SIMULATING JOINT INFORMATIONAL CONTENT TO BETTER ESTIMATE INTERACTION NETWORKS

Jan van Roozendaal
Student ID: 01803253

Promotor: Prof. Daniele Marinazzo
Copromotor: Prof. Yves Rosseel

A dissertation submitted to Ghent University in partial fulfilment of the requirements for the degree of Master of Science in Statistical Data Analysis.

Academic year: 2019 - 2020

Gent, June 19, 2020

The promotor,	The author,
Prof. Daniele Marinazzo	Jan van Roozendaal

ACKNOWLEDGEMENTS

I experienced enrolling for the Master's program in Statistical Data Analysis back in May 2018 as an impulsive, but most definitely positive move. With only one full year of work experience gained in data visualization after graduating before in the summer of 2016, the program offered me opportunities to gain more practical experience in programming and data science. A little known fact may be that I did not pass the assessment test; sheer motivation to prove myself during the first year by passing the compulsory courses earned me a chance to finish the program and obtain a degree in the second year, majoring in Computational Statistics. To me, the completion of this dissertation shows the confidence I have gained to tackle large-scale analytical-themed questions using both gained knowledge in statistical data analysis and programming. Combined with my previous studies in information systems & IT management, along with my gained work experience in business intelligence, I feel that this Master's program has been a great addition to my own personal development.

I would like to sincerely thank my supervisors prof. Daniele Marinazzo and prof. dr. Yves Rosseel from the Faculty of Psychological and Educational Sciences, Department of Data Analysis for their continued guidance and support, even during the quarantine period due to the COVID-19 outbreak which happened during the semester. Their efforts to always swiftly reply to my status updates even when they were challenged to change their way of working and daily lifestyle are greatly appreciated.

Furthermore, I would like to thank my parents, my sister and my sister-in-law for their moral support during the entire two-year period where I combined finishing the Master's program in Statistical Data Analysis with my professional job as IT consultant. I have never regretted taking the decision to follow this program, even if it meant to become more creative and flexible to combine work with education.

CONTENTS

ABSTRACT

Within the field of psychometrics, over the last decade a new approach using network visualizations is proposed to explore interactions between elements of a construct, such as a collection of symptoms for a disease of disorder, and explain the possible challenges of overlap or bridge connections between diseases of disorders which is formally known as comorbidity. Network visualizations allow to plot statistical dependencies between variables of such a construct to aid in exploratory data analysis. Estimation of such networks is typically done using node centrality indices to understand the role of importance for each variable present in the network. A challenge of such estimation is that the statistical dependencies are limited to those of two variables at a time, possibly conditioned on the presence of other variables. A concern addressed in this dissertation is that higher-order interaction terms in groups of three or more variables could influence statistical dependencies at a lower level, or the possible risk of false positive entries of statistical dependencies within the resulting network visualization.

This dissertation will discuss the possible role of joint information content on the estimation of interaction networks by introducing a simulation framework powered by Confirmatory Factor Analysis (CFA) to control the level of such higher-order interaction for specific groups of three variables as a parameter. This parameter consists of three levels: (a) a synergistic case where the presence of a third variable increases the statistical dependency of the first two variables; (b) a redundant case where information of the third variable may also be present in the first two variables, leading to a decrease of statistical dependency between the two original variables; and (c) a zero-case where no higher-order interaction term is forced on the variables. To achieve a sparse network representation of the data containing only significant statistical dependencies between variables, the graphical LASSO regularization method was considered. Several model layouts are also proposed for the simulation.

The results show that the level of joint informational content between specific groups of variables influences both occurrence and strength of statistical dependencies. This applies both for particular groups of variables sharing characteristics as for those for which their statistical dependency should be considered a false positive. These findings showcase possible new challenges regarding estimation of interaction networks.

To conform with the guidelines to allow for the results to be reproduced, an online repository has been made available which contains the complete library of used *R*-scripts of the simulation framework, its related functions and those used for visualization of the main results. It also contains multiple workspace files with data of the repetitions subjected to a fixed seed number (100). The repository can be found via the following URL:
https://github.com/jan-vanroozendaal/MaStat-Thesis-InteractionNetworks

Tags: psychometrics, network estimation, network visualization, joint informational content, interaction information

CHAPTER 1

RESEARCH OBJECTIVES

1.1 Introduction

The field of psychometrics is concerned with understanding and measuring psychological or social-related topics that are often complex by nature. Examples of such topics could be disorders for which the symptoms are known but for which their cause is still debatable. Either a collection of symptoms gives the diagnosis of the existence of a disorder or the sudden emergence of the disorder causes the symptoms to be present. Over the course of the last decade, network visualizations of such constructs have been proposed (Schmittmann et al., 2013; Borsboom and Cramer, 2013) to look at them from another perspective; to understand possible interactions of symptoms within a disorder system or between disorder systems. Interactions between symptoms using such data visualization approach is limited to those between two variables, either conditioned or unconditioned on the presence of other variables in the network. Such connections also depict statistical dependencies between variables in the network. The application of this new approach also demands for reliable estimation techniques to determine the importance of each point in the network, which is often done using node centrality indices. Despite this, a possible limitation of this approach is that it may not correctly address the fundamental issue of taking account for the possible joint information shared in groups of three or more variables, whether in a synergistic or redundant setting. The presence of a variable could either enhance or deteriorate statistical dependencies between other variables, possibly influencing results for network estimation when using the established centrality indices of strength, betweenness and closeness. Another possibility could be the existence of a statistical dependency between two variables which was first considered insignificant for other level settings of joint informational content between the related variables. This also increases the concern of detecting false positives regarding statistical dependencies during estimation. This dissertation will introduce a simulation framework to generate interaction networks where the amount of said joint information for a group of three variables will be controlled as a parameter. This simulation framework is written in the programming language *R* and relies heavily on the *lavaan* package (Rosseel, 2012).

1.2 Problem Statement & Research Objectives

The problem statement of the dissertation could be defined as follows: by manipulating the level of joint information content for a group of three variables to either a synergistic or redundant setting and comparing it to the 'zero'-level case, there is reason to believe that it influences statistical dependencies between variables and thus the results of the established estimation techniques using node centrality indices on the resulting networks. A challenge therefore is to derive to a sparse precision matrix for each level of joint informational content configured which conditions said statistical dependencies on the existence of the other variables in the model. It also drives insignificant statistical dependencies to zero to deliver the sparse representation of the network. The research objectives are thus to investigate to what extent these statistical dependencies change for each extreme level of joint information content. Another interest is whether its influence is only limited to a particular set of statistical dependencies between variables sharing some common characteristics, or if it could be considered random. To address the concern of possible false positives of statistical dependencies found in the resulting networks, key performance indicators such as specificity and sensitivity will be measured for different configurations of joint informational content and threshold options for the regularization method to come to a sparse representation of the data regarding statistical dependencies between variables.

1.3 Outline of dissertation

The following chapter will discuss the main theoretical background for the main topics of psychometrics, network visualization, and joint informational content which originates from information theory. Furthermore, the LASSO regularization technique to derive to a sparse network model is also discussed. Chapter 3 covers the methodology with regards to the modelling approach taken in order to control the level of joint informational content for each given set of variables within the model. Several model layouts are then proposed to explore whether its influence on the estimation of the resulting networks differs significantly. The fourth and final chapter covers the main results found with regards to the emergence or changes of (unexpected) statistical dependencies found in the resulting interaction networks. It is then followed by a brief discussion of these summarized results and a final conclusion regarding the role of joint informational content on the estimation of interaction networks. Several shortcomings and limitations of the dissertation are discussed which could provide further ideas and improvements on extending the simulation approach.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The theoretical background of three main topics will be discussed in this chapter. First, an overview of the motivation, use, visualization, estimation and related challenges of network psychometrics will be discussed in its own section. This is followed by coverage of several concepts related to the field of information theory in order to discuss possible definitions and accompanying calculations for joint informational content between variables. The final topic covered will be of the LASSO regularization method which is used to derive to a sparser collection of data with the intention of creating a network model with more interpretability. This technique could lead to a network model with possibly more significant or relevant interactions to explore further.

2.2 Network Psychometrics

This section will discuss the motivation for using an dependency network as a visual tool for exploratory data analysis in the field of psychometrics along with its core visual components and guidelines for interpretation. The topic of comorbidity shall also be briefly discussed to put emphasis on the possible interplay of symptoms across disorders challenging the traditional relationship models of symptoms and disorders as a possible argument to suggest the alternative, namely an dependency network approach. The Gaussian graphical model will be considered as the main template of such dependency network for this dissertation mainly because of its relation to the inverse covariance matrix and therefore also to the problem statement. Regarding the topic of suggested estimation methods of the accuracy of an dependency network or the importance of variables derived from the network approach, current limitations or embedded assumptions of the dependency network template and the suggested metrics for estimation of the model should be evaluated as well to motivate the need for attempting to include joint informational content for the intents and purposes of improving estimation of the network.

2.2.1 Motivation for a Network Approach

Psychometrics can be defined as a scientific discipline focused on the theory of how to objectively measure complex topics of psychological or social nature such as behavior, disorders or abilities. In the paper of Schmittmann et al. (2013) a summary is given of how the relationship between such psychological constructs and the variables (or symptoms) available for observation is mostly interpreted in one of two ways: either the construct causes the variables to be present (reflective model), or the appearance of a (set of) variable(s) determines the presence of the psychological construct (formative model). In both model types, the construct is represented as an unobserved or latent variable. Reflective models assume that correlation between symptoms should exist but only due to their common cause. Direct causal relationships between symptoms is disregarded to emphasize the idea that establishing the relationship between symptoms and psychological constructs is done via measurement. In formative models, for each symptom it is assumed they each capture separate parts that together explain the construct as a whole. Removal of one symptom would therefore directly change the severity or description of the construct. A limitation of such models is that it excludes the possible interplay and relationship between a set of symptoms that may occur within a given disease or disorder. In fact, the authors believe this to be a possible motivation why such psychological constructs are described as a single unit or entity. Instead of placing symptoms as a function to latent variables, both Schmittmann et al. (2013) and Borsboom and Cramer (2013) propose a third approach of psychometrics involving network analysis. A network of direct relationships between symptoms represents a causal, dynamic system of the psychological construct. The goal of the network is not to seek for a common cause of the construct; its emphasis is on the dynamics of the symptoms themselves. Schmittmann et al. (2013) stress that the dimension of time and its role on the symptoms is not well presented in reflective and formative models and that the proposed network models are to change over time. Borsboom and Cramer (2013) mention a possible advantage of using such network models with time-series data using lag-1 correlations to visualize the evolution of the emergence of symptoms, as building a network of all symptoms for one individual may not be suitable when the presence of one or more symptoms is missing at one point in time. Also, Schmittmann et al. (2013) discuss how outgoing effects or events in one's life may be derived by examining either the overall network or part of it. An example given is how a selection of interactions between symptoms may create a vicious circle within a person's lifeline possibly triggering a series of negative events along the way. The emphasis is to shift psychometrics more towards the dynamics of the symptoms when placed together in a psychological construct.

2.2.2 Role of Comorbidity

It should also be noted that the difficulty of how to differentiate constructs such as depression and anxiety can also be reflected via such network models. As both may share symptoms, one may ask whether the separate networks of the symptoms of both depression and anxiety could be linked together. This phenomena of symptoms related to multiple constructs is called comorbidity. Cramer et al. (2010) discuss the role of comorbodity in both the reflective model (in their paper referred to as the common cause hypothesis) and the network model approach. Comorbidity in a reflective model suggests a direct correlation between the latent variables while keeping the observable symptoms separated between each latent variable, eliminating the possibility to explore comorbidity in terms of exploring edges of nodes derived from distinct psychological constructs. The term '*bridge symptom*' is defined to identify symptoms that are shared across disorders or create overlap in the network model by possibly passing on effects (via correlations) to other nodes from each disorder when emerged. The authors also bring the argument that, compared to latent variable modelling, estimation of each of the symptoms (nodes) in the model is not considered to be equal. Instead, node centrality can be used as an estimation metric to understand the importance of one symptom in the network model compared to others, as will be discussed later in this section. In short, for the topic of comorbidity, the network approach along with its proposed estimation metrics for node centrality allows for a causal explanation why certain symptoms may lead to a greater risk of comorbidity and thus to an inequality of weight importance of symptoms. Cramer et al. (2010) also state that this inequality via node centrality puts a challenge to the cut-off approach of diagnosing disorders where the number of symptoms is used as a metric. Jones et al. (2019) define statistics specifically designed to measure the node centrality of such bridge symptoms, expanding on the defined node centrality estimation methods described by Epskamp et al. (2018). Both the conclusions of Cramer et al. (2010) and Jones et al. (2019) state that cases of comorbodity could be explored further via network modelling sets of symptoms between disorders without limiting to only directly linking latent variables acting as constructs of these disorders, providing a further argument to use this approach in psychometrics. Furthermore, the literature review of Fried et al. (2017) reviewing network studies including comorbidity research conducted between 2010 and 2016 mentions the implication derived from the network approach that the frequency of being diagnosed with one of either disorders may co-occur as a function dependent on the number of bridge symptoms. The authors state that such implication could not have been derived from the traditional concept of reflective models or the common-cause hypothesis. Referring back to the compatibility of time-series data as suggested by Borsboom and Cramer (2013) to understand the

evolution of emergence of symptoms within a network for each individual, combining this with the topic of comorbidity, Fried et al. (2017) do take into consideration that the occurrence of comorbidity may differ across individuals while being diagnosed similarly, allowing for further explanation via the network approach.

2.2.3 Data Visualization for Exploratory Data Analysis

Aspects in terms of data visualization for the network approach in psychometrics will now be discussed. The Gaussian graphical model will be used as the main graph template for visualization purposes. The main visual properties of the Gaussian graphical model is formally discussed in Lauritzen (1996) and consists of two essential building blocks of (a) nodes and (b) edges derived from a set of random variables where the data is assumed to hold a multivariate normal (or Gaussian) distribution to create an undirected network graph, often referred to as a Markov random field. Nodes typically represent variables from the given dataset for the study and are commonly graphically displayed with circles. Edges are the lines connecting nodes together based on a given relation. Such model is considered undirected as there is no restriction for the edges to follow a specific path or sequence to connect nodes together.

Borrowing the syntax for the formal notation of the properties of the data distribution and the accompanying Gaussian graphical model from Yuan and Lin (2007), one could start with the collection of variables notated as a random vector with length p referring to the number of dimensions (or variables) named X. The notation of the Guassian distribution is given using μ for the unknown mean and $\sum$ representing the covariance matrix.

$$X = (X^{(1)}, \ldots, X^{(p)}) \sim \mathcal{N}_p(\mu, \sum) \tag{2.1}$$

The relationship conveyed between two nodes in a Gaussian graphical model is the correlation of the two relevant variables after conditioning for all other present variables, giving the model an advantage of minimizing the risk of showcasing deceitful correlations (Bhushan et al., 2019). This conditioning is achieved by estimating the inverse of the covariance matrix C, which is often named the concentration matrix or precision matrix.

$$C = \sum^{-1} \tag{2.2}$$

Because of this property, it is not a prerequisite of the model to have all combinations of pairs of nodes connected with each other; however, Epskamp et al. (2018) discuss in their tutorial paper of estimating such networks that sampling variation derived from the multivariate normal distribution may introduce bias when measuring

centrality of nodes. While centrality will be discussed later in this section, for now it suffices to describe that it evolves around the presence of edges towards a node (Fried et al., 2017) or the presence of a node as a destination step when determining direct paths, most often the shortest, between a given pair of nodes in the network using edges to form a path (Costantini et al., 2019; Epskamp et al., 2018). Due to this variation, the observation of true zero partial correlations is considered rare (Bhushan et al., 2019), and most non-penalized network graphs are considered to be dense due to the high number of possible observed non-zero edges.

Using the definition of the Gaussian random vector X from Equation 2.1, the Guassian graphical model is defined as a graph G using the coordinates from the p-dimensions stored in V and a collection of edges E referring to the conditioned correlations between two variables:

$$G = (V, E) \tag{2.3}$$

For clarification, an edge included in E can be denoted as e_{ij} where i and j represent the two variables and lie between 1 and p. Referring back to the Gaussian random vector X, one could also describe it as the edge between $X^{(i)}$ and $X^{(j)}$. Because the edges are undirected, e_{ij} equals e_{ji} but the latter will be omitted due to redundancy. If an edge between $X^{(i)}$ and $X^{(j)}$ does not exist, this can be referred back to the precision matrix C where for the entry c_{ij} a value of 0 is found. The collection of edges E therefore include entries of e_{ij} where their corresponding conditional correlations $c_{ij} > 0$ and can be written as:

$$E = (e_{ij})_{1 \leq i < j \leq p} \tag{2.4}$$

Besides shapes and lines, other data visualization dimensions such as color and thickness of edges are used to convey information about the strength of the partial correlation between two nodes and whether it is considered positive or negative. An example of a final Gaussian graphical model is shown in Figure 2.1 and originates from the paper of Epskamp (2016) where a dataset *bfi* consisting of 25 variables grouped into five factors is used from the R-package *psych* (Revelle, 2011). While in this example a threshold is used to remove edges from the network with an absolute value smaller than 0.05, other missing edges could be derived from the observation that $c_{ij} = 0$.

When discussing similarities of the Gaussian graphical model to other modelling approaches, Bhushan et al. (2019) refer back to the single latent variable construct of the reflective model, this time described as a uni-dimensional factor model. It is critical to mention here that its purpose lies in exploratory data analysis to explore both presence of expected relationships between nodes based from theory and possible unexpected ones explored via data visualization, and is not to be used directly for sta-

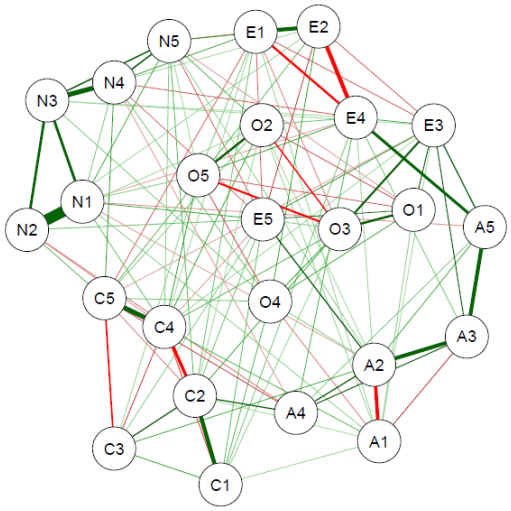


Figure 2.1: Example of a Guassian graphical model as seen in Epskamp (2016). Color and thickness are used as extra dimensions to convey more information about the partial correlations between nodes.

tistical inference. The authors do mention that comparison of network models across different groups of the population could be realized using a Hamming distance metric. Originating from information theory, this metric describes the number of positions where a change is required for an object to be in a specific required state. For string objects, this represents the number of characters to be replaced to spell out a given word or sequence. With regards to the network graphical models, the application of the Hamming distance refers to the difference of the presence or absence of edges across two models; however, the sign and level of the partial correlation embedded in the edges is ignored for comparison. This means that if an edge was present between nodes A and B with a strong positive partial correlation in network model 1 and a weak negative partial correlation in network model 2, the Hamming distance would not increase by one when comparing the two network models based on this edge comparison. Because of this limitation of ignoring the stability of the correlation between nodes across networks representing different target groups, bootstrapping methods have been proposed by Epskamp et al. (2018) to allow for confidence intervals of edge weights, as will be discussed later.

Jones et al. (2018) discuss possible misinterpretation of such network models from a data visualization perspective, focused mainly on node positioning. They discuss how most presentations of network studies utilize an algorithm for aesthetic purposes, named the Fruchterman-Reingold (FR) algorithm, as described in Fruchterman and Reingold (1991). In short, nodes are positioned as such that they do not overlap other nodes in the presentation view. Clusters of nodes with strong partial correlations are placed closer to each other. At the same time, nodes are placed as such that edges across the network are of approximate equal length. The network shown in Figure

2.1 shows the application of the FR-algorithm; nodes with thick edges are placed relatively close to each other. The clustering element of the algorithm would suggest that node positioning, or the distance between two particular nodes, captures some information about the likeliness of these nodes within the model. Bhushan et al. (2019) already warn that similarity of nodes is not conveyed in the model via the dimension of position; Jones et al. (2018) add to this that the coordination levels of the nodes along the X and Y axes do not contain any meaningful information about the characteristics of a node.

To allow exploratory data analysis via plots with meaning given to positions to inspect (dis)similarities of variables, the multidimensional scaling (MDS) method is suggested. Compatible with high-dimensional data sets, it allows to represent nodes in a low-dimensional space. The input for MDS is a proximity matrix of some sort, and the authors discuss how the network edges, each representing partial correlation between a pair of nodes, can be used for such application. Similarly to the techniques applied in the FR algorithm, nodes with strong associations will tend to be presented close to each other in the MDS plot; however, it is important to note that the Eucledian distance of nodes on the resulting MDS plot now does represent dissimilarities. The example MDS plot on Figure 2.2 shows the similarities of the nodes used in Figure 2.1 from the *bfi* dataset. The R-code snippet to recreate this plot can be found in Appendix A and refers to the tutorials as shown in Jones et al. (2018). The interpretation of the values of the two dimensions D_1 and D_2 to build the scatter plot is beyond the scope of this dissertation. It suffices to understand that positioning of the data points is now given a clear meaning compared to the node positioning in the network model.

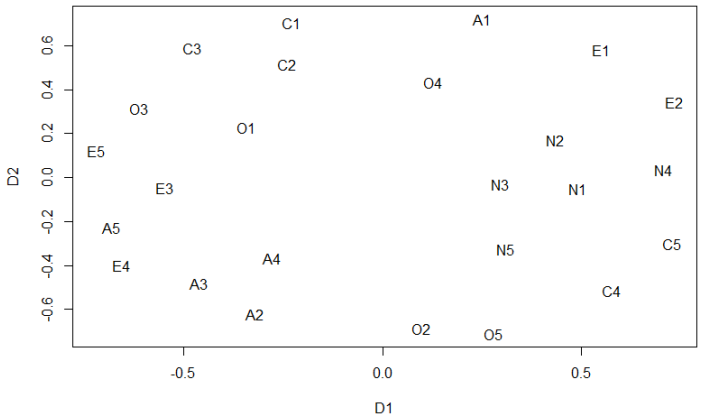


Figure 2.2: Example of MDS plot using the *bfi* dataset from the *psych* R-package.

A few clear examples to showcase the relation between the model network and the MDS plot are now given. The nodes *N1* through *N5* are positioned as such in the MDS plot that they could form a cluster. In Figure 2.1, these nodes are seen closely to each other on the upper-left corner and most edges are (strongly) positively related. The clustering of the data points in the MDS points therefore suggests that these

nodes are strongly similar or are closely related to each other. An example where dissimilarity is displayed can be done using the nodes *C1*, *C2* and *C3* against *C4* and *C5*. Starting from the network model, all five nodes are shown on the bottom-left corner. Negative weighted edges exist between *C3* and *C5* and between *C2* and *C4*. Because all other edges are (strongly) positive, there is a distance between nodes *C4* and *C5* from the remaining nodes of that factor. The MDS plot shows this clearly as nodes *C1* through *C3* are grouped on the upper-left corner while nodes *C4* and *C5* are on the opposite end of the plot on the bottom-right. The same phenomena is shown with node *A1* which is distant from the other A-nodes in the MDS plot due to its negative edge weights with nodes *A2* and *A3*.

With this insight, Jones et al. (2018) allowed for a technique where both edges and node positioning are given meaning, derived from the partial correlations and zero-order correlations, respectively. Alternatively, principal component analysis can be used to give meaning to the coordinates of the nodes on the X and Y axes. The covariance matrix could be used as input for eigenvalue decomposition. These axes are derived from the two main principal components representing most of the aggregated variance. Of course, as these two component may not capture all aggregated variance, information will be lost in this lower-dimensional space when plotting nodes. It can be interpreted as a metric of how well the complexity of the network can be represented in such two-dimensional spacing. If the two principal components collectively account for a low percentage of the aggregated variance explained, interpretation of the results derived from the plot should be taken with caution. Compared to MDS, the coordinates of the nodes on the X and Y axes on the PCA plot are now given meaning, but its trade-off is that the distance between nodes is not directly interpretable in terms of (dis)similarity, especially when the the two principal components account for a small percentage of the variance. Instead, the positional distance of two nodes in the PCA plot help to explain how one node may differ in one dimension or component related to the other. The main conclusion of Jones et al. (2018) is that plots representing the complexity of a network on a lower-dimensional space, either via MDS or PCA, may convey additional information about (dis)similarities of nodes using position as a data visualization dimension compared to the Gaussian graphical model utilizing the FR algorithm. Such additional plots may aid with the exploratory data analysis in the study.

The PCA plot for the *bfi* dataset using the first two principal components explaining most of the aggregated variance is shown in Figure 2.3; the R-code is also found in Appendix A. Between the two principal components, a total of 31.5% of the variance is explained meaning that caution is advised when relying on the insights of this plot. Because the coordinate points from the two principal components explaining most

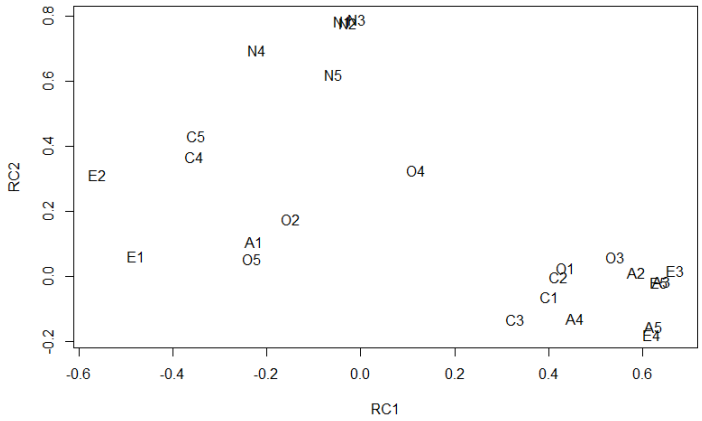


Figure 2.3: Example of PCA plot using the *bfi* dataset from the *psych* R-package. The first principal component *RC1* accounts for 18% of the aggregated variance; for *RC2* it is 13.5%.

of the variance has now been given a meaning, the positioning of each data point may have shifted when comparing it to the MDS model. Nevertheless, while distance between data points is not easily explained, the examples given of the N-nodes being closely together and a separation between nodes *C4* and *C5* and the remaining C-nodes are still noticeable from the PCA plot despite the low percentage of variance explained. This shows that there is potential to use both MDS and PCA plots alongside the network model visualization to support exploratory data analysis.

2.2.4 Estimation via Node Centrality

Regarding estimation techniques for the structure of psychometric networks, the measurement of node centrality via multiple indices of degree, betweenness and closeness is considered a popular method (Bringmann et al., 2019) and originate from the work of Freeman (1978). Important to note is that such indices were developed when dealing with social networks where nodes typically reflect people and edges the existence of an relationship or of a communication channel. Rather than the classification of the existence of an edge between two nodes, the centrality indices may also be compatible with edge weights (Opsahl et al., 2010) in a weighted network, such as with using the absolute partial correlation levels as has been described earlier. Such centrality indices taking into consideration edge weights has been considered a common standard (Bringmann et al., 2019) and has been used in the work of Epskamp et al. (2018) to estimate accuracy in psychometric networks. Furthermore, if edges were modelled as such that they were restricted to go into one direction, this dimension of edge direction influences the results on the three mentioned node centrality indices (Bringmann et al., 2019). For the scope of this literature review, only undirected edges will be considered.

Starting with the degree index, it describes the number of edges a node has (Freeman, 1978). Taking edge weights into consideration, an alternative method is to sum up the absolute edge weight values for this node. This degree index using edge weights is then renamed as 'strength'. The closeness index of a node is considered a more 'global' metric of the network and describes the level of distance a node has with all other nodes in the network, if such connection via edges can exist. The concept of closeness relates to efficiency of communication of information of a node to other nodes in the network (Freeman, 1978). Minimizing the distance between two nodes without considering edge weights is done by finding the lowest number of edges required to build the bridge between them; however, Bringmann et al. (2019) note that edge weights may be considered a proxy for connection speed or efficiency of communication, and that shorter paths with weak edge weights may not be preferred over longer paths with stronger edge weights when considering them. Betweenness tells for each node how many times they are included when mapping the shortest path (in this case, with edge weights considered) between a given set of two nodes (Freeman, 1978). This centrality index describes whether the node plays a significant role in information flow between nodes that are not directly connected in the network via an edge (Bringmann et al., 2019).

Before continuing with the assumptions made when applying these node centrality indices for estimation and the possible critique of how these may not be fit within the concept of psychometric networks due to their origins in social networks, the work of Epskamp et al. (2018) focuses further on this estimation by introducing accuracy of edge weights by creating a 95% confidence interval and stability of the aforementioned node centrality indices via non-parametric bootstrapping techniques. Their assumption is that, due to the rule edge weights play in the node centrality indices, wide confidence intervals of the edge weights lead to poorer accuracy to estimate node centrality. In combination, the authors introduce a stability metric for the indices by estimating them on subsets of the data by reducing the number of observations to be used to create the network while keeping the number of nodes constant. Stability is tested via correlation of the original values of the centrality indices and those estimated from the networks using a reduced number of observations. As a standard, in order to conclude stability of the indices, this correlation should be considered strong with a minimum threshold set to at least 0.7. The more observations can be dropped while maintaining this strong correlation with the original centrality indices, the stronger the stability of said indices.

Interestingly, Jones et al. (2019) discuss the combined topics of comorbidity and centrality indices to define a method for estimating bridge symptoms using altered versions of the strength, betweenness and closeness indices. Categorizing beforehand

the symptoms to their predefined communities (commonly disorders), the 'bridge strength' of a node can be defined as the sum of absolute edge weights of a node connected to other nodes originating from other communities. Similarly, bridge betweenness and bridge closeness only consider paths between nodes coming from different communities while taking into account edge weights. The main intention of introducing this variation of node centrality indices is to spot potential bridge symptoms for an individual to avoid comorbidity issues in diagnosis of disorders.

2.2.5 Node Centrality Assumptions

Because of the aforementioned origin of said indices (Freeman, 1978) using only the structure of social networks at the time, there has been discussion about their compatibility in psychometric networks (Bringmann et al., 2019; Hallquist et al., 2019). The word 'flow' was mentioned during the description of the betweenness centrality index. Borgatti (2005) describes three types of flow processes that would be considered applicable in social network structures: serial, parallel and transfer flows. In short, both serial and parallel flows are based on a duplication mechanism. A simple example would be a virus, where one person could infect others over time via coughing or sneezing. People infected with the virus would become immune over time, thus a recurring loop of people becoming infected is impossible allowing for a serial flow. A parallel flow could be of multiple people simultaneously spreading an e-mail to their direct connections due to a computer virus. Transfer flows focus on the question whether a flow, such as traffic, is designed to follow the fastest, shortest, or otherwise most efficient way to traverse from one node to another or prefers a non-deterministic approach. The example of a package delivery process is used as a route must be determined to allow for the most packages to be delivered in an area to gain time efficiency. The critique of Borgatti (2005) is that the betweenness and closeness centrality measures of Freeman (1978) directly make assumptions that the flow of information should only occur using the shortest paths possible and that revisiting of nodes is excluded. A concern could therefore be that using such indices to deduce node centrality (and thus its importance) on a network that uses a different approach of information flow would lead to misleading results. Instead, the essence of closeness and betweenness should be on the arrival time and frequency of arrival of information, respectively (Borgatti, 2005). The examples given to describe the flow types along with the redefined purposes of closeness and betweenness are applicable to social network structures; however, in the case of a psychological network depicting a order, while the visualization aspect allows one to conceptualize how symptoms may spread, it is hard to define the flow between symptoms using the edges as they do not carry information from one to another (Bringmann et al., 2019).

As for the assumption regarding betweenness and closeness using shortest paths, as symptoms are not meant to communicate information from one symptom to another to reach some end-state, in combination with the fact that edges only convey strength of relation and not a communication channel, it makes the value of these indices questionable.

Furthermore, Bringmann et al. (2019) mention that due to the use of length and distance metrics to calculate the node centrality indices, the information contained in the edge between nodes whether the partial correlation is negative or positive is ignored; therefore, the indices can only generally state something about the level of influence of a node, but not whether it is positive or negative. This can also be paired with the violation of the assumption that changing the representation of the node (a person in social networks and a symptom in psychological networks) would not change the interpretation of the indices. The DSM documentation (American Psychiatric Association, 2013) does classify symptoms in terms of severity. Such information is currently not conveyed in the centrality indices (Bringmann et al., 2019), meaning that the most central node may not always be considered the most important one to tackle if occurring when its severity level is considered relatively low compared to others close in the network. The context of node centrality is then not directly one of severity of the symptom.

Another important assumption about the centrality indices that is applicable much easier in social network structures is that each node refers to a distinct entity or person. On the other hand, symptoms are usually created by combining responses from multiple questions on a questionnaire. It is not unlikely to think that due to the setup of the questionnaire and the predetermined progression of the questions, multicollinearity may exist between these symptoms and can therefore not be interpreted as fully unique (Bulteel et al., 2016). If this assumption is not met, comparing centrality levels of two nodes becomes problematic when overlap of the two node constructs is possible (Bringmann et al., 2019). This assumption will be one of the central focus points throughout this dissertation.

Finally, as is applicable in multiple fields of research, one needs to assume that all relevant nodes are included for network analysis. Epskamp et al. (2018) pointed out in their bootstrap approach that stability of the betweenness and closeness indices often scores a lower rate when sub-setting the data compared to the strength index. Hallquist et al. (2019) find that sampling variability explains this instability of the two centrality indices and they are highly sensitive to spurious correlations between nodes. They refer this insight back to the concept of comorbidity and bridge symptoms, stating that the role of betweenness to identify bridge symptoms is put into danger. Furthermore, the authors suggest to use both the marginal and partial

correlation matrix and not rely fully on analysis via network visualization only. Bringmann et al. (2019) suggest to focus more on the dynamics of the network rather than node centrality to intervene with the development of a disorder. Indeed, the possibility of shared variance across nodes via multicollinearity is one to explore further to find better estimation techniques for psychometric networks to better understand interaction between symptoms.

2.3 Joint Informational Content

The measures discussed in this section are related to the information theory framework of Shannon (1948) of how to quantify information streams. The discussed network psychometrics are heavily reliable on the accuracy and stability of the correlations between two nodes after conditioning for all other variables to describe the similarity of nodes. It has been discussed how a low-dimensional approach via MDS has allowed for better interpretation of this similarity across nodes by using a coordinate system and the Euclidean distance as measure. An alternative approach to understand dependencies between variables based on information theory could be done via the metric of mutual information (Steuer et al., 2002). A limitation of the current network approach to better understand the established statistical dependencies between nodes is that it currently does not consider the high-order dependencies between three or more variables. Similarly to partial correlations, level of mutual information could be measured both before and after conditioning on the other variables. The shift of this measurement could then be interpreted as the interaction information metric. The core of information theory is that it uses the probability distributions of variables to determine the extent to which these variables are related to each other or the level of interaction between them. To better understand the concepts of mutual information and interaction between variables beyond the perspective of Pearson correlation values, it is best to start with the single-variable information metric of entropy derived from the framework of Shannon (1948).

