Iterative Graph Perturbation to Identify High Impact Nodes with Application to Genetic Regulation of Onset of Puberty

Basak Selcuk
M. S. Physics, University of Florida, 2008

Presented to the
Center for Spoken Language Understanding
within the Oregon Health & Science University
School of Medicine
in partial fulfillment of
the requirements for the degree
Doctor of Philosophy
in
Computer Science & Engineering

April 2020

Center for Spoken Language Understanding
School of Medicine
Oregon Health & Science University

CERTIFICATE OF APPROVAL

This is to certify that the Ph. D. dissertation of

Basak Selcuk

has been approved.

Kemal Sonmez, Thesis Advisor
Associate Professor

Peter A. Heeman
Associate Professor

Alejandro Lomniczi
Assistant Professor

Steven D. Bedrick
Associate Professor

Acknowledgements

This dissertation would not be possible without my thesis advisor, Kemal Sonmez, my thesis co-advisor Alejandro Lomniczi, and my graduate advisor Peter Heeman. I would like to thank Kemal for being the best mentor for me. He has been a continuous source of knowledge, direction, understanding and support. With his supervision, discussions and suggestions, this research project has evolved into its final shape.

I would like to thank Alejandro for his never-ending support. I am grateful to him for inviting me to his group meetings. The topics and discussions in those meetings helped me close the gaps in my understanding of complex biological systems and eventually shaped this thesis' biological basis. He has been a great source of knowledge and help for my thesis.

I would like to thank Peter. Without his persistence and supervision, this thesis would never have existed. He is an extraordinary graduate advisor who works above and beyond his duties. I also would like to thank him for helping me during the write-up of this thesis.

I would like to thank Sergio Ojeda, a world-renowned expert on puberty, who introduced me to this exciting project and its possibilities. I am grateful for his inputs early on in this research.

I would like to thank Hollis Wright who shared his vast knowledge on programming and systems biology, and helped me with my projects. I would like to thank Steven Bedrick for his explanations and help on cluster computing. I would like to thank my past and present committee members for their time and suggestions; without their support, this thesis would not have progressed.

I would like to thank to my father and my brothers for their incredible support and encouragement. I would especially like to thank my mother, my husband and my two sons for their help, inputs, ideas and, most importantly, their patience.

Contents

List of Tables

List of Figures

Abstract

Iterative Graph Perturbation to Identify High Impact Nodes with Application to Genetic Regulation of Onset of Puberty

Basak Selcuk

Doctor of Philosophy
Center for Spoken Language Understanding
within the Oregon Health & Science University
School of Medicine

April 2020
Thesis Advisor: Kemal Sonmez

Complex networks that connect hundreds or thousands of nodes together can function properly through the use of nodes interacting, affecting and regulating one another. Genetic networks underlying high-level hormonal changes are also believed to be complex and most of the time are not yet very well understood. There are also different networks with common genes that start to activate or deactivate at the same time, which also adds to the difficulty of the problem of understanding genetic networks. Therefore, finding genes transcriptionally active in a gene set that are responsible for a change in the human body is a key point in reaching the underlying network structure. Here, we present a new technique that searches for these nodes in a set of variables that connect to form a network. It is a stepwise greedy search that investigates the change in a chosen network when one node is taken out at a time. Our simulated genetic data results show that our method is successful for differentiating between transcriptionally active nodes and background nodes with a p-value of less than 0.05. As real biological data, we used rat RNA-seq data taken for initiation-of-puberty research. Our method found new genes as well as confirming previously known genes with significant enrichment results taking charge in functions such as transcritional binding, histone modifications, and reproductive development.

Chapter 1

Introduction

Networks are groups of similar or complementary components that interact with one another to function properly. Biological networks, such as gene regulatory networks or protein–protein interaction networks, are examples in which the variables—in this case genes and proteins—interact to control a genetic activity or to build, activate or stop a specific protein. Networks can also explain biological phenomena such as learning or remembering, by modeling how the neurons in our brains interact. Typically, hormone signaling networks regulate the hormonal activities of the body. There are also many examples in our daily lives in which networks express states, such as the flight routes of an airline, website links, roads of a city, or an email chain.

The elements of a network are called nodes or vertices. The connections between these nodes are called edges. Nodes can have different numbers of edges with different connection strengths depending on where they are located and how the network operates. In the above example of flight routes, the nodes would be the airports, and the airplane routes from one airport to another would be the edges of the network. For genetic networks, the nodes would be genes, and the edges would describe whether the genes interact with one another or not.

Although there can be small or randomly connected networks, real-life networks that operate at high levels show large-network characteristics with specific properties. These networks include node clusters and show high connectivity within these clusters but very low connectivity between clusters. Additionally, within a specific cluster there are hub nodes that have much higher edge counts than the average edge count for the network. Such networks are called scale-free networks.

Barabasi [9] hypothesizes that these scale-free networks are resilient to most outside perturbations, as long as their hubs are untouched. When they lose even a small number of their hubs, however, they cannot function properly. There are multiple statistical approaches for modeling these complex, real-life networks, but the mathematical representations resulting from these different approaches show unequal final structures, with different node clusters and edge counts, even though they use the same input data. These inconsistencies of the mathematical approaches, and

the differences between the real-life and emprical results show that current methods are not sophisticated enough. Thus, relying on the hub nodes identified by these algorithms might not give us a true understanding of how a scale-free network will behave when a change is introduced or let us know which nodes might be real hubs.

Here, we propose a new procedure for finding impact node sets—nodes whose removal would have the greatest effect on network functionality. Our process searches one node at a time by using local optima given a set of variables and their observable values. At each iteration, we take out one node from the input node list and construct a new network. We look at the impact of eliminating that node by comparing the new network to the complete network.

To construct networks and determine node clusters, we used previously published network inferences, clustering and similarity measures. Our algorithm ranked the potentially critical nodes according to their edge counts and edge strengths, which allowed us to search for hubs and for other nodes with fewer but still important connections. To rank the networks, we used comparison measures such as edges lost, nodes lost and the Jaccard index for cluster similarity.

To determine the efficiency of our procedure, we generated and used simulated gene expression data. We generated 100 networks with 100 genes each. Each of these 100 networks includes 10 transcriptionally active genes. By using two levels of comparisons, node level and network level, we ranked the nodes according to their impact in the network. By using our beam search algorithm, impact node finder (INF), we created node sets including 1 to 10 genes which showed highest potential to be critical when eliminated from the network. Our simulation results show that our process can select transcriptionally active nodes from among background nodes with a greater probability than random selection by using either Jaccard index or edges lost as network comparison measures. We compared our results with the nodes found by using node connectivity measure to find the active nodes from a network. Our results with jaccard index network comparison measure show that our algorithm can find more active nodes in a set of 10 genes than the node connectivity measure alone.

Biological processes such as protein–protein interaction are known to involve large, complex, scale-free networks with cluster of genes functioning together. Although there are some known networks of biological mechanisms for simple organisms such as E. coli, most complex biological processes in evolutionarily higher organisms have unknown underlying networks. One experimental technique to understand which genes might have a role in a genetic network is to silence or regulate chosen genes by using specific enzymes or small RNAs. This process needs prior knowledge for better results. In addition, the experiments can be expensive, and the samples may not by reproducible. Our approach, by contrast, serves as a mathematical method to compute and

predict possible sets of these knockdown genes to regulate beforehand.

One of these complex biological processes is the initiation of puberty in females. Puberty is the maturational process during which the body experiences the reactivation of the hypothalamic-pituitary axis and the maturation of secondary sexual characteristics, which include increased physical growth and changes in reproductive organs. The first menstrual period, menarche, signals the beginning of the capacity to reproduce and is clinically used as the milestone to identify the end of a maturational period called Age At Menarche (AAM).

Since the 18th century, AAM has fallen in industrialized countries. Today, the average AAM is 12–13 years of age, down from 16–17 in the 19th century. Research in this field suggests that both environmental and genetic factors influence the AAM. The environmental factors include endocrine disruptors such as BPA used in the production of plastics, which behaves like estrogen in the body; food additives used in packaged food, which behave like growth hormones; and life choices such as being overweight.

The early onset of puberty has lifelong physiological and psychological effects. The early maturation of the body without the proper maturation of the mind can cause lower self-esteem or even depression. It might also result in early menopausal age, high adult body weight and diabetes. Therefore, understanding puberty clinically, psychologically and physiologically, and investigating the genetic pathways that control the process, is key to devising new strategies to treat endocrine disorders of puberty and fertility.

However, the abovementioned complex changes in physical state suggest a complex neuroendocrine system, which includes a well-organized collaborative network of hormones, growth factors and various organs. This is controlled by a complex genetic network. Biological background also suggests that puberty-related genes also take part in other processes, such as tumor suppression, inflammation, metabolic control and stress, which adds another dimension to this complex mechanism. The complexity of the system hinders the findings of clear-cut results and solutions to the problem such as early AAM and disorders of puberty and fertility.

In this thesis, we present our results about the genetic control of puberty initiation from our impact node finder algorithm. Collaborating with Dr. Alejandro Lomniczi and his colleagues at the Oregon National Primate Research Center (ONPRC), we use high-throughput gene expression data taken from rat hypothalamic samples that include thousands of genes, covering the developmental stages from prepuberty to postpuberty ages.

The timing and duration of mammalian puberty varies from species to species. In mice and rats, for example, puberty starts about 21 days after birth and is completed by 33 to 36 days of age. In monkeys or humans, on the other hand, the beginning and end of puberty is measured in

years. Even though the onset and duration of puberty for these species show this much variance, the genetic pathway responsible for these changes shows incredible similarities from one species to another. Therefore, studies using animal models help our understanding of how human genetics works.

The onset of puberty is initiated by increased pulsatile release of the gonadotropin releasing hormone (GNRH) from the hypothalamus. Glial and neuronal populations in the hypothalamus regulate GNRH release form neuronal terminals. In general terms, glial cells are excitatory to GNRH neurons, whereas neurons can activate GNRH release (such as glutamatergic, kisspeptin and NPY neurons) or repress GNRH release (such as GABAergic, enkephalinergic and POMC neurons).

From a series of papers published by Dr. Lomniczi and his colleagues [158, 92, 62, 106, 107, 89, 91], a series of genes involved in the initiation of puberty are identified, such as EAP1 (enhanced-at-puberty-1), kisspeptin (KISS1) and thyroid transcription factor 1 (TTF1), whose increase is vital for the timely initiation of puberty. On the other hand, as puberty progresses, downregulation by puberty repressors, such as embryonic ectoderm development (EED) and chromobox homolog (CBX7), is required. The genes in this second group are epigenetic repressors of the pubertal process. Finally, a third group of genes, including zinc finger, BTB domain-containing protein 16 (ZBTB16), and B-cell lymphoma 6 (BCL6), are referred to as repressors of repressors. The expression of this group of repressors increases during pubertal maturation, and it is believed that they inhibit the expression of epigenetic repressors.

In this thesis, we construct and use a gene network to explain the onset of puberty. As explained above, puberty is also the result of a complex and robust network. It is mostly resilient to environmental effects, and it has many redundant pathways to ensure it responds properly under stress. As long as it maintains a number of specific genes that directly start the pubertal process, the network still achieves the expected outcome even after losing some genes on the way, producing pubertal delay. This suggests a scale-free topology with cluster formations and hub interactions.

Our first objective is to understand such a complex network and find a mathematical representation to explain the biological interactions between puberty-relevant genes. Our goal is to explain the experimental and biological findings and also discover new relations between puberty-related genes and genes with other functions by using mathematical predictions. We are also interested in environmental factors, such as obesity and nutrition, that affect gene expression and pubertal initiation.

Our second objective, by using the procedure introduced above, is to investigate the changes in network structure when one gene or a set of genes are taken out of the system. As mentioned

previously, a common methodology in systems biology research is to isolate the effect of one gene or a set of genes at a time to understand how the genes in the network interact with one another. The choice of which genes to take out of the system is a crucial point for this type of experiment and requires some a priori knowledge of the system. Here, our algorithm aims to identify the genes that would either result in puberty failure or delayed puberty if knocked down in a biological experiment.

The gene sets our algorithm found included KISS1R, which is a known puberty-triggering gene, and DNMT3B, an epigenetic modifier of interest. We used enrichment analysis to separate functional gene groups among our results to ensure our final gene-set results would include biologically meaningful subsets.

The specific contributions of this work are:

1. Proposed method for sorting networks using graph theoretic distances.
2. Selecting the perturbation with the greatest impact using the proposed graph sorting method.
3. Algorithm for iterative network perturbations to identify the list of most impactful nodes.
4. Demonstration of the algorithm on simulated gene expression data.
5. Application of the algorithm to real world data on regulation of initiation of puberty.

In this thesis, we aim to understand the changes in a network if a perturbation is introduced. In our backgoround chapters 2, 3, and 4 we are going to cover important literature work to understand our algorithm. In chapter 2, we first examine the structure of networks, their properties, and mathematical and statistical approaches to represent and analyze networks. We then explain the biological component of our research. We review the definition of genes, how they work, how their activity is measured, and how they are modeled in chapter 3. We also address the biological background of onset of puberty in chapter 4. In chapter 5, we will explain our algorithm in terms of its components and its metrics. In chapter 6, we will show our results for simulated gene expression data that confirms the use of our algorithm in idetifying transcriptionally active nodes in a network. In chapter 7, we will introduce our findings for the initiation of puberty data. In chapter 8 we will cover our future work for the improvements in the algorithm and for the biological data.

Chapter 2

Review of Network Theory

In our research, we are searching for the critical nodes of a complex network. Here, in chapter 2, we will first give a short description of networks in general, and their related attributes. We will look at important aspects of networks such as clustering in section 2.1. Then, in section 2.2 we will explain different types of networks, specifically complex networks and their properties. Last, in section 2.3, we will review different algorithms that are used to predict underlying network sturucture of a set of nodes.

2.1 Network Theory

Networks are efficient mechanisms when it comes to explaining large interacting components such as seen in biological systems. Network theory is based on graphs and graph attributes. In this research, we are using networks, their characteristics and their mathematical representations. This section introduces basic knowledge of graphs as found in [17, 20] unless noted otherwise.

