bioinformatics* 6.1, p. 227.

Wu, Qingyao, Yunming Ye, Michael K Ng, Shen-Shyang Ho, and Ruichao Shi (2014). "Collective prediction of protein functions from protein-protein interaction networks". In: *BMC Bioinformatics*. Vol. 15. 2. BioMed Central, S9.

Yabas, Mehmet, Hannah Elliott, and Gerard Hoyne (2016). "The role of alternative splicing in the control of immune homeostasis and cellular differentiation". In: *International Journal of Molecular Sciences* 17.1, p. 3.

Yan, Changhui, Feihong Wu, Robert L Jernigan, Drena Dobbs, and Vasant Honavar (2008). "Characterization of protein–protein interfaces". In: *The Protein Journal* 27.1, pp. 59–70.

Yang, Jaewon and Jure Leskovec (2013). "Overlapping community detection at scale: a nonnegative matrix factorization approach". In: *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*. ACM, pp. 587–596.

Yang, Xinping, Jasmin Coulombe-Huntington, Shuli Kang, Gloria M Sheynkman, Tong Hao, Aaron Richardson, Song Sun, Fan Yang, Yun A Shen, Ryan R Murray, et al. (2016). "Widespread expansion of protein interaction capabilities by alternative splicing". In: *Cell* 164.4, pp. 805–817.

Zhang, Chao and Sung-Hou Kim (2003). "Overview of structural genomics: from structure to function". In: *Current Opinion in Chemical Biology* 7.1, pp. 28–32.

Zhao, Shanrong, Wai-Ping Fung-Leung, Anton Bittner, Karen Ngo, and Xuejun Liu (2014). "Comparison of RNA-Seq and microarray in transcriptome profiling of activated T cells". In: *PloS One* 9.1, e78644.

Zitnik, Marinka, Marcus W Feldman, Jure Leskovec, et al. (2019). "Evolution of resilience in protein interactomes across the tree of life". In: *Proceedings of the National Academy of Sciences* 116.10, pp. 4426–4433.