2.3.1 Entropy & Mutual Information

If one knows that a variable can hold multiple values but may only return one at a time, each value could be mapped with a level of probability $p(a_i)$ when determining its distribution. The unpredictability of the result output from that variable for a given measurement test could be interpreted as the level of information one gains from reading said measurement result. To illustrate, if one knows that the result will always have the same outcome, only one value in the variable may exist with a probability

of 1 (or 100%); therefore, no additional information is gained as the outcome was to be expected. Using the same notation as seen in Steuer et al. (2002), going across all possible values of variable A, here given in the range $a_i, \ldots, a_{M_a}$, the average amount of information gain *H(A)* is named the entropy:

$$H(A) = -\sum_{i=1}^{M_a} p(a_i) \log p(a_i) \tag{2.5}$$

This concept of entropy can be expanded to two variables A and B by using their joint probability distribution, with a_i and b_j the states or outcomes of the variables and M_a and M_b the number of outcomes possible for variables A and B, respectively.

$$H(A, B) = -\sum_{i=1}^{M_a} \sum_{j=1}^{M_b} p(a_i, b_j) \log p(a_i, b_j) \tag{2.6}$$

In case of statistical independence between the two variables, such as when no edge has been visualized between two nodes in a network, the entropy levels of variables A and B may simply be added up to derive to *H(A,B)*. Otherwise, the calculation of the joint entropy *H(A,B)* has to include conditional probability in the form of $p(a_i|b_j)$ and its related conditional entropy *H(A|B)* defined as:

$$H(A|B) = -\sum_{i=1}^{M_a} \sum_{j=1}^{M_b} p(a_i, b_j) \log p(a_i|b_j) \tag{2.7}$$

The joint entropy is then calculated via the simple addition:

$$H(A, B) = H(A|B) + H(B) \tag{2.8}$$

Finally, using the property that the level of entropy of *H(A|B)* can only be smaller or equal to the level of entropy of *H(A)*, as knowing the outcome of variable B can only decrease the level of uncertainty of the outcome of variable A or leave it unchanged, the joint entropy *H(A,B)* holds the following mathematical rule:

$$H(A, B) \leq H(A) + H(B) \tag{2.9}$$

This property can then be used to define the level of mutual information *I(A,B)* between the variables A and B (Shannon, 1948) and cannot be negative:

$$I(A, B) = H(A) + H(B) - H(A, B) \geq 0 \tag{2.10}$$

Steuer et al. (2002) also discuss an alternative approach to entropy derived from Kullback (1959). Its original formula uses two probability distributions p and p_0. Rather than describing the average level of information gained from a measurement, Kullback's entropy equation explains how substituting an initial probability function p_0 with a more fitting distribution p changes the level of information gained. Again, using the notation as seen in Steuer et al. (2002):

$$K(p|p^0) = \sum p_i \log \frac{p_i}{p_i^0} \tag{2.11}$$

Steuer et al. (2002) rewrite this equation to refer back to the joint probability distribution of variables A and B. Here, p_0 has been substituted by the probability distributions of variables A and B separately, and the joint probability distribution is interpreted as the final fitting distribution p:

$$K(p|p^0) = \sum_{ij} p_{a_i,b_j} \log \frac{p(a_i, b_j)}{p(a_i)p(b_j)} \tag{2.12}$$

The authors mention that the measure of $K(p|p^0)$ can be used as a distance metric between the observed joint probability distribution and the assumption that the variables are statistically independent and can therefore be interpreted similarly as the mutual information *I(A,B)* found via the Shannon entropy equations. One important argument that Steuer et al. (2002) do make is that compared to Pearson correlation, the value of *I(A,B)* being 0 can directly be interpreted as such that the assumption of statistical independence between variables A and B holds. Another advantage is that the metric of mutual information is not limited to linear functions as it uses probability distributions and is not bound to a specific model approach (Timme et al., 2014). Where Pearson correlation is bound to linear dependencies, the mutual information measure captures correlation between the variables in a more generalized, broader sense (Steuer et al., 2002). This leads to their argument that a Pearson correlation value (close to) zero does not fully imply statistical independence between the variables.

Using the equation to understand the level of interaction between three or more variables can be done by grouping variables together as a set and applying it as a single vector against the target variable (Timme et al., 2014). If variables B and C were to be treated together within a set S to measure the mutual information with variable A as outcome Y, it can be rewritten as such:

$$I(Y, S) = \sum p(Y, B, C) \log \frac{p(Y, B, C)}{p(Y)p(B, C)} \tag{2.13}$$

Timme et al. (2014) do mention that a drawback of this approach is that the contribution of each of the variables within the set with regards to the mutual information between the set and the target variable cannot be derived directly. Instead, the calculation should be repeated excluding certain variables from the set at a time to deduce the influence of the presence of a variable in the set towards the resulting level of mutual information.

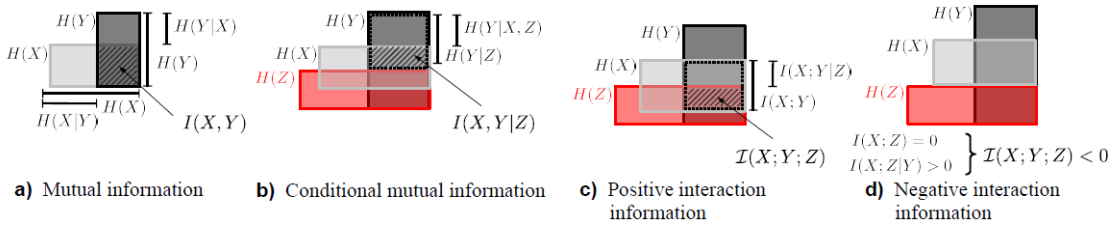
Initially, the numeric value of the calculated entropy may be hard to interpret. Up to now, the only conceptualization given is that a higher level of entropy describes, on average, a higher level of surprise or information gain from knowing the measurement of a variable. Ince et al. (2017) discuss how using the logarithm function with base-2 allows for an interpretation of this numeric value using the unit of bits. In their description, a bit represents a yes/no question. If the distribution is known and a series of questions need to be asked to know the outcome value for a given trial, entropy explains the average number of questions needed to guess that outcome value. Due to the usage of base 2 in the function, a reduction of one bit unit means that the level of uncertainty has been cut by half. An example given is when predicting the roll of a fair die, with entropy $\log_2 6$. If one already received information prior to guessing that the outcome was even, the number of possible outcomes is reduced by half, and the new entropy $log_2 3$ is exactly 1 bit less than $log_2 6$. When discussing mutual information, Ince et al. (2017) compare it to the level of explained variance in linear regression but state that while both are similarly interpreted in terms of context, mutual information is not dependent on the relationship type between variables and can thus be applied in non-linear relationships as well. This confirms with the description of entropy given by Timme et al. (2014). Due to the common scale of bits being applicable on many kinds of relationships between data, a view on mutual information could be to use it as a statistical test to determine independence between variables, similarly to how t-tests and tests of correlation are used (Ince et al., 2017).

2.3.2 Interaction Information

Similarly to how partial correlation is used in network psychometrics in order to account for the other variables present in the model, mutual information between two variables can be conditioned after knowing the content of the third variable (Timme et al., 2014; Cover and Thomas, 2012), which is usually given in the notation form *I(A;B|C)* with a non-negative property and is calculated using the following formula:

$$I(A;B|C) = \sum p(A,B,C) \log \frac{p(C)p(A,B,C)}{p(A,C)p(B,C)} \tag{2.14}$$

The concepts of mutual information and conditional mutual information are probably better explained via data visualization. Venn diagrams have been considered the most popular and useful approach to visualize relations between sets. In this case, each set can represent the existing entropy levels of a variable.



a) Mutual information **b)** Conditional mutual information **c)** Positive interaction information **d)** Negative interaction information

Figure 2.4: Entropy Venn Diagrams showcasing mutual information and interaction information, as found in Runge (2015).

Mutual information can be seen as the overlap of the two 'entropy sets' of variables X and Y, where in the conditional mutual information it is the unique part of overlap between X and Y that is not overlapped by variable Z. It is clear from these two visualizations that both metrics cannot be negative. Nevertheless, interpretation on the possible value shift when calculating the difference between the conditional and unconditional mutual information refers back to McGill (1954) and is referred to as interaction information. Ghassami and Kiyavash (2017) describe it as the multivariate generalization of mutual information. Additional information of a third variable Z may increase or decrease the level of information transmitted between the original variables X and Y. Using example C from Figure 2.4, the overlap of variables X, Y and Z can be notated as *I(X;Y;Z)* and is the remainder of the mutual information between X and Y minus the conditional mutual information of X and Y after conditioning for variable Z. Of course, the role of the variables can be interchanged, meaning that one could start with the mutual information of X and Z and measure it again after conditioning on variable Y; the value of the interaction information would remain the same, given that the entropy levels of the variables do not change. In notation form, this becomes:

$$I(X;Y;Z) = I(X;Y) - I(X;Y|Z) = I(Y;Z) - I(Y;Z|X) = I(X;Z) - I(X;Z|Y) \tag{2.15}$$

This equation allows for *I(X;Y;Z)* to be both positive or negative, which has given the opportunity to explain whether the relationship between two variables is of a synergistic or redundant nature (Timme et al., 2014). In short, synergy would suggest that two variables X and Y, when known together, allow for more information to be gained about a third variable Z compared to using only X and Y separately against Z. On the other hand, redundancy suggests that variables X and Y together deliver the same information of variable Z as when you inspect the information of variable Z explained

by X and Y alone. As the name suggests, when redundancy occurs there is no added value to have both X and Y be present together in a model with regards to the information delivered by X and Y alone. Despite this conceptualization, the properties of positive or (representing) negative interaction information is hard to interpret using Venn diagrams; the paper of Finn and Lizier (2020) addresses these challenges and proposes new compatible measures for those. Using examples C and D in Figure 2.4, Runge (2015) explains that both plots should not be over-interpreted. Overlap between X and Y ($I(X;Y)$) and X, Y and Z ($I(X;Y;Z)$) makes it easy to understand the difference between conditional and unconditional mutual information between X and Y and allows one to understand how the introduction of variable Z (and its entropy set) allows to partially explain the level of interaction and correlation between variables X and Y. On the other hand, when the entropy sets of variables X and Z do not overlap as shown in example D, it is straightforward to denote $I(X;Z) = 0$, but the additional equation $I(X;Z|Y) > 0$ cannot be directly derived from the visualization as there is still no overlap between X and Z. Instead, this should be interpreted as the case where the variables are, when variable Y is not present, unconditionally independent, but with the introduction of variable Y they become conditionally dependent (Runge, 2015). The work of Ghassami and Kiyavash (2017) explains how determining whether the interaction information is positive or negative can determine the skeleton of the DAG for a particular system. For a positive value, each variable is responsible for explaining (partially) the dependency between the two other variables. This is clearly represented in examples A and B in Figure 2.5. Also applicable in example C, variables X and Z are considered independent given the existence of variable Y.

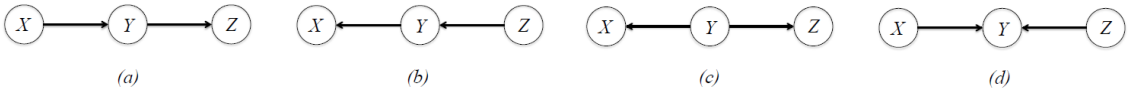


Figure 2.5: Skeleton structures of DAGs of three variables, as found in Ghassami and Kiyavash (2017)

This leads to the conclusion that for examples A, B and C, $I(X;Z|Y) = 0$ while $I(X;Z) \geq 0$, leading to a positive value for $I(X;Y;Z)$. Example D, which shows the conditional dependency of X and Z when Y is introduced, forces the inverse where $I(X;Z) = 0$ and $I(X;Z|Y) \geq 0$, leading to a negative value for $I(X;Y;Z)$. This scenario implies that knowing one variable strengthens the correlation between the two other variables (Ghassami and Kiyavash, 2017). For example D, knowing the value of X allows for a stronger correlation between Y and Z. One could place this in the context of an XOR gate where Y is the outcome variable. While $I(X;Z) = 0$, if Y is determined one knows that the input variable from variable X will automatically determine the input value of variable Z, and thus $I(X;Z|Y) \geq 0$. If Y was not known at the time, variables X and Z were still considered independent. An alternative approach where the value of interaction

information cannot be negative and proves how redundancy and synergy may co-exist together to produce a net result of zero interaction information is discussed in Appendix B and is named the Partial Information Decomposition method from the work of Williams and Beer (2010).

Estimation of the mutual information between two Gaussian variables could be simplified if the assumption of the given joint distribution is bivariate normally distributed (Gel'Fand and Yaglom, 1959). Used in the studies by Kraskov et al. (2004) for improving the estimation of mutual information using a k-nearest neighbours (KNN) approach rather than binning to determine the joint probability density and Ma and Sun (2011) using a variation via copula functions to estimate the mutual information, the equation utilizes the correlation coefficient ρ as such:

$$MI = -\frac{1}{2}\log(1-\rho^2) \tag{2.16}$$

This equation could also be considered compatible for the purposes of better estimating the described psychological network models as the edges are determined via partial correlations. Indeed, both conditional and unconditional mutual information could be calculated by using ρ both from the covariance matrix and its inverse to retrieve the partial correlation, leading to a possible value for the interaction information. A limitation here is that in the high-dimensional setting where the number of observations is lower than the number of variables, the inversion of the covariance matrix is not possible due to the matrix becoming singular. This issue may also occur outside the high-dimensional setting in the situation where one of the variables can be described as a linear function of other present variables, as is the case with collinearity. The matrix then becomes singular as embedded eigenvalues may equal to zero and will, at most, be positive semi-definitive.

2.4 Sparse Models via Graphical LASSO

As the number of interactions in the network may scale very quickly when increasing the number of variables relevant for the study, the number of edges possibly visualized in the network model will grow fast and could most likely hinder the exploratory data analysis phase of the research. Because of this, techniques for penalized regression and dimension reduction will be included within the scope of this review. In particular, the graphical LASSO penalized regression technique will be discussed in more detail as it is considered a broadly accepted approach with included flexibility of customizing its penalty parameter.

As the number of nodes grows and their partial correlations with other nodes may not be easily observed as exact zeroes (Costantini et al., 2019), the risk of having an overfitted model presenting biased results increases due to the model being forced to display all non-zero relationships between two variables. Instead, regularization methods such as LASSO (Tibshirani, 1996) can be used to find a sparse model to avoid this risk, and may also be applied in the setting of high-dimensional data. Because the Gaussian graphical model is based on the inverse of the covariance matrix, the goal of regularization is to estimate a sparse inversion of this matrix. As described in Friedman et al. (2008), if the assumption of the multivariate Gaussian distribution holds for the continuous data used in the model, given the rule that a zero partial correlation between two variables implies conditional independence between the two, introducing a penalty parameter allows to drive other partial correlations further down to zero to leave with only the sparse, strongly relevant partial correlations in the model.

As the lasso regularization technique is used to impose a penalty on the covariance matrix its type is classified as L_1. Using p again to define the number of variables or dimensions present, μ for the mean, S for the sample covariance and C for the inverse of the covariance matrix, the problem defined is one of maximizing a penalized log-likelihood. The following equation as shown in Friedman et al. (2008) describes the partially maximized Gaussian log-likelihood of the presence of the data when accounting for μ:

$$\log detC - trace(SC) - \lambda \sum |C| \tag{2.17}$$

The mechanics of this 'graphical LASSO' (GLASSO) could be summarized as follows. A penalty parameter λ_1 is chosen and multiplied by the sum of absolute values in Θ to compute the total penalty. By using the absolute values, both positive and negative values of partial correlation are treated equally (Epskamp et al., 2018). Using this penalty, the LASSO regularization method maximizes the problem of the log-likelihood function embedding this penalty as shown in Equation 2.17. Important for this problem is that the total sum of edge values originating from Θ is forced to be limited and thus, some edges will have to shrink towards zero to meet this condition. The result will be a sparse inverse covariance matrix and thus a sparser network model that will still attempt to represent the covariance present in the complete data set. Modifying the level of λ_1 changes the sparsity of the network; a higher penalty parameter leads to a sparser model. Typically, the level of λ_1 where all edges shrink down to zero is considered the maximum value for λ_1. Optimization methods to determine a fitting value for λ_1 can be done using the EBIC criteria and is derived from the work of (Chen and Chen, 2008). In short, this criteria is used to tackle the problem of variable selection given the complexity of the model due to the high number of possible covariates.

The optimal value of λ_1 is found where the EBIC criteria is minimized. The result of this regularization method is a sparse network model with non-zero partial correlations to gain better accuracy in model estimation and allow to easily identify strong relationships between variables after penalization. This approach differentiates from the regular LASSO approach as described in Tibshirani (1996) as a covariance matrix derived from a multivariate Gaussian distribution can be used as input.

An alternative method based on GLASSO is introduced by Celik et al. (2014) and focuses on the usage of 'modules', each representing a collection of variables. Compared to the earlier described approach by Friedman et al. (2008) which still considers the usage of (a selection of) original variables in the network model, the 'module GLASSO' (MGL) approach allows for even more aggressive dimensionality reduction in the model. This is shown in Figure 2.6 where only a network between the modules is considered. The motivation of this approach is derived from the idea that networks may be structured and that the independence assumption which drives the usage of regularization methods with penalization such as LASSO may not hold. The remark of Celik et al. (2014) stating networks may be structured refers to the idea that while an edge missing between two nodes may suggest mutual independence, it could be that on a higher-level concept there may be a form of interaction. Variables that are interacting heavily with each other could be bundled together in a module. The interaction between modules, and thus the estimation of conditional independence between modules, is then explored via MGL. This covers a limitation of the Gaussian graphical model where higher-level constructs to represent sets of variables is not made compatible and estimation is limited to the possible large number of edges.

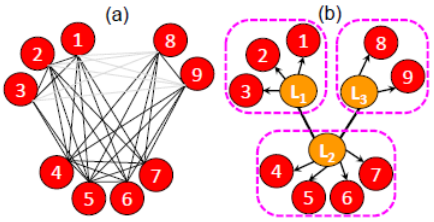


Figure 2.6: Left: a Gaussian graphical model representation using nine variables. Right: a module graphical lasso representation where the nine variables are divided into three modules. Each module is represented by a latent variable L_1, L_2 and L_3 and conditional independencies between modules are estimated by visualizing a network of the latent variables. Found in Celik et al. (2014).

Each module is represented via a latent variable, and MGL attempts to allocate each variable into one of the modules. In order to do so, the values of the set of latent variables L must be estimated first. MGL allows for the usage of the k-means clustering algorithm where cluster centroids will represent the latent variables. It can be used without having a prior assumption of a possible network structure between these latent variables. Because the visual representation of MGL suggests using edges be-

tween the latent variables, it also includes the step of estimating the inverse covariance matrix which is given the notation $\sum_L^{-1}$. Celik et al. (2014) have also provided proof that the estimation of the inverse covariance matrix $\sum_X^{-1}$ with original variables X can be derived when starting from $\sum_L^{-1}$. Once the latent variables have been determined, the Euclidean distance is used to assign original variables into modules.

This approach is interesting in combination with the topic of joint information content as it explores the interactions between variables from a higher level construct. Because the MGL approach also utilizes a partial correlation matrix $\sum_L^{-1}$ with possible estimation of $\sum_X^{-1}$, there is potential to utilize the concepts of mutual information and interaction information on modules. This also allows the opportunity to explore whether joint information content can be differentiated when exploring variables assigned within the same module or across different modules. An alternative method where latent variables are introduced in the Gaussian graphical model is described in Meng et al. (2014), where a distinction is made between global and remaining localized effects to explain the interactions between variables. Such global effects could affect many variables. An example is the oil price on the stock value of many companies which rely heavily on this resource, and the underlying political or geological factors that could affect the resource price. Their motivation is that achieving a sparse model is hindered by the presence of said global effects. Latent variables would capture the correlations of the global effects on the variables, and edges between variables found in the sparse model would suggest the remaining, localized level of interaction. As shown in Figure 2.7, representing global factors through the use of latent variables creates a model where the covariance matrix is not necessarily sparse, but the assumption is made that the number of latent variables is far lower than the original number of variables in the data set. Meng et al. (2014) describe it as a sparse plus low-rank matrix due to the combination of the original sparse matrix and the low rank inherited from the reduced number of latent variables. While this approach does not reference the usage of modules to assign variables together and the estimation of the latent variables is not performed via a k-means clustering algorithm, one could interpret the model approach as such that it might allow for an explanation of a higher-level interaction using latent variables as a proxy.

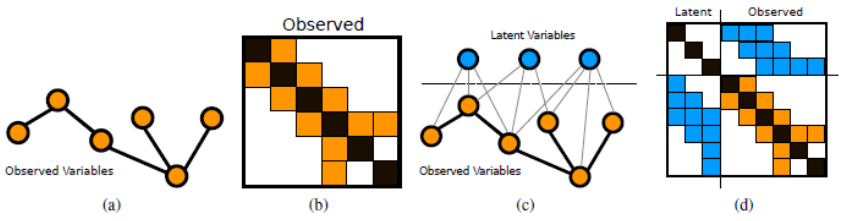


Figure 2.7: Visual representation of the Gaussian graphical model and its inverse covariance matrix (a & b) and the Latent Variable Gaussian Graphical Model (LVGMM) variation (c & d), as found in Meng et al. (2014).

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter will describe how a simulation framework was built in order to generate data from a specific created model and control the level of interaction information for a set of variables within this model. The configurations of the model would then serve as the baseline model or ground truth. Such configurations can be both the adjustment of the level of interaction information to manipulate correlations between given variables as well as including predefined residual correlations between variables from different sets. Once determined, the data generated from this model would be subjected to the graphical LASSO regularization approach in order to create a sparse model. The edges detected via this regularization method can then be visualized as a network model. The main interest is to determine whether spurious correlations are detected after applying graphical LASSO for each predetermined level of interaction information and how these findings change when shifting from a full-redundancy scenario up to a full-synergy scenario, including the case where little to none interaction information exists in the model. Spurious correlations are hereby defined as those found via graphical LASSO but which were not predefined in the baseline model. Detecting spurious correlations for a given level of interaction information included in (parts of) the model could have implications on how dependency networks should be estimated as the presence of one or more spurious correlations may influence the calculation of the node centrality indices.

The chapter is divided into the following sections: first, an overview is given of the model building approach using the R-package *lavaan* (Rosseel, 2012) in order to control the model design, configure the relationships between sets of variables and allow for simulating data sets; second, several model variations are presented to explore how the model design may influence the detection of spurious correlations when using similar levels of interaction information between the same variables in the model; and third, the presentation of the output in terms of average calculated level of interaction information determined for a given sample size of data sets and several KPIs related to spurious correlations found via graphical lasso regularization.

3.2 Modelling Approach

A latent variable approach, which in design may resemble the functionality from the module graphical lasso technique by (Celik et al., 2014), was selected for this study using the R-package *lavaan* (Rosseel, 2012). The idea is that for each latent variable included in the model, a new set of three variables *X*, *Y* and *Z* would be included in the network model. Each set of variables originating from the same latent variable will be referred to as a 'triplet'. The naming convention for the variables in this case would be the combination of the letter representing the latent variable pasted with the name of the variable present within the triplet. This means that for the first latent variable *A*, the variables *A.x*, *A.y* and *A.z* are created.

The implied correlations between the variables within a triplet can be stored in a matrix Σ and are derived from an equation considered central in the field of Confirmatory Factor Analysis (CFA) for which its theoretical background can be found in Bollen (1989). The equation is now shown with each component explained in the following paragraphs:

$$\Sigma = \Lambda \Psi \Lambda^T + \Theta \tag{3.1}$$

In essence, each latent variable in the model is constructed using three indicators (the variables *X*, *Y* and *Z*), each multiplied with a predetermined factor loading λ_1, λ_2 and λ_3. The mapping of which observed variables are loaded using which factors in a CFA model can be done via a matrix named Λ of size *p* x *m* (Bowen and Guo, 2011) with *p* the number of variables and *m* the number of latent variables specified in the model. The transposed matrix Λ^T is used as well in the equation. The matrix Ψ is symmetrical of size *m* x *m* and contains (co)variances of the latent variables which are encoded using ψ's. Finally, the matrix Θ is of size *p* x *p* and contains the residual variances or the variances of the error terms of the created variables *X*, *Y* and *Z* on the diagonal and their residual covariances off the diagonal. Using Equation 3.1, the matrix Σ will contain estimated population variances and covariances with the number of rows and columns equal to the number of observed variables derived from all latent variables in the model (Bowen and Guo, 2011).

Continuing with the example of a latent variable creating a triplet of variables, both the covariance structure and the implied covariance structure from CFA can be demonstrated. The accompanying matrix equation (3.2) is derived from Equation 3.1 and shows how to compute Σ for a given triplet. When using the *lavaan* model syntax, the visualization of the CFA model as shown in Figure 3.1 can be achieved using the R-package *semplot* (Epskamp and Stuber, 2014). Figure 3.2 shows the mapping of the (implied) covariances and were shown in Epskamp (2013).

$$\begin{bmatrix} \sigma_1^2 & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_2^2 & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_3^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} \begin{bmatrix} \psi_{11} \end{bmatrix} \begin{bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \end{bmatrix} + \begin{bmatrix} \theta_{11} & 0 & 0 \\ 0 & \theta_{22} & 0 \\ 0 & 0 & \theta_{33} \end{bmatrix}$$
$$= \begin{bmatrix} \lambda_1^2\psi_{11} + \theta_{11} & \lambda_1\lambda_2\psi_{11} & \lambda_1\lambda_3\psi_{11} \\ \lambda_2\lambda_1\psi_{11} & \lambda_2^2\psi_{11} + \theta_{22} & \lambda_2\lambda_3\psi_{11} \\ \lambda_3\lambda_1\psi_{11} & \lambda_3\lambda_2\psi_{11} & \lambda_3^2\psi_{11} + \theta_{33} \end{bmatrix} \tag{3.2}$$

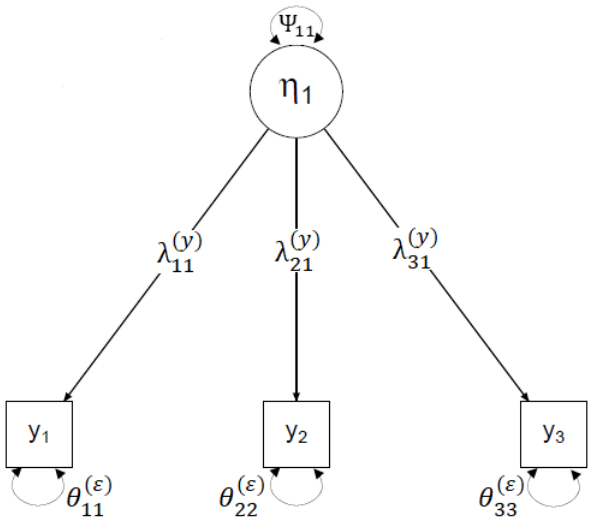


Figure 3.1: Visualization of the CFA modelling approach with mapping of the factor loadings λ_1, λ_2 and λ_3 coming from latent variable 1. ψ_{11} is the variance-covariance matrix of latent variable 1 with only one cell. θ_{11}, θ_{22} and θ_{33} are residual variances of the observed variables with error terms. As shown in Epskamp (2013).

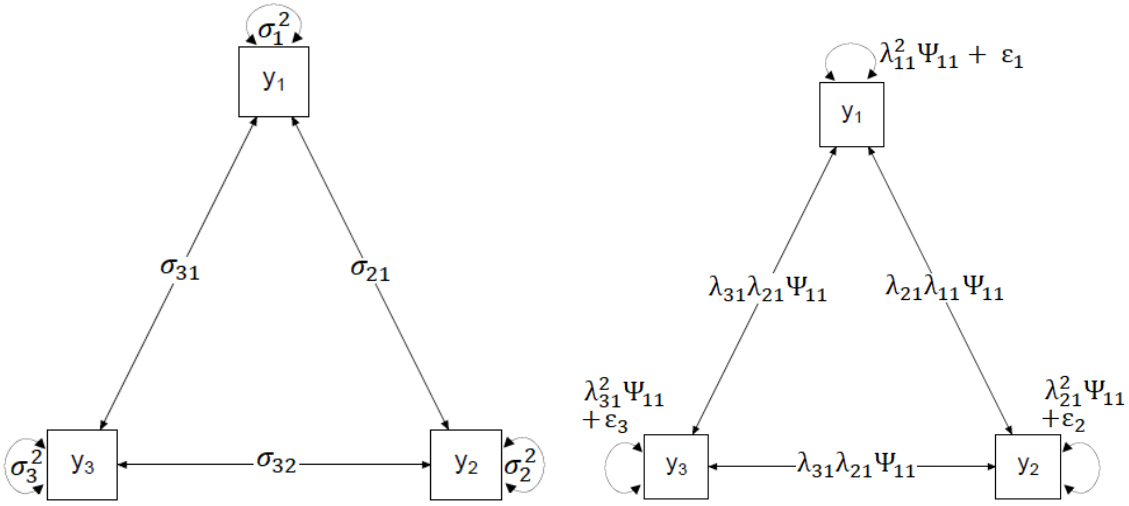


Figure 3.2: Left: covariance structure of the observed variables using the matrix on the left-hand side of Equation 3.2. Right: implied covariance structure using the resulting matrix on the right-hand side of Equation 3.2. As shown in Epskamp (2013).

In Appendix C.1, R-code of the *lavaan* model syntax is shown for the generation of data of a single triplet via one latent variable using the CFA model structure as shown in Figure 3.1. It includes the assignment of the factor loadings λ_1, λ_2 and λ_3 and the calculation of the (co)variances of the X, Y and Z-variables. For this study, the variance of the variable X is left fixed and is dependent on the factor loading λ_1. This is done to allow for the covariance between the other two variables Y and Z, and the underlying variances of variables Y and Z separately, to change. Changing the level of covariance of one pair in the triplet will allow for the level of interaction information to change. For clarification, the calculation of the level of interaction information present within the triplet is based on the difference of mutual information between variables X and Y in the unconditioned case and the conditioned case with the introduction of variable Z. Fixing the variance of variable X is therefore a necessary step. Combined with the factor loadings λ_2 and λ_3, the parameters for the level of variance of the variables Y and Z, named t_2s and t_3s respectively, are determined.

There is one noticeable difference in the model layout when comparing it to Figure 3.1. To achieve the effect of forcing a specific level of interaction information within a triplet, a second factor needs to be created including only loadings related to the variables Y and Z. This factor will be named *A.bf*, with 'bf' referring to the term 'bi-factor model' as each triplet of variables will be constructed using two factors. The loadings of the additional factor will not be the same as used to create the one-factor model for latent variable A. Instead, two new factor loadings l_2s and l_3s will also be indirectly related to the covariance of variables Y and Z. While this additional factor should be included in the model syntax, there should not be a correlation between the two defined factors. This can be forced in the model syntax by mapping the correlation of each factor to one multiplied by itself, and zero multiplied by the other factor.

A visualization of this model using the previous syntax is shown in Figure 3.3 and is also produced via the R-package *semplot*, where the additional factor is renamed to *A.b*. The lavaan model syntax and the creation of the variables *ecor*, t_2s, t_3s, l_2s and l_3s can then be repeated for each new triplet to be introduced in the model. This allows to expand the model with a given number of triplets with no embedded interaction between variables originating from different triplets. Such interaction can also be included in the model syntax if required; the next section of this chapter will discuss how residual correlations are used to create different layouts of the simulation model to explore the role of interaction information further. Having established a simple model including only one triplet, several different configurations of the variables λ_1, λ_2, λ_3 and *ecov* will be used to demonstrate their role on the covariance matrix of the triplet variables. While it is assumed that the factor loadings to construct the latent

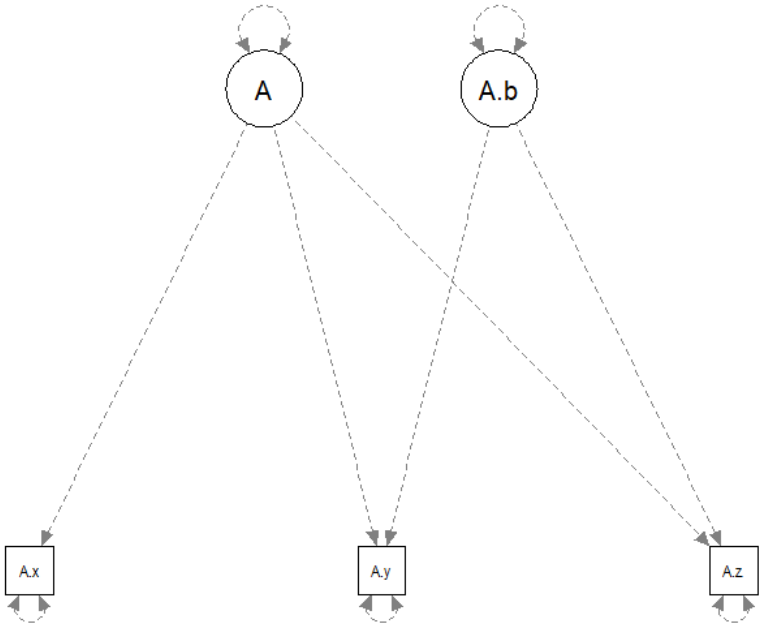


Figure 3.3: Visualization of the bi-factor lavaan model to create a triplet of variables with adjustable level of interaction information between variables X and Y.

variables will be indifferent for each triplet added to the model and will remain fixed once the complete model has been built, the *ecov* parameter may vary per triplet to introduce flexibility in the model and can be changed freely per triplet to set up a new scenario for simulation of the data.

For the baseline model, the factor loadings are fixed at $\lambda_1 = 0.99$, $\lambda_2 = 0.70$ and $\lambda_3 = 0.30$ as these are considered compatible for the model with the introduced flexibility of controlling the level of interaction information per introduced triplet. For the following examples, the *ecov* value is set at a given level so that approximately no interaction information exists within the triplet. The specifics regarding which values *ecov* can hold in the model and how this affects the level of interaction information follows shortly. For now, these examples demonstrate how each loading is primarily responsible for the covariance between one specific variable within the triplet against the two others. The factor loading λ_1 controls the covariances between both the variables *A.x* and *A.y* and between the variables *A.x* and *A.z*; a lower value decreases both covariances, albeit not at the same rate as demonstrated in Tables 3.1, 3.2 and 3.3. Similarly for λ_2, the covariances between *A.y* and the two other variables in the triplet are dependent on the factor loadings as shown in Tables 3.4, 3.5 and 3.6. Factor loading λ_3 does the same for the variable *A.z* as demonstrated in Tables 3.7, 3.8 and 3.9.