Networks are mathematical representations where variables either interact with one another to reach a final outcome or they express tangible or intangible states. The size and characteristics of networks vary greatly. No matter what size or specification, all networks can be described as some type of a graph where the variables are a set of specific nodes and the interactions between these nodes are a set of specific links, called edges. These edges can be any type of associations: correlations, dependencies, similarities, dissimilarities, or distance between nodes.

2.1.1 Graph Representation of a Network

According to the definition and specifications above, a network is a finite graph with a set of unique nodes (vertices), and a set of unique edges, represented as node pairs where the two nodes are different from one another. The simple graph representation ensures that there are no loops or multiple edges. A network can be either directed or undirected depending on whether the edges

have a direction, with an arrow specifying the source node and target node, or the edges have no direction. The edges also determine whether the network is unweighted or weighted. For a weighted network, the edges specify quantitatively how two nodes are related to each another. For an unweighted network, on the other hand, the edges show a connection between two nodes without showing the magnitude of the relation. In this thesis, we only use undirected and weighted graph representations. For simplicity, we are going to describe unweighted graphs first and explain weighted graphs afterward.

An undirected network can be represented mathematically as a symmetric, square matrix, called an adjacency matrix. The row and column identifiers of the adjacency matrix are identical sets representing the nodes of the network. The matrix element (i, j) specifies the edges. For unweighted networks, the matrix elements are 0 and 1. The input is zero when there is no connection between two nodes, and 1 when they are connected. The value of the diagonal would also be one, which is not represented as an edge in a network with simple graph representation.

The graph representation of a network can be described as

$$A = [a_{ij}] \tag{2.1}$$

For an undirected and weighted network, the matrix elements are values -1 to 1. An input of zero still means that there is no connection between two nodes. The boundaries -1, and 1, refer to perfect positive and negative associations, respectively. The value of diagonal elements are one, as in the unweighted graph case. The off-diagonal elements can be any values between the boundaries, depending on how correlated two connecting nodes are.

2.1.2 Network Characteristics

Once an adjacency matrix is formed, graph attributes can be used to describe how the elements of the network interact with one another and how the network is related to physical outcomes. The characteristics of a network depend on how the nodes are connected. Additionally, node and edge attributes can give mathematical clues about these characteristics.

An adjacency list specifies which nodes in the network are connected to each other. Two nodes with an edge connecting them in the list are said to be adjacent. Given a node, those nodes connected to it with a single direct edge are said to be first neighbors. These are the node's nearest neighbors, and they might have the closest relations and be most affected by a change of a state of the chosen node. Starting from a given node, those nodes that can be reached by tracing two edges are second neighbors, and the neighboring of a node continues in this fashion. A subgraph

is a subset of nodes and their connecting edges.

Connectivity, Centrality and Related Attributes

In a fully connected graph, all nodes and edges can be reached by a path started from a random origin and traced through the graph elements. Network connectivity refers to the minimum number of network elements to be removed to make a connected graph into disconnected subgraphs. The elements can either be a subset of nodes or a subset of edges.

Connectivity can be used to assess how well a network can respond to an external stimulus and continue to function properly. A highly connected network with many edges between nodes can still function even if it loses some of its edges due to some environmental effect. Thus, even if there is no direct information exchange between some nodes after some edges are lost, as long as the network stays connected, then the interactions between other nodes can still manage to overcome the effect.

A connectivity related measure, node degree, refers to the total number of edges a node has. Since the edge for each pair of nodes in an undirected network is counted twice when calculating the node degree, the sum of node degree of a network is twice as large as the total number of edges in the network. The maximum node degree specifies the highest number of edges a node in the network has. In network theory, hubs are nodes with a node degree value much higher than the average node degree of the network. It is not, however, specified how much of a difference from the average value makes a node become a hub.

For a weighted network, the above description of node degree still holds, but some edges in the network carry more weight than others, making those edges more important when it comes to interactions. In order to incorporate this characteristic, Barrat et al. [10] introduced the term node strength, s_i. For the node i, the unweighted node degree, k_i is as follows

$$k_i = \sum_{j=1}^{N} a_{ij} \tag{2.2}$$

where a_{ij} with values 0 or 1, are the elements of the adjacency matrix, A, with size NxN. Weighted degree, i.e. node strength, is

$$s_i = \sum_{j=1}^{N} a_{ij} \tag{2.3}$$

with weights a_{ij}.

The path length in a network also explains how connected a network is. For an unweighted network, the shortest path between two nodes is the minimum number of edges that must be

traced to move from one node to the other. For a weighted network, it is the path with minimum total weight of the edges traced. For a weighted network where edge weights can be negative, the shortest path might not exist. The average path length is the average of all the shortest paths in the network. On the other hand, the diameter of the network is the longest path in the network.

Edge density is the ratio of the total number of edges in a network or a subnetwork to the total number of edges if the network or subnetwork were maximally connected. For a network with an edge set of E, and a node set of V, the edge density E_D is

$$E_D = E/(V(V-1)/2) \tag{2.4}$$

The number E_D is between 0 and 1. The higher the number, the denser the graph. A value of one for a subnetwork means the subnetwork is fully connected. The number also shows how sparse a graph is as the ratio can reach zero. Therefore, edge density, also known as network density, if all the edges of a network are taken into account, shows how strongly the components of a network interact. Since the adjacency matrix summarizes the edges of the network, edge density is also calculated as the mean adjacency.

For both weighted and unweighted networks, centrality in a network is a measure of how important an element is compared to all others. Centrality of a node in a graph is a measure of how effective it is compared to all other nodes. Similarly, edge centrality is a measure to rank the edges. There are different types of centrality measures. Degree, and betweenness centrality measures are the most common ones. Degree centrality ranks the nodes according to their degrees. The node degree is defined as in the equations 2.2 and 2.3 for unweighted and weighted cases respectively.

Betweenness centrality is proportional to the total number of the shortest paths going through a node. For node x, define $\sigma_{st}(x)$ as the shortest path between two other nodes s and t going through x, then the betweenness centrality $BC(x)$ is the sum of the ratio of all the shortest paths between each pair of nodes s and t in the network that go through the node x, to the total number of shortest paths of each pair of nodes that go through the nodes s and t where the shortest paths can be defined as finite.

$$BC(x) = \sum_{st} \frac{\sigma_{st}(x)}{\sigma_{st}} \tag{2.5}$$

The betweenness centrality of a node shows the potential of that node's being a bridge between node clusters. Edge betweenness centrality similarly ranks the edges according to their corresponding betweenness scores.

In Chapter 5 we will use node degree and node strength to rank the nodes in a network to find which nodes have the most effect on the network. In Chapter 8, we are going to define node betweenness as one of the additional node ranking measures that we can also use to find other critically important nodes. At every iteration and for every new network we also look at the edge density value to see how the connectivity is affected by a perturbation done with removing nodes.

2.1.3 Clusters in a Network and Related Attributes

In most real-life networks, nodes tend to be within clusters. If a network can be separated into subnetworks, where in every subnetwork, the connectivity measure is greater than it is for the full network, then each subnetwork is said to be a cluster. In graph theory, the terms community, cluster, and module are used interchangeably.

One of the criteria we use in chapter 5 to decide whether a network is perturbed enough to make it unable to function properly is looking at how the clustering changes before and after the introduction of a perturbation. Here, in this section we will go over the different clustering measurements and the algorithms used to detect or predict clusters in a network.

Clustering Coefficient

Clustering coefficient is a measure to quantify the clustering behavior of a network. It is said that the higher the clustering coefficient, the more the nodes in a network tend to be within clusters in a network. Clustering coefficient can be measured globally, locally, or as an average. Here, we are going to focus on the global clustering coefficient. The method to calculate clustering coefficient is based on three-node groups and how strongly they are connected in a network. If all three nodes are connected to one another with an edge, they form a closed triplet. These closed triple nodes are called transitive triplets. The global clustering coefficient is the ratio of the number of these transitive triplets to total number of the connected triplet nodes. The resulting value of this calculation is a constant of a network. By using the formulation of Wasserman and Faust [165], the clustering coefficient C can be described as

$$C = \frac{|transitive \quad triplets|}{|triplets|} \tag{2.6}$$

Modularity

Real-life complex networks have clusters of nodes instead of random connections. Hub nodes in a real-life network tend to be in these clusters. Some even become the center of the clusters they belong to according to the above mentioned centrality measures. This behavior may be a result

of a common function, where the nodes work together to reach a common outcome. Therefore, these clusters are said to divide the network into functional modules. The modularity measure of a network indicates how well the network can be divided into these modules. The higher the modularity, the more structured the network modules are. Here, we are going to use the formulation adapted from Clauset [24] and Newman [113] on modularity.

Modularity is represented by partitioning. For a network partitioned into K different clusters, the set of clusters are $C = \{c_1, \ldots, c_K\}$. For a network with an edge set of E, the modularity Q is defined as [113]

$$Q = \frac{1}{4|E|} \sum_{ij} \left[a_{ij} - \frac{k_i k_j}{2|E|} \right] \delta(c_i, c_j) \tag{2.7}$$

The sum is over all node pairs i and j. c_i specifies the cluster i belongs to and c_j specifies the cluster j belongs to. The values k_i and k_j are the node degree values for nodes i and j, respectively. The adjaceny matrix element a_{ij} specifies the edge strength. The multiplication with the delta function $\delta(c_i, c_j)$ ensures that an edge between nodes i and j does not count if the two nodes are in different clusters.

Cluster Detection Algorithms

There are several algorithms that try to optimize a criterion function for dividing the network into clusters. One of these criterion functions is the modularity measure. Others can be any similarity or distance measure that is based on the network characteristics, such as Euclidean distance or cosine similarity [170].

There are hierarchical algorithms that try to maximize the network modularity or similarity measures, or to minimize distances. A top-down algorithm starts by first assuming all nodes are in one cluster. Then, according to the chosen measure, clusters are formed by dividing nodes into smaller groups such that a criterion is optimized. At every step the criterion is recalculated with the new clusters in mind. For a bottom-up algorithm, by contrast, all nodes are assumed to be one cluster each, and new clusters are formed by merging smaller clusters together to optimize the chosen criterion. This is recalculated at every step to obtain the best partitioning. The algorithm stops when the criterion does not change after iteration. Newman and Girvan [111] use betweenness centrality of edges to find those edges that fall between clusters. With a top-down algorithm they start with a network and calculate betweenness values for all edges. They eliminate edges with the highest edge betweenness values and recalculate the betweenness values for the remaining edges to find the next highest valued edge. A bottom-up algorithm, k-means clustering tries to minimize the sum of squared errors between cluster means and the data points.

Cluster Similarity and Network Comparisons

There are similarity measures that calculate how two cluster partitionings of the same set of nodes are similar to one another. The similarity measure covered in this thesis is the Jaccard index. It is the ratio of the number of nodes that clustered together for two partitionings to the total number of nodes in clusters for the same two partitionings. For two different clustering approaches of a set of nodes in a network, let's assume the sets of clusters are A and B. A and B can have different number of clusters with different sizes. The Jaccard index is the measure which shows how these two clustering approaches are similar or different from one another. So, for two different partitioning of A and B of a network, the Jaccard index is

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{2.8}$$

The higher the value of J, the more similar two partitionings A and B are. The nominator determines the number of nodes which cluster together in both partitionings, whereas the denominator is the total number of nodes in all clusters in both partitionings including nodes that cluster together and nodes that change clusters for the two partitionings.

2.2 Complex Networks

There are networks with large numbers of nodes and edges that exhibit characteristics different from those exhibited by small networks or man-made structures such as lattices. These complex networks show specific common features, such as having clusters of nodes with most network edges residing in these clusters, and preferential node choices that most of the other network nodes connect to. This section reviews the three types of real-life networks and the statistical models to explain their characteristics.

2.2.1 Types of Networks

Random Networks

Random networks, as described in [9], are networks in which the nodes are assumed to be connected together without any prior knowledge or preferences. The clustering coefficients of these networks are extremely low, and there are no hub nodes or other complex network structures. Hubs, as defined in 2.1.2 are nodes with much higher node degree values compared to mean node degree of the network. Node degree distribution of a random network is either a binomial or a Poisson distribution depending on the model and the number of edges in the network. Most real-life random

sparse networks are explained with a Poisson distribution. A coin-toss sequence or the information exchange between people who do not know each other show random network characteristics. Since random networks do not show any complex structures, the average path length of the random network tends to be short.

Small World Networks

Small-world networks, as described in [166], are networks that generate local clusters but they do not need to follow preferential node attachment choices. Their clustering coefficients are higher than what is found in random networks, and they have high modularity values, even though they do not include the same structures as complex networks. Their diameter is much smaller than those of complex networks, and they have a short average path length, the same as random networks. A complex network can be turned into a small-world network by adding new edges that would decrease its diameter drastically to enable shorter path length interactions between distant nodes. The diameter represents the size of the network, and decreasing it makes the network smaller. The new edges are chosen to be long-range edges to connect nodes with large path-length distance values.

Scale-free Networks

Some large real-life networks with complex structures cannot be modeled or understood with the above mentioned network types. The topological characteristics of these complex networks include high modularity values, large clusters with high connectivity rates, and hub nodes with a node connection probability much higher than that of a random connection. These networks are called scale-free or scale-invariant networks [147].

Scale-free means that the structure of the network looks the same from any distance. Scale-free networks develop in such a way to enable high clustering coefficients and high values of node preference instead of random connections, which makes them different from small-world and random networks. The probability that a new node will connect to an already connected node is higher than the probability of a random connection. Hence, the network favors nodes with high connectivity rates to have more nodes connected and stay with same clustering structure instead of starting a new structure. The probability of a new node connecting to a node i with node degree k_i is proportional to

$$p(\text{connecting to node i}) = \frac{k_i}{\sum_j k_j} \tag{2.9}$$

where the denominator is the sum of all the node degrees in the network, which is twice the total number of edges, as mentioned earlier. Node degree distribution in a scale-free network follows a power-law. The probability of selecting a node randomly follows a power-law curve. Power-law behavior is exhibited after the network evolves to include enough nodes and a minimum number of edges per node. Power-law distribution for a node degree k, and exponent α is,

$$p(k) \sim k^{-\alpha} \tag{2.10}$$

This equation implies that the degree distribution has no scale. No matter the number k, the probability always follows the above rule, even if it is increased by a scale variant. The distribution is independent of the initial values. The power law curve makes the node degree distribution an important attribute of the network. The network is composed of a large number of nodes with very low numbers of edges and a few nodes with very high numbers of edge connections.