As mentioned earlier, the factor loadings λ_1, λ_2 and λ_3 will remain fixed at the specified baseline levels. To have control over the amount of interaction information generated within a triplet between variables *A.x* and *A.y*, only one of the covariances

	A.x	A.y	A.z
A.x	1		
A.y	0.832	1	
A.z	0.545	0.308	1

Table 3.1: Baseline covariances

	A.x	A.y	A.z
A.x	1		
A.y	**0.748**	1	
A.z	**0.490**	0.308	1

Table 3.2: Change in $l_1 = 0.80$

	A.x	A.y	A.z
A.x	1		
A.y	**0.648**	1	
A.z	**0.424**	0.308	1

Table 3.3: Change in $l_1 = 0.60$

	A.x	A.y	A.z
A.x	1		
A.y	**0.944**	1	
A.z	0.545	**0.370**	1

Table 3.4: Change in $l_2 = 0.90$

	A.x	A.y	A.z
A.x	1		
A.y	**0.704**	1	
A.z	0.545	**0.237**	1

Table 3.5: Change in $l_2 = 0.50$

	A.x	A.y	A.z
A.x	1		
A.y	**0.545**	1	
A.z	0.545	**0.150**	1

Table 3.6: Change in $l_2 = 0.30$

	A.x	A.y	A.z
A.x	1		
A.y	0.832	1	
A.z	**0.832**	**0.550**	1

Table 3.7: Change in $l_3 = 0.70$

	A.x	A.y	A.z
A.x	1		
A.y	0.832	1	
A.z	**0.704**	**0.442**	1

Table 3.8: Change in $l_3 = 0.50$

	A.x	A.y	A.z
A.x	1		
A.y	0.832	1	
A.z	**0.315**	**0.115**	1

Table 3.9: Change in $l_3 = 0.10$

between variable *A.z* and another variable within the triplet should change. For this modelling approach, the level of *ecov* controls the covariance between variables *A.y* and *A.z*. This is achieved due to the inclusion of the additional latent variable within the model syntax; unlike the original latent variable, both of its factor loadings are dependent on *ecov* and are only related to *A.y* and *A.z*. Selection of this value should be done with care as it is a requirement with regards to analysis of the amount of interaction information generated for the covariance matrix to remain positive definitive. The next set of tables shows for three predetermined values of *ecov* the changes of the covariance between *A.y* and *A.z*. These chosen values for *ecov* happen to be solid cases to generate redundancy, synergy or zero interaction information within the triplet, excluding the risk of the covariance matrix not being invertible.

	A.x	A.y	A.z
A.x	1		
A.y	0.832	1	
A.z	0.545	**0.308**	1

Table 3.10: Zero interaction, with $ecov = -0.15$

	A.x	A.y	A.z
A.x	1		
A.y	0.832	1	
A.z	0.545	**0.068**	1

Table 3.11: Synergy, with $ecov = -0.39$

	A.x	A.y	A.z
A.x	1		
A.y	0.832	1	
A.z	0.545	**0.678**	1

Table 3.12: Redundancy, with $ecov = 0.22$

Assuming a bivariate normal distribution, the approach by Gel'Fand and Yaglom (1959) is applied using the correlation coefficient between *A.x* and *A.y* in both the conditioned and unconditioned case to measure the level of interaction information. For a single 3x3 matrix, the R-code of the function to calculate the level of interaction information is presented in Appendix C.2.

Using the covariance matrices from Tables 3.10, 3.11 and 3.12 along with the function to calculate interaction information, the values returned are approximately 0.00, 0.58

and -0.17, respectively. Interpretation of the output value is not as straightforward. For instance, unlike in the method suggested by Ince et al. (2017) to use a base-2 logarithm to allow a unit of bits to be used when calculating possible differences in entropy from observing a measurement, a natural logarithm is used instead. Negative values significantly distant from zero indicate a presence of redundancy while significantly positive values suggest synergy.

Even with a simple model consisting of only one triplet of variables, the influence of the level of interaction information can be demonstrated. Data based on the *lavaan* model can be generated to be subjected against the *glasso* regualization method. The influence of the level of interaction information on the results from *glasso* are visible when visualizing the related network model. For the visualization part of the workflow, the R-package *qgraph* (Epskamp et al., 2012) is used. In Figure 3.4, using the same values for *ecov* to force either synergy, redundancy, or zero interaction information as mentioned in Tables 3.10, 3.11 and 3.12, there are noticeable differences in terms of the strengths of the remaining partial correlations between variables and the shape of the network. It should be noted that the goal is not to represent or match the covariance matrices derived from the *lavaan* model; the *qgraph* visualization focuses on partial correlations found via a sparse inverse covariance matrix using the regularization method. This means that the edge weights found could indeed deviate from those observed in the original covariance matrix. The edge weight between variables *A.x* and *A.y* remains present and relatively consistent across the three cases. Examining from the direction of synergy towards redundancy, the edge weight between variables *A.y* and *A.z* steers more towards a stronger positive value. An opposite trend is found for the edge weight between variables *A.x* and *A.z* between the cases of synergy and zero interaction information; for the redundancy case, it is missing from the network. The structure of the network from the redundancy case could now be described better as a 'chain' rather than a fully connected cycle of a triplet. Without knowing the full details of the edge weights, other than knowing both are fairly positive by inspecting the visualization, explaining how the presence of redundant information found between variables *A.x* and *A.y* changed the network layout can be done as follows: as *A.y* is both positively correlated to *A.x* and *A.z* in this network, the unique information gained between *A.x* and *A.y* is reduced, because part of the relation with *A.z* will be embedded in *A.y*. As there is no other way of learning how *A.z* behaves with other variables, in this case because the edge with *A.x* is missing, redundant information is unavoidable as part of the information found between *A.y* and *A.z* will inevitably be present when exploring the information gained between *A.x* and *A.y*.

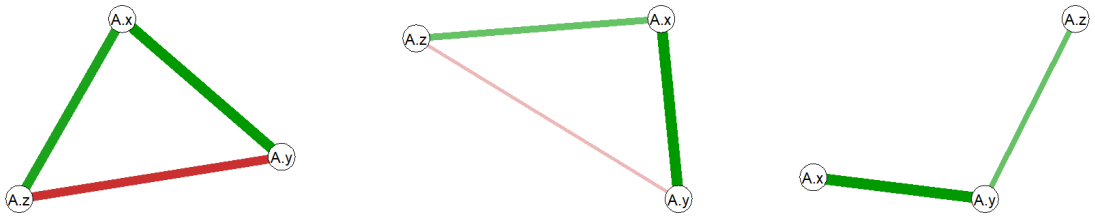


Figure 3.4: *qgraph* visualizations of the triplet. Both color intensity and edge thickness represent edge weight. Red indicates negative edge weight, green indicates positive edge weight. Left: synergy; Middle: Zero interaction information; Right: redundancy.

In the case of synergy, comparing the heavy edge weights of *A.x* to *A.z* and *A.y* to *A.z* and observing the difference of the sign, knowing both of these opposite correlations due to the presence of variable *A.z* brings more information about the characteristics of the positive relation between *A.x* and *A.y*, allowing for synergy to be present due to the introduction of *A.z*. In the case of zero interaction information, it can be noticed the two edges towards *A.z* both edges are steering towards the value of zero. For this particular set of edge weights found, it is inconclusive whether the relationship between *A.x* and *A.y* gives either more or less total insight given the presence of *A.z*.

The model can be expanded to include more triplets of variables, each assigned with the same factor loadings λ_1, λ_2, λ_3 for the latent variables and with the possibility to assign a different *ecov* value per triplet. Using the *lavaan* model syntax, the choice can be made which variables from one triplet may interact with those from other triplets by using residual correlations. The 'residual' part comes from the fact that each indicator was regressed on latent variables. The model expects a set of relationships, in this case edges, between the observed variables and only between those regressed from the same latent variable. As the correlation matrix from the data may differ from what is defined in the model, residual correlations may exist between other variables. The value of these correlations are forced into the model via the *lavaan* model syntax to create an edge between one variable regressed from one latent variable (or originating from one triplet) to another variable regressed from a second latent variable (or originating from another triplet).

Figure 3.5 shows how three triplets, with each triplet displaying either synergy, redundancy or zero interaction information, may have edges between them by forcing residual correlations between variables from different triplets. It is expected to inspect the edges in the network visualization of those representing the forced residual correlations in the model along with those between the observed variables originating from the same latent variable; however, spurious correlations between variables from different triplets may be observed as well that were not inherently programmed. An example of such spurious correlation could be the one between variables *A.z* and *C.x*. This modelling approach will therefore be used to understand the role of interac-

tion information on the presence of such spurious edges, and how it may impact the network estimation techniques used with regards to network centrality indices. The R-code to create such model using *lavaan* is shown in Appendix C.3.

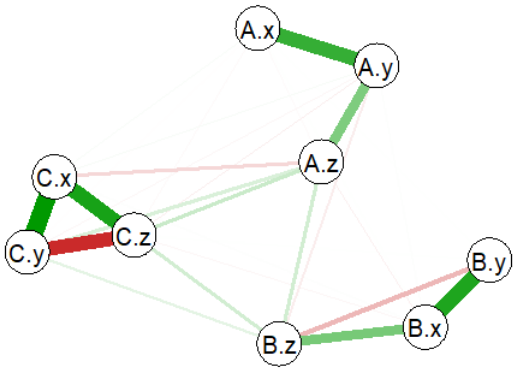


Figure 3.5: *qgraph* network visualization of three triplets with residual correlations between triplets. Triplet A includes redundancy; triplet B includes zero interaction information; triplet C includes synergy. The primary link between the triplets are done via the variables *A.z*, *B.z* and *C.z*. Residual correlations were set at 0.15.

A representation of the covariance matrix programmed in via the *lavaan* model syntax is shown in Table 3.13. The cursive values refer back to the demonstration of the influence of the *ecov* and represent the same values as shown in Tables 3.10, 3.11 and 3.12. The bold values are the newly introduced forced residual correlations between the *z*-variables from each triplet.

	A.x	A.y	A.z	B.x	B.y	B.z	C.x	C.y	C.z
A.x	1								
A.y	*0.832*	1							
A.z	*0.545*	*0.678*	1						
B.x	0	0	0	1					
B.y	0	0	0	*0.832*	1				
B.z	0	0	**0.150**	*0.545*	*0.308*	1			
C.x	0	0	0	0	0	0	1		
C.y	0	0	0	0	0	0	*0.832*	1	
C.z	0	0	**0.150**	0	0	**0.150**	*0.545*	*0.068*	1

Table 3.13: Covariance matrix representation of the example displayed in Figure 3.5 including the residual correlations as included in the *lavaan* model syntax in bold.

A limitation found with this modelling approach is the relatively weak residual correlations made possible when forcing synergy on each triplet and allowing for each triplet to be connected with at least one other triplet. For stable simulation runs with the factor loadings and *ecov*-values presented, residual correlations between *z*-variables are considered the most reliable, albeit with a maximum correlation level of 0.10. Raising these correlation levels causes the covariance matrix to not be positive definitive anymore. The less synergy is introduced within triplets in the model, the higher these residual correlations may be. Given the previous example using a cluster of three triplets connected to each other via their *z*-variables, in the case of

maximum redundancy allowed within the model for each triplet the residual correlations can be raised up to approximately 0.35. Any case between full synergy and full redundancy will accept maximum correlation levels between these two values. It has been demonstrated in the previous example that residual correlations of 0.15 are compatible within the model when using the combination of having one instance of synergy, redundancy and zero interaction information within a cluster of three interconnected triplets. To allow the feature of choosing per triplet the level of interaction information, any residual correlations added will be kept at the maximum level for which the model remains stable in the case of maximum possible generated synergy for each triplet.

Adding residual correlations between the *z*-variables of different triplets has shown to be the most stable and compatible form of adding additional forced correlations in the simulation framework model. Using the same approach for *x*- and *y*-variables, in the case of forcing the maximum allowed level of synergy per triplet, residual correlations between *x*-variables are not considered applicable in the model while a level of 0.04 maximum is recommended for residual correlations between *y*-variables. Continuing with possible combinations between the *x*-, *y*- and *z*-variables, only the residual correlations between a *y*-variable from one triplet and a *z*-variable from another triplet shows the most potential to be added in a model where maximum synergy is applied. Between all triplets, this allows for six additional residual correlations that all together can be used at a level of 0.06. These additional residual correlations can be used in combination with the ones using a pair of *z*-variables if the latter ones are reduced to a level of 0.05 maximum. Subsetting the possible combinations of residual correlations also allows for some flexibility. An example could be to have residual correlations between *z*-variables at 0.10 while using only a subset of possible residual correlations between *y*- and *z*-variables fixed at 0.06 for a layout using three triplets, each included with the maximum allowed level of synergy.

In the case of forcing maximum allowed redundancy within the model for each triplet, for this particular example the residual correlations between the *y*- and *z*-variables can be raised up to 0.23 while residual correlations between *z*-variables can be set to 0.36. Again, cases with mixed levels of interaction information per triplet allow for residual correlations with a level between these two extremes. Nevertheless, even with relatively weak residual correlations in the full synergy case, the glasso regularization method still detects spurious correlations that can be visualized within the network. A remaining question is whether different spurious correlations may be detected when changing the level of interaction information within one triplet, while keeping all other factors in the model constant. Using the *qgraph* package, an edgelist can be retrieved for each resulting network after applying the glasso regularization

method. Filtering out those edges that were expected to be included in the network, such as those occurring between variables of the same triplet and between variables that were used to force residual correlations into the model, results in a list of possible spurious correlations. Some of these may still be approximated close to zero and may be considered negligible; others can be considered significant and its existence and, possibly, strength is therefore influenced by the level of interaction information across triplets and the particular interaction given to them via residual correlations.

In summary, the complete workflow of the code can be described as follows: (1) a combination of compatible factor loadings and *ecov* values are prepared to create the variables needed for the *lavaan* model syntax per triplet; (2) the model syntax is written; (3) for a fixed number of repetitions (1000 for this approach) and a fixed number of the sample size of the generated data from the model (200 for this approach), for each triplet the level of interaction information is computed and the average is taken over all iterations to conclude the expected level of interaction information; (4) for each repetition a network model with glasso regularization applied is created and the complete edgelist stored in a table; and (5) both the individual edgelists from each repetition and the aggregated count of edges found from all repetitions are used to compute several key performance indicators. For each repetition, sensitivity and specificity are defined using the logic of knowing which edges were or were not expected to compute true positives and negatives along with false positives and negatives. This method has also been used in Epskamp (2016). Furthermore, the percentage of edges found which were (non-)programmed are also computed. As will be explained in the next section, the percentage of edges found within or between clusters (a collection of interconnected triplets) is also computed. As for the aggregated count of edges found, this list can be filtered based on whether the edges were (not) programmed in the model, or were occurring either within or between clusters. An aggregation of the average edge weight is also taken into account.

3.3 Model Variations

Having discussed the possible flexibility of the layout of the model via the mapping of the residual correlations, several model variations are now presented. The goal of introducing multiple layouts is to discover whether for each layout or group of similar layouts there is a noticeable pattern regarding the existence and possibly the weight of spurious correlations when changing the level of interaction information for one or more triplets. For this purpose, each of the suggested models has been scaled up to include between nine and twelve triplets. All of the following layouts will use one particular set of residual correlations, meaning that only residual correlations are used

between *z*-variables, or those only between *y*- and *z*-variables. A careful consideration has been made to have each triplet be connected to a fixed number of other triplets. The following layout designs show how each node from a triplet can be connected to two, three or four nodes originating from other triplets. A null model is also included for control of the experiment; this model contains no residual correlations, making each triplet independent. No spurious correlations are to be expected from this model regardless of the level of interaction information chosen per triplet.

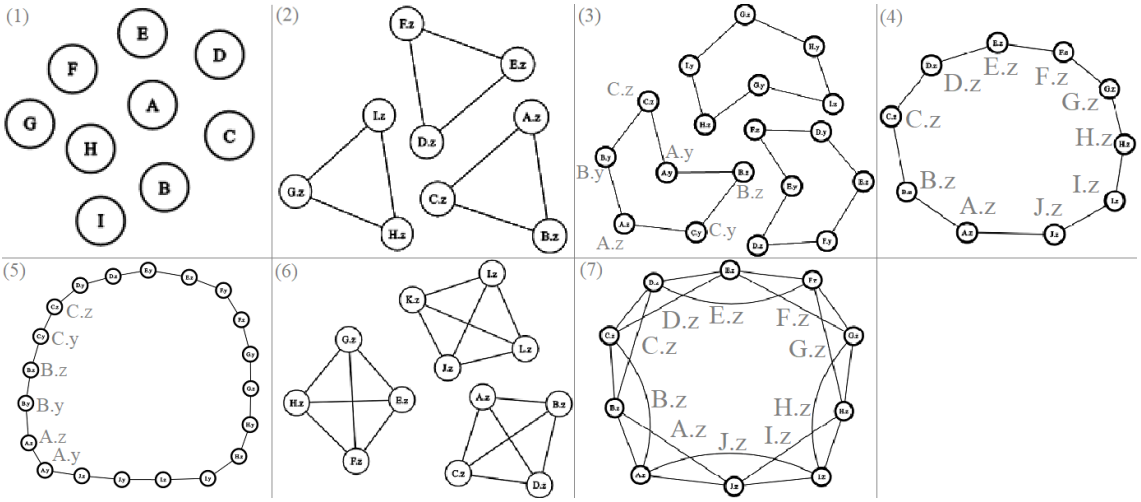


Figure 3.6: Visual overview of the presented model layouts.

Layout	Description
1	Null Model. Nine triplets, all considered independent. No interaction is expected between triplets.
2	Three clusters of three triplets. All triplets within a cluster are connected to two others via their *z*-variables.
3	Three clusters of three triplets. Each *z*-variable from one triplet is connected to the two *y*-variables from the other triplets.
4	Closed loop structure of 10 triplets in which each triplet connects to two others in a given sequence via their *z*-variables.
5	Closed loop structure of 10 triplets in which each triplet connects to two others in a given sequence via the *z*-variable of one triplet to the *y*-variable of the next one. Connections between the *y*- and *z*-variables within the same triplet are visualized to show the loop.
6	Expansion of Model Layout 2. Three clusters of four triplets. Each triplet within a cluster is connected to the three other related triplets via the *z*-variables.
7	A double closed-loop structure using the *z*-variables from each triplet. Following the alphabetic sequence, each triplet is connected to the two previous and the two next triplets in the sequence.

Table 3.14: Description of model layouts shown in Figure 3.6.

The R-codes for each model layout including the workflow as described earlier are found in Appendices D.1 and D.2. The levels of the residual correlations per model correspond to the maximum allowed in the case of forcing each triplet to include the maximum possible amount of synergy within the model.

CHAPTER 4

RESULTS

4.1 Main Results

For each model, twelve different threshold levels were applied when performing graphical LASSO regularization on the simulated data, ranging from 0.005 to 0.05 in steps of 0.005, including two Boolean values *True* and *False*. This was repeated for each 'case', referring to the setting of forcing either only synergy on all triplets, only redundancy on all triplets or no interaction information at all. Combining this with the seven different model layouts, for each combination of model layout, case and threshold level a simulation run was performed with 1000 repetitions. This brings a total of $7 * 3 * 12 * 1000 = 252.000$ repetitions for this study. For each unique simulation setting, the following KPIs were computed: (1) size of the model in terms of number of edges; (2) sensitivity and specificity based on the presence or absence of network edges that were (not) programmed; (3) percentage of edges present in the model which were expected; (4) for applicable Models 2, 3, and 6, the percentage of edges present within a cluster. Referring to Figure 3.6, a cluster in this context can be described as a separate unit of forced interconnected triplets via residual correlations; all applicable models have three clusters. For these models, this allows a further breakdown of potentially found spurious edges by whether these are present within a cluster or between variables from different clusters. Due to the extensive amount of summary statistics of those KPIs for each simulation setting, boxplot figures and their summary statistics are available in Appendix E.1. Across all models, general trends for sensitivity are fairly comparable when going through each threshold level setting in ascending order. Each model showcases how switching between cases changes both the initial sensitivity levels and influence of the threshold level on said levels. Sensitivity is considered lowest in the zero interaction case but remains approximately stable across all threshold levels, with the exception of the *True* Boolean value for the limit argument. For either the full synergy or redundancy case, sensitivity slowly decreases with each threshold level raised. Specificity hovers very closely to 1 in all models for the zero interaction and full redundancy case. Regarding the synergy case, all models follow the same pattern where at lower threshold levels specificity is

significantly below 1 but is most often closely reached starting from level 0.03. The percentage of edges found which were expected increase for each increment of the threshold level across all cases. This is to be expected; intentionally programmed or expected edges should have fairly strong partial correlations which would not shrink down to zero after applying regularization while spurious correlations are, in comparison, considered much weaker and likely to shrink down to zero. Starting percentages of programmed or expected edges found do differ across the cases for each model. The most notable difference is for the synergy case, where said starting percentages are well below 30% for all models except for Model 6. Higher threshold levels show that for the zero interaction and redundancy case, the percentage is steering closely towards 100%; for the synergy case, this does only happen when the Boolean value *True* is used in Model 1. A similar analysis is done regarding the percentage of edges found within a cluster, applicable to Models 2, 3, and 6. The trends shown from these results are fairly comparable to the ones shown for the percentage of expected edges found for all cases. Finally, considering the size of the model, as most spurious edges were found in the synergy case for all models the trends seen from the boxplots are considered logical. Network size decreases for each increased threshold level in all cases, where relatively little size shrinkage is found for the zero interaction and full redundancy case when comparing results of the two threshold extremes.

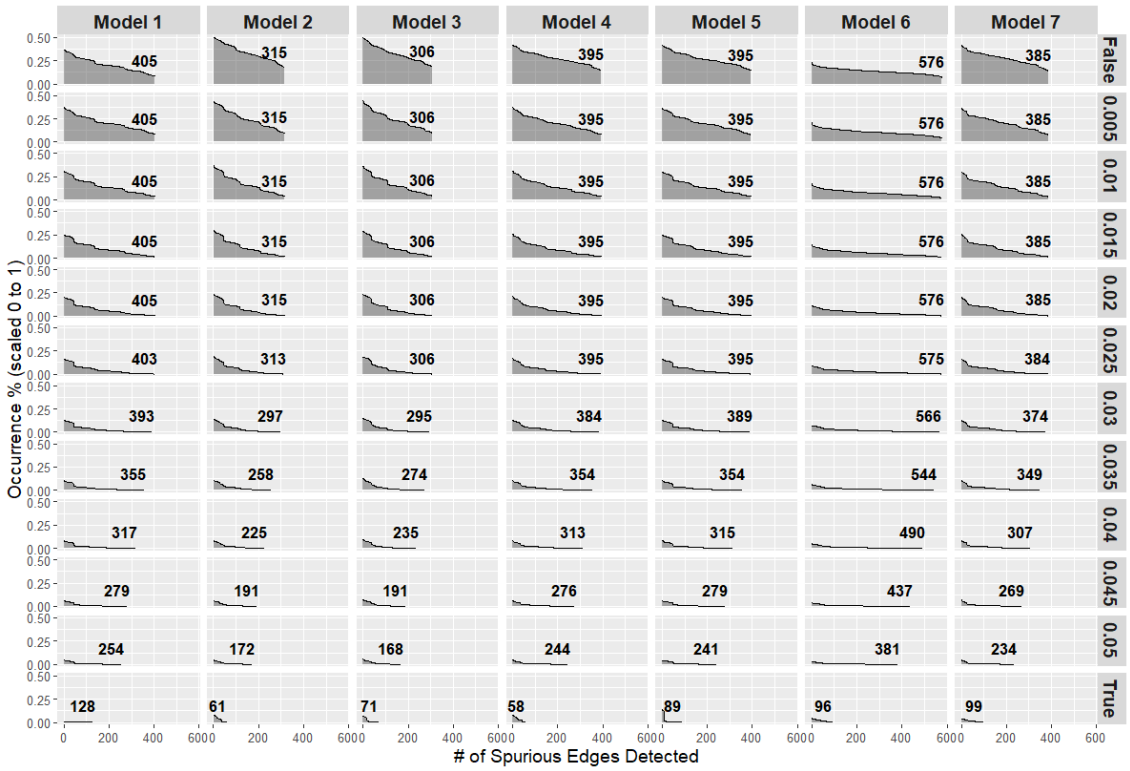


Figure 4.1: Area chart of discovered spurious edges in full synergy case. Split by threshold level (rows) and models (columns). Height of area represents % of occurrence for each edge. Edges are sorted descending by occurrence. Number shown is number of spurious edges found (width of area).

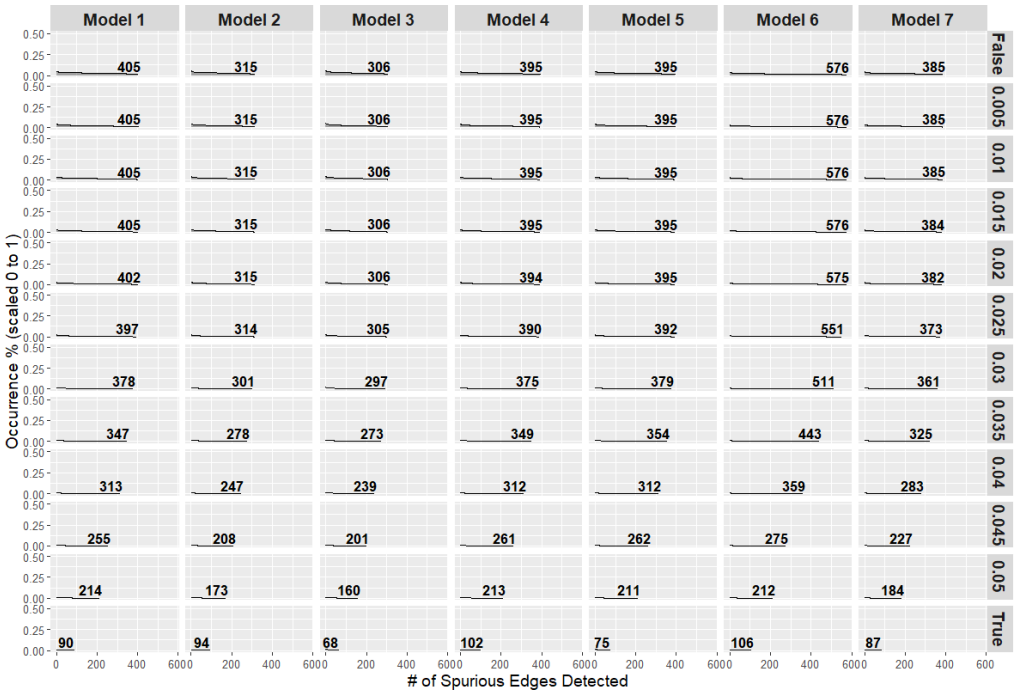


Figure 4.2: Idem, for the case of zero interaction.

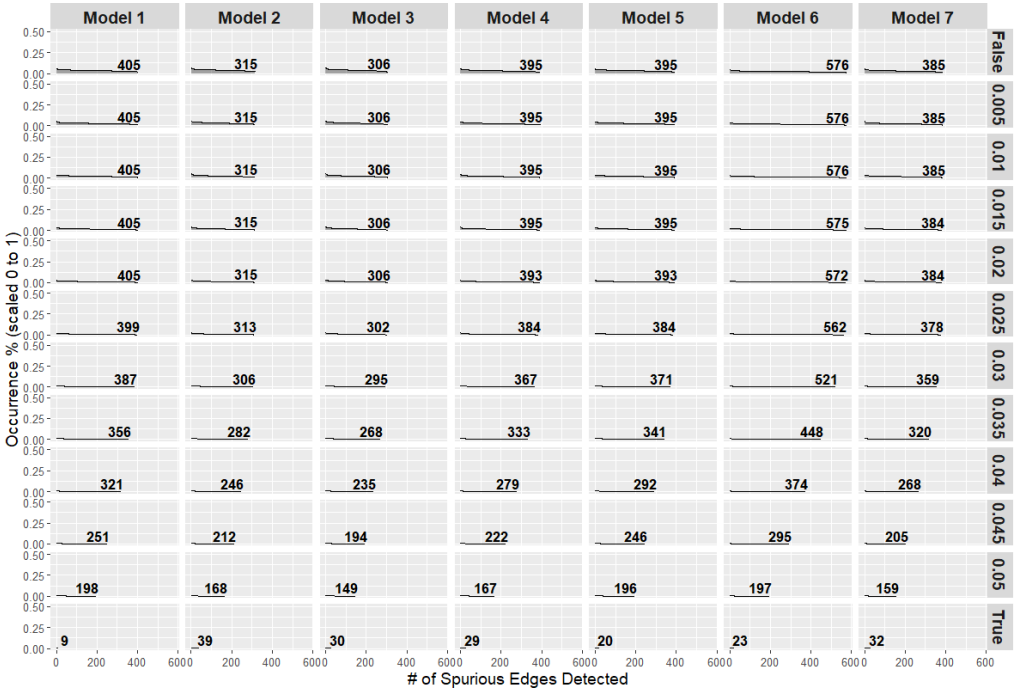


Figure 4.3: Idem, for the case of full redundancy.

Figures 4.1 to 4.3 showcase the number of spurious edges detected per set of configurations for the simulation along with the occurrence percentage per edge found across the 1000 simulations. At first glance the number of edges found across the cases are fairly similar for the lower threshold levels. Only from the threshold level of 0.03 and beyond, less spurious edges are detected in the synergy case. A similar trend can be said for the two other cases, where starting from 0.03 there are clear differences to be seen regarding the total number of edges found with each level increment. The

most noticeable observation is the much higher occurrence percentages found in the synergy case. It also showcases the different characteristics each model may hold, judging from comparing the shapes of the area per model. Several bumps are seen in the chart in the synergy case, most visibly for Models 2, 3 and 5 when starting from threshold level 0.01. These bumps suggest a sudden sharp decline in occurrence percentage, suggesting spurious edges in the synergy case could be categorized further. This is done by splitting these up into '*path types*'. For example, an edge between variable *A.x* and *B.y* is considered a *X-Y* path type. Because triplets were used, the path types possible are *X-X*, *X-Y*, *X-Z*, *Y-Y*, *Y-Z* and *Z-Z*. Mapping the range of occurrence percentages of each path type per model per case per threshold level is done in Appendix E.2. Because mapping the data per model, case, threshold level and path type leads to numerous tables, a summary is given for the occurrences of each path type per model in the synergy case and when applying no threshold.

Model	X to X	X to Y	X to Z	Y to Y	Y to Z	Z to Z
Model 1	9.90 (n=45)	14.05 (n=90)	19.60 (n=90)	20.20 (n=45)	26.85 (n=90)	34.00 (n=45)
Model 2	20.25 (n=36)	26.65 (n=72)	31.80 (n=72)	34.70 (n=36)	41.00 (n=72)	47.50 (n=27)
Model 3	20.30 (n=36)	25.55 (n=72)	31.25 (n=72)	32.95 (n=36)	40.10 (n=54)	46.60 (n=36)
Model 4	16.50 (n=45)	22.15 (n=90)	26.35 (n=90)	28.60 (n=45)	32.90 (n=90)	39.70 (n=35)
Model 5	16.40 (n=45)	21.70 (n=90)	25.60 (n=90)	27.50 (n=45)	32.60 (n=80)	38.70 (n=45)
Model 6	9.20 (n=66)	11.70 (n=132)	13.40 (n=132)	14.10 (n=66)	16.50 (n=132)	18.95 (n=48)
Model 7	16.70 (n=45)	22.00 (n=90)	27.80 (n=90)	29.30 (n=45)	32.60 (n=90)	39.00 (n=25)

Table 4.1: Median occurrence percentages (scaled 0-100) per model & path type in the full synergy case with no threshold. Includes number of uniquely related edges.

A common trend for all models is the increasing median occurrence percentages when going through the path types horizontally in Table 4.1; edges of path type *Z-Z* are more commonly found in all models compared to all other path types, despite the fact that the number of unique edges to be identified of this path type are considerably smaller compared to path types *X-Y*, *X-Z* and *Y-Z* across all models. Similarly, edges of path type *X-X* are found least frequently across all models. Because the edges are sorted in descending order of occurrence, the first bumps in the area charts seen in Models 2, 3 and 5 refer to the jump of edge occurrences between path types *Z-Z* and *Y-Z*; the second bump of edge occurrences between path types *Y-Z* and *Y-Y*. Taking Model 2 and 3 separately, their only difference regarding model layout is the configuration of forced residual correlations. Model 2 uses residual correlations using *Z-Z* type edges for triplets within a cluster; Model 3 uses *Y-Z* type edges instead. Despite this, no significant differences can be observed regarding median occurrence percentages of each path type when comparing both models. Similarly for Models 4 and 5 whose layouts form a closed-loop structure between all the triplets, there are minimal differences in occurrence percentages across all path types. This leads to another discussion point addressed further in the next section regarding the effect of the forced residual correlations and the effect of forcing different variations of a generic model layout.

As seen from Figures 4.2 and 4.3, occurrence percentages for spurious edges of all path types drop dramatically for each model, with the range between minimum and maximum occurrence percentages across path types per model being much smaller. In both cases, the occurrence values do not increase per path type when ordering them as shown in Table 4.1; values for *X-Z* are higher compared to *Y-Y*, the difference being greater in the full redundancy case. When only looking at Models 2 & 3 or Models 4 & 5 separately, no significant differences in edge occurrences per path type can be discovered.

Model	**X to X**	**X to Y**	**X to Z**	**Y to Y**	**Y to Z**	**Z to Z**
Model 1	0.70 (n=45)	0.90 (n=90)	1.30 (n=90)	1.00 (n=45)	1.60 (n=90)	2.20 (n=45)
Model 2	0.80 (n=36)	0.95 (n=72)	1.50 (n=72)	1.10 (n=36)	1.80 (n=72)	2.40 (n=27)
Model 3	0.70 (n=36)	1.00 (n=72)	1.50 (n=72)	1.20 (n=36)	1.70 (n=54)	2.55 (n=36)
Model 4	0.70 (n=45)	0.90 (n=90)	1.30 (n=90)	0.90 (n=45)	1.50 (n=90)	2.10 (n=35)
Model 5	0.70 (n=45)	0.90 (n=90)	1.30 (n=90)	1.10 (n=45)	1.40 (n=80)	2.10 (n=45)
Model 6	0.50 (n=66)	0.60 (n=132)	0.90 (n=132)	0.70 (n=66)	1.00 (n=132)	1.50 (n=48)
Model 7	0.70 (n=45)	0.80 (n=90)	1.10 (n=90)	1.00 (n=44)	1.30 (n=90)	1.80 (n=25)

Table 4.2: Idem, for the zero interaction case with no threshold.

Model	**X to X**	**X to Y**	**X to Z**	**Y to Y**	**Y to Z**	**Z to Z**
Model 1	1.40 (n=45)	1.00 (n=90)	1.90 (n=90)	0.80 (n=45)	1.40 (n=90)	2.30 (n=45)
Model 2	1.60 (n=36)	1.30 (n=72)	2.10 (n=72)	0.90 (n=36)	1.60 (n=72)	2.70 (n=27)
Model 3	1.50 (n=36)	1.20 (n=72)	2.10 (n=72)	0.85 (n=36)	1.70 (n=54)	2.50 (n=36)
Model 4	1.20 (n=45)	1.00 (n=90)	1.60 (n=90)	0.70 (n=45)	1.30 (n=90)	2.20 (n=35)
Model 5	1.20 (n=45)	1.10 (n=90)	1.80 (n=90)	0.70 (n=45)	1.50 (n=80)	2.10 (n=45)
Model 6	0.90 (n=66)	0.70 (n=132)	1.10 (n=132)	0.60 (n=65)	0.95 (n=132)	1.60 (n=48)
Model 7	1.10 (n=45)	1.00 (n=90)	1.45 (n=90)	0.70 (n=44)	1.10 (n=90)	2.00 (n=25)

Table 4.3: Idem, for the full redundancy case with no threshold.

So far, the focus of the presented results are related only to the frequency of the edges found in the network. More interesting is to combine both the occurrence percentage and average edge weight metrics to observe whether relatively highly occurring spurious edges carry, in proportion to other spurious edges, heavier weights. Optionally, for Models 2, 3 and 6 data from spurious edges could be split up into two groups; those found within and between clusters. The full synergy case is used as baseline due to the significantly higher occurrence percentages observed per path type for all models compared to the zero interaction and full redundancy case. Three main trends have been found between the two metrics in the full synergy case across all models and path types when iterating over the various threshold levels. For each related model and path type, these trends will be compared to the two other cases.

From hereon out, the focus will be set on one model at a time. Creating a scatter plot of the average edge weight and occurrence percentage allows to have each data point represent a specific edge between two variables. Splitting this data up into the aforementioned path types and threshold levels, there are three noticeable patterns to be found when scrolling through the plots in order of increasing threshold level. Common for all patterns is that the median occurrence percentage per edge decreases as

increasing threshold levels exclude more edges from the resulting networks. The first pattern shows a clustering of data points spread out in terms of occurrence but for which the average edge weight lies close to zero. At first sight there seems to be a balanced distribution in terms of positive and negative edge weights. As the threshold level increases, the data points are shifted to the left as the occurrence percentage drops. The range of observed average edge weights starts to widen. For those data points averaging significantly below or above zero, this could be explained by the average being computed of very few occurrences of an edge with weights well above the threshold with consistent sign (either positive or negative). Nevertheless, as the data points spread out some remain close to an average edge weight of zero. This seems to suggest the presence of an element of randomness in the edge weights for those edges across the repetitions in the simulation. As the threshold level rises, the number of data points starts to shrink as no occurrences of edges are captures with an absolute weight greater than the threshold. Occurrences are mostly no greater than 1% and often as low as 0.1%. This means, despite the larger edge weights detected even with high threshold levels, with such low occurrence percentages and few data points left it becomes improbable to find a pattern that may explain its influence on the estimation of network models. As for the last automatic threshold setting *True*, for some models and cases the number of data points or the spread on the average edge weights may be in contrast with the observed pattern.

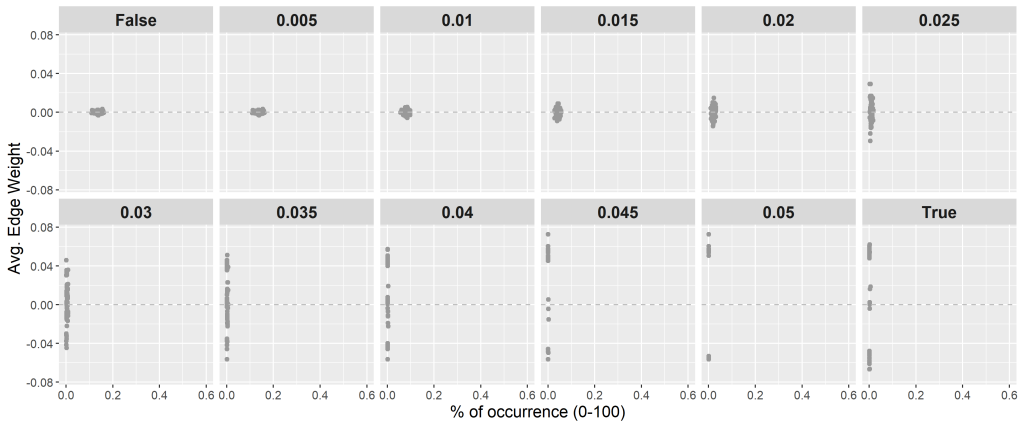


Figure 4.4: Scatter plots of occurrence percentage versus average edge weight for edges found in Model 1 with path type X-Y. Showcases the first described trend ("*A*") in result analysis.

The second observed trend is, when within a given path type, two distinct groupings of data points can be identified. Using Figure 4.5 as reference, the grey data points form one grouping and show a similar trend as explained earlier. The new grouping of data points, marked in black, start off with a different occurrence percentage, further away from the x-axis. As the threshold level increases, the data points remain closely together towards lower average edge weights. This shows that for, at least, a majority

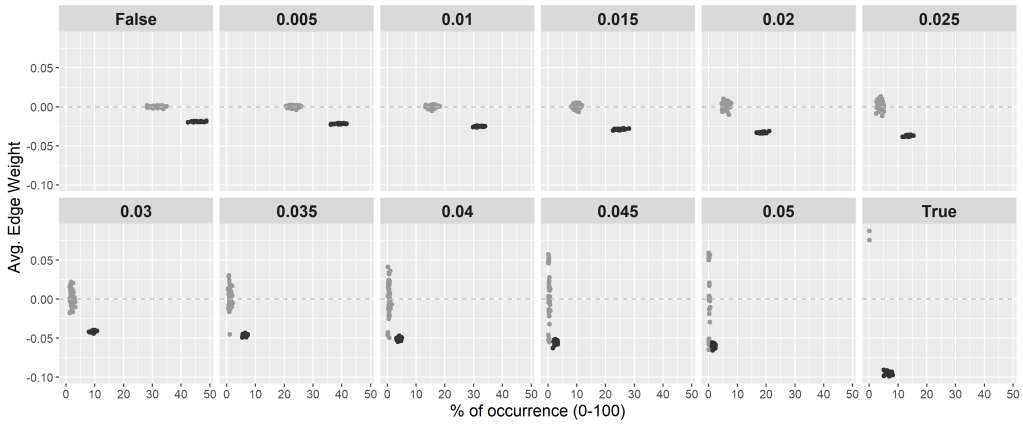


Figure 4.5: Idem, but for edges found in Model 2 with path type X-Z. Data points in black showcase the second described trend ("*B*").

	X-X	X-Y	X-Z	Y-Y	Y-Z	Z-Z
Model 1	A	A	A	A	A	A
Model 2	**B**	**C**	**B**	A	**C**	A
Model 3	**B**	**B**	**B**	**C**	A	**C**
Model 4	A	A	**B**	A	**C**	A
Model 5	A	**B**	**B**	A	A	A
Model 6	A	A	**B**	A	**C**	A
Model 7	A	A	**B**	A	**C**	A

Table 4.4: Trend types found per model and path type in the full synergy case.

of observations these edge weights were negative when present and significantly distinct from zero. The last threshold level shows show the data grouping has deviated from the trend, showing heavier edge weights and increased occurrence. Figure 4.6 shows the third, opposite observed trend where the average weight of the data points increases per threshold level raised. The next step is to seek common characteristics of edges found within such data groupings and how the metrics change in the other cases. For applicable models, it will be explained whether these trends are exclusive to edges discovered either within or between clusters.

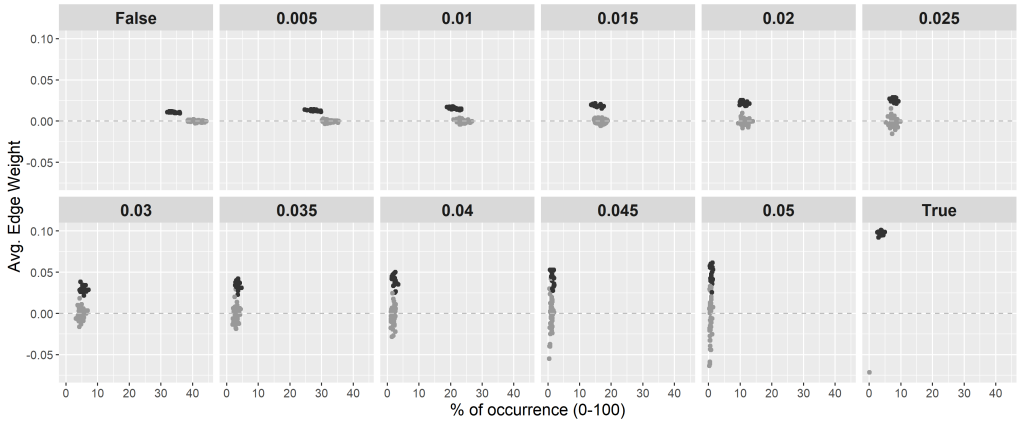


Figure 4.6: Idem, but for edges found in Model 2 with path type Y-Z. Data points in black showcase the third described trend ("*C*").

A good variation of trend discoveries is captured across all models and path types as seen in Table 4.4. For Models 2, 3 and 6 including clusters, when splitting the data points of edges into those occurring within a cluster or between clusters, the data points showing either trend 'B' or 'C' all came from edges within a cluster. Using Figures 4.5 and 4.6 with data from Model 2, the black data points represent edges within a cluster and the grey data points of those between clusters. Iterating per path type, for those models where either trend 'B' or 'C' was found, an overview is presented of condensed data for each case using a coordination system where the median of the occurrence percentages and average weights are taken for the grouping of related edges. In Models 2 and 3, the available data points for path type *X-X* that form a separate data grouping concerns all possible *X-X* edges within a cluster. As a cluster consists of three triplets, a maximum of three *X-X* edges between the three X-variables can exist, e.g. *A.x - B.x*, *A.x - C.x* and *B.x - C.x*. This yields a total of nine edges. For each threshold level and case, the data of the group of these nine edges are aggregated to find the median value for the occurrence rates and average edge weights. Appendix E.2 shows the specific values that form the coordinates for each setting; only scatter plots will be shown for this section. Figure 4.7 shows how the trend for the zero interaction case differs between Models 2 and 3. In most plots, data points that deviate much from the trend or have extreme values are usually found for the case then the threshold level is set to the Boolean value *True*. This behavior leaves room for further discussion. As expected, occurrence and number of edges detected decreases per threshold level raised. Finally, data for the redundancy case in Model 2 shows an atypical trend as the median edge weight for the edge group raises at later levels of the threshold.

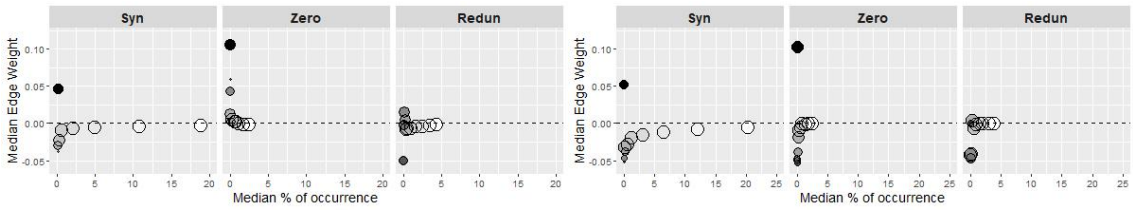


Figure 4.7: Coordinates of data grouping of path type *X-X*. Left: Model 2. Right: Model 3. Size represents distinct number of edges detected. Color gradient represents level of threshold applied (*False* is white, *True* is black).

Moving along with path type *X-Y*, again as seen in Table 4.4 in Models 2, 3 and 5 separate data groups have been identified showcasing a particular trend. The edges involved differ between the models. The edges included in the grouping for Models 2 and 3 contain all possible *X-Y* edges within a cluster of three triplets, with a maximum of 18 distinct edges. For Model 5, only one set of *X-Y* edges from neighbouring triplets are included. Going in alphabetical order, only edges where the *x*-variable from one triplet was connected to the *y*-variable from the next triplet are considered, yielding

10 distinct edges as 10 triplets were used. A clearer overview of included edges per model can be found in Appendix E.2 per path type.

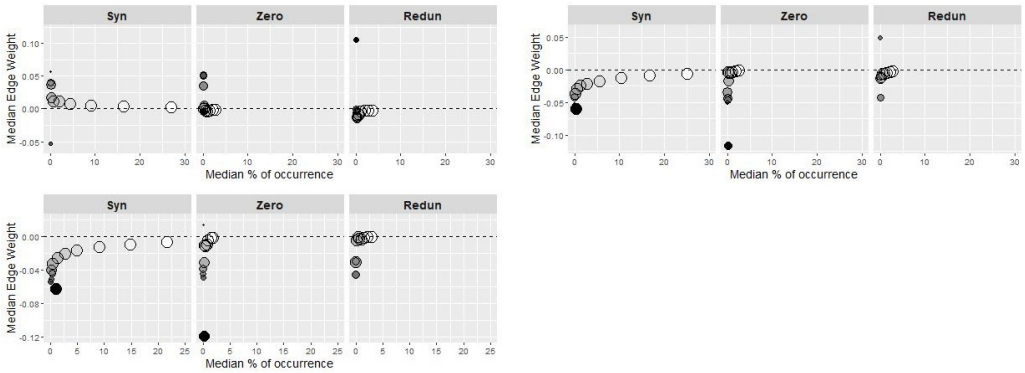


Figure 4.8: Idem, coordinates of data grouping of path type *X-Y*. Top Left: Model 2. Top Right: Model 3. Bottom Left: Model 5.

Trends for edge groups of path type *X-Z* are fairly consistent across all models and for all cases. The number and characteristics of edges included differ per model.

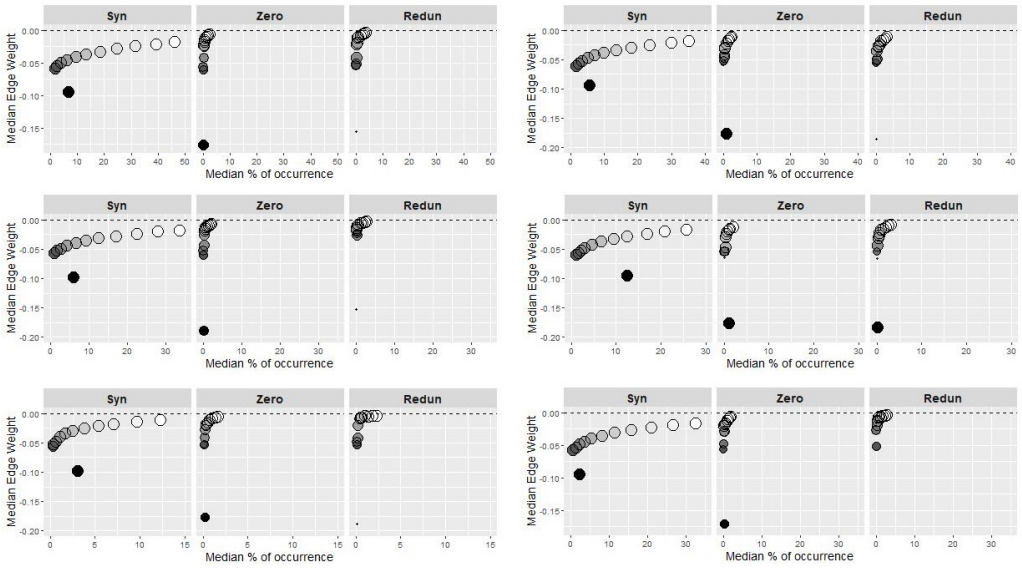


Figure 4.9: Idem, coordinates of data grouping of path type *X-Z*. Top Left: Model 2. Top Right: Model 3. Middle Left: Model 4. Middle Right: Model 5. Bottom Left: Model 6. Bottom Right: Model 7.

For Models 2 and 3, all six possible *X-Z* edges within a cluster are included, bringing a total of 18 edges for the full model. The same applies to Model 6 where twelve *X-Z* can occur within a cluster, yielding a total of 36. Model 4 includes twenty *X-Z* edges from neighboring triplets in both directions, meaning *A.x* to *B.z* and *A.z* to *B.x* are both allowed. Model 5 shares ten edges from Model 4 going in one specific direction; from *A.z* to *B.x*. Model 7 has forty edges included going in both directions between neighbouring triplets with distance ’2’ due to the double-loop layout. For instance, given triplet C, *X-Z* edges in both directions with triplets A, B, D, E are included.

Only for data analysis in Model 3 a separate data grouping was found related to the *Y-Y* path type. It includes all possible edges occurring within a cluster (*A.y* to *B.y*, *A.y* to *C.y* and *B.y* to *C.y*), yielding a total of nine edges as three clusters were used in the layout. Compared to the results from the other path types thus far, these results show unique trends across all cases.

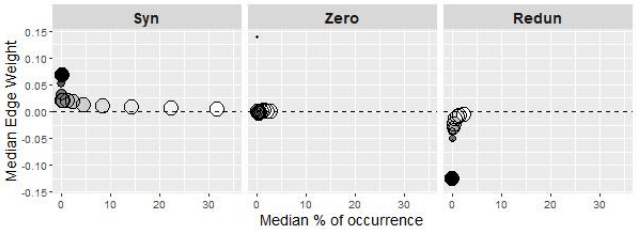


Figure 4.10: Idem, coordinates of data grouping of path type *Y-Y* for Model 3.

All possible *Y-Z* edges within a cluster were used for Model 2 and 6. Model 4 includes twenty *Y-Z* edges from neighboring triplets in both directions, meaning *A.y* to *B.z* and *A.z* to *B.y* are both allowed. Similar as in the previous path type, Model 7 has forty edges included with the same logic applied for filtering the relevant edges.

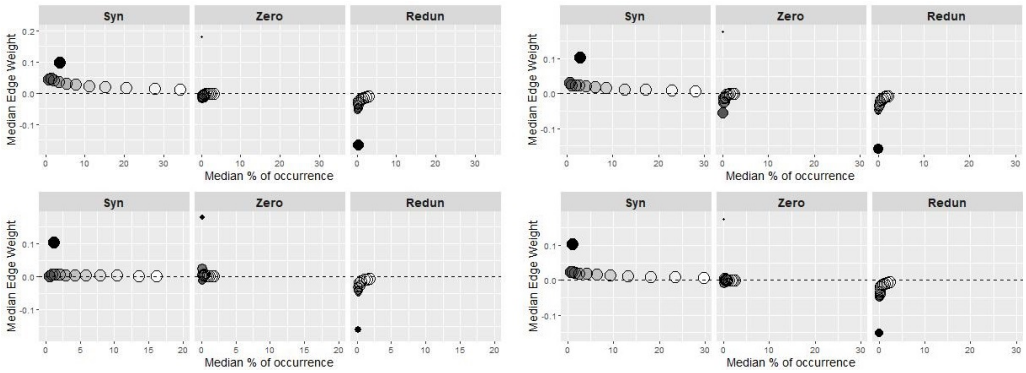


Figure 4.11: Idem, coordinates of data grouping of path type *Y-Z*. Top Left: Model 2. Top Right: Model 4. Bottom Left: Model 6. Bottom Right: Model 7.

Only for Model 3, a separate data group was found showcasing a different trend '*C*' for edges of path type *Z-Z*, consisting of those *Z-Z* found within a cluster (*A.z* to *B.z*, *A.z* to *C.z* and *B.z* to *C.z*). This brings the sample size for this data grouping to a total of nine distinct *Z-Z* edges for the model.

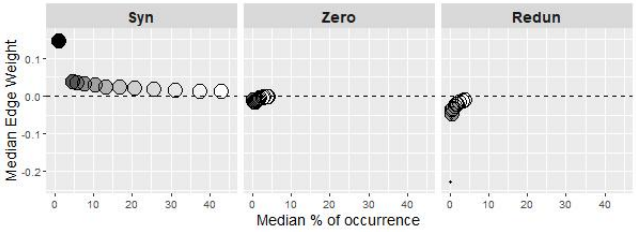


Figure 4.12: Idem, coordinates of data grouping of path type *Z-Z* for Model 3.

An enumerated summary of the findings is now given. The full synergy case allows for many more spurious correlations to be detected, both for variables within a construct and those between unique constructs when comparing the occurrence percentages between cases using Tables 4.1, 4.2 and 4.3. Consistent for all cases and models, occurrence percentages seem to differ per path type. The path type chosen for the forced residual correlations (either of type *Y-Z* or *Z-Z*) is unlikely to influence the occurrence of edges of specific path types when comparing figures for Model 2 versus 3 or Model 5 versus 6 giving closely similar results with occurrences of said residual correlations excluded. On the contrary, the influence of the model layout via mapping of residual correlations is best shown in Figure 4.7 where the trend in the zero interaction case heads in the opposite direction when comparing results from Models 2 & 3. Figure 4.8 repeats this where trends for Models 3 & 5, both using residual correlations of path type *Y-Z*, are opposite to Model 2 for the full synergy and zero interaction case. For path types *X-Z* and *Y-Z*, observed trends remain fairly stable for applicable models. It is important to stress that most spurious edges found follow trend *A* as seen in Figure 4.4. Occurrences of edges following this trend sharply drop as the threshold level is increased. Most edge weights average close to zero at the start of trend *A* and the range of values tends to spread out as occurrence percentages drop and the threshold level increases. It becomes difficult to determine both the general direction of edge weights per path type and cause of the spread in weight other than sampling variability. All data groupings found with trend *B* or *C* are either edges between neighboring triplets (or constructs) or, for Models 2, 5 and 6, edges found within a cluster. Configuration of the forced residual correlations to determine the model layout has also influenced the results of data groupings found. This is easily determined by comparing Model 2 to 3 or Model 4 to 5 where in the former case residual correlations of path type *Z-Z* were used and of path type *Y-Z* in the latter. The Boolean value *True* for the threshold setting often gives more extreme or contradictory results. The change of level of interaction information only changes the occurrences of spurious edges but not the general trend seen regarding the edge weights for some path types. Edges of path type *X-Z* all show a negative trend for increasing threshold levels. The most diverse results are found for path types *Y-Y* and *Z-Z* in Model 3 when residual correlations of path type *Y-Z* were used.

4.2 Discussion

Because each triplet (or construct) is considered identical in this modelling approach, regarding the different occurrence percentages per path type a possible, partial explanation could be given. Knowing that the presence of variable *Z* for each triplet is

responsible for generating the amount of interaction information, a possible explanation as of why many non-programmed *Z-Z* edges were discovered between constructs could be related to the theoretic role of comorbidity as discussed in Section 2.2.2. For each construct, variable *Z* is given influence on the behavior of the other variables *X* and *Y*. If such identical characteristics are embedded for each variable *Z*, it is not unlikely to think the possible *Z-Z* edges are those creating a bridge between symptoms from different constructs as seen in comorbidity. Again, in most cases these edges are less frequently observed with more randomized edge weights as the threshold level increases, thus this argument is also challenged as one may expect such bridge edges to be considered still significant after applying higher threshold levels with regularization. Nevertheless, generating synergy per construct for this modelling approach increases the potential of discovering more spurious edges with different occurrence frequencies per path type, allowing for more insights to be discovered how characteristics of variables in a full synergy case influence chances of occurrence. As for the sharp decline of edges found following trend *A*, it could mean that many edges found across repetitions were to be considered insignificant despite the application of GLASSO regularization. Referring to the edge characteristics found for data groupings following trends *B* and *C*, it allows for an argument that edges (or groupings) found outside these criteria showcasing trend *A* could be considered more as false positives instead of genuine discoveries and would support the results found in Model 1 as no connection between constructs was to be expected, but again this could be challenged by the relatively high occurrence percentages found per path type in the full synergy case. As for the results found via the *True* threshold setting often deviating from the found trends, documentation for the *qgraph* package explains this setting as a 'thresholded EBICglasso' approach where edge weights are reduced to zero if they are considered lower than the threshold value $\log(p * (p-1)/2))/\sqrt{n}$ with n the sample size used for the covariance matrix and p the number of dimensions. This is used to guarantee high specificity levels as possible false positives from the non-diagonal entries in the precision matrix due to sampling error are excluded from the network applying this threshold setting. There are cases, most notably for analysis of path types *X-X* and *X-Y* in Models 2 & 3, where the median occurrence and edge weight for the data grouping deviates from the observed trends. Again, given the very low occurrence percentages close to zero for these data points, it becomes difficult to state whether the trends are caused by the dominance of false positive edges within these data groupings which could explain the deviating results from the *True* threshold setting, or whether the setting is considered incompatible for this modelling approach given the sample size and number of variables included. Given the diversity of trends shown for path types *Y-Y* and *Z-Z* in Model 3 when using residual correlations of path type *Y-Z*, this may bring some evidence that a combination of the level of interac-

tion information and the intended forced model layout influence the direction of the trends shown regarding edge weights when increasing threshold levels. Nevertheless, it should be noted that the median edge weight of most of the data groupings when applying no threshold still hover close to zero, meaning that differences in networks across the cases may be harder to distinguish under this circumstance.

4.3 Conclusion

The simulation approach has shown that across all presented model layouts with interaction between constructs, each consisting of a triplet of variables, both the amount and, to some extent, the average weight of spurious edges between constructs can be influenced by the level of interaction information. The findings shown introduce a new challenge with regards to estimating network models if using the node centrality indices, especially if the edge weights are not considered. The presence of a spurious edge may influence said centrality indices for multiple nodes at once. While the presence of only one spurious edge may only lead to minimum changes for the centrality indices when dealing with larger and complex networks, the inclusion of more variables within such network also increases the presence of (dynamic) values of interaction information for each selected triplet of variables from the data, which in return may increase the chance of having spurious edges present when no threshold logic is applied. Beyond the interaction terms between two variables described via (conditioned) correlation, higher-order interaction terms where joint information is shared in groups of three or more variables, whether in a synergistic or redundant setting, also have their influence on the presence of edges and thus on the estimation of the resulting network and the grade of importance of its nodes. The discussed embedded assumptions when using the original centrality indices of Freeman (1978) are still applicable when using the level of interaction information as a parameter for simulation purposes to create and estimate a network model. This means that, if the constructs would represent disorders or diseases and its nodes the related symptoms, additional information regarding the severity of symptoms is still ignored when using centrality indices as an estimation technique. The discussion point made by Bulteel et al. (2016) and Bringmann et al. (2019) that the risk of multicollinearity may exist for symptoms if their presence are both determined by combining responses from a questionnaire is also considered relevant for the results found. The discussed possible overlap of two node constructs could be enforced if synergy is achieved after the introduction of a third, closely related node, further influencing the possible results found after estimating the network model. The findings of this simulation approach also bring more emphasis on the suggestions made in Epskamp et al. (2018) regard-

ing the inference of the accuracy of edge weights and the stability of the centrality indices via non-parametric bootstrapping, both which are dependent on the level of occurrence of edges found across the many iterations performed. Due to the shown influence of the level of interaction information set via higher-order interaction terms on the occurrence percentages and in some cases the shift of sign of the edge weight, such stability metrics could also depend whether the levels of interaction information found for particular groups of variables is considered consistent across all iterations given the data set at hand. In short, this study has shown an introduction of possible proof that higher-order interaction terms limited to a group of three variables may indeed influence the layout and thus the estimation of network models, and that the level of interaction information for particular triplets should be considered in exploratory data analysis prior to further statistical inference to check for possible causes or assumptions regarding importance of nodes in psychometrics.

4.4 Limitations

The manipulation of the level of interaction information is done as such that only 'pure' synergy or redundancy is achieved per triplet. The Partial Information Decomposition approach by Williams and Beer (2010) is discussed in Appendix B and explains how a combination of synergy and redundancy could be present when measuring interaction information for a set of two variables. No combinations of synergy, zero interaction and redundancy across triplets in the same model were considered for simulation. Residual correlations were set at the maximum level compatible for the synergy case, which range from 0.06 to 0.10. More distinguishable results are to be expected when using a simulation setting where said residual correlations could be increased significantly. Another improvement could be to apply the techniques used on a real data set or case along with detailed analysis of possible changes of node centrality indices. Alternative approaches regarding regularization methods and thresholds to derive to a sparse precision matrix could also be proposed, such as the MPT2 algorithm by Lauritzen et al. (2019) which deviates from GLASSO and model selection via the EBIC criteria.

4.5 Acknowledgments

Credit is given to prof. dr. Yves Rosseel for providing the first R-code snippets regarding the CFA modelling approach in *lavaan* and the function to calculate interaction information for a given triplet.

BIBLIOGRAPHY

American Psychiatric Association (2013). *Diagnostic and statistical manual of mental disorders (DSM-5®)*. American Psychiatric Pub.

Bhushan, N., Mohnert, F., Jans, L., Sloot, D., Albers, C., and Steg, L. (2019). Using a gaussian graphical model to explore relationships between items and variables in environmental psychology research. *Frontiers in psychology*, 10:1050.

Bollen, K. A. (1989). Structural equations with latent variables wiley. *New York*.

Borgatti, S. P. (2005). Centrality and network flow. *Social networks*, 27(1):55–71.

Borsboom, D. and Cramer, A. O. (2013). Network analysis: an integrative approach to the structure of psychopathology. *Annual review of clinical psychology*, 9:91–121.

Bowen, N. K. and Guo, S. (2011). *Structural equation modeling*. Oxford University Press.

Bringmann, L. F., Elmer, T., Epskamp, S., Krause, R. W., Schoch, D., Wichers, M., Wigman, J. T., and Snippe, E. (2019). What do centrality measures measure in psychological networks? *Journal of Abnormal Psychology*, 128(8):892.

Bulteel, K., Tuerlinckx, F., Brose, A., and Ceulemans, E. (2016). Using raw var regression coefficients to build networks can be misleading. *Multivariate behavioral research*, 51(2-3):330–344.

Celik, S., Logsdon, B., and Lee, S.-I. (2014). Efficient dimensionality reduction for high-dimensional network estimation. In *International Conference on Machine Learning*, pages 1953–1961.

Chen, J. and Chen, Z. (2008). Extended bayesian information criteria for model selection with large model spaces. *Biometrika*, 95(3):759–771.

Costantini, G., Richetin, J., Preti, E., Casini, E., Epskamp, S., and Perugini, M. (2019). Stability and variability of personality networks. a tutorial on recent developments in network psychometrics. *Personality and Individual Differences*, 136:68–78.

Cover, T. M. and Thomas, J. A. (2012). *Elements of information theory*. John Wiley & Sons.

Cramer, A. O., Waldorp, L. J., Van Der Maas, H. L., and Borsboom, D. (2010). Comorbidity: A network perspective. *Behavioral and brain sciences*, 33(2-3):137–150.

Epskamp, S. (2013). Network visualizations of statistical relationships and structural equation models. In *The R User Conference, useR! 2013 July 10-12 2013 University of Castilla-La Mancha, Albacete, Spain*, volume 10, page 118.

Epskamp, S. (2016). Brief report on estimating regularized gaussian networks from continuous and ordinal data. *arXiv preprint arXiv:1606.05771*.

Epskamp, S., Borsboom, D., and Fried, E. I. (2018). Estimating psychological networks and their accuracy: A tutorial paper. *Behavior Research Methods*, 50(1):195–212.

Epskamp, S., Cramer, A. O., Waldorp, L. J., Schmittmann, V. D., Borsboom, D., et al. (2012). qgraph: Network visualizations of relationships in psychometric data. *Journal of Statistical Software*, 48(4):1–18.

Epskamp, S. and Stuber, S. (2014). semplot: Path diagrams and visual analysis of various sem packages' output. *R package version*, 1(1).

Finn, C. and Lizier, J. T. (2020). Generalised measures of multivariate information content. *Entropy*, 22(2):216.

Freeman, L. C. (1978). Centrality in social networks conceptual clarification. *Social networks*, 1(3):215–239.

Fried, E. I., van Borkulo, C. D., Cramer, A. O., Boschloo, L., Schoevers, R. A., and Borsboom, D. (2017). Mental disorders as networks of problems: a review of recent insights. *Social Psychiatry and Psychiatric Epidemiology*, 52(1):1–10.

Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441.

Fruchterman, T. M. and Reingold, E. M. (1991). Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164.

Gel'Fand, I. and Yaglom, A. (1959). Abouta random function contained in another such function". *Eleven Papers on Analysis, Probability and Topology*, 12:199.

Ghassami, A. and Kiyavash, N. (2017). Interaction information for causal inference: The case of directed triangle. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pages 1326–1330. IEEE.

Hallquist, M. N., Wright, A. G., and Molenaar, P. C. (2019). Problems with centrality measures in psychopathology symptom networks: Why network psychometrics cannot escape psychometric theory. *Multivariate behavioral research*, pages 1–25.

Ince, R. A., Giordano, B. L., Kayser, C., Rousselet, G. A., Gross, J., and Schyns, P. G. (2017). A statistical framework for neuroimaging data analysis based on mutual information estimated via a gaussian copula. *Human brain mapping*, 38(3):1541–1573.

Jones, P. J., Ma, R., and McNally, R. J. (2019). Bridge centrality: A network approach to understanding comorbidity. *Multivariate behavioral research*, pages 1–15.

Jones, P. J., Mair, P., and McNally, R. J. (2018). Visualizing psychological networks: A tutorial in r. *Frontiers in Psychology*, 9:1742.

Kraskov, A., Stögbauer, H., and Grassberger, P. (2004). Estimating mutual information. *Physical review E*, 69(6):066138.

Kullback, S. (1959). Statistics and information theory.

Lauritzen, S., Uhler, C., Zwiernik, P., et al. (2019). Maximum likelihood estimation in gaussian models under total positivity. *The Annals of Statistics*, 47(4):1835–1863.

Lauritzen, S. L. (1996). *Graphical models*, volume 17. Clarendon Press.

Ma, J. and Sun, Z. (2011). Mutual information is copula entropy. *Tsinghua Science & Technology*, 16(1):51–54.

McGill, W. (1954). Multivariate information transmission. *Transactions of the IRE Professional Group on Information Theory*, 4(4):93–111.

Meng, Z., Eriksson, B., and Hero, A. (2014). Learning latent variable gaussian graphical models. In *International Conference on Machine Learning*, pages 1269–1277.

Opsahl, T., Agneessens, F., and Skvoretz, J. (2010). Node centrality in weighted networks: Generalizing degree and shortest paths. *Social networks*, 32(3):245–251.

Revelle, W. (2011). An overview of the psych package. *Department of Psychology Northwestern University. Accessed on March*, 3(2012):1–25.

Rosseel, Y. (2012). Lavaan: An r package for structural equation modeling and more. version 0.5–12 (beta). *Journal of statistical software*, 48(2):1–36.

Runge, J. (2015). Quantifying information transfer and mediation along causal pathways in complex systems. *Physical Review E*, 92(6):062829.

Schmittmann, V. D., Cramer, A. O., Waldorp, L. J., Epskamp, S., Kievit, R. A., and Borsboom, D. (2013). Deconstructing the construct: A network perspective on psychological phenomena. *New ideas in psychology*, 31(1):43–53.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423.

Steuer, R., Kurths, J., Daub, C. O., Weise, J., and Selbig, J. (2002). The mutual information: detecting and evaluating dependencies between variables. *Bioinformatics*, 18(suppl_2):S231–S240.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.

Timme, N., Alford, W., Flecker, B., and Beggs, J. M. (2014). Synergy, redundancy, and multivariate information measures: an experimentalist's perspective. *Journal of computational neuroscience*, 36(2):119–140.

Williams, P. L. and Beer, R. D. (2010). Nonnegative decomposition of multivariate information. *arXiv preprint arXiv:1004.2515*.

Yuan, M. and Lin, Y. (2007). Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35.