Most biological networks, such as protein-protein interaction networks, show scale-free properties. Complex biological systems require clustered gene groups and hierarchical structures with hubs to function properly. In our research we are interested in complex, large, structured and functional biological networks which are scale-free. In chapter 5, we will go over our algorithm and discuss the steps of testing the scale-free property of each generated network.

2.2.2 Scale-Invariance and Power Law

This section follows the derivations adapted from Newman [112] on scale-invariance and power law. Here, we examine the relationship between being scale-free and having a power-law curve for probability distribution. As stated earlier, networks with scale-free characteristics follow a power-law distributions. By assuming the probability $p(x)$ being scale free, then the probability should satisfy,

$$p(bx) = g(b)p(x) \tag{2.11}$$

for any b and any scaling function g. For $x = 1$, the above equation becomes,

$$g(b) = \frac{p(b)}{p(1)} \tag{2.12}$$

Substituting this into the first equation and taking the first derivative with respect to b since the equation is true for any b, gives us

$$xp(bx) = \frac{p'(b)p(x)}{p(1)} \tag{2.13}$$

For $b = 1$, and putting x, and $p(x)$ on two sides of the equation, this becomes,

$$\frac{dp}{p} = \frac{p'(1)}{p(1)} \frac{dx}{x} \tag{2.14}$$

Integrating the above derivative equation gives the solution as,

$$\ln p(x) = \frac{p(1)}{p'(1)} \ln x + constant \tag{2.15}$$

with the constant $ln\ p(1)$ found by setting $x = 1$. So, by taking the exponential, the scale-free probability distribution above becomes

$$p(x) = p(1)x^{-\alpha} \tag{2.16}$$

where $\alpha = -p(1)/p'(1)$. This derivation shows that the power-law distribution is the only distribution which satisfies the scale-free property. Thus, any network whose node degree distribution shows a power-law curve shows scale-free characteristics.

2.2.3 Complex Network Statistics and Measuring Power Laws

Identifying complex, scale-free networks requires determining whether the node degree distribution of the network fits to a power law. The easiest way to measure a power-law is by plotting the log-transformed degree data against the log-transformed probability distribution using random samples from the data. A network with a power-law distribution would produce a straight line with a negative slope of α. However, this procedure is not very accurate, since the higher the node degree of the data gets, the lower the number of samples. A more reliable way of finding the network statistics is fitting the data to a model by using its cumulative distribution function (CDF).

Maximum Likelihood Estimate of the Power-law Fit

This section follows the derivations adapted from Newman [113] on maximum likelihood estimate of the power-law fit.

The power law probability density function (PDF) that explains a straight line with a negative slope can be written as

$$p(x) = Cx^{-\alpha} \tag{2.17}$$

where C is the normalization constant, and α is the exponent of power law, called the scaling parameter. To fit a power-law model to the data, the maximum likelihood estimate (MLE) can be used. The above probability distribution with a normalization constant becomes

$$p(x) = \frac{\alpha - 1}{x_{min}}(\frac{x}{x_{min}})^{-\alpha} \tag{2.18}$$

for $x \geq x_{min}$. Accordingly, the log-likelihood of the above probability that the data would fit to the model becomes

$$Ł = \ln \prod_{i=1}^{n} \frac{\alpha - 1}{x_{min}}(\frac{x_i}{x_{min}})^{-\alpha} \tag{2.19}$$

where x_i is the i^{th} data point of data with size n, given $x_i \geq x_{min}$.This likelihood function is maximized at the MLE value of α, where the data have the highest probability to be taken from the chosen model. Solving the differential equation with respect to α gives the estimate as

$$\hat{\alpha} = 1 + n(\sum_{i=1}^{n} \ln \frac{x_i}{x_{min}})^{-1} \tag{2.20}$$

with a standard deviation of

$$\sigma = \frac{\alpha^n}{\sqrt{n}} + O(n^{-1}) \tag{2.21}$$

This error on $\hat{\alpha}$ shows that the above estimate is only stable for large n. For reliable parameter estimates of $\hat{\alpha}$ and x_{min}, there should be enough nodes with degree equal to or higher than x_{min}. Previous work shows that for a value of $n \geq 50$ the above fit and estimates are close enough to be statistically significant [15, 162].

The above estimate assumes that x_{min} of the data is known, at which value the data points start to behave like a power-law distribution. However, it usually is the case that x_{min} also needs to be estimated. Using a lower than expected x_{min} value would introduce bias in the estimate, and the model fit would drift from the real data. Using an x_{min} value higher than expected, however, would exclude data points.

Kolmogorov Smirnov Test

The one-sample Kolmogorov Smirnov test (KS-test) calculates the distance between the CDFs of the data and the specified model. By using the above defined MLE value $\hat{\alpha}$ for the power-law model parameter, the KS-test distance D between the model and the empirical data would be at a minimum when

$$\hat{\alpha} = argmin_{\alpha} D_{\alpha} \tag{2.22}$$

where

$$D_{\alpha} = max_{x \geq x_{min}} |F(x) - y(x)| \tag{2.23}$$

For an experiment node degree data set of $y(x)$, and a power-law model fit of $F(x)$, the test statistic, D, is the maximum distance between the two. The p-value of the test quantifies the probability of D at least as large as the one calculated if the two data points were coming from the same distribution. The test is assumed to be accepted where the $p\text{-}value > 0.05$.

Our approach, discussed in chapter 5, searches for critical nodes. One of the criteria we use to decide whether the chosen nodes have the desired perturbation effect on the final outcome is to look whether the resulting network is still scale-free or not. We use KS-test and the p-value after every iteration to decide whether each generated network can still assumed to be scale-free.

2.3 Statistical Approaches for Network Inference

Network inference is the idea of fitting a model to quantitatively predict the interactions of a set of variables. For a variable set of population size p, with experimental observations of size n, a network inference model will estimate a graph representation of nodes and edges. For a weighted and undirected graph, the model will predict how many variables are connected to one another and with what strength.

2.3.1 Modeling Small Networks

Covariance and Correlation Estimates

For a set of data where $p \sim n$, the first step to understand whether two variables of the same population have a connection to each other would be to investigate their joint behavior in observable samples. For a variable X, with discreet measurements $(x_1, ..., x_n)$ the measurement mean of X would be

$$\mu_x = E[X] = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{2.24}$$

with a variance value of

$$\sigma_x^2 = E[(x - \mu_x)^2] = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu_x)^2 \tag{2.25}$$

where $E[X]$ is the expected value of X. Thus, the covariance estimate, which is also called the joint variability of two variables X and Y, is

$$cov(x, y) = E[(X - \mu_x)(Y - \mu_y)] = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y) \tag{2.26}$$

The sign of this equation determines the linear relationship of the two variables. By using this covariance estimate and normalizing it with the standard deviation we can reach the Pearson correlation moment

$$r_{xy} = \frac{cov(xy)}{\sqrt{\sigma_x^2 \sigma_y^2}} \tag{2.27}$$

The Pearson correlation moment estimates both the strength and the direction of the linear correlation of two dependent variables, X and Y. For a positive large Pearson correlation value, the two variables will have a strong interaction and will be connected with an edge in the network. Their increasing and decreasing trends over samples will also be positively related. A value of zero means no linear correlation, but it is not enough to conclude no relationship exists between two variables. In biology, gene co-expression networks (GCNs) specify a specific group of mathematical representations where nodes are genes and edges connect genes with significant co-epression relationships. The level of co-expression is usually determined by Pearson correlation moment [148].

Graphical Gaussian Model and Partial Correlation

Network edges found by using Pearson correlation moments will include both direct and indirect interactions of the whole network. To focus on only the direct interaction between two variables, partial correlation can be calculated. For variables X and Y, where a third variable Z is kept constant the partial correlation would be [31]

$$r_{xy,z} = \frac{r_{xy} - r_{xz}r_{yz}}{\sqrt{(1 - r_{xy}^2)(1 - r_{yz}^2)}} \tag{2.28}$$

To find the partial correlation weights for the edges in a network, the above equation should be generalized for a set of variables that includes all controlled variables of the network. The Gaussian graphical model (GGM) assumes that the data points are sampled from a normal distribution [167]. For this model, there are two ways to estimate partial correlation coefficients, discussed next.

Linear Regression and Least Square Estimates:

This section follows the derivations of Hastie et al [58] on linear regression and least square estimates. A linear regression model assumes a dependent variable y_i can be expanded as a linear combination of all other measurements of x_i of the same observable. As the GGM assumes normal distribution of p variables of each of n observables, this means a chosen variable y_i with n measurements can be defined as n linear regression equations.

For a measurement of vector $y = (y_1, \ldots, y_n)^T$, and all other measurements of $x_i = (x_{i1}, \ldots, x_{ip})^T$ from the same population, a linear regression estimation of any element of the y-vector would have the form

$$\hat{y}_i = \beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip} + \varepsilon_i \qquad i = 1, ..., n \tag{2.29}$$

Here, the β_j are the regression coefficients, and will be estimated by fitting the data into the model. The last term is an error term. The first coefficient can be set to zero for centered data where there is no intercept. If we define a data matrix of $X^T = (X_1, \ldots, X_p)$ where each X_j is an n-dimensional measurement vector, the above equation simplifies to

$$f(x) = \sum_{j=1}^{p} x_j \beta_j + \varepsilon_i \tag{2.30}$$

For a linear regression model, the residual is defined as the difference between the predicted estimate and the observed data. The smaller the residuals, the better the model fit to the real data. The coefficients may be estimated by using least squares regression in terms of the residuals. We can write the residual sum of squares as

$$RSS(\beta) = \sum_{i=1}^{n} (y_i - f(x_i))^2 \tag{2.31}$$

and in matrix form

$$RSS(\beta) = (y - x_\beta)^T (y - x_\beta) \tag{2.32}$$

where, $f(X) = XB$ is used. For the small size case where $n \geq p$ and the derivatives are well defined, by setting the first derivative to zero to find the minimum residual difference, we can find the β estimate as,

$$\frac{d(RSS)}{d\beta} = -2X^T(y - X_\beta) = 0 \tag{2.33}$$

$$\hat{\beta} = (X^T X)^{-1} X^T y \tag{2.34}$$

The partial correlation between two variables can be found by using the above estimate of β. Li et al [87] defines partial correlation as follows: For two variables X and Y when all other variables of set Z are controlled, partial correlation can be calculated as the correlation of the linear regression residuals of X on Z and of Y on Z. Thus, by using the above formulation and this definition we can estimate partial correlation by finding the β estimate for both X, and Y, and calculating the correlation between these values. By using the formulation of Kramer et al [82], this results in the partial correlation definition of

$$\hat{r}_{xy} = sign(\hat{\beta}_x)\sqrt{\hat{\beta}_x \hat{\beta}_y} \tag{2.35}$$

Matrix Representation and Inverse Covariance Matrix:

The below formulation is adapted from Kramer et al [82] on matrix representation and inverse covariance matrix. It is possible to estimate partial correlations by using the covariance matrix of the data. If we use the above defined data matrix X^T of (p x n) dimensions, the covariance matrix estimate of (p x p) dimensions would be

$$\widehat{\Sigma} = \frac{1}{n-1} X^T X \tag{2.36}$$

The inverse of this matrix is the precision matrix with elements $\hat{\omega}_{xy}$.

$$\hat{\Omega} = \{\hat{\omega}_{xy}\} = \widehat{\Sigma}^{-1} \tag{2.37}$$

The partial correlation of two variables x and y is

$$\hat{r}_{xy} = -\frac{\hat{\omega}_{xy}}{\sqrt{\hat{\omega}_{xx}\hat{\omega}_{yy}}} \tag{2.38}$$

The two formulations of partial correlation coefficient are equivalent, and the regression coefficient estimates can be represented in terms of the precision matrix elements.

2.3.2 Modeling Large Networks

The above estimate for the regression coefficients and covariance formulation are well defined for a set of variables with a size comparable to the set of observations. If the variables are much larger than the observables, which is the case for most high-throughput biological data, both the above definitions of partial correlation do not explain the data correctly because of the overfitting of the linear regression model and the singularity of the covariance matrix [82]. So, for the case $p \gg n$, the partial correlation estimates need to be regularized.

This section covers different approaches to overcome the overfitting and singularities introduced with the large datasets. We will review the algorithms that use regularization with the regression estimates mentioned above, such as lasso and ridge as well as regularized version of the above mentioned precision matrix. We will also discuss information theory based models and how to choose one optimum solution among all. Our algorithm, explained in chapter 5, uses lasso regularized regression model with cross validation to reach network predictions at each step for a set of chosen nodes.