APPENDIX A

R-CODE OF MDS & PCA PLOTS WITH 'BFI' DATASET

```r
# as seen in Jones et al. (2018)
# "Visualizing Psychological Networks: A Tutorial in R"

library(psych)
library(smacof)

bfi_data <- bfi
bfi_data[26:28] <- list(NULL) #dropping last three columns (age/gender/education)
bfi_data <- na.omit(bfi_data) #removing observations with missing data

COR <- cor(bfi_data)

#MDS
bfi_diss <- sim2diss(COR) #converting similarities into dissimilarities
bfi_MDS <- mds(bfi_diss)

plot(bfi_MDS$conf, type="n")
text(bfi_MDS$conf, colnames(bfi_data))

#PCA
PCA_adult <- principal(COR, nfactors = 2) #eigen value decomposition

plot(PCA_adult$loadings, type="n")
text(PCA_adult$loadings, colnames(bfi_data))
```

APPENDIX B

PARTIAL INFORMATION DECOMPOSITION

Negative interaction information can still be difficult to interpret when applying to real-world examples. As an alternative, Williams and Beer (2010) opt for a non-negative measure approach, named the partial information decomposition, as the interpretation of negative information being passed along from one variable to another is considered unclear. In order to achieve this, a new definition of redundancy is given, namely it being the minimum information that any variable, entropy set, or otherwise considered information source can provide about the outcome of the target variable, averaged over all possible outcomes (Williams and Beer, 2010). All used information sources may share common information, which is considered the minimum information provided, while at the same time different sources may deliver information with regards to different outcomes of the target variable. The main idea is to split the total information of a set of variables towards a target variable into synergy, redundancy and unique information, which are referred to as the atoms of the total information. This differs from the earlier described usage of synergy and redundancy in the sense that they may coexist in the partial information decomposition approach. Another characteristic is that, instead of bundling the input variables together to find the average interaction information, the partial information decomposition takes into account possible subsets of these input variables.

Timme et al. (2014) provide a simplification of the mutual information values that can be calculated in a three-variable scenario as shown in Figure B.1, being the mutual information of the collection of input variables against the target variable S, here denoted as R_1 and R_2, and the mutual information between each input variable separately against the target variable S:

$$I(S; R_1, R_2) = Syn(S; R_1, R_2) + Unq(S; R_1) + Unq(S; R_2) + Rdn(S; R_1, R_2) \quad \text{(B.1)}$$

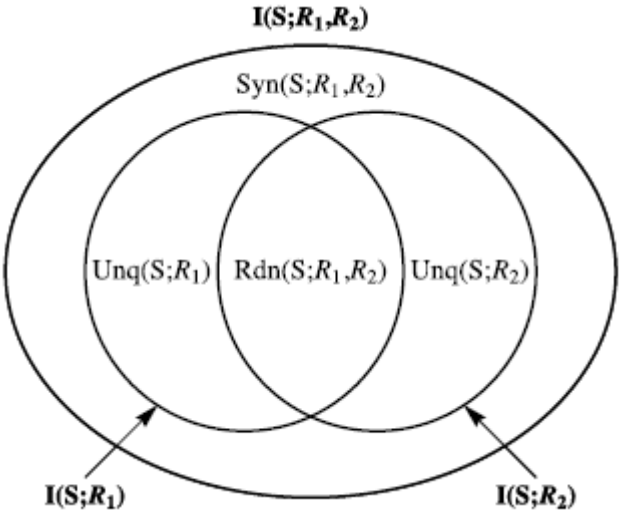


Figure B.1: Overview of the atoms of synergy, redundancy, and unique information, as found in Williams and Beer (2010)

$$I(S;R_1) = Unq(S;R_1) + Rdn(S;R_1,R_2) \tag{B.2}$$

$$I(S;R_2) = Unq(S;R_2) + Rdn(S;R_1,R_2) \tag{B.3}$$

The first step in this approach is to find $Rdn(S;R_1,R_2)$. Given the new definition of redundancy by Williams and Beer (2010), for each input variable and each outcome of the target variable the specific information captured needs to be measured. This is done by the following formula:

$$I(S=s;X) = \sum_x p(x|s)\left[\log\frac{1}{p(s)} - \log\frac{1}{p(s|x)}\right] \tag{B.4}$$

Knowing that the term $\frac{1}{p(s)}$ captures the surprise of reading the value s as a measurement, the information term *I(S=s;X)* describes the information delivered by variable X for each outcome s possible in outcome variable S, whereas *I(S;X)* would only describe it as the average or expected value of delivered information calculated across all possible outcomes of variable S. To apply this to two or more input variables to find redundancy across them towards output variable S, one could simply sum up the minimum information delivered by each input variable given the output s:

$$I_{min}(S;X_1,X_2,\ldots,X_k) = \sum_{s\in S} p(s) min_{X_i} I(S=s;X_i) \tag{B.5}$$

According to Williams and Beer (2010), this summation of the minimum information delivered by each input variable equals the redundancy of information delivered towards the outcome variable, therefore $I_{min}(S;R_1,R_2) = Rdn(S;R_1,R_2)$. Calculating the atoms of unique information per input variable and the synergy created by both input variables is now made possible. Focusing only on synergy, it becomes interesting how from Figure B.1 the redundancy term plays a role in finding the level of synergy as they are treated as two separate quantities:

$$Syn(S;R_1,R_2) = I(S;R_1,R_2) - I(S;R_1) - I(S;R_2) + Rdn(S;R_1,R_2) \tag{B.6}$$

The redundancy term has to be added back, as from Figure B.1 it becomes clear that subtracting both $I(S;R_1)$ and $I(S;R_2)$ means that the overlapping area, being the redundancy, is subtracted twice and must be corrected by adding the redundancy term back once. Calculating the unique information atoms $Unq(S;R_1)$ and $Unq(S;R_2)$ can be derived from equations B.2 and B.3 when knowing $Rdn(S;R_1,R_2)$.

Finally, both Williams and Beer (2010) and Timme et al. (2014) explain how negative values for the traditional interaction information measure can be explained by looking further into the interplay of synergy and redundancy. If the interaction information formula from equation 2.15 is rewritten as shown below, it becomes possible to plug in the different mutual information measures as described in the atoms of synergy, redundancy and unique information:

$$I(S;R_1;R_2) = I(S;R_1,R_2) - I(S;R_1) - I(S;R_2) \tag{B.7}$$

Plugging in the values gives the following insights: the unique information terms $Unq(S;R_1)$ and $Unq(S;R_2)$ are present in both $I(S;R_1,R_2)$ with a positive sign and in $I(S;R_1)$ and $I(S;R_2)$ with a negative sign, weighing each other out. The redundancy term occurs twice over $I(S;R_1)$ and $I(S;R_2)$ and is subtracted from the single redundancy term found in $I(S;R_1,R_2)$, leaving only the following terms:

$$I(S;R_1;R_2) = Syn(S;R_1,R_2) - Rdn(S;R_1,R_2)) \tag{B.8}$$

Once again this equation shows that the atoms of synergy and redundancy are to be considered together rather than exclusive, as was done before in the earlier interpretation of interaction information where the sign of the value would dictate whether only synergy or redundancy occurs. It also gives a more general and understandable explanation why negative values for interaction information could occur. While the association between positive/negative values for interaction information and syner-

gy/redundancy is still considered logical, the equation does not exclude the possibility that systems could exist with both non-zero contributions towards synergy and redundancy while yielding a zero value for interaction information, meaning that $I(S; R_1, R_2)$ can confound the atoms of synergy and redundancy (Williams and Beer, 2010). This also means that a system with negative interaction information can still hold interactions between variables that yield synergy and vice versa.

APPENDIX C

MODELLING APPROACH IN R

C.1 Lavaan Model Syntax for Single Triplet

```
library(lavaan)

#set factor loadings and covariance within triplet
l1 <- sqrt(0.99)
l2 <- sqrt(0.70)
l3 <- sqrt(0.30)
A.ecov <- -0.38
t1 <- 1 - l1^2 #variance
t2 <- 1 - l2^2
t3 <- 1 - l3^2

A.ecor <- A.ecov / (sqrt(t2) * sqrt(t3)) #correlation
A.t2s  <- t2 - abs(A.ecor)*t2 #used in lavaan syntax
A.t3s  <- t3 - abs(A.ecor)*t3 #used in lavaan syntax
A.l2s  <- 1 * sqrt(abs(A.ecor)*t2)
A.l3s  <- sign(A.ecor) * sqrt(abs(A.ecor)*t3)

pop.model <- c("
    # model A
    A  =~ (", l1, ")*A.x + (", l2, ")*A.y + (", l3, ")*A.z
    A.bf =~ (", A.l2s, ")*A.y + (", A.l3s, ")*A.z

    A.x ~~ (", t1,  ")*A.x
    A.y ~~ (", A.t2s, ")*A.y
    A.z ~~ (", A.t3s, ")*A.z

    A    ~~ 1*A
    A.bf ~~ 1*A.bf
    A    ~~ 0*A.bf
    ")
fit <- lavaan(pop.model)

#visualize lavaan SEM
semPaths(fit)
```

C.2 Function to Calculate Interaction Information in Triplet

```
lav_interaction_information_cor_triplet <- function(triplet.cor = NULL) {
  # unconditioned case - mutual information between A.x and A.y in cell [2,1]
  cor.xy <- triplet.cor[2,1]
  mi.xy <- -1/2 * log(1 - (cor.xy*cor.xy))
  # conditioned case - mutual information between A.x and A.y
  mi.xy_z <- as.numeric(NA)
  #2x2 matrix of A.x and A.y minus cross-product of transpose of matrix
  #transpose contains only correlations of A.z with other two variables A.x and A.y
  #tcrossprod() returns 2x2 matrix
  res.cov <- ( triplet.cor[1:2,1:2] - tcrossprod(triplet.cor[1:2,3]) )
  if(all(diag(res.cov) > 0)) {
  #from this new 2x2 matrix, extract new conditioned correlation between A.x and A.y
    res.cor <- cov2cor(res.cov)[2,1]
    #explicit check whether correlation value falls in [-1, 1]
    if(abs(res.cor) < 0.999) {
      mi.xy_z <- -1/2 * log(1 - (res.cor*res.cor))
    }
  }
  #difference in mutual information in the conditioned minus unconditioned case
  mi.xy_z - mi.xy
}
```

C.3 Model of Three Triplets with Varying Levels of Interaction Information

```
library(lavaan)

#set factor loadings and covariance within triplet
l1 <- sqrt(0.99)
l2 <- sqrt(0.70)
l3 <- sqrt(0.30)

A.ecov <- 0.22 #redundancy
B.ecov <- -0.15 #zero-interaction
C.ecov <- -0.39 #synergy

t1 <- 1 - l1^2 #variance
t2 <- 1 - l2^2
t3 <- 1 - l3^2

A.ecor <- A.ecov / (sqrt(t2) * sqrt(t3)) #correlation
A.t2s  <- t2 - abs(A.ecor)*t2 #used in lavaan syntax
```

```
A.t3s  <- t3 - abs(A.ecor)*t3 #used in lavaan syntax
A.l2s  <- 1 * sqrt(abs(A.ecor)*t2)
A.l3s  <- sign(A.ecor) * sqrt(abs(A.ecor)*t3)

B.ecor <- B.ecov / (sqrt(t2) * sqrt(t3))
B.l2s  <-            1 * sqrt(abs(B.ecor)*t2)
B.l3s  <- sign(B.ecor) * sqrt(abs(B.ecor)*t3)
B.t2s  <- t2 - abs(B.ecor)*t2
B.t3s  <- t3 - abs(B.ecor)*t3

C.ecor <- C.ecov / (sqrt(t2) * sqrt(t3))
C.l2s  <-            1 * sqrt(abs(C.ecor)*t2)
C.l3s  <- sign(C.ecor) * sqrt(abs(C.ecor)*t3)
C.t2s  <- t2 - abs(C.ecor)*t2
C.t3s  <- t3 - abs(C.ecor)*t3


pop.model <- c("

    # model A
    A  =~ (", l1, ")*A.x + (", l2, ")*A.y + (", l3, ")*A.z
    A.bf =~ (", A.l2s, ")*A.y + (", A.l3s, ")*A.z

    A.x ~~ (", t1,  ")*A.x
    A.y ~~ (", A.t2s, ")*A.y
    A.z ~~ (", A.t3s, ")*A.z

    A    ~~ 1*A
    A.bf ~~ 1*A.bf
    A    ~~ 0*A.bf

    # model B
    B  =~ (", l1, ")*B.x + (", l2, ")*B.y + (", l3, ")*B.z
    B.bf =~ (", B.l2s, ")*B.y + (", B.l3s, ")*B.z

    B.x ~~ (", t1,  ")*B.x
    B.y ~~ (", B.t2s, ")*B.y
    B.z ~~ (", B.t3s, ")*B.z

    B    ~~ 1*B
    B.bf ~~ 1*B.bf
    B.bf ~~ 0*B

    # model C
    C  =~ (", l1, ")*C.x + (", l2, ")*C.y + (", l3, ")*C.z
    C.bf =~ (", C.l2s, ")*C.y + (", C.l3s, ")*C.z

    C.x ~~ (", t1,  ")*C.x
    C.y ~~ (", C.t2s, ")*C.y
```

```
    C.z ~~ (", C.t3s, ")*C.z

    C    ~~ 1*C
    C.bf ~~ 1*C.bf
    C.bf ~~ 0*C

    # residual correlations
    A.z ~~ 0.15*B.z
    B.z ~~ 0.15*C.z
    A.z ~~ 0.15*C.z
")
fit <- lavaan(pop.model)

Data1 <- simulateData(pop.model, sample.nobs = 200)
qgraph(cor(Data1), layout="spring", graph="glasso", sampleSize=200, cut=0)
```

APPENDIX D

R-CODE OF MODEL LAYOUTS

D.1 R-Code Template for Model 1 - Null Model

```r
# EXAMPLE 1: 10 LATENT VARIABLES, ALL TRIPLETS, NO INTERACTION BETWEEN NODES FROM DIFFERENT TRIPLETS ("NULL MODEL")

# Loading packages
library(lavaan)
library(semPlot)
library(qgraph)
library(dplyr)

#set seed for reproducability
set.seed(100)

#parameters for simulation
REP <- 1000L #repetitions
N <- 200L #sample size

#=-=-CASE SELECTION=-=-=-#
#e.covs are now arranged in a vector of size 10
#input of ii_choice in order: redundancy/zero interaction/synergy
ii_choice = c(0.22, -0.15, -0.39)

#CASE - ZERO INTERACTION
#ecov <- rep(-0.15, 10L)

#CASE - REDUNDANCY
#ecov <- rep(0.22, 10L)

#CASE - SYNERGY
ecov <- rep(-0.39, 10L)

#CASE - CUSTOM
#ecov <- c(0.22, -0.15, -0.39, 0.22, -0.15, -0.39, 0.22, -0.15, -0.39, 0.22)

#CASE - RANDOM
#ecov = sample(ii_choice, 10, replace = TRUE)
```

```
#=-=-FUNCTIONS=-=-=-#
#Function 1: calculate II from 3x3 correlation matrix
lav_interaction_information_cor_triplet <- function(triplet.cor = NULL) {
  # mi.xy
  cor.xy <- triplet.cor[2,1]
  mi.xy <- -1/2 * log(1 - (cor.xy*cor.xy))

  # mi.xy_z
  mi.xy_z <- as.numeric(NA)
  res.cov <- ( triplet.cor[1:2,1:2] -
                 tcrossprod(triplet.cor[1:2,3]) * (1/triplet.cor[3,3]) )
  if(all(diag(res.cov) > 0)) {
    res.cor <- cov2cor(res.cov)[2,1]
    if(abs(res.cor) < 0.999) {
      mi.xy_z <- -1/2 * log(1 - (res.cor*res.cor))
    }
  }

  mi.xy_z - mi.xy
}


#=-=-CREATING MODEL VARIABLES-=-=-=-=-=#
#L and T-values
l1 <- sqrt(0.99)
l2 <- sqrt(0.70)
l3 <- sqrt(0.30)

t1 <- 1 - l1^2
t2 <- 1 - l2^2
t3 <- 1 - l3^2

#alphabet string is prepared, substring function to be in used in for-loop to create
      variable names for the model
alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

#for-loop to create the model components
for(i in 1:length(ecov)) {
  #first, take the appropriate capital letter of the alphabet
  letter = substr(alphabet, i, i)

  #temporary variables to store the new variable names in
  #first iteration will create A.ecor, A.l2s, ... , second iteration will create B.
      ecor, B.l2s, ... , etc.
  nam_ecor <- paste(letter, ".ecor", sep="")
  nam_l2s <- paste(letter, ".l2s", sep="")
  nam_l3s <- paste(letter, ".l3s", sep="")
  nam_t2s <- paste(letter, ".t2s", sep="")
  nam_t3s <- paste(letter, ".t3s", sep="")
```

```
82   #convert the string of the variable names ("A.ecor") to actual variables (A.ecor)
         with numeric values attached to them
83   assign(nam_ecor, ecov[i] / (sqrt(t2) * sqrt(t3)))
84   assign(nam_l2s, 1 * sqrt(abs(eval(as.name(paste(nam_ecor))))*t2))
85   assign(nam_l3s, sign(eval(as.name(paste(nam_ecor)))) * sqrt(abs(eval(as.name(paste
         (nam_ecor))))*t3))
86   assign(nam_t2s, t2 - abs(eval(as.name(paste(nam_ecor))))*t2)
87   assign(nam_t3s, t3 - abs(eval(as.name(paste(nam_ecor))))*t3)
88
89   #clear the workspace of 'junk' variables after the last iteration
90   if(i == length(ecov)) {
91     rm(i, letter, nam_ecor, nam_l2s, nam_l3s, nam_t2s, nam_t3s)
92   }
93 }
94
95 #=-=-LAVAAN MODEL SYTNAX-=-=-=-=-=-#
96 pop.model <- c("
97
98     # model A
99     A  =~ (", l1, ")*A.x + (", l2, ")*A.y + (", l3, ")*A.z
100    A.bf =~ (", A.l2s, ")*A.y + (", A.l3s, ")*A.z
101
102    A.x ~~ (", t1,  ")*A.x
103    A.y ~~ (", A.t2s, ")*A.y
104    A.z ~~ (", A.t3s, ")*A.z
105
106    A    ~~ 1*A
107    A.bf ~~ 1*A.bf
108    A    ~~ 0*A.bf
109
110    # model B
111    B  =~ (", l1, ")*B.x + (", l2, ")*B.y + (", l3, ")*B.z
112    B.bf =~ (", B.l2s, ")*B.y + (", B.l3s, ")*B.z
113
114    B.x ~~ (", t1,  ")*B.x
115    B.y ~~ (", B.t2s, ")*B.y
116    B.z ~~ (", B.t3s, ")*B.z
117
118    B    ~~ 1*B
119    B.bf ~~ 1*B.bf
120    B.bf ~~ 0*B
121
122    # model C
123    C  =~ (", l1, ")*C.x + (", l2, ")*C.y + (", l3, ")*C.z
124    C.bf =~ (", C.l2s, ")*C.y + (", C.l3s, ")*C.z
125
126    C.x ~~ (", t1,  ")*C.x
127    C.y ~~ (", C.t2s, ")*C.y
128    C.z ~~ (", C.t3s, ")*C.z
```

```
129     C    ~~ 1*C
130     C.bf ~~ 1*C.bf
131     C.bf ~~ 0*C
132
133     # model D
134     D  =~ (, l1, )*D.x + (, l2, )*D.y + (, l3, )*D.z
135     D.bf =~ (, D.l2s, )*D.y + (, D.l3s, )*D.z
136
137     D.x ~~ (, t1,  )*D.x
138     D.y ~~ (, D.t2s, )*D.y
139     D.z ~~ (, D.t3s, )*D.z
140
141     D    ~~ 1*D
142     D.bf ~~ 1*D.bf
143     D.bf ~~ 0*D
144
145     # model E
146     E  =~ (, l1, )*E.x + (, l2, )*E.y + (, l3, )*E.z
147     E.bf =~ (, E.l2s, )*E.y + (, E.l3s, )*E.z
148
149     E.x ~~ (, t1,  )*E.x
150     E.y ~~ (, E.t2s, )*E.y
151     E.z ~~ (, E.t3s, )*E.z
152
153     E    ~~ 1*E
154     E.bf ~~ 1*E.bf
155     E.bf ~~ 0*E
156
157     # model F
158     F  =~ (, l1, )*F.x + (, l2, )*F.y + (, l3, )*F.z
159     F.bf =~ (, F.l2s, )*F.y + (, F.l3s, )*F.z
160
161     F.x ~~ (, t1,  )*F.x
162     F.y ~~ (, F.t2s, )*F.y
163     F.z ~~ (, F.t3s, )*F.z
164
165     F    ~~ 1*F
166     F.bf ~~ 1*F.bf
167     F.bf ~~ 0*F
168
169     # model G
170     G  =~ (, l1, )*G.x + (, l2, )*G.y + (, l3, )*G.z
171     G.bf =~ (, G.l2s, )*G.y + (, G.l3s, )*G.z
172
173     G.x ~~ (, t1,  )*G.x
174     G.y ~~ (, G.t2s, )*G.y
175     G.z ~~ (, G.t3s, )*G.z
176
177
```

```
    G    ~~ 1*G
    G.bf ~~ 1*G.bf
    G.bf ~~ 0*G

    # model H
    H  =~ (", l1, ")*H.x + (", l2, ")*H.y + (", l3, ")*H.z
    H.bf =~ (", H.l2s, ")*H.y + (", H.l3s, ")*H.z

    H.x ~~ (", t1,  ")*H.x
    H.y ~~ (", H.t2s, ")*H.y
    H.z ~~ (", H.t3s, ")*H.z

    H    ~~ 1*H
    H.bf ~~ 1*H.bf
    H.bf ~~ 0*H

    # model I
    I  =~ (", l1, ")*I.x + (", l2, ")*I.y + (", l3, ")*I.z
    I.bf =~ (", I.l2s, ")*I.y + (", I.l3s, ")*I.z

    I.x ~~ (", t1,  ")*I.x
    I.y ~~ (", I.t2s, ")*I.y
    I.z ~~ (", I.t3s, ")*I.z

    I    ~~ 1*I
    I.bf ~~ 1*I.bf
    I.bf ~~ 0*I

    # model J
    J  =~ (", l1, ")*J.x + (", l2, ")*J.y + (", l3, ")*J.z
    J.bf =~ (", J.l2s, ")*J.y + (", J.l3s, ")*J.z

    J.x ~~ (", t1,  ")*J.x
    J.y ~~ (", J.t2s, ")*J.y
    J.z ~~ (", J.t3s, ")*J.z

    J    ~~ 1*J
    J.bf ~~ 1*J.bf
    J.bf ~~ 0*J
")

fit <- lavaan(pop.model)
Sigma <- lavInspect(fit, "Sigma")

#visualize lavaan SEM
semPaths(fit)
```

```
#=-=-BASELINE EDGELIST-=-=-=-=-=-=#
# get indices lower-half of Sigma
idx <- lav_matrix_vech_idx(n = nrow(Sigma), diagonal = FALSE)
node_from  <- col(Sigma)[idx]
node_to    <- row(Sigma)[idx]
# programmed: non-zero edge
programmed <- ifelse(abs(Sigma[idx]) > 0, 1, 0)
# create df_edgelist
df_edgelist <- data.frame(node_from, node_to, programmed)

#=-=-SIMULATION PROCESS-=-=-=-=-=-=#
#prepare master dataframe
df_master <- data.frame(node_from=as.integer(),
                        node_to=as.integer(),
                        programmed=as.integer(),
                        weight=as.numeric(),
                        run_id=as.integer())

#prepare list of vectors
ii_list <- list()

#prepare vectors for KPIs
sensitivity_vector <- specificity_vector <- programmed_vector <- nonprogrammed_
    vector <- size_vector <- c(numeric(REP))

for(i in 1:length(ecov)){
  #create dynamic string for variable name for vector
  nam_ii <- paste("ii",i,sep="")
  #append vector to list
  ii_list[[i]] <- numeric(REP)
}

for(j in seq_len(REP)) {
  #STEP 1: simulate 'REP' times a dataset of size N and find the correlation matrix
  Data <- simulateData(pop.model, sample.nobs = N)
  COR <- cor(Data)


  #STEP 2: calculate interaction information per triplet
  #length of ecov also translates in the number of triplets in the model
  for(i in 1:length(ecov)){
    #for each triplet
    #find index numbers to subset the correlation matrix into the relevant 3x3
        matrix (per triplet)
    m_low <- ((i-1) * 3) + 1
    m_high <- m_low + 2

    #use function lav_interaction_information_cor_triplet
    #and assign for each iteration of REP the value into the 'dynamic value'
```

```
274     ii_list[[i]][j] <- lav_interaction_information_cor_triplet(COR[m_low:m_high, m_
            low:m_high])
275   }
276
277
278   #STEP 3: retrieve edgelist from glasso
279   qgraph_glasso <- qgraph(cor(Data), layout="spring", graph="glasso", sampleSize=N,
280                           threshold=0.015, DoNotPlot=TRUE)$Edgelist
281
282   glasso_edges <- data.frame(qgraph_glasso$from, qgraph_glasso$to, qgraph_glasso$
          weight)
283   #rename column names to match with df_edgelist
284   colnames(glasso_edges) <- c("node_from", "node_to", "weight")
285
286
287   #STEP 4: merge the glasso edges with the baseline edgelist, left outer join
288   df_edgelist_merged <-merge(x=df_edgelist,y=glasso_edges, all.x=TRUE)
289   #add run_id to know from which iteration the data comes from
290   df_edgelist_merged$run_id <- j
291
292
293   #STEP 5: save results of iteration in master dataframe
294   df_master <- rbind(df_master, df_edgelist_merged)
295
296
297   #STEP 6: calculate KPIs per iteration
298   #prepare a 2x2 matrix for sensitivity/specificity
299   kpi_matrix <- matrix(c(0,0,0,0),nrow=2,ncol=2)
300
301   # (A) True Positives - programmed = 1 and weight != NA
302   kpi_matrix[1,1] <- length(which(df_edgelist_merged$programmed == 1
303                                   & !is.na(df_edgelist_merged$weight)))
304
305   # (B) False Negatives - programmed = 1 and weight = NA
306   kpi_matrix[2,1] <- length(which(df_edgelist_merged$programmed == 1
307                                   & is.na(df_edgelist_merged$weight)))
308
309   # (C) True Negatives - programmed = 0 and weight = NA
310   kpi_matrix[2,2] <- length(which(df_edgelist_merged$programmed == 0
311                                   & is.na(df_edgelist_merged$weight)))
312
313   # (D) False Positives - programmed = 0 and weight != NA
314   kpi_matrix[1,2] <- length(which(df_edgelist_merged$programmed == 0
315                                   & !is.na(df_edgelist_merged$weight)))
316
317   #calculate sensitivity & specificity
318   sensitivity_vector[j] <- kpi_matrix[1,1] / (kpi_matrix[1,1] + kpi_matrix[2,1])
319   specificity_vector[j] <- kpi_matrix[2,2] / (kpi_matrix[2,2] + kpi_matrix[1,2])
320
```

```
321   #calculate percentage of edges found that were programmed/non-programmed
322   #edges programmed: TP / (TP + FP)
323   programmed_vector[j] <- kpi_matrix[1,1] / (kpi_matrix[1,1] + kpi_matrix[1,2])
324   nonprogrammed_vector[j] <- 1 - programmed_vector[j]
325
326   #calculate number of edges to represent size of network
327   size_vector[j] <- length(which(!is.na(df_edgelist_merged$weight)))
328
329   print(j) #to keep track in console
330 }
331
332 #store KPI vectors into list
333 list_kpi <- list(size = size_vector, sensitivity = sensitivity_vector, specificity =
        specificity_vector,
334                  '%_programmed' = programmed_vector, '%_nonprogrammed' =
                       nonprogrammed_vector)
335
336 #convert KPI vectors into data-frame
337 df_kpi <- as.data.frame(do.call(cbind, list_kpi))
338 df_kpi <- format(df_kpi, digits=3, nsmall=0)
339 df_kpi <- sapply(df_kpi, as.numeric)
340
341 df_kpi_avg <- colMeans(df_kpi)
342 df_kpi_avg <- format(df_kpi_avg, digits=3, nsmall=0)
343
344
345 #=-=-RENAMING NODES FROM NUMBERS TO VARIABLE NAMES-=-=-=-=-=-=#
346 #transform numeric values for nodes in edgelist to actual variable names of model
347 #sequential order, 1/2/3 are A.x/A.y/A.z, etc.
348 number <- seq(3*length(ecov))
349
350 for(i in 1:length(ecov)) {
351   #first, take the appropriate capital letter of the alphabet
352   letter <- substr(alphabet, i, i)
353
354   #inner loop, create three variables for each iteration
355   for (j in 1:3) {
356     if (j == 1){
357       #create A.x, B.x, etc.
358       name <- paste(letter, ".x", sep="")
359     }
360     else if (j == 2){
361       #create A.y, B.y, etc.
362       name <- paste(letter, ".y", sep="")
363     }
364     else {
365       #create A.z, B.z, etc.
366       name <- paste(letter, ".z", sep="")
367     }
```

```
368     #replace each occurrence of current number (first one in vector) to newly
            created name
369     df_master$node_from[df_master$node_from == number[1]] <- name
370     df_edgelist$node_from[df_edgelist$node_from == number[1]] <- name
371     #repeat the same for the 'node_to' column
372     df_master$node_to[df_master$node_to == number[1]] <- name
373     df_edgelist$node_to[df_edgelist$node_to == number[1]] <- name
374     #delete first value of vector, similar to number += 1 in Python
375     number <- number[-1]
376   }
377   if (length(number) == 0) { #if all iterations are complete
378     #remove 'junk' variables from the workspace
379     rm(number, i, j, name, letter, alphabet)
380   }
381 }
382
383
384 #=-=-OVERALL KPIs-=-=-=-=-=-=#
385 #having calculated KPIs per iteration, now to calculate KPIs from df_master after
        all iterations
386 #KPIs could be related to subsets of the master dataframe
387
388 # (A) % of replications where particular edge was found (via group by)
389 #filter the master dataframe with only records including weights
390 df_master_filtered <- df_master[!is.na(df_master$weight),1:2]
391
392 #aggregate edges by number of occurrences, named 'count'
393 df_master_filtered_agg <- aggregate(df_master_filtered, by=list(df_master_filtered$
        node_from, df_master_filtered$node_to),
394                                     FUN=length)[1:3]
395 #rename columns
396 colnames(df_master_filtered_agg) <- c("node_from", "node_to", "count")
397
398 #calculate % of occurrence, named 'occur'
399 for (i in 1:nrow(df_master_filtered_agg)) {
400   df_master_filtered_agg$occur[i] <- df_master_filtered_agg$count[i] / REP
401 }
402
403 #merge grouped-by dataframe of edges with metadata of 'programmed'
404 df_master_filtered_agg <-merge(x=df_edgelist,y=df_master_filtered_agg, all.x=TRUE)
405
406 #some edges may have a 'NA' value for columns 'count' and 'occur', which is
        plausible
407 #replace 'NA' in columns 'count' and 'occur' with 0
408 df_master_filtered_agg$count[is.na(df_master_filtered_agg$count)] <- 0
409 df_master_filtered_agg$occur[is.na(df_master_filtered_agg$occur)] <- 0
410
411
412
```

```
413 # (A-1) full list of all edges with occurrence, descending order
414 df_master_filtered_agg_occurlist <- df_master_filtered_agg[order(-df_master_filtered
        _agg$occur),][,c(1,2,5)]
415
416 # (A-2) occurrences split by programmed and non-programmed edges
417 df_master_filtered_agg_programmed <- df_master_filtered_agg[df_master_filtered_agg$
        programmed == 1,c(1,2,5)]
418 #descending sort by occurrence %
419 df_master_filtered_agg_programmed <- df_master_filtered_agg_programmed[order(-df_
        master_filtered_agg_programmed$occur),]
420
421 df_master_filtered_agg_nonprogrammed <- df_master_filtered_agg[df_master_filtered_
        agg$programmed == 0,c(1,2,5)]
422 #descending sort by occurrence %
423 df_master_filtered_agg_nonprogrammed <- df_master_filtered_agg_nonprogrammed[order(-
        df_master_filtered_agg_nonprogrammed$occur),]
424
425 # (B) overall specificity and sensitivity KPIs
426 kpi_matrix_agg <- matrix(c(0,0,0,0),nrow=2,ncol=2)
427
428 # (1) True Positives - sum of edge counts where programmed = 1
429 kpi_matrix_agg[1,1] <- length(which(df_master$programmed == 1
430                                     & !is.na(df_master$weight)))
431
432 # (2) False Negatives - number of records where programmed = 1 and count = 0
433 kpi_matrix_agg[2,1] <- length(which(df_master$programmed == 1
434                                     & is.na(df_master$weight)))
435
436 # (3) True Negatives - number of records where programmed = 0 and count = 0
437 kpi_matrix_agg[2,2] <- length(which(df_master$programmed == 0
438                                     & !is.na(df_master$weight)))
439
440 # (4) False Positives - sum of edge counts where programmed = 0
441 kpi_matrix_agg[1,2] <- length(which(df_master$programmed == 0
442                                     & is.na(df_master$weight)))
443
444 #calculate sensitivity & specificity
445 sensitivity_agg <- kpi_matrix_agg[1,1] / (kpi_matrix_agg[1,1] + kpi_matrix_agg[2,1])
446 specificity_agg <- kpi_matrix_agg[2,2] / (kpi_matrix_agg[2,2] + kpi_matrix_agg[1,2])
447
448 # (C) Average weight of edges
449 df_master_filtered_weight <- df_master[!is.na(df_master$weight),c(1,2,4)]
450 #aggregate edges by average weight
451 df_master_filtered_weight_agg <- aggregate(df_master_filtered_weight[,3], by=list(df
        _master_filtered_weight$node_from, df_master_filtered_weight$node_to), FUN=mean)
452 df_master_filtered_weight_agg[,3] <- format(df_master_filtered_weight_agg[,3],
        digits=3, nsmall=0)
453 colnames(df_master_filtered_weight_agg) <- c("node_from", "node_to", "avg_weight")
454
```

```
455 #add metadata about % of occurrences in list
456 df_master_filtered_weight_agg <- merge(x=df_master_filtered_weight_agg,y=df_master_
        filtered_agg[,c(1,2,5)], all.x=TRUE)
457 #descending sort by occurrence %
458 df_master_filtered_weight_agg <- df_master_filtered_weight_agg[order(-df_master_
        filtered_weight_agg$occur),]
459
460 # (C-2) Filter by only significant edges (< -0.02 or > 0.02)
461 df_weight_sig <- df_master_filtered_weight_agg[abs(as.numeric(df_master_filtered_
        weight_agg$avg_weight)) >= 0.02,]
462
463
464 #=-=-CALCULATING II PER TRIPLET-=-=-=-=-=-=#
465 #preparing dataframe for the output, two columns for scores and description of level
         of II intended
466 df_ii <- data.frame(ii_score=as.numeric(10),
467                     ii_programmed=character(10),
468                     stringsAsFactors = FALSE)
469
470 for(i in 1:length(ecov)){
471   #add mean II values per triplet
472   df_ii$ii_score[i] <- mean(ii_list[[i]])
473
474   #add description of programmed intention of level of II
475   if (ecov[i] == ii_choice[1]) {
476     df_ii$ii_programmed[i] <- "redundancy"
477   }
478   else if (ecov[i] == ii_choice[2]) {
479     df_ii$ii_programmed[i] <- "zero interaction"
480   }
481   else if (ecov[i] == ii_choice[3]) {
482     df_ii$ii_programmed[i] <- "synergy"
483   }
484   else {
485     df_ii$ii_programmed[i] <- "custom"
486   }
487 }
488
489 #=-=-FINAL LIST OF RESULTS-=-=-=-=-=-=#
490 list_results <- list('Iteration Summary (Avg. KPIs)' = df_kpi_avg, 'Iteration KPIs'
        = df_kpi, 'II per Triplet' = df_ii, 'Occurrences of All Edges' = df_master_
        filtered_agg_occurlist, 'Occurrences of All Programmed Edges' = df_master_
        filtered_agg_programmed, 'Occurrences of All Non-Programmed Edges' = df_master_
        filtered_agg_nonprogrammed, 'Avg. Edge Weights & Occurrences' = df_master_
        filtered_weight_agg, 'Only Significant Avg. Edge Weights & Occurrences' = df_
        weight_sig)
491
492 print(list_results$‘Iteration Summary (Avg. KPIs)‘)
493 print(list_results$‘II per Triplet‘)
```