Regularized Regression Estimator Models

To find an estimator for the regression coefficients vector $\beta = (\beta_1, \ldots, \beta_p)$ the residual sum of squares needs to be minimized as mentioned above [59]. For the column centered data, without the intercept the least squares estimate is written as

$$minimize \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} x_{ij}\beta_{ij})^2 \tag{2.39}$$

There might be more than one optimum solution to the minimization problem. The least absolute shrinkage and selection operator (Lasso) algorithm uses an l1-regularization parameter to find the optimum solution.

$$minimize \frac{1}{2N} \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} x_{ij}\beta_{ij})^2 \text{ subject to } \|\beta\|_1 \leq t \tag{2.40}$$

with $\|\beta\|_1 = \sum_{j=1}^{p} |\beta_j| \leq t$ as a data-specific upper boundary for the model fit. Another algorithm, ridge regression, follows the same optimization procedure, but uses l2-regularization instead as follows

$$minimize \frac{1}{2N} \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} x_{ij}\beta_{ij})^2 \text{ subject to } \sum_{j=1}^{p} \beta_j^2 \leq t \tag{2.41}$$

The lasso and ridge regression models both give stability under the condition of $p \gg n$. Lasso estimates result in models where the coefficients shrink and some of the coefficients become 0. The above regularization limits are equivalent to an addition of a penalized term to the equations. So, the β coefficients can be estimated for lasso and ridge as

$$\beta_{lasso} = argmin_{\beta_j} \{ \frac{1}{2N} \sum_{i=1}^{N} (y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |(\beta_j)| \} \tag{2.42}$$

$$\beta_{ridge} = argmin_{\beta_j} \{ \frac{1}{2N} \sum_{i=1}^{N} (y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} (\beta_j)^2 \} \tag{2.43}$$

The regularization parameter λ that satisfies the previously set condition is $\lambda > 0$, and it is usually between 0 and 1. The optimal solution is unique under the condition of the variable vector being full-rank. Fan and Li [46] showed that the lasso shrinkage is biased when the coefficients are large. Hui Zou [176] introduced a two-stage adaptive lasso procedure to overcome this bias which includes weights to the penalty term. The above equation becomes

$$\beta_{adaptive-lasso} = argmin_{\beta_j}\{\frac{1}{2N}\sum_{i=1}^{N}(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda\sum_{j=1}^{p}\hat{\omega_i}\,|\beta_j|\} \tag{2.44}$$

The first stage uses the usual lasso algorithm to estimate a $\hat{\beta}$. Then, using this estimate, the weights are calculated as

$$\hat{\omega}_i = \frac{1}{\left|\hat{\beta}_{i,lasso}\right|} \tag{2.45}$$

The weight estimates become small when the coefficients become large. Plugging these weight estimates back into equation 2.45 above gives the adaptive lasso coefficient estimate. Equation 2.45 gives either the same or in most cases sparcer models when compared with lasso model.

For all the above regularized regression models, the covariance selection and network construction aim to find the zeros in the estimates. The network, then, will be all the nodes left with non-zero entries.

The partial correlations resulting from the above regression method estimates can be summarized as

$$\hat{r}_{ij} = sign({\beta_j}^{(i)})min\left\{1, \sqrt{{\beta_j}^{(i)}{\beta_i}^{(j)}}\right\} \tag{2.46}$$

if $sign{\beta_j}^{(i)} = sign{\beta_i}^{(j)}$ and 0 otherwise. This eliminates the singularity and makes the partial correlation well-defined between the boundaries -1 and 1.

Regularized Matrix Estimator Models

Another possible way to regularize and estimate partial correlations for the $p \gg n$ case would be using the precision matrix. Hastie et al. [58] defines the Graphical lasso (Glasso) model that optimizes the precision matrix. For the same case where the data can be estimated as a multivariate Gaussian distribution with sample covariance matrix Σ, and the inverse of it as precision matrix Ω, the Glasso method optimizes the following equation

$$\hat{\Omega}_{Glasso} = argmin_{\Omega}\{< \Omega, \Sigma > -logdet(\Omega) + \lambda\,\|\Omega\|\} \tag{2.47}$$

Here, $< \Omega, \Sigma >$ is the trace of the matrix multiplication of Ω, and Σ. And, $||\Omega|| = \Sigma_{ij}\omega_{ij}$. The above equation is the same as the optimization problem

$$min \, \|\Omega\| \text{ subject to } |\Omega^{-1} - \Sigma| \leq \lambda \tag{2.48}$$

For this precision matrix estimator, the partial correlation can still be represented by using the elements of the estimated precision matrix as in Equation 2.38

Information Theory-Based Estimator Models

Information theory-based estimator methods use either mutual information or mutual information ratio to calculate the dependencies between nodes in a network. For variables X and Y, the mutual information I is defined as the change in entropy H [103]

$$I(X;Y) = H(X) + H(Y) - H(X,Y) = H(X) - H(X|Y) \tag{2.49}$$

$$H(X) = -\sum_{x \epsilon X} p(x) log p(x) \tag{2.50}$$

where p is a probability measure. The mutual information ratio is the ratio of $I(X;Y)/H(X)$. A network can be inferred using these information matrices. These methods do not rely on monotonic relationships such as seen in Pearson correlation, so they can detect non-linear interactions. There are also methods that apply both partial correlation and mutual information models together [134].

Finding the Unique Solution After Regularization

The above mentioned regularization methods all depend on the choice of the penalty term λ, which is usually between 0 and 1. There will be different solutions with different regularization parameters. A high number of regularization may correspond to a network with no edges at all which is not a reasonable result for further investigation of the data. On the contrary, a low number of regularization may result in a network that has not even been regularized and fully-connected which also does not depict a real-life problem. So, choosing the right λ is the key to deducing the most reliable network solution.

The method of k cross-validation is one way to choose a penalty term. It separates the data into k subunits and uses $k-1$ parts for the optimization and the last part for the cross-validation procedures. Other methods use the Bayesian information criterion (BIC), or the Akaike information criterion (AIC), which both look at the log-likelihood of the data and the penalty. There is also extended BIC (EBIC) [43]. The EBIC method uses a user-specified tuning parameter, y, to decide the optimum λ parameter. The y parameter is set between 0 and 0.5. The equations for these parameters are as follows

$$AIC = -2L + 2E \tag{2.51}$$

$$BIC = -2L + Elog(n) \tag{2.52}$$

$$EBIC = -2L + Elog(n) + 4yElog(P) \tag{2.53}$$

Here, L is the log likelihood of the data; E is the number of estimated parameters, which is the number of nonzero edges in the case of network models; n is the sample size; and P is the number of nodes. All the above equations choose the network that minimizes the criterion value.

Chapter 3

Regulatory Networks for Gene Expression

In this dissertation, we use gene expression data. In sections 3.1, 3.2, and 3.3 we will give a short overview of molecular biology concepts related to gene expression and gene regulatory networks, which control the expression of genes based on tissue type and external signals. We will then present an introductory explanation about how gene expression is measured via sequencing (RNA-seq). In section 3.4 we will go over different model building approaches for gene regulatory network estimation. Finally, in section 3.5 we will discuss commonly used biological tools and algorithms for pre-processing of expression and network estimation that we used in our research.

The techniques covered in this chapter are well established in the literature for processing and normalizing RNA-seq data and building gene networks based on expression data. These methods will form the basis for our contribution, an iterative graph perturbation algorithm that will identify nodes (genes) of high importance by starting from gene networks estimated by partial correlation, which is described in Section 3.4.2.

3.1 Gene Expression

Using 4 nucleotides (adenine, thymine, cytosine and guanine) the DNA molecule carries the genetic code needed for cellular function. In the mammalian genome there are around 34,000 genes. These gene sequences are composed of exonic and intronic sequences. Exons ultimately codify for mature messenger RNAs (mRNAs) that in turn are translated into amino acid sequences of complex proteins. Introns, on the other hand, serve as non codifying sequences or spacers that are taken out during mRNA maturation in a process called splicing. This gives a cell the possibility of rearraging the pattern of intron and exon skipping to alter the mRNA coding sequence in a process called alternative splicing. This process allows for the production of a variety of different

proteins from a single gene.

Gene expression is the process whereby the genetic code is used to synthesize an RNA molecule or RNA transcript. When a gene needs to be expressed a series of transcription factors (TF) and adaptor molecules bring the RNA polymerase enzyme to the gene's transcriptional start site (TSS). The RNA polymerase opens up the two strands of DNA, and starting from the 3' end, it binds nucleic acids together complementary to the sequence of the template DNA strand, generating an exact copy of the coding strand. The transcription terminator, located at the 3' end of the coding region of the DNA strand, stops RNA polymerase when the transcription is completed. This section follows the explanations of the books [4, 139] about gene expression, gene regulation and gene regulatory networks unless otherwise noted.

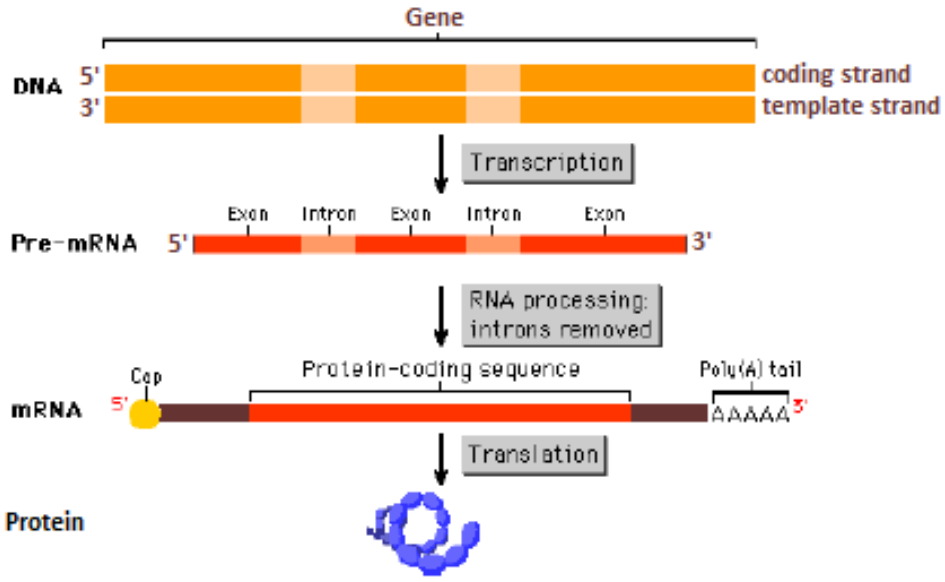

Figure 3.1: Gene expression steps. Modified from Alberts et al. [4]

Figure 3.1 shows the mechanism of gene expression. The transcription and mRNA processing levels take place in the cell nucleus. The processing of mRNA, in which the pre-mRNA is turned into a stable mRNA that travels and binds to a ribosome site for protein synthesis, has three steps. The first step is 5' capping, in which a methylated guanine nucleotide without the phosphate group is added to the 5' side to protect the newly synthesized molecule from degrading before the process finishes. The next step is polyadenylation, in which a small chain of adenines around 200 bases long, known as a poly(A) tail, is added to the other end of the strand. This step, as with the first step, protects the mRNA. The cap also enables the mRNA to travel through the nucleus wall and

initiate the next part of the process.

The last step of mRNA processing is splicing. In RNA splicing, all introns are removed from the molecule and exons are selectively connected to one another, making the mature mRNA. The spliceosome, a large enzymatic complex made of catalytic proteins and non-coding RNAs catalyzes this event. Alternative splicing occurs at this time enabling for one gene to encode more than one final product. The different mRNAs that are the products of alternative splicing are gene isoforms since they are coded by the same gene.

The whole body of RNA transcripts of an organism is called its transcriptome. After mRNA processing, the mRNA molecule is exported from the nucleus to the cytoplasm, to be translated into an amino acid (AA) chain. Every three-nucleotide chain of mRNA, called a codon, translates to an AA. The start codon is the first codon of an mRNA transcript to be translated by the Ribosome. In mammals the start codon is always ATG and codifies for the AA methionine. Starting on ATG the Ribosome links AA together by reading the mRNA sequence and linking AA polymers with help of transfer RNAs (tRNA)s that carry the corresponding AAs. This AA chain then folds into its natural three-dimensional structure with help of chaperones, making it a stable and properly functioning protein.

3.1.1 Gene Expression Regulation

Genes are not expressed at a constant rate. When a cell is in need of a protein, a gene is activated and starts to be expressed at a higher rate, increasing the concentration of the specific mRNA and thus, of the protein. When the gene is repressed however, the transcription rate of the mRNA falls below the basal level. The regulation of gene expression is accomplished at specific noncoding regions of DNA and with the assistance of specific RNAs and proteins that form the transcriptional complex. Genotype is the organism's complete hereditary information and is encoded in it's genome. Phenotype is the organism's actual properties like morphology, behavior and metabolic rate that respond to environmental changes with observable variations in mRNA and protein content. Therefore, understanding gene expression regulation and making quantitative assumptions about it helps better explain genotype-phenotype relations.

By using the DNA coding strand as reference, we define the first triplet as the transcriptional start site (TSS). The elements 5' to the TSS are called upstream, whereas the 3' elements are said to be downstream. The first of the upstream regulatory elements is the promoter. The RNA polymerase binding rate to the promoter is the first step of gene regulation. With the help of transcription factors and associated proteins the RNA polymerase binds to the promoter region and actively starts transcription when needed by the cell.

Enhancers and silencers also regulate the activity rate of the RNA polymerase. These short sequenced elements may be located upstream of the promoter or downstream of the gene. TFs bind to enhancers or silencers and are involved in the transcription phase of gene expression. When a gene needs to be activated, a transcriptional activator will bind to the enhancer site and increase the recruitment of RNA polymerase to the gene's TSS in order to enchance mRNA production. When gene expression needs to be repressed, transcriptional repressors are recruited to gene silencers that ultimately block the RNA polymerase's access to the gene's TSS, diminishing mRNA production. Activators and repressors also regulate the alternative splicing process of gene expression.

The three-dimensional packaging structure of DNA can affect the binding regions that regulate gene expression. Other than activators and repressors, various other transcription factors influence RNA polymerase activity rate while RNA polymerase is active. The expression rate continues to be controlled after the transcription phase is finished. The cap regulates the rate at which mRNAs diffuse from the nucleus and also the rate at which mRNAs bind to ribosomes.

Translation control is another regulatory mechanism. At the product level, even after protein synthesis, both the mRNA and protein concentration in cell plasma are controlled by the enzyme proteases and by siRNA, both of which take part in the process of degrading protein molecules if the production is too high.

3.1.2 Gene Regulatory Networks

For a cell to carry out a function, it needs to synthesize many proteins, and many genes are activated at the same time. Thus, transcription factors usually control the expression of a set of genes. The genes that, in turn, synthesize these regulatory transcription factors are called regulatory genes. The genes that activate other genes are called activator genes, and those that repress other genes are called repressor genes.

According to above, there is a hierarchical organization when it comes to the cell's protein synthesis process. The level of hierarchy among the gene subsets depends on the cell type or specific function. There are cell functions that are maintained by only one gene or one gene subset, whereas there are other functions controlled by a cascade of regulatory gene subsets. It is our goal to identify the groups of genes that fuction as a core in a complex gene regulatory network. Using an unbiased informatic method and without any "a priori" information of gene function, we intend to identify the most important genes in a network of thousands or even tens of thousands of genes.