D.1.1 Model Syntax for Model 4

Model 4 uses a closed loop structure via *Z*-variables of two neighboring triplets. To force the layout, the following residual correlations are added in the *lavaan* model syntax:

```
# residual correlations
    A.z ~~ 0.10*B.z
    B.z ~~ 0.10*C.z
    C.z ~~ 0.10*D.z
    D.z ~~ 0.10*E.z
    E.z ~~ 0.10*F.z
    F.z ~~ 0.10*G.z
    G.z ~~ 0.10*H.z
    H.z ~~ 0.10*I.z
    I.z ~~ 0.10*J.z
    J.z ~~ 0.10*A.z
```

D.1.2 Model Syntax for Model 5

Model 5 uses a closed loop structure using the *Y*-variable from one triplet to a *Z*-variable of its neighboring triplet, going in alphabetical order. The residual correlations are changed as follows:

```
# residual correlations
    A.z ~~ 0.06*B.y
    B.z ~~ 0.06*C.y
    C.z ~~ 0.06*D.y
    D.z ~~ 0.06*E.y
    E.z ~~ 0.06*F.y
    F.z ~~ 0.06*G.y
    G.z ~~ 0.06*H.y
    H.z ~~ 0.06*I.y
    I.z ~~ 0.06*J.y
    J.z ~~ 0.06*A.y
```

D.1.3 Model Syntax for Model 7

Model 7 uses a double closed loop structure using the *Z*-variables between two triplets. Going in alphabetical order, each triplet is connected to four triplets in total: two triplets positioned before and two triplets positioned after the triplet of question. For instance, triplet C is connected to triplets A, B, D and E, all via edges between their *Z*-variables. The residual correlations are thus split up into two separate loops, as follows:

```
# residual correlations
    #loop 1
    A.z ~~ 0.1*B.z
    B.z ~~ 0.1*C.z
    C.z ~~ 0.1*D.z
    D.z ~~ 0.1*E.z
    E.z ~~ 0.1*F.z
    F.z ~~ 0.1*G.z
    G.z ~~ 0.1*H.z
    H.z ~~ 0.1*I.z
    I.z ~~ 0.1*J.z
    J.z ~~ 0.1*A.z

    #loop 2
    A.z ~~ 0.09*C.z
    B.z ~~ 0.09*D.z
    C.z ~~ 0.09*E.z
    D.z ~~ 0.09*F.z
    E.z ~~ 0.09*G.z
    F.z ~~ 0.09*H.z
    G.z ~~ 0.09*I.z
    H.z ~~ 0.09*J.z
    I.z ~~ 0.09*A.z
    J.z ~~ 0.09*B.z
```

D.2 R-Code Template for Model 2

Compared to the R-code shown in Appendix D.1, several smaller code chunks have been adapted to include metrics of edges occurring either within or between clusters. The *lavaan* model syntax now creates only nine triplets *A* to *I* and forced residual correlations have been added:

```
    # residual correlations
    A.z ~~ 0.10*B.z
    B.z ~~ 0.10*C.z
    A.z ~~ 0.10*C.z
    D.z ~~ 0.10*E.z
    E.z ~~ 0.10*F.z
    D.z ~~ 0.10*F.z
    G.z ~~ 0.10*H.z
    H.z ~~ 0.10*I.z
    G.z ~~ 0.10*I.z
```

The section "*Baseline Edgelist*" is edited as such:

```
    #=-=-BASELINE EDGELIST-=-=-=-=-=-=#
# get indices lower-half of Sigma
idx <- lav_matrix_vech_idx(n = nrow(Sigma), diagonal = FALSE)
node_from  <- col(Sigma)[idx]
node_to    <- row(Sigma)[idx]
# programmed: non-zero edge
programmed <- ifelse(abs(Sigma[idx]) > 0, 1, 0)
# within-cluster edges
set1 <- 1:9
set2 <- 1:9 + 9
set3 <- 1:9 + 9 + 9
in_cluster <- ifelse((node_from %in% set1 & node_to %in% set1) |
                       (node_from %in% set2 & node_to %in% set2) |
                       (node_from %in% set3 & node_to %in% set3), 1, 0)
# create df_edgelist
df_edgelist <- data.frame(node_from, node_to, programmed, in_cluster)
```

Additional vectors are prepared to store KPI data in regarding the percentage of edges found within or between clusters.

```
    #prepare vectors for KPIs
    sensitivity_vector <- specificity_vector <- incluster_vector <- outcluster_
        vector <-
  programmed_vector <- nonprogrammed_vector <- size_vector <- c(numeric(REP))
```

Before the code chunk in lines 321-324, the following is added:

```
1     #calculate percentage of edges found within/outside cluster
2   edges_incluster <- length(which(df_edgelist_merged$in_cluster == 1 & !is.na(df_
        edgelist_merged$weight)))
3   incluster_vector[j] <- edges_incluster / length(which(!is.na(df_edgelist_merged$
        weight)))
4   outcluster_vector[j] <- 1 - incluster_vector[j]
```

The additional vectors are added to the list $list_{kpi}$:

```
1     #store KPI vectors into list
2 list_kpi <- list(size = size_vector, sensitivity = sensitivity_vector, specificity =
       specificity_vector, '%_programmed' = programmed_vector, '%_nonprogrammed' =
      nonprogrammed_vector, '%_incluster' = incluster_vector, '%_outcluster' =
      outcluster_vector)
```

Before line 425, the following code chunk is added:

```
1     # (A-3) split by those within and outside cluster
2 df_master_filtered_agg_incluster <- df_master_filtered_agg[df_master_filtered_agg$in
      _cluster == 1,c(1,2,6)]
3 #descending sort by occurrence %
4 df_master_filtered_agg_incluster <- df_master_filtered_agg_incluster[order(-df_
      master_filtered_agg_incluster$occur),]
5
6 df_master_filtered_agg_outcluster <- df_master_filtered_agg[df_master_filtered_agg$
      in_cluster == 0,c(1,2,6)]
7 #descending sort by occurrence %
8 df_master_filtered_agg_outcluster <- df_master_filtered_agg_outcluster[order(-df_
      master_filtered_agg_outcluster$occur),]
9
10
11 # (A-4) combination: non-programmed, in cluster
12 #those that are non-programmed and outside of cluster are already covered by '
      outcluster'
13 #because no edges between clusters would be programmed with intention
14 df_master_filtered_agg_nonprogrammed_incluster <- df_master_filtered_agg[df_master_
      filtered_agg$programmed == 0  & df_master_filtered_agg$in_cluster == 1 ,c(1,2,6)
      ]
15 #descending sort by occurrence %
16 df_master_filtered_agg_nonprogrammed_incluster <- df_master_filtered_agg_
      nonprogrammed_incluster[order(-df_master_filtered_agg_nonprogrammed_incluster$
      occur),]
```

The final list of results is larger as KPI data is now available about edges occurring either within a cluster or between clusters:

```
1     #=-=-FINAL LIST OF RESULTS-=-=-=-=-=-=#
2 list_results <- list('Iteration Summary (Avg. KPIs)' = df_kpi_avg, 'Iteration KPIs'
      = df_kpi, 'II per Triplet' = df_ii, 'Occurrences of All Edges' = df_master_
```

```
filtered_agg_occurlist, 'Occurrences of All Programmed Edges' = df_master_
filtered_agg_programmed, 'Occurrences of All Non-Programmed Edges' = df_master_
filtered_agg_nonprogrammed, 'Occurrences of Edges within Cluster' = df_master_
filtered_agg_incluster, 'Occurrences of Edges between Clusters' = df_master_
filtered_agg_outcluster, 'Occurrences of Non-Programmed Edges within Cluster' =
df_master_filtered_agg_nonprogrammed_incluster, 'Avg. Edge Weights & Occurrences
' = df_master_filtered_weight_agg, 'Only Significant Avg. Edge Weights &
Occurrences' = df_weight_sig)
```

D.2.1 Model Syntax for Model 3

Model 3 uses the same setup as Model 2 with its only difference being the forced weaker residual correlations linked between a *Z*-variable from one triplet towards the *Y*-variables from the other triplets in its cluster:

```
# residual correlations
    A.z ~~ 0.06*B.y
    A.z ~~ 0.06*C.y

    B.z ~~ 0.06*A.y
    B.z ~~ 0.06*C.y

    C.z ~~ 0.06*A.y
    C.z ~~ 0.06*B.y

    D.z ~~ 0.06*E.y
    D.z ~~ 0.06*F.y

    E.z ~~ 0.06*D.y
    E.z ~~ 0.06*F.y

    F.z ~~ 0.06*D.y
    F.z ~~ 0.06*E.y

    G.z ~~ 0.06*H.y
    G.z ~~ 0.06*I.y

    H.z ~~ 0.06*G.y
    H.z ~~ 0.06*I.y

    I.z ~~ 0.06*G.y
    I.z ~~ 0.06*H.y
```

D.2.2 Model Syntax for Model 6

Model 6 is quite similar to Model 2 as it uses residual correlations between *Z*-variables for all triplets in the cluster. The only difference is that four triplets are presents within each cluster instead of three. Assuming the additional triplets have been prepared in the *lavaan* model syntax (as twelve triplets are used instead of nine requiring the creation of triplet J, K and L), this changes the list of residual correlations required as follows:

```
A.z ~~ 0.10*B.z
A.z ~~ 0.10*C.z
A.z ~~ 0.10*D.z
B.z ~~ 0.10*C.z
B.z ~~ 0.10*D.z

C.z ~~ 0.10*D.z

E.z ~~ 0.10*F.z
E.z ~~ 0.10*G.z
E.z ~~ 0.10*H.z

F.z ~~ 0.10*G.z
F.z ~~ 0.10*H.z

G.z ~~ 0.10*H.z

I.z ~~ 0.10*J.z
I.z ~~ 0.10*K.z
I.z ~~ 0.10*L.z

J.z ~~ 0.10*K.z
J.z ~~ 0.10*L.z

K.z ~~ 0.10*L.z
```

APPENDIX E

SIMULATION RESULTS

E.1 Summary KPIs

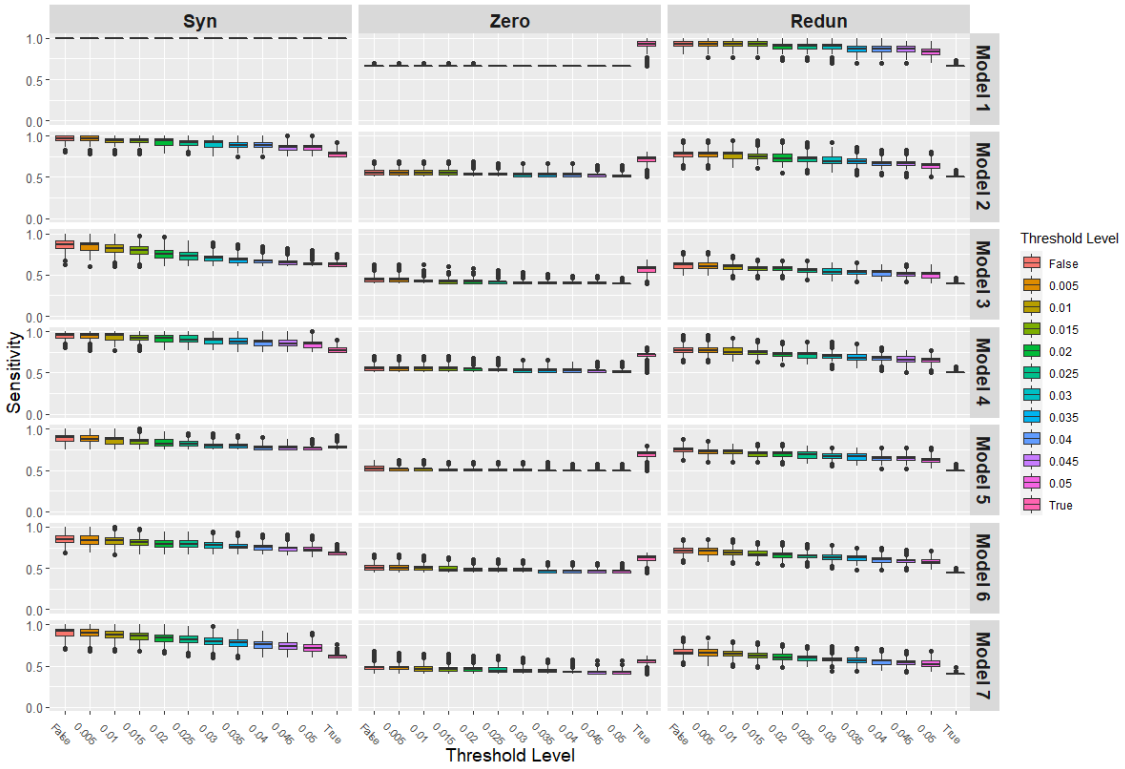


Figure E.1: Boxplot of Sensitivity levels per combination of model layout, case and threshold level.

Threshold	Syn	Zero	Redun
False	100.00 (100.00-100.0)	66.70 (66.70-66.7)	93.30 (90.00-96.7)
0.005	100.00 (100.00-100.0)	66.70 (66.70-66.7)	93.30 (90.00-96.7)
0.01	100.00 (100.00-100.0)	66.70 (66.70-66.7)	93.30 (90.00-96.7)
0.015	100.00 (100.00-100.0)	66.70 (66.70-66.7)	93.30 (90.00-96.7)
0.02	100.00 (100.00-100.0)	66.70 (66.70-66.7)	90.00 (86.70-93.3)
0.025	100.00 (100.00-100.0)	66.70 (66.70-66.7)	90.00 (86.70-93.3)
0.03	100.00 (100.00-100.0)	66.70 (66.70-66.7)	90.00 (86.70-93.3)
0.035	100.00 (100.00-100.0)	66.70 (66.70-66.7)	86.70 (83.30-90.0)
0.04	100.00 (100.00-100.0)	66.70 (66.70-66.7)	86.70 (83.30-90.0)
0.045	100.00 (100.00-100.0)	66.70 (66.70-66.7)	86.70 (83.30-90.0)
0.05	100.00 (100.00-100.0)	66.70 (66.70-66.7)	83.30 (80.00-86.7)
True	100.00 (100.00-100.0)	93.30 (90.00-96.7)	66.70 (66.70-66.7)

Table E.1: Model 1 - Sensitivity KPIs

Threshold	Syn	Zero	Redun
False	97.20 (94.40-100.0)	55.60 (52.80-58.3)	77.80 (75.00-80.6)
0.005	97.20 (94.40-100.0)	55.60 (52.80-58.3)	77.80 (75.00-80.6)
0.01	94.40 (91.70-97.2)	55.60 (52.80-58.3)	77.80 (72.20-80.6)
0.015	94.40 (91.70-97.2)	55.60 (52.80-58.3)	75.00 (72.20-77.8)
0.02	94.40 (88.90-97.2)	52.80 (52.80-55.6)	72.20 (69.40-77.8)
0.025	91.70 (88.90-94.4)	52.80 (52.80-55.6)	72.20 (69.40-75.0)
0.03	91.70 (86.10-94.4)	52.80 (50.00-55.6)	69.40 (66.70-75.0)
0.035	88.90 (86.10-91.7)	52.80 (50.00-55.6)	69.40 (66.70-72.2)
0.04	88.90 (86.10-91.7)	52.80 (50.00-55.6)	66.70 (63.90-69.4)
0.045	86.10 (83.30-88.9)	52.80 (50.00-52.8)	66.70 (63.90-69.4)
0.05	86.10 (83.30-88.9)	50.00 (50.00-52.8)	63.90 (61.10-66.7)
True	77.80 (75.00-80.6)	72.20 (69.40-75.0)	50.00 (50.00-50.0)

Table E.2: Model 2 - Sensitivity KPIs

Threshold	Syn	Zero	Redun
False	86.70 (82.20-91.1)	44.40 (42.20-46.7)	62.20 (57.80-64.4)
0.005	86.70 (80.00-88.9)	44.40 (42.20-46.7)	60.00 (57.80-64.4)
0.01	82.20 (77.80-86.7)	42.20 (42.20-44.4)	60.00 (57.25-62.2)
0.015	80.00 (75.60-84.4)	42.20 (40.00-44.4)	57.80 (55.60-60.0)
0.02	75.60 (71.10-80.0)	42.20 (40.00-44.4)	57.80 (55.60-60.0)
0.025	73.30 (68.90-77.8)	42.20 (40.00-42.2)	55.60 (53.30-57.8)
0.03	71.10 (66.70-73.3)	40.00 (40.00-42.2)	53.30 (51.10-57.8)
0.035	68.90 (64.40-71.1)	40.00 (40.00-42.2)	53.30 (51.10-55.6)
0.04	66.70 (64.40-68.9)	40.00 (40.00-42.2)	53.30 (48.90-55.6)
0.045	64.40 (62.20-66.7)	40.00 (40.00-42.2)	51.10 (48.90-53.3)
0.05	62.20 (62.20-64.4)	40.00 (40.00-40.0)	51.10 (46.70-53.3)
True	62.20 (60.00-64.4)	57.80 (53.30-60.0)	40.00 (40.00-40.0)

Table E.3: Model 3 - Sensitivity KPIs

Threshold	Syn	Zero	Redun
False	95.00 (92.50-97.5)	55.00 (52.50-57.5)	77.50 (75.00-80.0)
0.005	95.00 (92.50-97.5)	55.00 (52.50-57.5)	77.50 (75.00-80.0)
0.01	95.00 (90.00-97.5)	55.00 (52.50-57.5)	75.00 (72.50-80.0)
0.015	92.50 (90.00-95.0)	55.00 (52.50-57.5)	75.00 (72.50-77.5)
0.02	92.50 (87.50-95.0)	55.00 (52.50-55.0)	72.50 (70.00-75.0)
0.025	90.00 (87.50-95.0)	52.50 (52.50-55.0)	72.50 (67.50-75.0)
0.03	90.00 (85.00-92.5)	52.50 (50.00-55.0)	70.00 (67.50-72.5)
0.035	87.50 (85.00-92.5)	52.50 (50.00-55.0)	67.50 (65.00-72.5)
0.04	87.50 (82.50-90.0)	52.50 (50.00-55.0)	67.50 (65.00-70.0)
0.045	85.00 (82.50-90.0)	52.50 (50.00-52.5)	65.00 (62.50-70.0)
0.05	85.00 (80.00-87.5)	50.00 (50.00-52.5)	65.00 (62.50-67.5)
True	77.50 (75.00-80.0)	72.50 (70.00-72.5)	50.00 (50.00-50.0)

Table E.4: Model 4 - Sensitivity KPIs

Threshold	Syn	Zero	Redun
False	90.00 (85.00-92.5)	52.50 (50.00-55.0)	75.00 (72.50-77.5)
0.005	87.50 (85.00-92.5)	52.50 (50.00-52.5)	72.50 (70.00-75.0)
0.01	87.50 (82.50-90.0)	52.50 (50.00-52.5)	72.50 (70.00-75.0)
0.015	85.00 (82.50-87.5)	50.00 (50.00-52.5)	70.00 (67.50-72.5)
0.02	82.50 (80.00-87.5)	50.00 (50.00-52.5)	70.00 (67.50-72.5)
0.025	82.50 (80.00-85.0)	50.00 (50.00-52.5)	68.75 (65.00-72.5)
0.03	80.00 (77.50-82.5)	50.00 (50.00-52.5)	67.50 (65.00-70.0)
0.035	80.00 (77.50-82.5)	50.00 (50.00-50.0)	67.50 (62.50-70.0)
0.04	77.50 (75.00-80.0)	50.00 (50.00-50.0)	65.00 (62.50-67.5)
0.045	77.50 (75.00-80.0)	50.00 (50.00-50.0)	65.00 (62.50-67.5)
0.05	77.50 (75.00-77.5)	50.00 (50.00-50.0)	62.50 (60.00-65.0)
True	77.50 (77.50-80.0)	70.00 (67.50-72.5)	50.00 (50.00-50.0)

Table E.5: Model 5 - Sensitivity KPIs

Threshold	Syn	Zero	Redun
False	85.20 (81.50-88.9)	50.00 (48.10-53.7)	70.40 (68.50-74.1)
0.005	83.30 (79.60-88.9)	50.00 (48.10-53.7)	70.40 (66.70-74.1)
0.01	83.30 (79.60-87.0)	50.00 (48.10-51.9)	68.50 (66.70-72.2)
0.015	81.50 (77.80-85.2)	48.10 (46.30-51.9)	66.70 (64.80-70.4)
0.02	79.60 (75.90-83.3)	48.10 (46.30-50.0)	66.70 (63.00-68.5)
0.025	79.60 (75.90-83.3)	48.10 (46.30-50.0)	64.80 (63.00-66.7)
0.03	77.80 (74.10-81.5)	48.10 (46.30-50.0)	63.00 (61.10-66.7)
0.035	75.90 (74.10-79.6)	46.30 (44.40-48.1)	63.00 (59.30-64.8)
0.04	75.90 (72.20-77.8)	46.30 (44.40-48.1)	61.10 (57.40-63.0)
0.045	74.10 (70.40-75.9)	46.30 (44.40-48.1)	59.30 (57.40-61.1)
0.05	72.20 (70.40-75.9)	46.30 (44.40-46.3)	57.40 (55.60-61.1)
True	68.50 (66.70-68.5)	63.00 (59.30-64.8)	44.40 (44.40-44.4)

Table E.6: Model 6 - Sensitivity KPIs

Threshold	Syn	Zero	Redun
False	92.00 (86.00-94.0)	48.00 (46.00-50.0)	66.00 (64.00-70.0)
0.005	90.00 (86.00-94.0)	48.00 (46.00-50.0)	66.00 (62.00-70.0)
0.01	88.00 (84.00-92.0)	46.00 (44.00-50.0)	64.00 (62.00-68.0)
0.015	86.00 (82.00-90.0)	46.00 (44.00-48.0)	62.00 (60.00-66.0)
0.02	84.00 (80.00-88.0)	46.00 (44.00-48.0)	60.00 (58.00-64.0)
0.025	82.00 (78.00-86.0)	44.00 (42.00-48.0)	60.00 (56.00-62.0)
0.03	80.00 (76.00-84.0)	44.00 (42.00-46.0)	58.00 (56.00-60.0)
0.035	78.00 (74.00-82.0)	44.00 (42.00-46.0)	56.00 (54.00-60.0)
0.04	76.00 (72.00-80.0)	42.00 (42.00-44.0)	56.00 (52.00-58.0)
0.045	74.00 (70.00-78.0)	42.00 (40.00-44.0)	54.00 (52.00-56.0)
0.05	72.00 (68.00-76.0)	42.00 (40.00-44.0)	52.00 (50.00-56.0)
True	62.00 (60.00-62.0)	56.00 (54.00-58.0)	40.00 (40.00-40.0)

Table E.7: Model 7 - Sensitivity KPIs

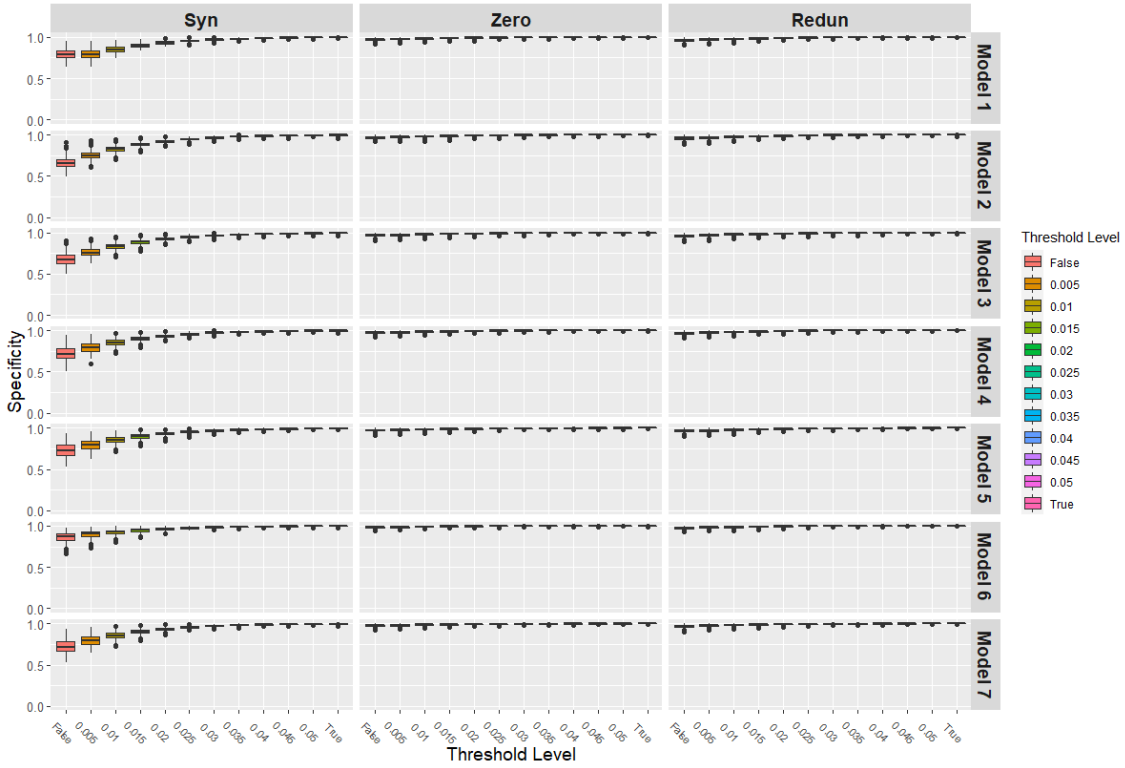


Figure E.2: Boxplot of Specificity levels per combination of model layout, case and threshold level.

Threshold	Syn	Zero	Redun
False	78.90 (75.10-83.70)	97.30 (96.30-98.00)	96.80 (95.60-97.80)
0.005	78.90 (75.10-83.70)	97.80 (97.00-98.50)	97.50 (96.30-98.30)
0.01	85.20 (82.70-88.40)	98.50 (97.50-99.00)	98.00 (97.30-98.80)
0.015	89.90 (88.10-91.90)	98.80 (98.30-99.30)	98.80 (98.00-99.30)
0.02	93.10 (92.10-94.60)	99.30 (98.80-99.50)	99.00 (98.50-99.50)
0.025	95.60 (94.60-96.30)	99.50 (99.00-99.80)	99.50 (99.00-99.80)
0.03	97.00 (96.30-97.50)	99.50 (99.30-99.80)	99.50 (99.30-99.80)
0.035	98.00 (97.50-98.50)	99.80 (99.50-100.00)	99.80 (99.50-100.00)
0.04	98.80 (98.30-99.00)	99.80 (99.50-100.00)	99.80 (99.80-100.00)
0.045	99.30 (98.80-99.50)	100.00 (99.80-100.00)	100.00 (99.80-100.00)
0.05	99.50 (99.30-99.80)	100.00 (99.80-100.00)	100.00 (99.80-100.00)
True	100.00 (100.00-100.00)	100.00 (100.00-100.00)	100.00 (100.00-100.00)

Table E.8: Model 1 - Specificity KPIs

Threshold	Syn	Zero	Redun
False	65.70 (61.82-70.50)	97.10 (96.20-98.10)	96.20 (94.90-97.50)
0.005	75.20 (72.10-78.40)	97.80 (96.80-98.40)	97.10 (95.90-98.10)
0.01	83.20 (80.60-85.40)	98.40 (97.50-99.00)	97.80 (96.80-98.70)
0.015	88.60 (87.00-90.20)	98.70 (98.10-99.40)	98.40 (97.80-99.00)
0.02	92.40 (91.10-93.70)	99.00 (98.70-99.70)	99.00 (98.40-99.40)
0.025	94.90 (94.00-95.90)	99.40 (99.00-99.70)	99.40 (98.70-99.70)
0.03	96.80 (95.90-97.50)	99.70 (99.40-100.00)	99.70 (99.00-100.00)
0.035	98.10 (97.50-98.40)	99.70 (99.40-100.00)	99.70 (99.40-100.00)
0.04	98.70 (98.10-99.00)	100.00 (99.70-100.00)	100.00 (99.70-100.00)
0.045	99.00 (98.70-99.40)	100.00 (99.70-100.00)	100.00 (99.70-100.00)
0.05	99.40 (99.00-99.70)	100.00 (99.70-100.00)	100.00 (99.70-100.00)
True	99.70 (99.00-100.00)	100.00 (100.00-100.00)	100.00 (100.00-100.00)

Table E.9: Model 2 - Specificity KPIs

Threshold	Syn	Zero	Redun
False	67.00 (62.70-72.50)	97.10 (96.10-98.00)	96.40 (95.10-97.40)
0.005	75.80 (72.50-79.70)	97.70 (96.70-98.70)	97.10 (96.10-98.00)
0.01	83.30 (81.00-85.90)	98.40 (97.40-99.00)	98.00 (97.10-98.70)
0.015	88.90 (86.90-90.50)	98.70 (98.00-99.30)	98.70 (97.70-99.00)
0.02	92.50 (91.20-93.80)	99.00 (98.70-99.70)	99.00 (98.40-99.30)
0.025	95.10 (94.10-96.10)	99.30 (99.00-99.70)	99.30 (99.00-99.70)
0.03	96.70 (96.10-97.40)	99.70 (99.30-100.00)	99.70 (99.30-99.70)
0.035	97.70 (97.40-98.40)	99.70 (99.30-100.00)	99.70 (99.30-100.00)
0.04	98.70 (98.00-99.00)	99.70 (99.70-100.00)	100.00 (99.70-100.00)
0.045	99.00 (98.70-99.30)	100.00 (99.70-100.00)	100.00 (99.70-100.00)
0.05	99.30 (99.00-99.70)	100.00 (99.70-100.00)	100.00 (99.70-100.00)
True	99.70 (99.30-100.00)	100.00 (100.00-100.00)	100.00 (100.00-100.00)

Table E.10: Model 3 - Specificity KPIs

Threshold	Syn	Zero	Redun
False	71.60 (66.80-78.50)	97.50 (96.50-98.20)	97.00 (95.90-98.00)
0.005	79.00 (75.20-83.58)	98.00 (97.15-98.70)	97.70 (96.70-98.50)
0.01	85.60 (82.95-88.40)	98.50 (97.70-99.00)	98.20 (97.50-99.00)
0.015	90.10 (88.40-92.20)	99.00 (98.20-99.50)	98.70 (98.20-99.20)
0.02	93.40 (92.20-94.70)	99.20 (98.70-99.50)	99.20 (98.70-99.50)
0.025	95.70 (94.70-96.50)	99.50 (99.00-99.70)	99.50 (99.00-99.70)
0.03	97.20 (96.50-97.70)	99.70 (99.20-99.70)	99.70 (99.50-100.00)
0.035	98.00 (97.50-98.50)	99.70 (99.50-100.00)	99.70 (99.50-100.00)
0.04	98.70 (98.20-99.20)	99.70 (99.70-100.00)	100.00 (99.70-100.00)
0.045	99.20 (98.70-99.50)	100.00 (99.70-100.00)	100.00 (99.70-100.00)
0.05	99.50 (99.20-99.70)	100.00 (99.70-100.00)	100.00 (99.70-100.00)
True	99.70 (99.20-100.00)	100.00 (100.00-100.00)	100.00 (100.00-100.00)

Table E.11: Model 4 - Specificity KPIs

Threshold	Syn	Zero	Redun
False	72.70 (66.80-79.20)	97.50 (96.65-98.20)	97.00 (95.70-97.70)
0.005	79.70 (75.20-84.30)	98.00 (97.20-98.70)	97.50 (96.50-98.27)
0.01	85.60 (82.95-88.90)	98.50 (97.70-99.00)	98.20 (97.20-99.00)
0.015	90.10 (88.40-92.20)	99.00 (98.20-99.20)	98.70 (98.00-99.20)
0.02	93.40 (92.20-94.70)	99.20 (98.70-99.50)	99.20 (98.70-99.50)
0.025	95.40 (94.70-96.50)	99.50 (99.00-99.70)	99.50 (99.00-99.70)
0.03	97.00 (96.20-97.70)	99.70 (99.20-99.70)	99.50 (99.20-99.70)
0.035	98.00 (97.50-98.50)	99.70 (99.50-100.00)	99.70 (99.50-100.00)
0.04	98.70 (98.20-99.00)	99.70 (99.70-100.00)	99.70 (99.70-100.00)
0.045	99.20 (98.70-99.50)	100.00 (99.70-100.00)	100.00 (99.70-100.00)
0.05	99.50 (99.20-99.70)	100.00 (99.70-100.00)	100.00 (99.70-100.00)
True	99.70 (99.50-100.00)	100.00 (100.00-100.00)	100.00 (100.00-100.00)

Table E.12: Model 5 - Specificity KPIs

Threshold	Syn	Zero	Redun
False	87.20 (83.00-90.50)	98.30 (97.60-98.80)	97.70 (96.90-98.40)
0.005	90.10 (86.60-92.70)	98.60 (97.90-99.10)	98.30 (97.60-98.80)
0.01	92.70 (90.25-94.65)	99.00 (98.40-99.30)	98.80 (98.10-99.30)
0.015	94.80 (92.90-96.20)	99.30 (98.80-99.70)	99.10 (98.60-99.50)
0.02	96.40 (95.00-97.40)	99.50 (99.10-99.70)	99.50 (99.10-99.70)
0.025	97.60 (96.50-98.30)	99.70 (99.50-99.80)	99.70 (99.30-99.80)
0.03	98.30 (97.60-98.80)	99.80 (99.70-99.80)	99.80 (99.50-99.80)
0.035	98.80 (98.40-99.10)	99.80 (99.70-100.00)	99.80 (99.70-100.00)
0.04	99.10 (98.80-99.50)	100.00 (99.80-100.00)	100.00 (99.80-100.00)
0.045	99.50 (99.10-99.70)	100.00 (99.80-100.00)	100.00 (99.80-100.00)
0.05	99.70 (99.50-99.80)	100.00 (99.80-100.00)	100.00 (100.00-100.00)
True	99.80 (99.70-100.00)	100.00 (100.00-100.00)	100.00 (100.00-100.00)

Table E.13: Model 6 - Specificity KPIs

Threshold	Syn	Zero	Redun
False	71.70 (66.50-78.20)	97.70 (96.90-98.40)	97.10 (96.10-98.20)
0.005	79.20 (75.30-83.90)	98.20 (97.40-99.00)	97.70 (96.90-98.40)
0.01	85.50 (82.90-88.60)	98.70 (98.20-99.20)	98.40 (97.70-99.00)
0.015	90.25 (88.30-92.20)	99.00 (98.40-99.50)	99.00 (98.40-99.50)
0.02	93.50 (92.20-94.80)	99.20 (99.00-99.70)	99.20 (99.00-99.70)
0.025	95.80 (94.80-96.60)	99.50 (99.20-99.70)	99.50 (99.20-99.70)
0.03	97.10 (96.60-97.90)	99.70 (99.50-100.00)	99.70 (99.50-100.00)
0.035	98.20 (97.70-98.70)	99.70 (99.50-100.00)	99.70 (99.70-100.00)
0.04	98.70 (98.40-99.20)	100.00 (99.70-100.00)	100.00 (99.70-100.00)
0.045	99.20 (99.00-99.50)	100.00 (99.70-100.00)	100.00 (99.70-100.00)
0.05	99.50 (99.20-99.70)	100.00 (99.70-100.00)	100.00 (100.00-100.00)
True	99.70 (99.50-100.00)	100.00 (100.00-100.00)	100.00 (100.00-100.00)

Table E.14: Model 7 - Specificity KPIs

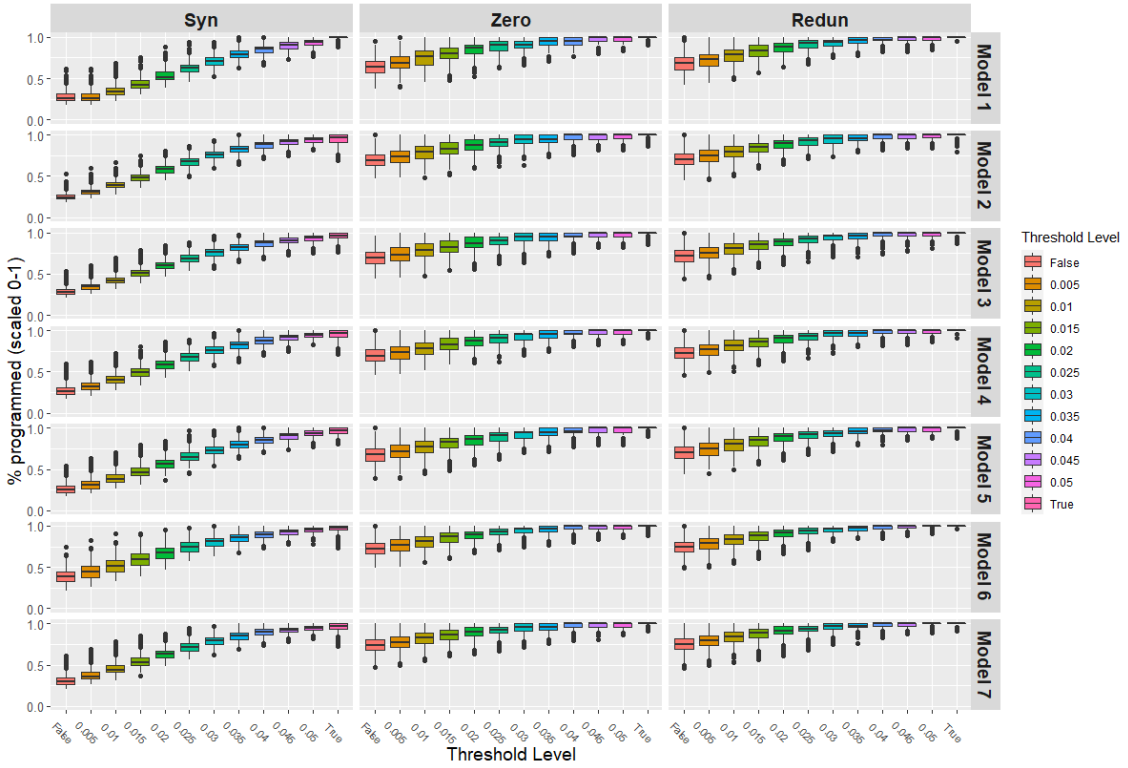


Figure E.3: Boxplot of percentage levels of edges found which were programmed or expected per combination of model layout, case and threshold level.