3.2 Measurement of Expression

The mRNA content of the cell gives quantitative information about the activation and repression of genes, since the production rate of mRNAs changes according to the cell's needs. The increase or decrease of gene expression is a response of the cells to environmental and hormonal cues. Systematic recording of these gene expression changes gives clues about the gene's function in cellular homeostasis. In this section we will cover one way to determine gene expression changes by genome wide high troughput mRNA sequencing. This section follows the explanations of the books [78, 163] about the measurement of expression unless otherwise noted.

With RNA sequencing, it is possible to reach quantitative results, even if the gene sequence is not known, although the most commonly used methods are still performed on known genome or transcriptome sequences. This process gives information about gene splice regions and gene families with common functions. Illumina, Inc. http://www.illumina.com is one of the widely used platforms for RNA-seq technology.

3.2.1 RNA Processing

mRNA-seq is a type of RNA-seq where mRNAs are used for data gathering. RNA-seq experiments start with either a living or frozen cell or tissue sample. The transcriptome from the sample is dissected, for that the mRNAs are purified and ribosomal and other RNAs are eliminated from the sample.

For the mRNA-specific purification, the polyA tail of each mRNA is used. After organic extraction, the total RNA from a sample is mixed with polyT oligomers bound to a substrate such as magnetic bead. polyT beads bind to the polyA tails of mRNAs, the sample is then washed and depleted of all other RNA types followed by elution of enriched polyA containing mRNAs.

The RNA-seq process can be performed in any of three different ways, as shown in Figure 3.2. Only the first process is explained below, although the steps described here are interchangeable in the three processes shown in the figure. For the efficiency of the sequencing steps, large-sized mRNAs are fragmented into smaller pieces. These small mRNA fragments are reverse transcribed into copy-DNA (cDNA) using a set of random oligonucleotides bound to specific adapter molecules and a reverse transcriptase enzyme capable of converting mRNA molecules into cDNA. At the end of this process all the mRNA molecules were degraded, and double cDNAs were made in a 1:1 ratio. Linear amplification of cDNAs is performend by polymerase chain reaction (PCR) amplification. PCR amplification takes a small DNA segment and generates thousands of copies of the same molecule, making the measurements of sample concentrations easier to achieve. The Illumina

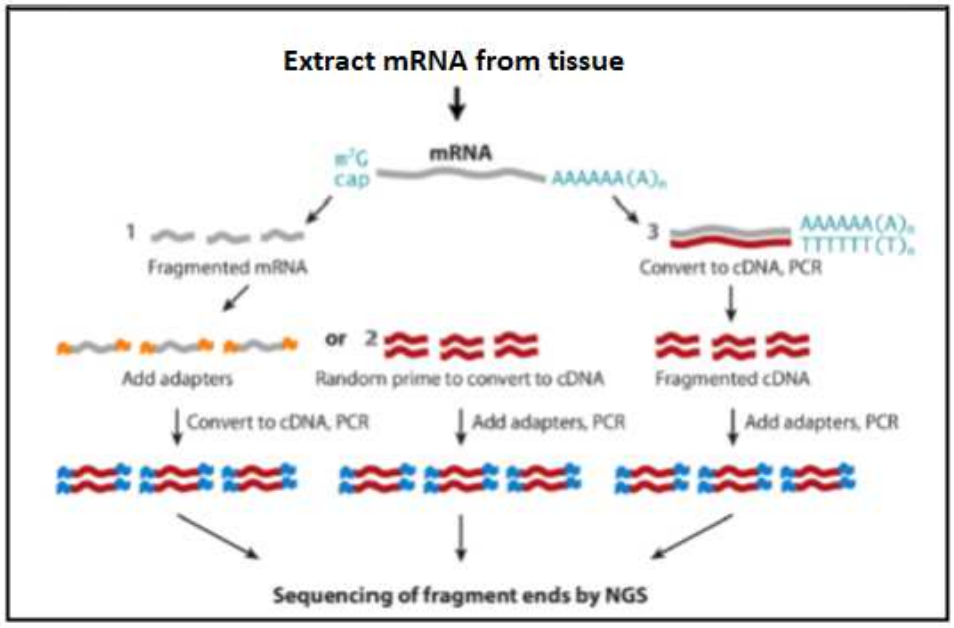

Figure 3.2: mRNA-seq workflow. Adapted from Simon et al. [144]

sequencing method is based on reversible dye-terminators that enable the identification of single bases as they are introduced into the DNA strands, a process called sequence by synthesis. This platform enables to sequence millions of strands of DNA at once in a high-throughput way.

3.2.2 Read Alignment and Normalization

To know which genes are highly expressed, the sequencing reads should be mapped to genes. This can be done by aligning the sequencing information with either the known genome or the transcriptome of the organism. First a quality assessment and culling is performed. Reads with low quality are eliminated, including ribosomal RNAs that remain in the sample and DNA strand sequences that were not fully sequenced. The whole sample is also checked for uniform nucleotide base content and sufficient number of good quality reads.

There are various approaches for read alignment. A database may be prepared before the mapping by using the sample exome from the reference genome and adding it to the already known exon sites from databases. The reads are then mapped by making a search through the database. Another method uses the reference genome first to find the variants without alternative splicing, and then, by using these newly found regions as a database, those sequences that are not mapped are searched to find the spliced variants.

After the reads are aligned with the reference, the number of alignments is counted to generate

the raw data. At this point, the experiment exhibits significant bias. It has been shown that the raw reads data exhibits transcript length bias, the longer the transcript, the greater the number of copies that would be read. Nucleotide concentration also produces bias. Finally, depending on the RNA processing step, there could also be 3' end bias depending on the primers used for reverse transcription. For all these reasons, normalization steps are taken before further use of the data. The sequencing depth should also be taken into account. Sequencing depth—the total number of mapped reads—shows how well a sample is sequenced. Therefore all datasets are normalized by the total number of reads obtained.

There are different approaches to normalize the read counts for each gene. Reads per kilobase per million reads mapped (RPKM) and fragments (for paired-end reads) per kilobase per million reads mapped (FPKM) are two of the most common normalization procedures. There are some derivations of these that differ in scaling factor or in reads per number of bases counted. For normalization using a scaling factor of per million reads mapped, the equation of the normalized expression level of a gene is:

Normalized expression level = (Raw counts x 10^9) / (sequencing depth x length of the gene)

Another approach, trimmed mean of M values (TMM), assumes that most genes are not expressed in the sample. M values here are log expression ratios. This approach takes into account that we do not know the length of each gene without an error in the above equation. Instead of trying to estimate the total RNA production, this method looks at the fold changes between samples and, by using gene-wise log-fold changes, it excludes highly expressed genes from the normalization value calculation [78].

3.3 Gene Expression Profiling Statistics

High throughput gene expression data usually includes whole genome expression values given the samples. However, the analysis step usually concentrates on a specific group of genes where the data shows drastic variability over the samples. Gene expression profiling focuses on finding these genes for further investigation.

For an initial exploration, the log-fold change can be used. Log-fold change looks at the difference between the logarithms of the expression values for different samples. If the difference is above a certain threshold, such as a 2-fold difference, then the gene would be considered differentially expressed. This method is not very reliable, and it is difficult to compare genes to one another, since the fold-change might mean different changes because of the drastic differences in gene expressions [70]. A more reliable approach would use means and variances.

3.3.1 Gaussian Model

This section follows the derivations adapted from Draghici [34] of the Gaussian Model.

For the most simplistic case, where the population mean and population variance of genes are known and a set of measurements is done for a gene, a research question could be whether this gene is upregulated or downregulated compared to the population mean. For a population size of N, the population mean and variance are given by

$$\mu = \frac{1}{N}\sum_{i=1}^{N} X_i \tag{3.1}$$

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N} (X_i - \mu)^2 \tag{3.2}$$

Assuming the population is large enough, every measurement becomes a sample from a normal distribution. The probability density function (PDF) for this case fits the Gaussian distribution as given below:

$$p(X_i) = \frac{1}{\sigma\sqrt{2\pi}} exp[(-\frac{1}{2})(\frac{X_i - \mu}{\sigma})^2] \tag{3.3}$$

If we define a variable Z which depends on X_i as

$$Z_i = \frac{X_i - \mu}{\sigma} \tag{3.4}$$

The above PDF becomes

$$p(Z_i) = \frac{1}{\sqrt{2\pi}} exp[-\frac{{Z_i}^2}{2}] \tag{3.5}$$

Equation 3.5 is the standard Gaussian distribution with mean value, ν, of 0 and variance σ^2, of 1.

To answer the question of whether a gene is, for example, upregulated compared to the general population mean, the CDF should be calculated. The null hypothesis states that the gene expression level should be at the mean value. The CDF of the gene expression value being higher than the mean can be assumed to represent a total probability, and can be calculated as the integral of the PDF, the area under the curve for specific boundary conditions, if the distribution is continuous, and as the sum if it is discreet.

This CDF value shows the probability of a gene expression value obtained by chance. If this probability is lower than a previously chosen significance level, then the null hypothesis would be rejected, and the gene would be assumed to be upregulated. For defining the differential gene expression (DGE), the p-value significance level is usually chosen as 0.05.

Equations 3.2 and 3.4 assume that all measurements are done for the same experiment condition—for example, a sample of treatment, case, or control. Usually, the same sample includes more than one replicate. For a set of n replicates of the same sample, the mean of the measurements can be used instead of measurements X_i.

$$\bar{X}_i = \frac{1}{n}\sum_{j=1}^{n} x_{ij} \tag{3.6}$$

The variance is also corrected for the n replicates case as

$$\sigma^2_{\bar{X}_i} = \frac{\sigma^2}{n} \tag{3.7}$$

The corrected Z score for n replicates is

$$Z_{\bar{X}_i} = \frac{\bar{X}_i - \mu}{\frac{\sigma}{\sqrt{n}}} \tag{3.8}$$

3.3.2 t-distribution

There are cases where the population mean and variance are unknown, but instead there is a representative gene set with known sample mean and variance. In this case, the Z score becomes a t-test score, and the distribution becomes a t-distribution.

$$t_i = \frac{X_i - m}{\frac{s}{\sqrt{n}}} \tag{3.9}$$

where the sample mean and variance are given by

$$m = \sum_{i=1}^{p} X_i \tag{3.10}$$

$$s^2 = \frac{1}{p-1}\sum_{i=1}^{p} (X_i - m)^2 \tag{3.11}$$

for a sample of p variables.

The PDF, CDF and p-value would still be calculated as above, but this time by using a t-distribution instead of a normal distribution.

Gene expression data can be collected to understand DGE values for two experimental setups, for example, cases and controls. For two experimental samples with means m_1 and m_2, and variances s_1 and s_2, the t-test score for a gene X_i is given by

$$t_i = \frac{(X_{i1} - m_1) - (X_{i2} - m_2)}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}} \tag{3.12}$$

It is shown that the ratio of the sample variances in this two sample case follows an F-distribution, and this can be used to estimate the contrasts between samples. The F-statistics value is given by

$$F = \frac{S_1^2}{S_2^2} \tag{3.13}$$

with degrees of freedom specified as

$$\nu = \frac{(\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2})^2}{\frac{(\frac{S_1^2}{n_1})^2}{n_1 - 1} + \frac{(\frac{S_2^2}{n_2})^2}{n_2 - 1}} \tag{3.14}$$

3.3.3 Bayesian Estimates

This section follows the derivations of Baldi and Long [7] of Bayes estimates.

Even though the t-test equation derived above works for small sample sizes, where the number of genes is similar to the number of sample replicates, most high-throughput data include very large number of genes per sample. This case poses a threat to the approximation, since the sample mean and variance could deflect from the true mean and variance when there are not enough samples. So, these values need to be corrected. One way is to use Bayesian estimates for these values in the formula. Bayes theorem states that the posterior probability of an unknown model, M, given data D, is proportional to the combination of the probability of the model and the probability of data given the model.

$$P(M|D) = \frac{P(D|M)P(M)}{P(D)} \tag{3.15}$$

Here, $P(M|D)$ is the posterior probability of M. $P(D)$ is the probability of the data which is a proportionality constant. $P(D|M)$ is the data likelihood function given the model. $P(M)$ is the prior probability distribution function. For the gene expression data, assuming a large number of genetic variables and that all the genes and samples are independent from one another, a Gaussian model can be chosen. The model M would have two identifier parameters where $M = (\lambda, \sigma)$ for each sample. So, by using Equation 3.15, the data likelihood can be expressed as

$$P(D|\mu, \sigma^2) \approx \prod_{i=1}^{n} N(X_i; \mu, \sigma^2) = C(\frac{1}{\sigma^2})^{\frac{n}{2}} exp - [\sum_{i=1}^{n} \frac{(X_i - \mu)^2}{2\sigma^2}] \tag{3.16}$$

for a normalization constant, C. It can be shown that the prior probability has the same functional form as the posterior and can be defined as a normal distribution for a set of hyperparameters, assuming the model parameters (λ, σ^2) are not independent. If we expand the model probability, again by using Bayes theorem, as

$$P(M) = P(\mu, \sigma^2) = P(\mu|\sigma^2)P(\sigma^2) \tag{3.17}$$

with a set of hyperparameters

$$\alpha = (\mu_0, \lambda_0, \nu_0, \sigma_0^2) \tag{3.18}$$

the prior conditional distribution becomes

$$P(\mu|\sigma^2) = N(\mu; \mu_0, \frac{\sigma^2}{\lambda_0}) \tag{3.19}$$

The probability $P(\sigma^2)$ is an inverse gamma function defined as,

$$P(\sigma^2) = I(\sigma^2; \nu_0, \sigma_0^2) = \frac{(\nu_0/2)^{(\nu_0/2)}}{\Gamma(\nu_0/2)}(\sigma_0)^{\nu_0}(\sigma^2)^{-(\frac{\nu_0}{2}+1)}exp - [\frac{\nu_0\sigma_0^2}{2\sigma^2}] \tag{3.20}$$

Here, the hyperparameter set α can either be found from the prior probability distribution or be estimated from data.

Combining all the findings above, the posterior probability of the model, given the data becomes

$$P(\mu, \sigma^2|D, \alpha) = N(\mu; \mu_n, \frac{\sigma^2}{\lambda_n})I(\sigma^2; \upsilon_n, \sigma_n^2) \tag{3.21}$$

$$\mu_n = \frac{\lambda_0}{\lambda_0 + n}\mu_0 + \frac{n}{\lambda_0 + n}m \tag{3.22}$$

$$\lambda_n = \lambda_0 + n \tag{3.23}$$

$$\nu_n = \nu_0 + n \tag{3.24}$$

$$\nu_n\sigma_n^2 = \nu_0\sigma_0^2 + (n-1)s^2 + \frac{\lambda_0 n}{\lambda_0 + n}(m - \mu_0)^2 \tag{3.25}$$

Here, ν specifies the degrees of freedom, μ means, σ^2 variances, and s the scale parameter of the model posterior and prior. λ value determines how broad the distribution is. The model estimates

for (μ, σ^2) can be found by using the mean of the posterior. If $\mu_0 = m$, sample mean, and using Eq.20, 22, and 23, the model mean and variance become

$$\mu = m \quad and \quad \sigma^2 = \frac{\nu_n \sigma_n^2}{\nu_n - 2} = \frac{\nu_0 \sigma_0^2 + (n-1)s^2}{\nu_0 + n - 2} \tag{3.26}$$

Thus, using these point estimates for the calculation of t-test score for each gene will correct the result for the small sample, large data case.

3.3.4 Generalized Linear Models and Moderated t-statistics

This section follows the derivations adapted from Smyth [70] of generalized linear models and moderated t-statistics.

Section 3.3.3 formulation can be generalized for data cases with more than two samples and those where the gene expressions do not have to be assumed to be a sampling from a normal distribution. For a gene g, and number of samples n, the expected value of the expression set $y_g = (y_{g1}, \ldots, y_{gn})$ can be represented as a generalized linear model:

$$E(y_g) = X\alpha_g \tag{3.27}$$

where X is the design matrix with columns specifying sample groups and rows specify the replicates in each sample group. The design matrix X should have n rows, the same as the number of columns in the data matrix. The normalized response vector y_g consists of the log-intensity measurements of a gene for each sample replicate case. The α_g is a vector of linear coefficients. Sample contrasts can be constructed from the design matrix X. So, for a contrast matrix C^T, the coefficients of linear estimates become

$$\beta_g = C^T \alpha_g \tag{3.28}$$

For a generalized linear model, the contrast coefficient estimates $\beta_g = (\beta_{g1}, ..., \beta_{gn})$ for each sample replicate are assumed to be normally distributed with mean β_g. For an unscaled covariance data matrix, V_g, the variance of the β_g is given by

$$var(\hat{\beta_g}) = C^T V_g C s_g^2 \tag{3.29}$$

where s_g^2 is the residual variance between data and the model. Again, by using Bayesian estimates for posterior probability of the model, the expected value of the model variance given the sample variance s_g^2 is

$$\tilde{s_g^2} = E(\sigma_g^2 | s_g^2) = \frac{\nu_0 \sigma_0^2 + \nu_n s_g^2}{\nu_0 + \nu_n} \tag{3.30}$$

The moderated t-test score using the estimate of the sample mean and for the j^{th} sample replicate is

$$\tilde{t}_{gj} = \frac{\tilde{\beta_{gj}}}{\tilde{S}_g \sqrt{v_{gj}}} \tag{3.31}$$

where v_{gj} is the j^{th} diagonal element of $C^T V_g C$.

3.3.5 Multiple Comparisons and p-value Adjustment

The p-values calculated using the above moderated t-test scores should also be adjusted for thousands of DGE comparisons. Here, the adjusted p-values are corrected by using the Benjamini-Hochberg procedure, which is also called false discovery rate (FDR) correction [34].

According to FDR correction, the calculated p-value for each gene is first ranked from highest to lowest. If the total number of ranked genes is R, and the chosen significance level is α, a gene k with calculated p-value p_k would only be chosen to be a DGE if,

$$p_k = \frac{k}{R}\alpha \tag{3.32}$$

with an adjusted p-value of

$$(p_{adjusted})_k = \frac{R p_k}{k} \tag{3.33}$$

3.4 Correlation Networks

Regulation of gene expression is a very complex field of study with many model building approaches as reviewed in Ay and Arnosti [6]. Network theory and network inference models are used more and more to explain the complex behavior of gene interactions. In this section, we will review two approaches that will form the starting point for our perturbation analysis in chapter 5: (i) weighted gene co-expression network analysis (WGCNA) [84], and (ii) partial correlation network analysis [80].

Also, we will review other approaches which are used to find the underlying network structures of gene regulatory networks. We will go over how these networks are used to identify the key genes of a network and how they can be used to rank the genes in the order of importance. For the rest

of this section we will use weighted edges to identify the interaction strength between nodes in a gene network where the nodes will be assumed to be genes.

3.4.1 Weighted Gene Co-expression Network Analysis (WGCNA)

Gene co-expression Networks (GCNs) aim to identify and group together genes that show similar trends of decreasing and increasing expression values over a set of samples. These co-expressed genes are usually assumed to have a common function in a cell and therefore show similar changes across samples with different physical conditions, such as treatment and control. The first step of WGCNA [84] is to find the weights associated with the pairwise interactions between genes. The association is measured with Pearson correlation as in Equation 2.27. The resulting square matrix captures the correlation values among genes which would be related to their co-expression strength.

WGCNA assumes that large gene data sets coming from high-throughput gene expression measurements such as RNA-seq, or microarrays (another type of high-throughput measurement), are elements of networks that show scale-free topology. This means even though the correlation values show how genes interact with one another, not all the values show important edges in the network. So, the second step is eliminating low-value edges by using a power-law equation. To find out the optimum power to the equation a soft-thresholding approach is used. So, by changing the power in the equation and looking at the node degree connectivity of genes in the network, one can reach an optimum mathematical representation of a network that retains enough edges to explain the important interactions.

Then, this matrix is turned into an unweighted adjacency graph and WGCNA calculates the cluster of genes which show the closest expression value changes across samples. Some genes that do not fit into any of the clusters would be grouped together and would be excluded from the rest of the analysis. Once these clusters, or so-called modules in WGNCA, are found then the genes can be investigated in terms of how they are grouped together.

The first step is to find the hubs of each module which would have the highest connectivity inside the modules. Then, the module eigengene is calculated which is the first principal component of the gene expression levels that belong to the same module. The correlation to the module eigengene would show how similar a gene expression to the eigengene is. Since module eigengene is assumed to represent the weighted average of the expression profile, genes with high eigengene correlations would lead the module trend. There is also gene significance measure that incorporates external phenotype information such as known biological interactions into the calculated co-expression network to find a ranking among genes of significance.

WGCNA is a well-established and well-used algorithm since its release date. There are some drawbacks to the algorithm that we came across during our research. The first one is that it is data dependent. The modules might not be as informative as it is intended to be. If there are distinct types of trends in the data, then the modules could only show those relations which is already obvious from the start. We came across networks that divided thousands of genes into only two modules: one showing a profile of straight decreasing trend and the other showing a profile of straight increasing trend across samples.

We feel the above drawback is the result of the use of Pearson correlation. WGCNA is strong at finding linear relations among genes, but it might not be sophisticated enough to explain the interactions in a network. Pearson correlation only measures whether two genes are correlated or not, without a distinction of the correlation. So, the network includes edges that show both direct and indirect interactions. Further more, even after the thresholding and grouping, there are a lot of edges to investigate afterwards.

After the modules are specified the next step for a biological data scientist is to go to databases and compare it by using measures such as the known interactions, genetic pathways, or known tissue co-localizations. We will go over this step more in section 3.4.4. Since the modules might be hundreds of genes large, it might also be hard to identify a concentrated relation between known functions and genes in the same module. Most of the time, the results include general gene functions which do not help the focus of the search to a specific group.

Additionally, we have seen that some of the genes which were excluded from the modules were in fact biologically important which gave the idea of whether the excluded group should really be excluded from the co-expression network. The module eigengene profiles also did not always fit to the average gene profiles of a module.

3.4.2 Partial Correlation Network Analysis

Instead of Pearson correlation, one can also use partial correlations to specify the edges, as we mentioned in chapter 2. Here, the algorithms usually use the first order partial correlations since the partial correlations tend to be very small numbers. There are a few approaches to calculate partial correlations, as mentioned before. The approach we use to construct networks in this thesis is the lasso regularized partial correlation approach which makes use of the generalized linear models with an l1 regularization coefficient instead of the regularized precision matrix approach [80]. We covered this approach and how it is calculated in section 2.3. Our approach perturbs partial correlation networks and we compare our results with the initial, unperturbed partial correlation networks of each input data to see whether our algorithm finds more critical networks

or not.

Partial correlation approach only includes edges with direct interactions. After the weight calculation of the edges and constructing the matrix, the rest of the analysis follows the same steps as WGCNA. Since the resulting network with only direct edges is a sparse matrix which is already regularized to find the optimum choice, there is no need to use a power-law approach. To find the modules, there are cluster algorithms that can be used. Even, WGCNA steps could be used to find the modules, eigengenes and significant genes.

The most important drawback of this approach is the sparsity of the matrix. Even though it only includes direct edges, there are important interactions between genes that are skipped when this algorithm is used. Also, because of the same reason, there are more genes that can be excluded when the network construction step is done.

3.4.3 Other approaches

There are algorithms that make use of mutual information approaches. MINET (Mutual Information Networks) [103] uses the MI equations of section 2.3 to get the edge weights. Aracne (Algorithm for the Reconstruction of Accurate Cellular Networks) [97] uses triplet of genes and calculates the pairwise mutual information strengths. For a given additive tolerance value, the algorithm excludes the edges among the three if it is lower than a threshold. There are other algorithms such as CLR (Context Likelihood or Relatedness Network) [44], and MRNET (Maximum Relevance Minimum Redundancy) [104], that also use a form of MI in their equations to reach the relevant gene interaction networks. All these network approaches result in dense networks with a lot of edges.

Adaptive Lasso [80] can also be used for gene correlation networks which combines both MI and partial correlation. It gives much sparser network results compared to partial correlation networks. PCIT (Partial Correlation and Information Theory) [134] method starts from a Pearson correlation network and uses partial correlation and MI equations to eliminate edges with low values. Kadarmideen and Watson-haigh [71] compared WGCNA and PCIT and found that PCIT eliminated biologically important edges that were present in a WGCNA network.

3.4.4 Searching for critical genes in a network

In all of the algorithms in section 3.4, the main idea of constructing a network is to reach the interacting genes that can explain the underlying changes observable in different samples. These interacting genes are often the ones that participate in the same protein synthesis, or are common

elements of a biological pathway or may be controlled by the same transcriptional factor. Therefore, for a regulation network the interacting genes might be the transcriptionally active genes that show highly differential changes across samples. If there is a perturbation or a disease, these genes are the ones that are affected the most, and show the most changes, and response to the perturbation.

However, not all the genes in a network are these active elements. Some interact with one another for reasons that are unrelated to the observable change. There are experimental errors that include genes with unreliable expression values. So, a network includes a set of genes with a range of interaction strengths. To find the highly active elements, there have been different approaches. Usually, the underlying network representation is investigated by focusing on small subnetworks and is coupled with other related data, such as known protein interactions, to reach biological results [128].

Poirel *et al* [128] explains a ranking system of genes in microarray data that starts with the p-values of DGE. They use machine learning and information retrieval approaches, such as Vanilla Algorithm [174], PageRank [120], GeneMania [105], and Heat Kernel [22], to calculate an additional correction term and incorporate this into the approach to re-rank the genes since the p-values are not always reliable in the large data sets. Their starting point is a biological human protein-protein interaction network. Bourdakou et al [18] uses the approach Poirel *et al* [128] developed, but starts with different network algorithms to prioritize genes in a network. They use breast cancer microarray data and compare networks constructed by using the above mentioned algorithms, such as Aracne, parcor, WGCNA etc. They re-rank the log-fold changes from the expression data set by using these network weights the same way Poirel et al explained. They look at the top 100 significant genes and their functional enrichment. Even though this method incorporates a more statistical approach, literature [80] shows that log-fold changes are not reliable to use for gene-by-gene eliminations.

Kadarmideen and Watson-Haigh [71] compare WGCNA and PCIT in terms of highly differentially ranked genes. They look at the hubs for each of different data sets they used for the same set of genes and they conclude a gene as a highly differentially ranked if the gene is highly connected in at least one of the constructed networks. The literature shows that the functional enrichment analysis of a list of genes which only includes hub genes show disconnected functions and pathways which do not lead to real biological conclusions [128]. So, only focusing on the hub genes of any network inference algorithm would be an incomplete explanation of a biological phenomenon.

Here, we are introducing a perturbation algorithm to search for transcriptionally active genes in a network. These genes would be the ones which show the most change when a perturbation is introduced. Although our algorithm would work with any of the above-mentioned network

inference methods, to only focus on direct interactions, we chose to use the partial correlation network. Since all the above-mentioned algorithms use different types of interactions between genes, they each end up with different networks. So, for correct comparison of our approach and see whether it reaches statistically significant differences, we will compare the resulting genes with the most connected high ranked hub genes of the partial correlation network for the same gene set.

3.5 Bioinformatics Methods and Tools for Gene Expression Data Analysis

We used R [131] statistical language and its Bioconductor [5] packages for our calculations and figure creations. R is a free and open source statistical language that is a part of the GNU project. It is a powerful language with mathematical, statistical and data manipulation tools, and it supports data structures such as vectors, matrices and data frames. Data frames enable the user to work on different types of inputs at the same time such as numeric vectors and character strings. It has matrix arithmetic abilities similar to those of MATLAB. File, directory, and system manipulations are possible in R, and it allocates memory automatically for a new data structure. It supports opening, viewing, and changing of tools and adding new algorithms. Other programming languages such as C++ and Python can use R scripts from within. R loads and saves data quickly, which means it can be used to easily manage large-scale biological data sets.