Threshold	Syn	Zero	Redun
False	26.00 (22.90-31.20)	64.50 (57.10-71.40)	68.20 (60.90-75.00)
0.005	26.00 (22.90-31.20)	69.00 (62.50-76.90)	73.00 (65.00-79.40)
0.01	33.30 (30.00-39.00)	76.90 (66.70-83.30)	78.40 (71.40-84.80)
0.015	42.30 (38.50-47.60)	80.00 (74.10-87.00)	83.90 (77.10-90.08)
0.02	51.70 (48.40-57.70)	87.00 (80.00-90.90)	87.90 (82.40-93.30)
0.025	62.50 (57.70-66.70)	90.90 (83.30-95.20)	92.60 (87.10-96.40)
0.03	71.40 (66.70-75.00)	90.90 (87.00-95.20)	93.50 (89.93-96.60)
0.035	78.90 (75.00-83.30)	95.20 (90.90-100.00)	96.30 (92.90-100.00)
0.04	85.70 (81.10-88.20)	95.20 (90.90-100.00)	96.60 (96.00-100.00)
0.045	90.90 (85.70-93.80)	100.00 (95.20-100.00)	100.00 (96.20-100.00)
0.05	93.80 (90.90-96.80)	100.00 (95.20-100.00)	100.00 (96.30-100.00)
True	100.00 (100.00-100.00)	100.00 (100.00-100.00)	100.00 (100.00-100.00)

Table E.15: Model 1 - Percentage of Programmed or Expected Edges KPIs

Threshold	Syn	Zero	Redun
False	24.50 (22.68-27.10)	69.00 (62.90-76.18)	70.70 (64.30-77.50)
0.005	30.80 (28.30-33.70)	73.30 (66.70-80.80)	75.00 (68.30-81.80)
0.01	39.10 (36.30-42.40)	79.20 (71.40-86.40)	80.00 (73.70-86.70)
0.015	48.50 (45.30-52.20)	83.30 (76.90-90.50)	84.80 (79.40-90.30)
0.02	58.50 (54.50-62.30)	87.00 (81.80-94.70)	89.70 (83.90-93.10)
0.025	68.00 (63.60-71.80)	90.90 (86.40-95.20)	92.90 (87.50-96.30)
0.03	76.20 (72.10-80.00)	94.70 (90.00-100.00)	95.90 (90.30-100.00)
0.035	82.90 (79.50-86.80)	95.00 (90.90-100.00)	96.20 (92.90-100.00)
0.04	88.60 (84.60-91.40)	100.00 (94.70-100.00)	100.00 (95.80-100.00)
0.045	91.70 (88.60-94.40)	100.00 (95.00-100.00)	100.00 (96.00-100.00)
0.05	94.10 (91.40-96.92)	100.00 (95.15-100.00)	100.00 (96.30-100.00)
True	96.40 (90.52-100.00)	100.00 (100.00-100.00)	100.00 (100.00-100.00)

Table E.16: Model 2 - Percentage of Programmed or Expected Edges KPIs

Threshold	Syn	Zero	Redun
False	28.10 (26.00-31.10)	69.00 (62.50-76.90)	71.55 (65.00-78.40)
0.005	34.30 (31.80-37.50)	73.10 (65.60-81.80)	75.70 (69.00-82.40)
0.01	42.10 (39.60-45.70)	78.30 (71.30-86.40)	80.60 (74.30-86.70)
0.015	51.30 (47.90-54.40)	82.60 (76.00-90.00)	85.70 (79.40-90.00)
0.02	60.00 (56.50-63.80)	87.00 (81.80-94.70)	89.30 (84.30-93.10)
0.025	68.60 (64.70-72.50)	90.50 (85.70-95.00)	92.60 (88.50-96.20)
0.03	76.10 (72.08-80.00)	94.70 (90.00-100.00)	95.80 (91.70-96.60)
0.035	82.30 (78.40-85.70)	95.00 (90.50-100.00)	96.20 (92.60-100.00)
0.04	87.50 (83.30-90.90)	95.00 (94.70-100.00)	100.00 (95.70-100.00)
0.045	90.90 (87.50-93.80)	100.00 (94.70-100.00)	100.00 (95.80-100.00)
0.05	93.50 (90.52-96.60)	100.00 (94.70-100.00)	100.00 (96.20-100.00)
True	96.70 (93.50-100.00)	100.00 (100.00-100.00)	100.00 (100.00-100.00)

Table E.17: Model 3 - Percentage of Programmed or Expected Edges KPIs

Threshold	Syn	Zero	Redun
False	25.50 (22.70-30.60)	69.00 (62.50-76.70)	72.70 (66.00-79.18)
0.005	31.50 (28.30-36.60)	73.30 (65.80-80.80)	76.70 (69.80-83.30)
0.01	39.80 (36.10-44.45)	78.60 (71.40-85.20)	82.10 (75.60-88.20)
0.015	48.70 (44.78-53.62)	83.30 (76.70-90.90)	86.30 (80.60-91.20)
0.02	58.50 (54.38-63.00)	87.50 (81.50-92.08)	90.30 (85.22-93.90)
0.025	67.80 (63.30-72.00)	91.30 (85.20-95.50)	93.50 (88.82-96.70)
0.03	75.60 (71.93-80.00)	95.20 (88.50-95.80)	96.30 (92.60-100.00)
0.035	82.20 (78.68-86.40)	95.50 (91.30-100.00)	96.60 (93.30-100.00)
0.04	87.50 (84.10-91.40)	95.70 (95.20-100.00)	100.00 (96.30-100.00)
0.045	91.70 (88.40-94.40)	100.00 (95.50-100.00)	100.00 (96.40-100.00)
0.05	94.40 (91.70-97.10)	100.00 (95.50-100.00)	100.00 (96.70-100.00)
True	96.90 (91.40-100.00)	100.00 (100.00-100.00)	100.00 (100.00-100.00)

Table E.18: Model 4 - Percentage of Programmed or Expected Edges KPIs

Threshold	Syn	Zero	Redun
False	25.00 (22.00-30.10)	67.70 (60.60-75.00)	70.50 (63.30-77.50)
0.005	30.60 (27.10-35.62)	71.40 (64.70-80.00)	75.00 (67.40-81.80)
0.01	37.80 (34.30-43.40)	76.90 (70.00-84.00)	80.60 (73.20-86.70)
0.015	46.60 (42.90-52.20)	83.30 (76.50-87.50)	85.30 (78.80-90.60)
0.02	56.10 (51.92-61.10)	87.00 (80.00-91.30)	89.70 (83.90-93.50)
0.025	64.80 (61.08-70.20)	90.90 (84.60-95.20)	92.90 (87.90-96.40)
0.03	73.20 (68.90-77.50)	95.20 (87.50-95.72)	93.90 (90.30-96.70)
0.035	80.00 (76.70-84.20)	95.20 (90.90-100.00)	96.40 (93.10-100.00)
0.04	85.70 (82.10-89.20)	95.50 (95.20-100.00)	96.60 (96.00-100.00)
0.045	90.90 (86.10-93.80)	100.00 (95.20-100.00)	100.00 (96.20-100.00)
0.05	93.80 (90.90-96.80)	100.00 (95.20-100.00)	100.00 (96.30-100.00)
True	96.90 (94.10-100.00)	100.00 (100.00-100.00)	100.00 (100.00-100.00)

Table E.19: Model 5 - Percentage of Programmed or Expected Edges KPIs

Threshold	Syn	Zero	Redun
False	38.10 (32.40-44.70)	72.60 (65.90-79.40)	74.50 (68.50-80.40)
0.005	44.05 (37.70-51.20)	77.10 (70.30-83.90)	78.70 (72.50-85.40)
0.01	51.20 (44.80-58.60)	81.60 (75.00-87.50)	83.70 (77.80-89.70)
0.015	59.50 (52.30-66.20)	86.70 (80.00-92.30)	88.10 (82.98-92.50)
0.02	67.70 (60.60-73.80)	89.70 (84.80-93.50)	91.90 (87.20-94.90)
0.025	74.60 (68.70-80.00)	92.60 (88.90-96.30)	94.30 (90.20-97.20)
0.03	80.80 (75.88-85.40)	96.00 (92.30-96.62)	96.80 (92.50-97.40)
0.035	85.70 (81.75-90.00)	96.30 (92.90-100.00)	97.10 (94.60-100.00)
0.04	89.80 (86.30-93.20)	100.00 (96.00-100.00)	100.00 (96.90-100.00)
0.045	93.00 (89.40-95.50)	100.00 (96.20-100.00)	100.00 (97.10-100.00)
0.05	95.10 (92.70-97.50)	100.00 (96.40-100.00)	100.00 (100.00-100.00)
True	97.30 (94.70-100.00)	100.00 (100.00-100.00)	100.00 (100.00-100.00)

Table E.20: Model 6 - Percentage of Programmed or Expected Edges KPIs

Threshold	Syn	Zero	Redun
False	29.70 (26.60-34.40)	73.50 (67.60-80.60)	75.00 (69.60-82.10)
0.005	36.00 (32.80-41.00)	77.40 (71.40-84.10)	79.35 (73.50-85.70)
0.01	44.25 (40.90-49.40)	82.80 (76.42-88.50)	83.80 (78.60-89.55)
0.015	53.50 (50.00-58.22)	86.60 (81.20-92.00)	88.60 (83.30-93.80)
0.02	62.90 (58.90-67.20)	90.00 (85.20-95.50)	91.40 (87.80-96.40)
0.025	71.70 (67.20-75.50)	92.30 (88.50-95.80)	93.90 (90.60-96.90)
0.03	79.20 (75.00-83.30)	95.50 (91.30-100.00)	96.60 (93.30-100.00)
0.035	85.00 (81.20-88.90)	95.80 (92.30-100.00)	96.80 (96.20-100.00)
0.04	89.40 (86.00-92.90)	100.00 (95.50-100.00)	100.00 (96.40-100.00)
0.045	92.70 (89.70-95.10)	100.00 (95.70-100.00)	100.00 (96.60-100.00)
0.05	95.00 (92.50-97.40)	100.00 (96.20-100.00)	100.00 (100.00-100.00)
True	96.90 (93.80-100.00)	100.00 (100.00-100.00)	100.00 (100.00-100.00)

Table E.21: Model 7 - Percentage of Programmed or Expected Edges KPIs

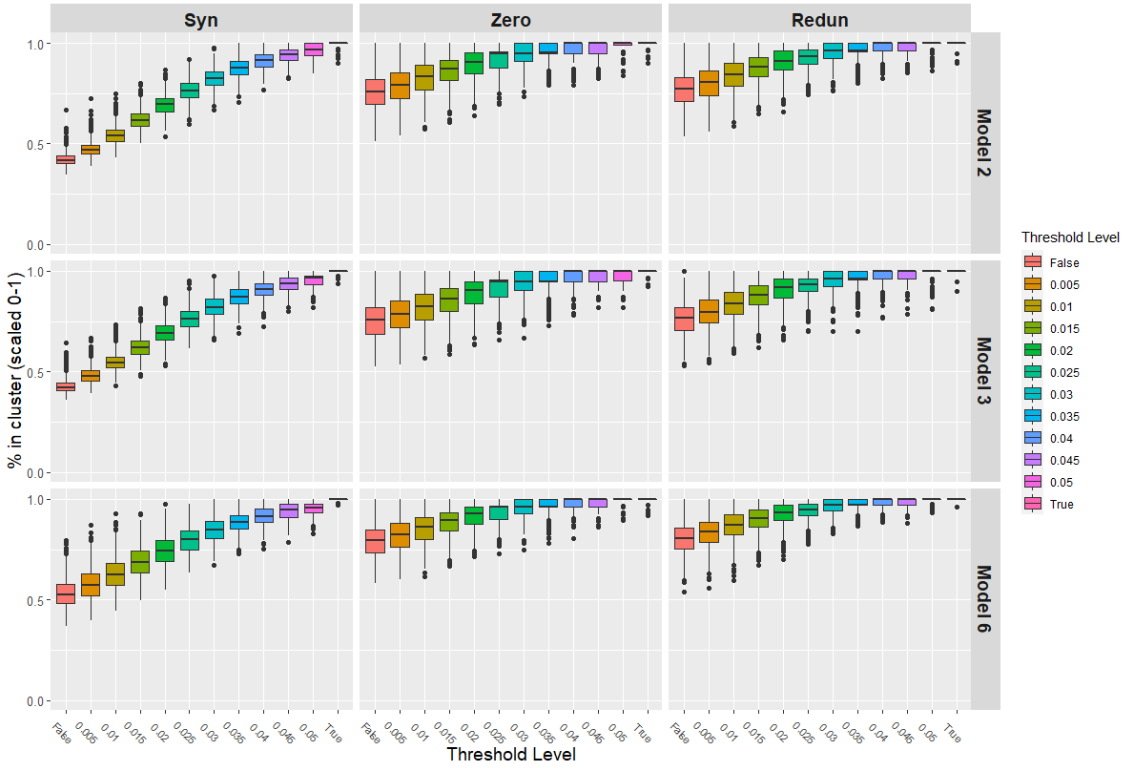


Figure E.4: Boxplot of percentage levels of edges found within their related cluster per combination of model layout, case and threshold level.

Threshold	Syn	Zero	Redun
False	41.75 (40.00-43.80)	75.90 (69.70-82.10)	77.10 (71.10-82.90)
0.005	47.05 (44.80-49.40)	79.30 (72.40-85.20)	80.60 (73.90-86.10)
0.01	53.80 (51.18-56.72)	83.30 (76.90-88.90)	84.40 (78.60-90.00)
0.015	61.40 (58.50-64.80)	87.00 (81.50-91.70)	88.20 (83.30-93.10)
0.02	69.40 (65.60-72.40)	90.50 (84.60-95.20)	91.20 (86.70-96.20)
0.025	76.50 (72.90-80.00)	94.70 (87.50-95.70)	93.30 (89.70-96.60)
0.03	82.50 (79.20-86.00)	95.00 (90.90-100.00)	96.20 (92.60-100.00)
0.035	87.50 (84.20-90.90)	95.20 (94.70-100.00)	96.30 (95.70-100.00)
0.04	91.40 (88.20-94.30)	100.00 (95.00-100.00)	100.00 (96.00-100.00)
0.045	94.10 (91.40-96.90)	100.00 (95.00-100.00)	100.00 (96.20-100.00)
0.05	96.80 (93.80-100.00)	100.00 (98.95-100.00)	100.00 (100.00-100.00)
True	100.00 (100.00-100.00)	100.00 (100.00-100.00)	100.00 (100.00-100.00)

Table E.22: Model 2 - Percentage of Spurious Edges within Cluster KPIs

Threshold	Syn	Zero	Redun
False	42.15 (40.60-44.50)	75.80 (68.60-82.10)	76.50 (70.77-81.87)
0.005	47.60 (45.50-50.40)	78.60 (71.90-85.20)	79.50 (74.40-85.70)
0.01	54.40 (52.10-57.40)	82.60 (76.00-88.50)	83.90 (78.40-89.70)
0.015	61.80 (58.80-65.40)	86.40 (80.00-91.30)	87.90 (83.30-92.90)
0.02	69.20 (65.70-72.90)	90.50 (84.00-95.00)	91.85 (86.70-96.20)
0.025	76.10 (72.50-80.00)	94.70 (87.00-95.50)	93.30 (90.00-96.40)
0.03	82.10 (78.40-86.00)	95.00 (90.50-100.00)	96.00 (92.60-100.00)
0.035	87.15 (83.70-90.90)	95.00 (94.70-100.00)	96.30 (95.70-100.00)
0.04	91.05 (87.90-93.90)	100.00 (94.70-100.00)	100.00 (96.00-100.00)
0.045	93.80 (90.90-96.80)	100.00 (95.00-100.00)	100.00 (96.00-100.00)
0.05	96.60 (93.30-97.40)	100.00 (95.20-100.00)	100.00 (100.00-100.00)
True	100.00 (100.00-100.00)	100.00 (100.00-100.00)	100.00 (100.00-100.00)

Table E.23: Model 3 - Percentage of Spurious Edges within Cluster KPIs

Threshold	Syn	Zero	Redun
False	52.40 (48.10-57.92)	79.40 (73.30-84.80)	80.40 (75.40-85.77)
0.005	57.10 (51.98-62.90)	82.55 (76.45-87.90)	83.70 (78.40-88.68)
0.01	62.50 (57.30-68.30)	86.10 (80.00-90.90)	87.20 (82.47-92.35)
0.015	68.55 (63.30-74.20)	89.70 (84.20-93.50)	90.50 (86.40-94.90)
0.02	74.30 (69.20-79.70)	92.90 (87.50-96.30)	93.25 (89.70-97.20)
0.025	80.00 (75.00-84.50)	96.00 (90.00-96.60)	94.90 (92.10-97.40)
0.03	84.60 (80.60-88.95)	96.30 (92.90-100.00)	97.10 (94.30-100.00)
0.035	88.60 (85.05-91.85)	96.40 (96.00-100.00)	97.30 (96.90-100.00)
0.04	91.50 (88.48-95.20)	100.00 (96.20-100.00)	100.00 (97.10-100.00)
0.045	94.90 (91.10-97.50)	100.00 (96.30-100.00)	100.00 (97.10-100.00)
0.05	95.50 (93.20-97.60)	100.00 (100.00-100.00)	100.00 (100.00-100.00)
True	100.00 (100.00-100.00)	100.00 (100.00-100.00)	100.00 (100.00-100.00)

Table E.24: Model 6 - Percentage of Spurious Edges within Cluster KPIs

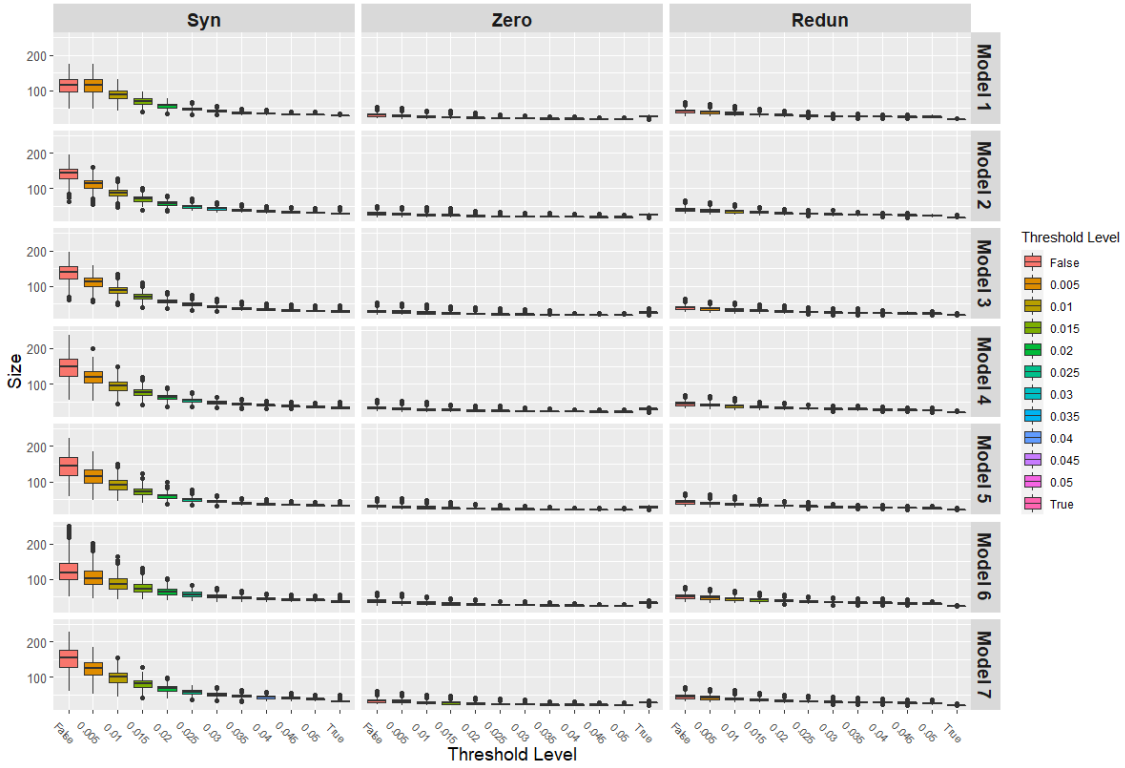


Figure E.5: Boxplot of number of edges found per combination of model layout, case and threshold level.

Threshold	Syn	Zero	Redun
False	115.5 (96.00-131.00)	31.0 (28.00-35.00)	42.0 (38.00-46.00)
0.005	115.5 (96.00-131.00)	29.0 (26.00-32.00)	39.0 (36.00-43.00)
0.01	90.0 (77.00-100.00)	26.0 (24.00-30.00)	36.0 (33.00-39.00)
0.015	71.0 (63.00-78.00)	25.0 (23.00-27.00)	33.0 (31.00-36.00)
0.02	58.0 (52.00-62.00)	23.0 (22.00-25.00)	31.0 (29.00-33.00)
0.025	48.0 (45.00-52.00)	22.0 (21.00-24.00)	30.0 (28.00-31.00)
0.03	42.0 (40.00-45.00)	22.0 (21.00-23.00)	28.0 (27.00-30.00)
0.035	38.0 (36.00-40.00)	21.0 (20.00-22.00)	28.0 (26.00-29.00)
0.04	35.0 (34.00-37.00)	21.0 (20.00-22.00)	27.0 (26.00-28.00)
0.045	33.0 (32.00-35.00)	20.0 (20.00-21.00)	26.0 (25.00-27.00)
0.05	32.0 (31.00-33.00)	20.0 (20.00-21.00)	26.0 (24.00-27.00)
True	30.0 (30.00-30.00)	28.0 (27.00-29.00)	20.0 (20.00-20.00)

Table E.25: Model 1 - Number of Edges within Network KPIs

Threshold	Syn	Zero	Redun
False	143.0 (127.00-155.00)	29.0 (26.00-33.00)	40.0 (36.00-45.00)
0.005	113.0 (102.00-123.00)	27.0 (24.00-31.00)	37.0 (34.00-42.00)
0.01	88.0 (80.00-95.00)	25.0 (23.00-28.00)	35.0 (31.00-38.00)
0.015	70.0 (64.00-76.00)	24.0 (22.00-26.00)	32.0 (30.00-35.00)
0.02	58.0 (53.00-62.00)	22.0 (21.00-24.00)	30.0 (28.00-32.00)
0.025	49.0 (45.00-52.00)	21.0 (20.00-23.00)	28.0 (27.00-30.00)
0.03	43.0 (40.00-46.00)	20.5 (19.00-22.00)	27.0 (25.00-29.00)
0.035	39.0 (36.00-41.00)	20.0 (19.00-21.00)	26.0 (24.00-27.00)
0.04	36.0 (34.00-38.00)	19.0 (19.00-21.00)	25.0 (24.00-26.00)
0.045	34.0 (32.00-36.00)	19.0 (18.00-20.00)	24.0 (23.00-25.00)
0.05	32.0 (31.00-34.00)	19.0 (18.00-20.00)	23.0 (22.00-25.00)
True	29.0 (27.00-32.00)	26.0 (25.00-27.00)	18.0 (18.00-18.00)

Table E.26: Model 2 - Number of Edges within Network KPIs

Threshold	Syn	Zero	Redun
False	140.0 (122.00-156.00)	29.0 (25.75-33.00)	39.0 (35.00-43.00)
0.005	112.0 (99.00-124.00)	27.0 (24.00-31.00)	36.0 (33.00-40.00)
0.01	88.0 (79.00-96.00)	25.0 (22.00-28.00)	33.0 (30.00-37.00)
0.015	70.0 (63.00-77.00)	23.0 (21.00-26.00)	31.0 (28.00-33.00)
0.02	57.0 (52.00-62.00)	22.0 (20.00-24.00)	29.0 (27.00-31.00)
0.025	48.0 (44.00-52.00)	21.0 (19.00-22.00)	27.0 (26.00-29.00)
0.03	42.0 (39.00-45.00)	20.0 (19.00-21.00)	26.0 (25.00-27.00)
0.035	37.0 (35.00-40.00)	19.0 (19.00-20.00)	25.0 (24.00-26.00)
0.04	34.0 (32.00-36.00)	19.0 (18.00-20.00)	24.0 (23.00-25.00)
0.045	32.0 (30.00-34.00)	19.0 (18.00-19.00)	24.0 (22.00-25.00)
0.05	30.0 (29.00-32.00)	18.0 (18.00-19.00)	23.0 (22.00-24.00)
True	29.0 (27.00-31.00)	26.0 (24.00-27.00)	18.0 (18.00-18.00)

Table E.27: Model 3 - Number of Edges within Network KPIs

Threshold	Syn	Zero	Redun
False	149.0 (122.00-170.00)	32.0 (29.00-36.00)	43.0 (39.00-48.00)
0.005	120.0 (102.00-136.00)	30.0 (27.00-34.00)	40.0 (37.00-45.00)
0.01	95.0 (82.00-106.00)	28.0 (25.00-31.00)	37.0 (34.00-41.00)
0.015	76.0 (67.00-84.00)	26.0 (24.00-29.00)	34.0 (32.00-37.00)
0.02	63.0 (57.00-69.00)	25.0 (23.00-27.00)	32.0 (30.00-35.00)
0.025	54.0 (49.00-58.00)	24.0 (22.00-26.00)	31.0 (29.00-33.00)
0.03	47.0 (44.00-51.00)	23.0 (21.00-24.00)	29.0 (28.00-31.00)
0.035	43.0 (40.00-45.25)	22.0 (21.00-23.00)	29.0 (27.00-30.00)
0.04	40.0 (38.00-42.00)	22.0 (21.00-23.00)	28.0 (26.00-29.00)
0.045	37.0 (36.00-39.00)	21.0 (20.00-22.00)	27.0 (26.00-28.00)
0.05	36.0 (34.00-38.00)	21.0 (20.00-22.00)	26.0 (25.00-27.00)
True	32.0 (31.00-35.00)	29.0 (28.00-29.00)	20.0 (20.00-20.00)

Table E.28: Model 4 - Number of Edges within Network KPIs

Threshold	Syn	Zero	Redun
False	144.0 (117.00-168.00)	31.0 (28.00-35.00)	42.0 (38.00-47.00)
0.005	115.5 (96.75-134.00)	29.0 (26.00-32.00)	39.0 (36.00-43.00)
0.01	91.0 (78.00-103.00)	27.0 (24.00-30.00)	36.0 (33.00-40.00)
0.015	73.0 (64.00-81.00)	25.0 (23.00-27.00)	33.0 (31.00-36.00)
0.02	60.0 (54.00-65.00)	24.0 (22.00-26.00)	31.0 (30.00-34.00)
0.025	50.0 (46.00-54.00)	23.0 (21.00-24.00)	30.0 (28.00-31.00)
0.03	44.0 (41.00-47.00)	22.0 (21.00-23.00)	29.0 (27.00-30.00)
0.035	40.0 (37.00-42.00)	21.0 (20.00-22.00)	28.0 (27.00-29.00)
0.04	37.0 (35.00-39.00)	21.0 (20.00-22.00)	27.0 (26.00-28.00)
0.045	34.0 (33.00-36.00)	20.0 (20.00-21.00)	26.0 (25.00-27.00)
0.05	33.0 (32.00-34.00)	20.0 (20.00-21.00)	26.0 (24.00-27.00)
True	32.0 (31.00-34.00)	29.0 (27.00-30.00)	20.0 (20.00-20.00)

Table E.29: Model 5 - Number of Edges within Network KPIs

Threshold	Syn	Zero	Redun
False	120.0 (100.00-146.00)	38.0 (34.00-42.00)	52.0 (47.00-57.00)
0.005	103.0 (86.00-123.00)	35.0 (32.00-39.00)	49.0 (44.00-53.00)
0.01	87.0 (74.00-102.00)	33.0 (30.00-37.00)	45.0 (41.00-49.00)
0.015	74.0 (65.00-86.00)	31.0 (28.00-34.00)	42.0 (39.00-45.00)
0.02	64.0 (57.00-73.00)	29.0 (27.00-32.00)	39.0 (37.00-42.00)
0.025	57.0 (52.00-64.00)	28.0 (26.00-30.00)	37.0 (35.00-40.00)
0.03	52.0 (48.00-57.00)	27.0 (26.00-29.00)	36.0 (34.00-38.00)
0.035	48.0 (45.00-52.00)	26.0 (25.00-28.00)	35.0 (33.00-36.00)
0.04	45.0 (43.00-48.00)	26.0 (25.00-27.00)	34.0 (32.00-35.00)
0.045	43.0 (41.00-45.25)	25.0 (24.00-26.00)	33.0 (31.00-34.00)
0.05	42.0 (40.00-44.00)	25.0 (24.00-26.00)	32.0 (30.00-33.00)
True	38.0 (36.00-40.00)	34.0 (32.00-35.00)	24.0 (24.00-24.00)

Table E.30: Model 6 - Number of Edges within Network KPIs

Threshold	Syn	Zero	Redun
False	154.5 (127.00-176.00)	33.0 (29.00-37.00)	44.5 (40.00-49.25)
0.005	125.5 (105.00-141.00)	31.0 (27.75-35.00)	42.0 (37.00-46.00)
0.01	100.0 (85.00-112.00)	29.0 (26.00-32.00)	38.0 (35.00-42.00)
0.015	81.0 (71.00-90.00)	27.0 (24.00-30.00)	36.0 (33.00-39.00)
0.02	67.5 (60.00-74.00)	25.0 (23.00-28.00)	33.0 (31.00-36.00)
0.025	58.0 (53.00-63.00)	24.0 (22.00-26.00)	32.0 (30.00-34.00)
0.03	51.0 (47.00-55.00)	23.0 (22.00-25.00)	30.0 (29.00-32.00)
0.035	46.0 (43.00-49.00)	23.0 (21.00-24.00)	29.0 (28.00-31.00)
0.04	43.0 (40.00-46.00)	22.0 (21.00-23.00)	28.0 (27.00-30.00)
0.045	40.0 (38.00-43.00)	22.0 (21.00-23.00)	27.0 (26.00-29.00)
0.05	38.0 (36.00-40.00)	21.0 (20.00-22.00)	26.0 (25.00-28.00)
True	31.0 (30.00-33.00)	28.5 (27.00-29.00)	20.0 (20.00-20.00)

Table E.31: Model 7 - Number of Edges within Network KPIs

E.2 Trend Analysis Data

For each path type, a table of coordinates per model, threshold level and case is presented for which the included data is used for the visualizations presented in Section 4.1 of the dissertation.

The set of *X-X* edges considered for Models 2 and 3 ($n = 9$) were the following:

Models 2 & 3
A.x to B.x
A.x to C.x
B.x to C.x
D.x to E.x
D.x to F.x
E.x to F.x
G.x to H.x
G.x to I.x
H.x to I.x

Table E.32: Set of *X-X* edges considered per model that showcase a particular trend subjected to further analysis.