The genomic data tools that operate in R are gathered in Bioconductor [5]. Bioconductor is an open source software that enables users to download different biological and genetic data statistics packages to analyze and deduce comprehensive results. It has packages to fit, model, test and simulate data. These packages, called libraries, can be loaded to the R environment in the same way a data set can be loaded. We used R version 3.1 and its corresponding Bioconductor version throughout our analysis.

In the following sections 3.5.1 and 3.5.2, we go over the generic biological tools to preprocess a high troughput gene expression data as described in detail in [85, 93]. In section 3.5.3 we review the parcor [80] package that calculates regularized partial correlations. In section 3.5.4 we explain the Java application SynTRen [160] that generates simulated gene expressions. In section 3.5.5 we go over the enrichment analysis tools and data bases we used in chapter 7, and in section 3.5.6 we introduce the packages for plotting and statistical calculations.

3.5.1 Searching for Human Ortholog Genes

Animal models are an important part of biological research since they can shed light on human traits without posing a risk to humans. Mammalians, such as monkeys, mice, and rats, are usually used for animal models, especially for puberty and related research. The high number of ortholog genes these animals have, genes believed to have the same function in the body regardless of the species, is one of the reasons why these animals are used in research. Their shorter lifespan is another reason. Researchers compare the data collected from these animal model experiments to findings from humans and deduce solutions that may explain human traits.

The first step after collecting animal data is to make it comparable to data from humans. Biomart (www.biomart.org) is a free database that makes this possible. It connects many of the biological data repositories so that interspecies searches are possible. Based on search criteria, Biomart results include gene names orthologous from one species to another. These results also include gene information about the gene's ontology, different names, gene IDs according to different platforms, and so on.

Biomart also makes it possible to identify and eliminate superfluous genes by looking at their names, descriptions and gene ontology information in the data before analysis is performed. This reduces the size of the gene pool being analyzed and thus increases the possibility of finding a true biological result. The Bioconductor package that enables R to connect the Biomart website, download species data, make comparisons and load platform-specific gene IDs is called bioMart [38, 39].

3.5.2 Finding Differential Gene Expressions

As described in [85, 93], for RNA-seq data, Bioconductor package limma (linear models for microarrays) [135] can be used to find differentially expressed genes. Limma uses moderated t-statistics and empirical Bayesian method to fit linear models to the data. It also uses lcpm (log counts per million) values as the normalized gene expression values.

One can use these lcpm values and construct design and contrast matrices. To eliminate duplicate gene names from the data so that networks would have unique node names, F-scores can be used to rank the genes. Only the one with the highest F-score among the same-name genes is kept. In chapter 7 we follow the same approach.

3.5.3 Partial Correlation Matrices

The parcor package [80] calculates partial correlations between genes. This package uses lasso regularization to estimate partial correlations for a data set with the number of variables much greater than number of observables. Its input is the matrix with genes as rows, and different sample expression values as columns which is the output structure of the limma package.

Partial correlation values should be between -1 and 1 with the boundaries representing perfect positive or negative correlations. The only values of 1 should be the diagonal elements of the matrix where the genes are correlated with themselves. The non diagonal elements, which correspond to the weighted edge values of the networks, should be small values, given that they show the correlation between two genes when all other effects of the network are controlled. Thus, to check whether the partial correlation network matrix is reliable or not, one can check the non-diagonal elements and discarded the networks with unreliable partial correlation matrices. One can set a threshold for unreliable matrices as those with more than one half of their elements having values equal to or larger than 0.8.

The optimal lambda of the lasso calculation is determined by cross validation with k=10. The partial correlation network is the corresponding network with the optimum lambda. In chapters 6 and 7 we used parcor package to construct networks at each step of the beam search iterations.

3.5.4 Simulation of gene expression

Java application SynTReN (Synthetic Transcriptional Regulatory Networks) [160] generates simulated gene expression data. It uses emprically known network topologies from gene regulatory networks of E.coli and S.cerevisiae organisms as the source of interaction generation between nodes. Once the number of transcriptionally active nodes and background nodes to be generated is given as initial input the algorithm uses the known topologies and generates the interaction network by using either neighbor addition or cluster addition to add new nodes.

Once the interactions are set, the coefficients are calculated by using clustering coefficient, as mentioned in section 2.1.3. The nearest neighbors, also mentioned in section 2.1.2, have the highest interaction strength. The expression values between nearest neighbors are highly correlated, whereas the correlation decreases as the neighboring degree increases.

For the active node set, the nodes may show activator or repressor type interactions. All gene expression values are normalized between 0 and 1. The higher the gene expression value the higher the transcription is. A value of 1 shows maximal level of transcription. The number of activators and repressors in the active node set are randomly chosen from predefined distributions by a range

so that the generated variation is likely to occur in a true network.

After setting the interaction coefficients, the gene expression data is generated by using Michaelis-Menten and Hill kinetics. There are three gene expression rate variations to consider by using input parameters of noise levels. The first is the variation among replicates of the same experimental stage. The variations among replicates of the same biological conditions are only set by using noise. For example the relicates can be different samples coming from the same experimental stage. The second variation is the variations between experimental stages for background nodes. These variations are set by using slight random increases or decreases of the basal levels of gene interactions. The third one is the variations between experimental stages for transcriptionally active nodes. For these, the variations are distinct from one stage to another to mimic the time course of true experiments such as gene overexpression measurements.

3.5.5 Enrichment Analysis

Gene set enrichment analysis (GSEA) [149] uses the idea of statistically comparing previously known gene sets to a gene set of interest. There are databases such as Enrichr https://amp.pharm.mssm.edu/Enrichr/ and GSEA https://www.gsea-msigdb.org/gsea/index.jsp , that store gene sets found from different experiments. The gene sets are grouped into categories such as common biological function, chromosomal location, or sequence information. So, by using the gene set in question and comparing the number of shared genes in the databases statistical conclusions can be made. If there are a number of common genes which are already found from the previously known mechanisms or disease phenotypes, then the gene set in question might have a connection with these phenotypes.

We used the enrichR [83] package to find the enriched gene sets in our results by comparing our empirical gene sets with gene sets from known databases. For the p-value calculation of the enrichment, the package uses the size of the gene set from the experiment and of the database, and the size of the resulting intersection. The p-value is then adjusted for the large dataset. If the adjusted p-value is smaller than 0.05, the result is accepted to be enriched.

3.5.6 Graph Tools

For the scale-free network calculation, the igraph [27] power-law-fit function can be used to test whether a network shows scale-free characteristics or not. This function calculates simultaneously both a minimum node-degree value and an alpha value that fits the power-law curve by using MLE. It also gives the distance value D between the fit and the real curve where alpha is fit. The KS-test

was used to determine whether the alpha corresponds to scale-free power law model exponent or not. A p-value less than or equal to 0.05 is included.

To find the clusters of a network, the igraph package [27] can be used. The cluster detection algorithm of igraph we used is the fast-greedy algorithm, a bottom-up, greedy method that tries to optimize the modularity equation. It looks for local optimal clustering of nodes to achieve the greatest increase in modularity.

Network similarity can be evaluated using the clusteval [132] package for Jaccard index calculations. It uses the same node list as one of the inputs, and two different clustering membership vectors of these input node lists as the other input. It calculates the Jaccard index by looking at whether the same nodes are grouped together.

Chapter 4

Biology of Initiation of Female Puberty

Puberty is the postnatal developmental stage where the body goes through physiological changes to attain the ability for reproduction. It is well established in the literature that the age at which the teenagers reach the milestones of puberty is getting younger. This change is more evident in young females than males. The age at menarche (AAM), which is the sign of the beginning of the capability to reproduce decreased over the years in developed countries [133]. The study of the initiation of puberty tries to shed light into the understanding of this phenomenon. In this chapter, we will review the hormonal, genetic and epigenetic findings of the initiation of puberty.

4.1 Neuroendocrine Control of the Onset of Puberty

Like every other hormonal process going on in the body, neurons control the initiation and the continuation of the changes in puberty stages. There are billions of neurons located in our brain. Only 800 - 1000 of these neurons show activity during the initiation of puberty. These gonadotropin-releasing hormone (GNRH) secreting neurons are located in the medial basal hypothalamus(MBH). Puberty is achieved when GNRH is re-activated in the body. GNRH then stimulates the synthesis and release of the two pituitary gonadotropins follicle stimulating hormone (FSH) and luteinizing hormone (LH). GNRH secretion and FSH and LH stimulation is a pulsatile event which gives the name "pulse generator" to GNRH [126]. In humans, there is a short activation of GNRH at the beginning of infancy which is called "mini-puberty". Even though the purpose of this hormonal period is not completely understood, recent studies show an association between growth rate and testosterone secretion [74]. After this period all three hormones go into a quiescence phase where the GNRH neuronal activity is inhibited till the early puberty stage of the development is reached.

Both types of hypothalamic cells, neurons and glial cells, take charge in the neuroendocrine control of puberty. These cells work to both excite and inhibit mechanisms that regulate the onset of puberty. During infancy the balance is tilted toward inhibition as mentioned above, whereas at

puberty there is a loss of inhibition with a simultaneous increase in excitatory signals. [126, 154].

4.1.1 Neuronal Mechanisms

The transsynaptic control of GNRH neurons include both inhibitory and excitatory neuronal mechanisms. The strongest excitatory input comes from neurons that secrete kisspeptin peptides, that in turn increase GNRH [114, 29]. Inactivating mutations in the KISS1 gene or KISS1 receptor (KISS1R) induce pubertal failure [141, 32], where a mature body is not able to attain a reproductive state at the age of puberty or later. The kisspeptin neurons located in the arcuate nucleus (ARC) of hypothalamus are called KNDy neurons. KNDy neurons release kisspeptin, neurokynin B, and dynorphin [110, 161]. Neurokynin B (NKB) has the ability to stimulate KNDy neurons to release kisspeptins which in turn stimulate GNRH neurons [12], while dynorphin inhibits kisspeptin release. Additionally, glutamergic neurons provide excitatory input to GNRH neurons. These neurons produce glutamate, the most common neurotransmitter used by the brain for signaling. Glutamate excites GNRH neurons to release GNRH [126, 116].

There are three neuronal types that provide inhibitory transsynaptic regulation of GNRH neurons. Opiatergic neurons have the ability to inhibit GNRH secretion both directly and indirectly. Opioid peptides can reduce GNRH directly, or by affecting kisspeptin release which in turn inhibits GNRH secretion [154, 77, 37, 116, 109]. RFamide-related peptide (RFRP)-containing neurons can inhibit GNRH neurons directly by using two peptides, RFRP1 and RFRP3 which bind to a receptor GPR147, which is expressed in GNRH neurons [65, 159]. GABAergic neurons release the neurotransmitter gamma aminobutyric acid (GABA)neurons. These neurons also take charge in the negative regulation of GNRH secretion [154, 63]. GABA is a common inhibitor in the central nervous system which affects the expression levels of GABAA and GABAB receptors in GNRH neurons [116, 154, 88].

4.1.2 Glial Inputs

Glial cells surround the nerve cells to protect and provide them the right environment for proper function. Glial cells also contribute to the hypothalamic control of puberty [90, 130] via a stimulatory mode of control. Glial cells control GNRH function via diffusible growth factors, cell adhesion molecules and small molecules, such as ATP and protaglandin E2 (PGE2). Studies show that these molecules act on GNRH neurons and enhance GNRH secretion [90, 23, 130].

Cell-to-cell adhesive interactions are mediated by the use of adhesion molecules. The glial cell adhesion molecules synaptic cell adhesion molecule 1 (SynCAM1) [138, 137] and sialylated neural

cell adhesion molecule NCAM (PSA-NCAM) [122, 124] regulate GNRH activity by changing the glial to neuronal response area. Synaptic Cell Adhesion Molecule 1 (SynCAM1) interacts with other SynCAM1 molecules expressed in GNRH neurons modulating GNRH function [138, 137]. Neurons also express contactin protein which is another cell adhesion molecule. Parent et al [121] showed that during active stages of GNRH neurons 80 percent of cells show high expression of the contactin gene (CNTN) expression. Contactin binds to the glial recognition molecule receptor-like protein tyrosine phosphatase-β (RPTPβ). SnyCAM1 and RPTPβ also have cell to cell signaling capabilities which enable them to mediate the GNRH to glial cell communication indirectly.

Growth factors synthesized in glial cells also act on GNRH neuron receptors and take part in signaling and stimulation of GNRH neurons. These growth factors include FGFb, IGF, and TGFβ. These growth factors are synthesized and released by glial cells. Their action is executed through specific receptors present in GNRH neurons. Most of these receptors use tyrosine or serine kinases as signaling transduction mechanisms. Ultimately, the activation of the growth factor signalling cascade increase PGE2 production or intracellular Ca2+ which in turn increases GNRH release [90].

4.2 Genetics of Puberty Initiation

Idiopathic hypogonadotropic hypogonadism (IHH) is a range of disorders where reproductive function and sexual development is affected. It manifests as a puberty failure due to deficient gonadotropin release. IHH has two types: anosmic and normosmic. In humans, GNRH neurons are not initially located in MBH. Instead, in embroyonic stages, the GNRH neurons migrate from the nasal placode regions through the vomeronasal nerve and reach the brain [169]. Kallman syndrome (KS) is a subtype of IHH where this migration is interrupted producing anosmia and infertility. Genes that are related to KS include KAL1, FGFR1, FGF8, PROK2, PROKR2, CHD7, WDR11, SEMA3 and NELF [141, 32, 156, 155].

Normosmic IHH (nIHH) is the second type of IHH and also related to single gene mutations. Most of the genes mutated in patients with nIHH are involved in the stimulation of the GNRH pulse generator at puberty. Those are: GNRH, GNRHR, KISS1, KISS1R, TAC3 and TAC3R. The genes LEP, encoding leptin, and LEPR, encoding leptin receptor, whose mutations result in nIHH are also related to obesity. These genes are essential for the regulation of energy metabolism and the onset of puberty [2, 40]. Mutations in genes involved in intracellular phospholipid regulation, and regulation of membrane trafficking proteins show brain, eye and endocrine abnormalities which was linked to nIHH. These genes include: PNPLA6, POLR3A, POLR3B, RAB3GAP1, RAB3GAP2,

RAB18 and TBC1D20 [157].