Model	Threshold	Syn	Zero	Redun
Model 2	False	(18.9, -0.0026), n=9	(2.5, -0.0018), n=9	(4.4, -0.0024), n=9
Model 2	0.005	(10.8, -0.0043), n=9	(1.8, -0.0018), n=9	(3.5, -0.0031), n=9
Model 2	0.01	(5.0, -0.0061), n=9	(1.1, -0.0010), n=9	(2.5, -0.0044), n=9
Model 2	0.015	(2.1, -0.0072), n=9	(0.8, 0.0019), n=9	(1.6, -0.0040), n=9
Model 2	0.02	(0.7, -0.0100), n=9	(0.6, 0.0024), n=9	(1.0, -0.0070), n=9
Model 2	0.025	(0.3, -0.0230), n=8	(0.3, 0.0050), n=9	(0.6, -0.0072), n=9
Model 2	0.03	(0.1, -0.0302), n=5	(0.1, 0.0127), n=7	(0.4, -0.0085), n=9
Model 2	0.035	(0.1, -0.0384), n=2	(0.2, 0.0016), n=5	(0.3, 0.0039), n=7
Model 2	0.04	NA	(0.1, 0.0426), n=5	(0.2, 0.0146), n=7
Model 2	0.045	NA	(0.1, 0.0594), n=2	(0.1, -0.0015), n=6
Model 2	0.05	NA	(0.1, 0.0594), n=2	(0.1, -0.0507), n=5
Model 2	True	(0.1, 0.0464), n=6	(0.1, 0.1058), n=7	(0.1, 0.0030), n=2
Model 3	False	(20.4, -0.0054), n=9	(2.5, -0.0005), n=9	(3.9, -0.0002), n=9
Model 3	0.005	(12.1, -0.0082), n=9	(1.9, -0.0010), n=9	(3.2, -0.0006), n=9
Model 3	0.01	(6.6, -0.0122), n=9	(1.4, -0.0014), n=9	(2.1, -0.0009), n=9
Model 3	0.015	(3.1, -0.0160), n=9	(0.8, -0.0011), n=9	(1.6, -0.0003), n=9
Model 3	0.02	(1.3, -0.0198), n=9	(0.6, -0.0064), n=9	(1.0, -0.0021), n=9
Model 3	0.025	(0.6, -0.0284), n=9	(0.3, -0.0102), n=9	(0.7, -0.0076), n=9
Model 3	0.03	(0.2, -0.0323), n=9	(0.2, -0.0181), n=8	(0.4, 0.0030), n=9
Model 3	0.035	(0.2, -0.0382), n=4	(0.2, -0.0393), n=5	(0.2, -0.0416), n=9
Model 3	0.04	(0.1, -0.0471), n=2	(0.1, -0.0478), n=3	(0.1, -0.0430), n=9
Model 3	0.045	(0.1, -0.0523), n=1	(0.1, -0.0511), n=3	(0.1, -0.0462), n=6
Model 3	0.05	(0.1, -0.0523), n=1	(0.1, -0.0530), n=2	(0.1, 0.0008), n=2
Model 3	True	(0.1, 0.0513), n=5	(0.1, 0.1030), n=8	NA

Table E.33: Data coordinates for a specific group of edges with path type X-X in Models 2 and 3. The x-coordinate is the median of occurrence percentages of the included edges; the y-coordinate is the median of the averages of all included edge weights. *n* stands for the number of unique edges included in the group of interest.

The set of *X-Y* edges considered for Models 2, 3 (n = 18) and 5 (n = 10) were the following:

Models 2 & 3	Model 5
A.x to B.y & A.y to B.x	A.x to B.y
A.x to C.y & A.y to C.x	B.x to C.y
B.x to C.y & B.y to C.x	C.x to D.y
D.x to E.y & D.y to E.x	D.x to E.y
D.x to F.y & D.y to F.x	E.x to F.y
E.x to F.y & E.y to F.x	F.x to G.y
G.x to H.y & G.y to H.x	G.x to H.y
G.x to I.y & G.y to I.x	H.x to I.y
H.x to I.y & H.y to I.x	I.x to J.y
	J.x to A.y

Table E.34: Set of *X-Y* edges considered per model that showcase a particular trend subjected to further analysis.

Model	Threshold	Syn	Zero	Redun
Model 2	False	(26.95, 0.0024), n=18	(2.85, -0.0018), n=18	(3.80, -0.0026), n=18
Model 2	0.005	(16.50, 0.0034), n=18	(2.25, -0.0025), n=18	(2.90, -0.0028), n=18
Model 2	0.01	(9.15, 0.0046), n=18	(1.50, -0.0032), n=18	(2.05, -0.0037), n=18
Model 2	0.015	(4.50, 0.0075), n=18	(1.00, -0.0047), n=18	(1.25, -0.0050), n=18
Model 2	0.02	(2.05, 0.0112), n=18	(0.65, -0.0035), n=18	(0.90, -0.0095), n=18
Model 2	0.025	(0.85, 0.0107), n=18	(0.30, -0.0003), n=18	(0.55, -0.0090), n=18
Model 2	0.03	(0.30, 0.0167), n=17	(0.25, 0.0020), n=14	(0.30, -0.0139), n=17
Model 2	0.035	(0.20, 0.0364), n=13	(0.20, 0.0062), n=12	(0.20, -0.0116), n=14
Model 2	0.04	(0.10, 0.0403), n=7	(0.10, 0.0354), n=10	(0.20, -0.0136), n=10
Model 2	0.045	(0.10, -0.0529), n=2	(0.10, 0.0500), n=7	(0.20, -0.0035), n=7
Model 2	0.05	(0.10, -0.0529), n=2	(0.10, 0.0515), n=6	(0.10, -0.0012), n=6
Model 2	True	(0.10, 0.0574), n=1	(0.10, -0.0048), n=6	(0.10, 0.1062), n=3
Model 3	False	(25.20, -0.0071), n=18	(2.60, -0.0020), n=18	(2.95, -0.0031), n=18
Model 3	0.005	(16.70, -0.0097), n=18	(1.95, -0.0029), n=18	(2.30, -0.0038), n=18
Model 3	0.01	(10.55, -0.0134), n=18	(1.25, -0.0038), n=18	(1.60, -0.0054), n=18
Model 3	0.015	(5.55, -0.0178), n=18	(0.85, -0.0059), n=18	(1.05, -0.0072), n=18
Model 3	0.02	(2.80, -0.0222), n=18	(0.50, -0.0056), n=18	(0.70, -0.0074), n=18
Model 3	0.025	(1.40, -0.0253), n=18	(0.30, -0.0049), n=18	(0.30, -0.0120), n=17
Model 3	0.03	(0.60, -0.0304), n=18	(0.30, -0.0170), n=17	(0.15, -0.0138), n=16
Model 3	0.035	(0.20, -0.0377), n=18	(0.10, -0.0356), n=15	(0.10, -0.0090), n=12
Model 3	0.04	(0.10, -0.0433), n=11	(0.10, -0.0443), n=13	(0.10, -0.0423), n=8
Model 3	0.045	(0.10, -0.0537), n=5	(0.10, -0.0464), n=9	(0.10, 0.0486), n=5
Model 3	0.05	(0.10, -0.0542), n=4	(0.10, -0.0502), n=5	(0.10, 0.0004), n=4
Model 3	True	(0.50, -0.0602), n=18	(0.20, -0.1173), n=11	NA
Model 5	False	(21.70, -0.0072), n=10	(1.90, -0.0024), n=10	(3.05, -0.0014), n=10
Model 5	0.005	(14.95, -0.0097), n=10	(1.60, -0.0026), n=10	(2.30, -0.0016), n=10
Model 5	0.01	(9.25, -0.0130), n=10	(1.05, -0.0053), n=10	(1.65, -0.0017), n=10
Model 5	0.015	(4.95, -0.0173), n=10	(0.80, -0.0099), n=10	(1.15, -0.0045), n=10
Model 5	0.02	(2.85, -0.0210), n=10	(0.45, -0.0117), n=10	(0.75, -0.0029), n=10
Model 5	0.025	(1.40, -0.0260), n=10	(0.30, -0.0112), n=10	(0.45, -0.0012), n=10
Model 5	0.03	(0.55, -0.0327), n=10	(0.20, -0.0308), n=9	(0.20, -0.0053), n=10
Model 5	0.035	(0.20, -0.0396), n=9	(0.10, -0.0388), n=6	(0.10, -0.0314), n=10
Model 5	0.04	(0.30, -0.0439), n=5	(0.10, -0.0452), n=4	(0.10, -0.0302), n=6
Model 5	0.045	(0.20, -0.0514), n=3	(0.10, -0.0493), n=3	(0.10, -0.0463), n=6
Model 5	0.05	(0.10, -0.0540), n=3	(0.10, 0.0134), n=2	(0.10, -0.0044), n=2
Model 5	True	(1.10, -0.0626), n=10	(0.20, -0.1184), n=9	NA

Table E.35: Data coordinates for a specific group of edges with path type X-Y in Models 2, 3 and 5. The x-coordinate is the median of occurrence percentages of the included edges; the y-coordinate is the median of the averages of all included edge weights. n stands for the number of unique edges included in the group of interest.

The set of *X*-*Y* edges considered for Models 2, 3 ($n = 18$), 4 ($n = 20$), 5 ($n = 10$), 6 ($n = 36$) and 7 ($n = 40$) were the following:

Models 2 & 3	Model 4	Model 5	Model 6	Model 7
A.x to B.z	A.x to B.z	A.z to B.x	A.x to B.z	A.x to B.z
A.z to B.x	A.z to B.x	B.z to C.x	A.z to B.x	A.z to B.x
A.x to C.z	B.x to C.z	C.z to D.x	A.x to C.z	A.x to C.z
A.z to C.x	B.z to C.x	D.z to E.x	A.z to C.x	A.z to C.x
B.x to C.z	C.x to D.z	E.z to F.x	A.x to D.z	B.x to C.z
B.z to C.x	C.z to D.x	F.z to G.x	A.z to D.x	B.z to C.x
D.x to E.z	D.x to E.z	G.z to H.x	B.x to C.z	B.x to D.z
D.z to E.x	D.z to E.x	H.z to I.x	B.z to C.x	B.z to D.x
D.x to F.z	E.x to F.z	I.z to J.x	B.x to D.z	C.x to D.z
D.z to F.x	E.z to F.x	J.z to A.x	B.z to D.x	C.z to D.x
E.x to F.z	F.x to G.z		C.x to D.z	C.x to E.z
E.z to F.x	F.z to G.x		C.z to D.x	C.z to E.x
G.x to H.z	G.x to H.z		E.x to F.z	D.x to E.z
G.z to H.x	G.z to H.x		E.z to F.x	D.z to E.x
G.x to I.z	H.x to I.z		E.x to G.z	D.x to F.z
G.z to I.x	H.z to I.x		E.z to G.x	D.z to F.x
H.x to I.z	I.x to J.z		E.x to H.z	E.x to F.z
H.z to I.x	I.z to J.x		E.z to H.x	E.z to F.x
	J.x to A.z		F.x to G.z	E.x to G.z
	J.z to A.x		F.z to G.x	E.z to G.x
			F.x to H.z	F.x to G.z
			F.z to H.x	F.z to G.x
			G.x to H.z	F.x to H.z
			G.z to H.x	F.z to H.x
			I.x to J.z	G.x to H.z
			I.z to J.x	G.z to H.x
			I.x to K.z	G.x to I.z
			I.z to K.x	G.z to I.x
			I.x to L.z	H.x to I.z
			I.z to L.x	H.z to I.x
			J.x to K.z	H.x to J.z
			J.z to K.x	H.z to J.z
			J.x to L.z	I.x to J.z
			J.z to L.x	I.z to J.x
			K.x to L.z	I.x to A.z
			K.z to L.x	I.z to A.x
				J.x to A.z
				J.z to A.x
				J.x to B.z
				J.z to B.x

Table E.36: Set of *X*-*Z* edges considered per model that showcase a particular trend subjected to further analysis.

Model	Threshold	Syn	Zero	Redun
Model 2	False	(46.20, -0.0188), n=18	(2.50, -0.0073), n=18	(3.85, -0.0047), n=18
Model 2	0.005	(39.40, -0.0218), n=18	(2.00, -0.0080), n=18	(3.15, -0.0053), n=18
Model 2	0.01	(31.60, -0.0250), n=18	(1.55, -0.0107), n=18	(2.45, -0.0071), n=18
Model 2	0.015	(25.00, -0.0287), n=18	(1.10, -0.0122), n=18	(2.00, -0.0087), n=18
Model 2	0.02	(18.60, -0.0331), n=18	(0.80, -0.0161), n=18	(1.20, -0.0100), n=18
Model 2	0.025	(13.55, -0.0370), n=18	(0.60, -0.0184), n=18	(0.85, -0.0123), n=18
Model 2	0.03	(9.55, -0.0412), n=18	(0.45, -0.0232), n=18	(0.70, -0.0195), n=18
Model 2	0.035	(6.35, -0.0461), n=18	(0.30, -0.0242), n=17	(0.40, -0.0223), n=18
Model 2	0.04	(4.25, -0.0503), n=18	(0.30, -0.0420), n=15	(0.30, -0.0421), n=18
Model 2	0.045	(2.70, -0.0548), n=18	(0.20, -0.0565), n=13	(0.20, -0.0515), n=17
Model 2	0.05	(1.80, -0.0589), n=18	(0.20, -0.0607), n=11	(0.10, -0.0536), n=15
Model 2	True	(7.05, -0.0952), n=18	(0.25, -0.1762), n=16	(0.15, -0.1556), n=2
Model 3	False	(35.35, -0.0187), n=18	(2.60, -0.0123), n=18	(3.40, -0.0120), n=18
Model 3	0.005	(29.95, -0.0217), n=18	(2.20, -0.0136), n=18	(2.85, -0.0146), n=18
Model 3	0.01	(23.70, -0.0260), n=18	(1.80, -0.0166), n=18	(2.35, -0.0170), n=18
Model 3	0.015	(18.25, -0.0300), n=18	(1.50, -0.0195), n=18	(1.65, -0.0191), n=18
Model 3	0.02	(13.60, -0.0344), n=18	(1.10, -0.0226), n=18	(1.30, -0.0239), n=18
Model 3	0.025	(10.05, -0.0389), n=18	(0.70, -0.0296), n=18	(0.85, -0.0268), n=18
Model 3	0.03	(7.20, -0.0436), n=18	(0.60, -0.0312), n=18	(0.75, -0.0284), n=18
Model 3	0.035	(5.40, -0.0477), n=18	(0.40, -0.0416), n=17	(0.45, -0.0353), n=18
Model 3	0.04	(3.55, -0.0525), n=18	(0.30, -0.0459), n=17	(0.30, -0.0489), n=17
Model 3	0.045	(2.50, -0.0568), n=18	(0.20, -0.0517), n=15	(0.25, -0.0515), n=16
Model 3	0.05	(1.75, -0.0620), n=18	(0.10, -0.0561), n=12	(0.20, -0.0561), n=14
Model 3	True	(5.80, -0.0946), n=18	(1.05, -0.1762), n=18	(0.15, -0.1848), n=10
Model 4	False	(33.65, -0.0182), n=20	(2.25, -0.0076), n=20	(2.85, -0.0034), n=20
Model 4	0.005	(28.15, -0.0207), n=20	(2.00, -0.0086), n=20	(2.40, -0.0039), n=20
Model 4	0.01	(22.55, -0.0245), n=20	(1.60, -0.0103), n=20	(1.75, -0.0057), n=20
Model 4	0.015	(17.15, -0.0282), n=20	(1.10, -0.0115), n=20	(1.35, -0.0067), n=20
Model 4	0.02	(12.70, -0.0321), n=20	(0.80, -0.0134), n=20	(0.80, -0.0087), n=20
Model 4	0.025	(9.45, -0.0360), n=20	(0.65, -0.0157), n=20	(0.60, -0.0100), n=20
Model 4	0.03	(6.80, -0.0402), n=20	(0.40, -0.0184), n=20	(0.40, -0.0169), n=20
Model 4	0.035	(4.45, -0.0443), n=20	(0.30, -0.0259), n=19	(0.30, -0.0149), n=20
Model 4	0.04	(2.80, -0.0495), n=20	(0.20, -0.0434), n=19	(0.20, -0.0258), n=17
Model 4	0.045	(1.75, -0.0533), n=20	(0.10, -0.0525), n=15	(0.20, -0.0234), n=12
Model 4	0.05	(1.10, -0.0577), n=20	(0.10, -0.0601), n=13	(0.20, -0.0211), n=9
Model 4	True	(6.10, -0.0988), n=20	(0.20, -0.1898), n=16	(0.10, -0.1529), n=3
Model 5	False	(25.70, -0.0174), n=10	(2.10, -0.0130), n=10	(3.15, -0.0091), n=10
Model 5	0.005	(20.90, -0.0208), n=10	(1.45, -0.0153), n=10	(2.60, -0.0106), n=10
Model 5	0.01	(16.95, -0.0242), n=10	(1.15, -0.0180), n=10	(2.10, -0.0128), n=10
Model 5	0.015	(12.45, -0.0284), n=10	(0.80, -0.0207), n=10	(1.55, -0.0153), n=10
Model 5	0.02	(9.55, -0.0336), n=10	(0.65, -0.0244), n=10	(1.05, -0.0180), n=10
Model 5	0.025	(6.75, -0.0378), n=10	(0.50, -0.0302), n=10	(0.70, -0.0226), n=10
Model 5	0.03	(4.65, -0.0427), n=10	(0.35, -0.0466), n=10	(0.60, -0.0276), n=10
Model 5	0.035	(3.05, -0.0482), n=10	(0.20, -0.0546), n=9	(0.45, -0.0314), n=10
Model 5	0.04	(2.30, -0.0528), n=10	(0.20, -0.0546), n=9	(0.20, -0.0447), n=10
Model 5	0.045	(1.65, -0.0578), n=10	(0.10, -0.0546), n=9	(0.10, -0.0544), n=8
Model 5	0.05	(1.25, -0.0607), n=10	(0.15, -0.0649), n=6	(0.10, -0.0657), n=6
Model 5	True	(12.60, -0.0952), n=10	(1.20, -0.1767), n=10	(0.20, -0.1841), n=10

Table E.37: Data coordinates for a specific group of edges with path type X-Z in Models 2, 3, 4 and 5. The x-coordinate is the median of occurrence percentages of the included edges; the y-coordinate is the median of the averages of all included edge weights. *n* stands for the number of unique edges included in the group of interest.

Model	Threshold	Syn	Zero	Redun
Model 6	False	(12.30, -0.0122), n=36	(1.70, -0.0065), n=36	(2.40, -0.0040), n=36
Model 6	0.005	(9.70, -0.0152), n=36	(1.40, -0.0079), n=36	(1.90, -0.0049), n=36
Model 6	0.01	(7.10, -0.0185), n=36	(1.10, -0.0094), n=36	(1.40, -0.0064), n=36
Model 6	0.015	(5.40, -0.0219), n=36	(0.80, -0.0114), n=36	(1.10, -0.0053), n=36
Model 6	0.02	(3.80, -0.0256), n=36	(0.60, -0.0161), n=36	(0.70, -0.0071), n=36
Model 6	0.025	(2.50, -0.0296), n=36	(0.40, -0.0186), n=36	(0.60, -0.0075), n=36
Model 6	0.03	(1.70, -0.0342), n=36	(0.30, -0.0206), n=35	(0.40, -0.0093), n=35
Model 6	0.035	(1.10, -0.0403), n=36	(0.20, -0.0270), n=32	(0.25, -0.0208), n=34
Model 6	0.04	(0.70, -0.0486), n=36	(0.20, -0.0411), n=27	(0.20, -0.0420), n=31
Model 6	0.045	(0.40, -0.0528), n=36	(0.10, -0.0526), n=20	(0.10, -0.0480), n=28
Model 6	0.05	(0.30, -0.0560), n=35	(0.10, -0.0532), n=19	(0.10, -0.0528), n=21
Model 6	True	(3.05, -0.0988), n=36	(0.20, -0.1762), n=24	(0.10, -0.1884), n=2
Model 7	False	(32.60, -0.0164), n=40	(2.10, -0.0071), n=40	(3.00, -0.0044), n=40
Model 7	0.005	(26.75, -0.0195), n=40	(1.70, -0.0080), n=40	(2.55, -0.0054), n=40
Model 7	0.01	(21.05, -0.0232), n=40	(1.35, -0.0102), n=40	(1.85, -0.0060), n=40
Model 7	0.015	(16.00, -0.0270), n=40	(1.05, -0.0135), n=40	(1.30, -0.0078), n=40
Model 7	0.02	(11.55, -0.0312), n=40	(0.70, -0.0167), n=40	(1.00, -0.0070), n=40
Model 7	0.025	(8.20, -0.0358), n=40	(0.50, -0.0188), n=40	(0.75, -0.0081), n=40
Model 7	0.03	(5.55, -0.0404), n=40	(0.30, -0.0214), n=40	(0.45, -0.0131), n=40
Model 7	0.035	(3.65, -0.0445), n=40	(0.30, -0.0286), n=38	(0.30, -0.0137), n=39
Model 7	0.04	(2.40, -0.0490), n=40	(0.20, -0.0301), n=33	(0.20, -0.0195), n=37
Model 7	0.045	(1.50, -0.0536), n=40	(0.10, -0.0483), n=27	(0.15, -0.0273), n=30
Model 7	0.05	(0.80, -0.0576), n=40	(0.10, -0.0564), n=18	(0.10, -0.0518), n=24
Model 7	True	(2.40, -0.0948), n=40	(0.20, -0.1700), n=25	(0.10, -0.0158), n=2

Table E.38: Data coordinates for a specific group of edges with path type X-Z in Models 6 and 7. The x-coordinate is the median of occurrence percentages of the included edges; the y-coordinate is the median of the averages of all included edge weights. *n* stands for the number of unique edges included in the group of interest.

The set of *Y-Y* edges considered for Model 3 ($n = 9$) were the following:

Model 3
A.y to B.y
A.y to C.y
B.y to C.y
D.y to E.y
D.y to F.y
E.y to F.y
G.y to H.y
G.y to I.y
H.y to I.y

Table E.39: Set of *Y-Y* edges considered per model that showcase a particular trend subjected to further analysis.

Threshold	Syn	Zero	Redun
False	(31.7, 0.0049), n=9	(3.0, 0.0003), n=9	(2.4, -0.0064), n=9
0.005	(22.4, 0.0068), n=9	(2.3, 0.0004), n=9	(1.7, -0.0077), n=9
0.01	(14.4, 0.0085), n=9	(1.6, 0.0012), n=9	(1.1, -0.0106), n=9
0.015	(8.6, 0.0106), n=9	(1.2, -0.0009), n=9	(0.7, -0.0146), n=9
0.02	(4.6, 0.0126), n=9	(0.7, -0.0014), n=9	(0.5, -0.0233), n=9
0.025	(2.5, 0.0179), n=9	(0.6, -0.0024), n=9	(0.2, -0.0294), n=8
0.03	(1.3, 0.0204), n=9	(0.4, -0.0034), n=9	(0.2, -0.0318), n=6
0.035	(0.5, 0.0201), n=9	(0.3, 0.0010), n=9	(0.1, -0.0368), n=3
0.04	(0.2, 0.0200), n=9	(0.2, -0.0010), n=8	(0.1, -0.0497), n=2
0.045	(0.1, 0.0312), n=6	(0.2, -0.0021), n=5	(0.1, -0.0497), n=2
0.05	(0.1, 0.0510), n=3	(0.1, -0.0037), n=3	(0.1, -0.0533), n=1
True	(0.3, 0.0688), n=9	(0.1, 0.1386), n=1	(0.1, -0.1255), n=9

Table E.40: Data coordinates for a specific group of edges with path type Y-Y in Model 3. The x-coordinate is the median of occurrence percentages of the included edges; the y-coordinate is the median of the averages of all included edge weights. *n* stands for the number of unique edges included in the group of interest.

The set of *Y-Z* edges considered for Models 2(n = 18), 4 (n = 20), 6 (n = 36) and 7 ($n = 40$) were the following:

Model 2	Model 4	Model 6	Model 7
A.y to B.z	A.y to B.z	A.y to B.z	A.y to B.z
A.z to B.y	A.z to B.y	A.z to B.y	A.z to B.y
A.y to C.z	B.y to C.z	A.y to C.z	A.y to C.z
A.z to C.y	B.z to C.y	A.z to C.y	A.z to C.y
B.y to C.z	C.y to D.z	A.y to D.z	B.y to C.z
B.z to C.y	C.z to D.y	A.z to D.y	B.z to C.y
D.y to E.z	D.y to E.z	B.y to C.z	B.y to D.z
D.z to E.y	D.z to E.y	B.z to C.y	B.z to D.y
D.y to F.z	E.y to F.z	B.y to D.z	C.y to D.z
D.z to F.y	E.z to F.y	B.z to D.y	C.z to D.y
E.y to F.z	F.y to G.z	C.y to D.z	C.y to E.z
E.z to F.y	F.z to G.y	C.z to D.y	C.z to E.y
G.y to H.z	G.y to H.z	E.y to F.z	D.y to E.z
G.z to H.y	G.z to H.y	E.z to F.y	D.z to E.y
G.y to I.z	H.y to I.z	E.y to G.z	D.y to F.z
G.z to I.y	H.z to I.y	E.z to G.y	D.z to F.y
H.y to I.z	I.y to J.z	E.y to H.z	E.y to F.z
H.z to I.y	I.z to J.y	E.z to H.y	E.z to F.y
	J.y to A.z	F.y to G.z	E.y to G.z
	J.z to A.y	F.z to G.y	E.z to G.y
		F.y to H.z	F.y to G.z
		F.z to H.y	F.z to G.y
		G.y to H.z	F.y to H.z
		G.z to H.y	F.z to H.y
		I.y to J.z	G.y to H.z
		I.z to J.y	G.z to H.y
		I.y to K.z	G.y to I.z
		I.z to K.y	G.z to I.y
		I.y to L.z	H.y to I.z
		I.z to L.y	H.z to I.y
		J.y to K.z	H.y to J.z
		J.z to K.y	H.z to J.z
		J.y to L.z	I.y to J.z
		J.z to L.y	I.z to J.z
		K.y to L.z	I.y to A.z
		K.z to L.y	I.z to A.y
			J.y to A.z
			J.z to A.y
			J.y to B.z
			J.z to B.y

Table E.41: Set of *Y-Z* edges considered per model that showcase a particular trend subjected to further analysis.

Threshold	Syn	Zero	Redun	
Model 2	False	(34.40, 0.0105), n=18	(3.35, -0.0026), n=18	(3.00, -0.0106), n=18
Model 2	0.005	(27.95, 0.0126), n=18	(2.65, -0.0029), n=18	(2.45, -0.0129), n=18
Model 2	0.01	(20.70, 0.0154), n=18	(2.30, -0.0033), n=18	(1.85, -0.0158), n=18
Model 2	0.015	(15.35, 0.0184), n=18	(1.80, -0.0045), n=18	(1.40, -0.0186), n=18
Model 2	0.02	(11.15, 0.0219), n=18	(1.35, -0.0036), n=18	(1.00, -0.0223), n=18
Model 2	0.025	(7.85, 0.0257), n=18	(1.00, -0.0072), n=18	(0.55, -0.0287), n=18
Model 2	0.03	(5.30, 0.0299), n=18	(0.80, -0.0106), n=18	(0.40, -0.0344), n=18
Model 2	0.035	(3.55, 0.0352), n=18	(0.50, -0.0075), n=18	(0.30, -0.0396), n=15
Model 2	0.04	(2.20, 0.0411), n=18	(0.40, -0.0102), n=18	(0.20, -0.0457), n=13
Model 2	0.045	(1.50, 0.0440), n=18	(0.40, -0.0074), n=15	(0.20, -0.0500), n=10
Model 2	0.05	(1.00, 0.0439), n=18	(0.30, -0.0180), n=15	(0.10, -0.0547), n=8
Model 2	True	(3.60, 0.0978), n=18	(0.10, 0.1813), n=4	(0.30, -0.1653), n=17
Model 4	False	(28.25, 0.0063), n=20	(2.60, -0.0018), n=20	(2.35, -0.0087), n=20
Model 4	0.005	(23.00, 0.0078), n=20	(2.20, -0.0021), n=20	(1.95, -0.0100), n=20
Model 4	0.01	(17.30, 0.0096), n=20	(1.70, -0.0018), n=20	(1.50, -0.0113), n=20
Model 4	0.015	(12.65, 0.0113), n=20	(1.35, -0.0034), n=20	(1.10, -0.0162), n=20
Model 4	0.02	(8.70, 0.0138), n=20	(1.00, -0.0053), n=20	(0.70, -0.0180), n=20
Model 4	0.025	(6.25, 0.0165), n=20	(0.75, -0.0107), n=20	(0.40, -0.0204), n=19
Model 4	0.03	(4.30, 0.0206), n=20	(0.50, -0.0136), n=20	(0.30, -0.0332), n=19
Model 4	0.035	(2.75, 0.0212), n=20	(0.50, -0.0108), n=19	(0.20, -0.0371), n=17
Model 4	0.04	(2.05, 0.0226), n=20	(0.40, -0.0242), n=19	(0.10, -0.0476), n=13
Model 4	0.045	(1.20, 0.0234), n=20	(0.20, -0.0271), n=19	(0.10, -0.0516), n=10
Model 4	0.05	(0.75, 0.0291), n=20	(0.10, -0.0545), n=18	(0.10, -0.0543), n=7
Model 4	True	(2.90, 0.1014), n=20	(0.10, 0.1778), n=7	(0.10, -0.1586), n=17
Model 6	False	(16.25, 0.0009), n=36	(1.90, -0.0005), n=36	(1.90, -0.0075), n=36
Model 6	0.005	(13.60, 0.0011), n=36	(1.65, -0.0003), n=36	(1.50, -0.0092), n=36
Model 6	0.01	(10.50, 0.0018), n=36	(1.30, 0.0007), n=36	(1.10, -0.0111), n=36
Model 6	0.015	(8.00, 0.0023), n=36	(1.00, 0.0000), n=36	(0.70, -0.0146), n=36
Model 6	0.02	(5.95, 0.0032), n=36	(0.80, 0.0011), n=36	(0.45, -0.0203), n=36
Model 6	0.025	(4.30, 0.0031), n=36	(0.50, 0.0040), n=36	(0.40, -0.0272), n=35
Model 6	0.03	(3.00, 0.0032), n=36	(0.40, 0.0042), n=35	(0.20, -0.0335), n=30
Model 6	0.035	(2.10, 0.0046), n=36	(0.25, 0.0027), n=34	(0.20, -0.0396), n=25
Model 6	0.04	(1.40, 0.0054), n=36	(0.20, 0.0051), n=32	(0.10, -0.0423), n=21
Model 6	0.045	(0.95, 0.0059), n=36	(0.10, 0.0235), n=29	(0.10, -0.0523), n=13
Model 6	0.05	(0.60, 0.0014), n=36	(0.10, -0.0131), n=21	(0.10, -0.0559), n=9
Model 6	True	(1.30, 0.1030), n=36	(0.10, 0.1782), n=12	(0.10, -0.1624), n=16
Model 7	False	(29.85, 0.0060), n=40	(2.60, -0.0018), n=40	(2.40, -0.0073), n=40
Model 7	0.005	(23.65, 0.0072), n=40	(2.30, -0.0020), n=40	(1.90, -0.0088), n=40
Model 7	0.01	(18.15, 0.0085), n=40	(1.80, -0.0021), n=40	(1.40, -0.0113), n=40
Model 7	0.015	(13.30, 0.0102), n=40	(1.30, -0.0026), n=40	(0.90, -0.0134), n=40
Model 7	0.02	(9.45, 0.0124), n=40	(1.00, -0.0012), n=40	(0.60, -0.0155), n=40
Model 7	0.025	(6.45, 0.0141), n=40	(0.75, -0.0021), n=40	(0.40, -0.0207), n=40
Model 7	0.03	(4.20, 0.0176), n=40	(0.60, 0.0005), n=40	(0.30, -0.0339), n=37
Model 7	0.035	(2.70, 0.0182), n=40	(0.40, 0.0017), n=40	(0.20, -0.0395), n=36
Model 7	0.04	(1.75, 0.0193), n=40	(0.20, 0.0008), n=38	(0.10, -0.0488), n=27
Model 7	0.045	(1.20, 0.0222), n=40	(0.20, -0.0099), n=33	(0.10, -0.0501), n=21
Model 7	0.05	(0.70, 0.0227), n=40	(0.20, 0.0016), n=23	(0.10, -0.0520), n=16
Model 7	True	(1.10, 0.1010), n=40	(0.10, 0.1753), n=10	(0.10, -0.1523), n=25

Table E.42: Data coordinates for a specific group of edges with path type Y-Z in Models 2, 4, 6 and 7. The x-coordinate is the median of occurrence percentages of the included edges; the y-coordinate is the median of the averages of all included edge weights. *n* stands for the number of unique edges included in the group of interest.

The set of *Z*-*Z* edges considered for Model 3($n = 9$) were the following:

Model 3
A.z to B.z
A.z to C.z
B.z to C.z
D.z to E.z
D.z to F.z
E.z to F.z
G.z to H.z
G.z to I.z
H.z to I.z

Table E.43: Set of *Z*-*Z* edges considered per model that showcase a particular trend subjected to further analysis.

Threshold	Syn	Zero	Redun
False	(43.00, 0.0107), n=9	(4.20, -0.0029), n=9	(4.00, -0.0128), n=9
0.005	(37.70, 0.0122), n=9	(3.80, -0.0031), n=9	(3.30, -0.0154), n=9
0.01	(31.10, 0.0145), n=9	(3.20, -0.0034), n=9	(2.70, -0.0172), n=9
0.015	(25.70, 0.0166), n=9	(2.70, -0.0060), n=9	(2.00, -0.0214), n=9
0.02	(20.80, 0.0189), n=9	(2.10, -0.0065), n=9	(1.70, -0.0221), n=9
0.025	(16.80, 0.0215), n=9	(1.60, -0.0089), n=9	(1.40, -0.0288), n=9
0.03	(13.20, 0.0238), n=9	(1.40, -0.0128), n=9	(0.80, -0.0355), n=9
0.035	(10.50, 0.0285), n=9	(1.10, -0.0138), n=9	(0.70, -0.0368), n=9
0.04	(7.90, 0.0313), n=9	(0.80, -0.0158), n=9	(0.60, -0.0489), n=9
0.045	(5.90, 0.0344), n=9	(0.70, -0.0157), n=9	(0.40, -0.0542), n=8
0.05	(4.70, 0.0370), n=9	(0.40, -0.0104), n=9	(0.25, -0.0589), n=8
True	(1.20, 0.1438), n=9	NA	(0.10, -0.2278), n=8

Table E.44: Data coordinates for a specific group of edges with path type Z-Z in Model 3. The x-coordinate is the median of occurrence percentages of the included edges; the y-coordinate is the median of the averages of all included edge weights. *n* stands for the number of unique edges included in the group of interest.