A recent study showed that mutation in the MKRN3 gene, encoding the protein makorin ring finger protein 3, is related to early puberty, thus it is assumed to be involved in the inhibitory control of puberty [1]. In addition to the single gene mutation studies, genome wide assocation studies (GWASs) have found more than 380 genetic loci variations associated with female pubertal onset [30, 119, 125, 150, 61, 41, 26, 152]. These genes include the above mentioned mutated genes such as GNRH, KISS1, and MKRN3. Additionally, GWASs results included enrichment of genes with different functions in the central nervous system as well as in the pituary gland. These wide range of functional genes support the idea of puberty initiation as a coordinated work of functional modules of genes in a larger gene network. It also supports the idea that the timing of the initiation of puberty is under significant genetic influence [25, 56, 57].

4.3 Transcriptional Control of Puberty

Transcriptional factors (TFs) are proteins that regulate gene expression by interacting with upstream DNA regulatory regions, as mentioned in section 3.1.1. TFs can either activate or repress gene expression, thus their balance is key for gene activity and function at specific developmental stages.

The presence of a transcriptional repressive mode of control regulating puberty activating (PA) genes was first proposed by Dr. Ojeda's lab in studies showing a tumor-related gene (TRG) network whose central nodes were able to repress the transcriptional activity of KISS1 [62, 106]. TRGs are related to tumor formation or suppression regardless of their wide-range of cellular functions. Studies show that these genes and their interacting neighbors form a well-organized network with a central core of repressive regulators and a peripherally situated transcriptionally regulated genes. These central core genes include EAP1, OCT2 and TTF1 [98, 62, 115], all activating TFs.

Experimental evidence by Matagne et al showed that thyroid transcription factor (TTF1) increased expression levels vary diurnally and reach its maximum just before GNRH maximum expression level is reached in female rat hypothalamus [99]. OCT2 has been shown to control TGFα expression in the hypothalamus [115]. Mueller et al showed that EAP1 takes charge in trans-activating GNRH promoter, and also indirectly KISS1 promoter. The same study also shows that EAP1 transcriptional activity, in turn, is regulated by itself, TTF1, CDP, and YY1 [106].

Other trancriptionally active genes, such as NELL2 and NRG, related to puberty have shown similar interactions [50]. The trithorax group (TrxG) member MLL3, which we will explain in section 4.4 below, has been shown to serve as a central hub by regulating the expression of 39

downstream genes [50]. These experimental evidence and network connectivity strongly suggest that pubertal onset is the result of different gene regulatory networks working together.

4.4 Epigenetics and Puberty Initiation

Epigenetics refers to hereditable changes that are not carried by the DNA sequence. These modifications affect the DNA accessibility and also the interactions with chromatin structure, which in turn results in different control mechanisms of gene expression. There are three types of epigenetic regulation mechanisms: DNA Methylation and Hydroxymethylation, Histone Post-translational Modifications, and Non-coding RNAs. Each of these three methods is discussed below. At the last section we review the findings on how these epigenetic methods affect the initiation of puberty.

DNA Methylation and Hydroxymethylation

The first mechanism of epigentic regulation is DNA Methylation and Hydroxymethylation. DNA methylation is the catalytic activity where a methyl group (CH3) is added to cytosine at the C5 position. This chemical reaction targets the CG base pairs in DNA [67, 16]. Methylation is mediated by the enzyme family called DNA methyltransferases (DNMTs). In mammals, these enzymes are DNMT1, DNMT3a and DNMT3b. These enzymes take charge in the processes where 5-methylcytosine (5-mC) is formed. On the other hand, the TET family of dio-oxygenase enzymes change 5-mC into 5-hydroxymethylcytosine (5-hmC) in oxidation processes [151, 76]. While repression of gene expression is associated with high levels of 5-mC at gene regulatory regions, gene activation is associated with increased 5-hmC [47, 54].

Histone Post-translational Modifications

The second mechanism of epigentic regulation is modifications of the chromatin structure caused by posttranslational modifications (PTMs) of histones. Chromatin is a macromolecule including DNA, proteins and RNA which is formed to package and protect DNA, control gene expression and DNA replication processes, and facilitate mitosis. Histones are the most common proteins found in chromatin. They bind to one another to enable DNA packaging. The smallest chromatin structure, euchromatin is formed by a part of a DNA sequence packaged into a denser molecule including four histone proteins (H2A, H2B, H3 and H4) in the center. The nucleosome is the smallest unit of chromatin, composed of a single histone octamer core and around 126 bp of double stranded DNA. PTMs of these histone proteins occur on their N-terminus tails [79, 8]. These tails can be changed by PTMs such as acetylation, methylation, phosphorylation, ubiquitination, and

sumoylation among other modifications [79].

Histone acetylation is the enzymatic reaction where an acetyl (CH_3CO) group is added to the lysine (K) of a histone in the nucleosome core. The removal of the acetyl group is called deacetylation. The enzymes that facilitate these two reactions are histone acetyltransferases (HATs), and histone deacetylases (HDACs), respectively [79, 173, 136]. Acetylation is related to activation of gene transcription, and deacetylation with transcriptional repression [79]. Histones can be methylated with the same process we mentioned above. It can result in either transcriptional activation or repression depending on the methylation site and how many times the residue is metylated [68]. Experimental evidence suggests that methylation of lysines 9 and 27 of H3 (H3K9me and H3K27me) is associated with transcriptional repression, whereas trimethylation of H3 at lysine 4 (H3K4me3) is associated with transcriptional activation sites [79, 123].

Non-coding RNAs

The third mechanism of epigentic regulation is through non-coding RNAs (ncRNAs), providing epigenetic information as either miRNAs or as long-intergenic noncoding RNAs (lincRNAs). Non-coding RNAs are the types of RNAs where the information copied from DNA does not include genetic codes. There are several types of non-coding RNAs which carry epigenetic information. Small non-coding RNAs are microRNAs (miRNAs) [66, 48], endo-small inhibitory RNAs (endo siRNAs) [118], and piwi RNAs (piRNAs) [73, 52]. The experimental evidence support that these RNAs are epigenetic silencers [66]. They control gene expression stages of transcription, RNA processing and translation [19]. miRNAs have been shown to affect DNA methylation and histone PTMs. Long non-coding RNAs are long intergenic non-coding RNAs (lincRNAs). These RNAs have shown to bind to the molecules that control chromatin structure which in turn control gene expression [11, 53].

Epigenetic Regulation of Puberty Initiation

Research in non-human primates supports that the regulation of puberty is achieved by changes in gene expressions resulting from DNA metylation and demethylation reactions [171]. The latter corresponds to the chemical reaction where the methyl group is lost. Studies with GNRH neurons show evidence that a decrease in methylation levels at sites with CG pairs upstream from the GNRH transcription starting site (TSS) that result in a repressive influence, thus activating GNRH transcription [69].

Messina and colleagues [102] demonstated in their mice studies that miRNAs epigenetically regulate the GNRH gene during the infantile-to-juvenile period. They showed that impaired miRNA

synthesis leads to reduced plasma LH and FSH, hypogonadotropic hypogonadism and infertility due to a reduction in GNRH trancriptional activity.

Lomniczi et al. [89] showed that two central members of the polycomb group (PcG) of transcriptional repressors, EED and CBX7, are expressed within ARC kisspeptin neurons, and their protein products are associated to the KISS1 promoter during prepubertal development. At the end of juvenile stages the methylation at the promoter sites of these genes increased resulting in decreased EED and CBX7 expressions. This decrease is accompanied with a reorganization of chromatin structure at the KISS1 locus and an increase in histone PTMs at H3K9ac, H3K14ac and H3K4me3, which are modifications associated with gene activation. This in turn resulted in an increase in hypothalamic KISS1 expression.

The fact that not only the PcG epigenetic-silencing components are lost but also the epigenetic-activating components, such as trimethylation and acetylation of H3, increase during pre-pubertal development suggests a co-occurance of an activating complex which controls the epigenetic excitatory arm. In a subsequent paper from the Lomniczi lab, two genes, MLL1 and MLL3 - two members of the TrxG complex, are identified as the trans-activational input on promoter and enhancer regions of the KISS1 gene [158]. These genes were shown to take charge in essential histone acetylation PTMs hence changing the epigentic control of KISS1 to an active state when puberty starts. Their experiments also showed evidence that members of the zinc finger (ZNF) family of transcriptional repressors are important components of the puberty network [91]. This family of genes controls GNRH secretion levels during the pre-pubertal development in a repressive manner, by preventing the premature re-awakening of GNRH pulse generator.

Chapter 5

Impact Node Finder Algorithm

In many domains where graphs are used to capture relationships among variables of interest, such as genes and proteins in biology, it is of great practical interest to identify which graph nodes have the biggest impact in the outcome of interest. For example, in a gene interaction network for a particular biological pathway, the scientific question usually involves which genes are essential to the function of the pathway. Specifically, for the biological application domain that we use in this dissertation, the most important question is which genes control the onset of puberty in such a way that if they did not perform their regulatory function, the menarche would not start. This is a motivating example for the algorithmic question in this dissertation: How can we identify which nodes carry the largest impact? To answer this question, we develop a technique for perturbing a network by removing nodes and estimating the impact of the perturbation in an iterative manner to identify a sequence of nodes that influence the outcome maximally. We name our algorithm Impact Node Finder (INF) and describe it in detail in this chapter. The following two chapters are devoted to the application of the algorithm: first to simulate gene expression data in which we determine the nodes of impact and then demonstrate that the INF algorithm successfully identifies them, and second to real expression data from rats collected for the onset of puberty project to identify a set of genes that are impactful for starting the onset of menarche.

This chapter covers the components and the workflow of the INF algorithm. First, we explain the main components of the algorithm and how they function. Second, we describe how the components are integrated for a beam search of impact nodes in a network. A beam search over possible impact node sets works similar to a beam search algorithm in automatic speech recognition (ASR) that searches possible word sequences with the best fit to a given input. Unlike ASR, here the order of the chosen nodes do not affect the output. The impact nodes can be considered as the nodes which are the active or critical nodes, in a network. They are the ones that show the most changes when a perturbation is introduced, and they are the ones that would impact the functionality and the topology of the network the most when taken out.

We are interested in both functional and structural changes in networks. For functionality, our approach is observing how the clustering changes in a network when a perturbation is introduced. Since nodes cluster together to function properly, the change in the clustering of nodes gives clues about the change or failure in the functionality of the network. For the structure, we compare networks according to the number of edges or number of nodes they are able to keep when a perturbation is introduced. Since the structure of a network also affects its functionality when it changes considerably, the functionality of a network would be affected when it loses most of its edges or nodes. We will investigate the results of perturbations according to clustering and node and edge values in chapter 6 with simulated data.

5.1 Objective of the Algorithm

The goal of the INF algorithm is to find the critical nodes of a clustered, scale-free network. The algorithm searches node subsets, such that, when removed, each one has the potential of disrupting the structure or the functionality of the network. Our algorithm introduces a way to measure how impactful a set of nodes is. It finds the set of nodes that results in a graph that shows the most change by a measure when compared to the created graph with all of the nodes in it.

There are two levels of comparisons when deciding the critical nodes of a network. The first level looks at the node characteristics to decide which nodes in the network have the highest impact potential when eliminated. At the node level, at each iteration, for each network, the algorithm ranks the nodes according to their total edges and edge strengths and chooses the nodes with highest values as critical. The second level compares the resulting networks after the elimination to decide which of the first level chosen nodes ended up impacting the network the most. At the network level, at each iteration, the algorithm compares the newly created networks with the one with all nodes and chooses the most different ones according to a network comparison measure. Below, in sections 5.2.3 and 5.2.4 we will explain more in depth about node level ranking and the network level comparison measures.

A brute force approach by trying all possible subsets of nodes in a network to find the critical nodes would take exponential time in the number of nodes. So, a more efficient way to find the set of nodes is important. One could do a greedy approach, find the one node that causes the graphs to be the most different, then find the next node that when added to the first node are the most disruptive, and keep going. However, that approach could easily get trapped in a local minimum. Instead, our algorithm does a beam search over subsets of excluded nodes. At each iteration a set of most possible node sets are chosen according to a network comparison measure and the rest is

eliminated from the search space. Even with this approach, network comparison step is still really slow when all nodes are considered. So, we do a two-step search as explained above. We first use a heuristic measure to select the most likely nodes B_i of at most some constant in size. This gives us a small set of very likely nodes to then do the full test on.

Another difference from a beam search for ASR mentioned above is the result of the algorithm. In an ASR beam search, the final outcome of the algorithm is the best contender of the beam width. In our biological data there are thousands of nodes in a network. However, our beam width is chosen by taking into account the processing steps we need to do between iterations. So, in our case, the beam width was much smaller and the pruning was much more stringent compared to an ASR beam search. To overcome this short coming, we considered all the contenders at each iteration as a possible outcome and investigated further with a biological gene selection step for the real data case.

In addition, because of the same reason of too small beam width compared to the data size, there is a randomness of the contenders that needs to be taken into account. As the number of iterations increase, the pruning criterion rankings become very close to one another. This makes it statistically impossible to distinguish between the last chosen contenders of a beam and the first unchosen candidates for higher number of iterations. Therefore, we stopped our algorithm after a set number of iterations even if no final criteria is met.

5.2 Components of the Algorithm

Figure 5.1 shows the workflow of INF algorithm. We start with an input data of n nodes. The first step is creating the initial network, G_0. The contenders of the first iteration of the beam search are chosen by using this initial network. As a first level of search, by using the potential node list of G_0, we find the first subset of nodes to be excluded, the initial beam. For each node subset, the node subset is eliminated from the initial input data and a new network, G_S, is constructed. Every new network is checked for final criteria thresholds, and if none of the final criteria are met, at the second level of the search, each G_S is compared to the initial network. This is the part where the contenders of the beam is ranked and the beam pruned according to a network comparison metric. For a specific beam width, only the networks at the top ranks are chosen for the next iteration. The next likely nodes to add to the node subsets to be eliminated are chosen by using these chosen networks, and the first level comparison step. They are added to the contenders of the next iteration and the algorithm repeats the steps.

The potential node list of lists include the sets of nodes as elements. At each iteration, every

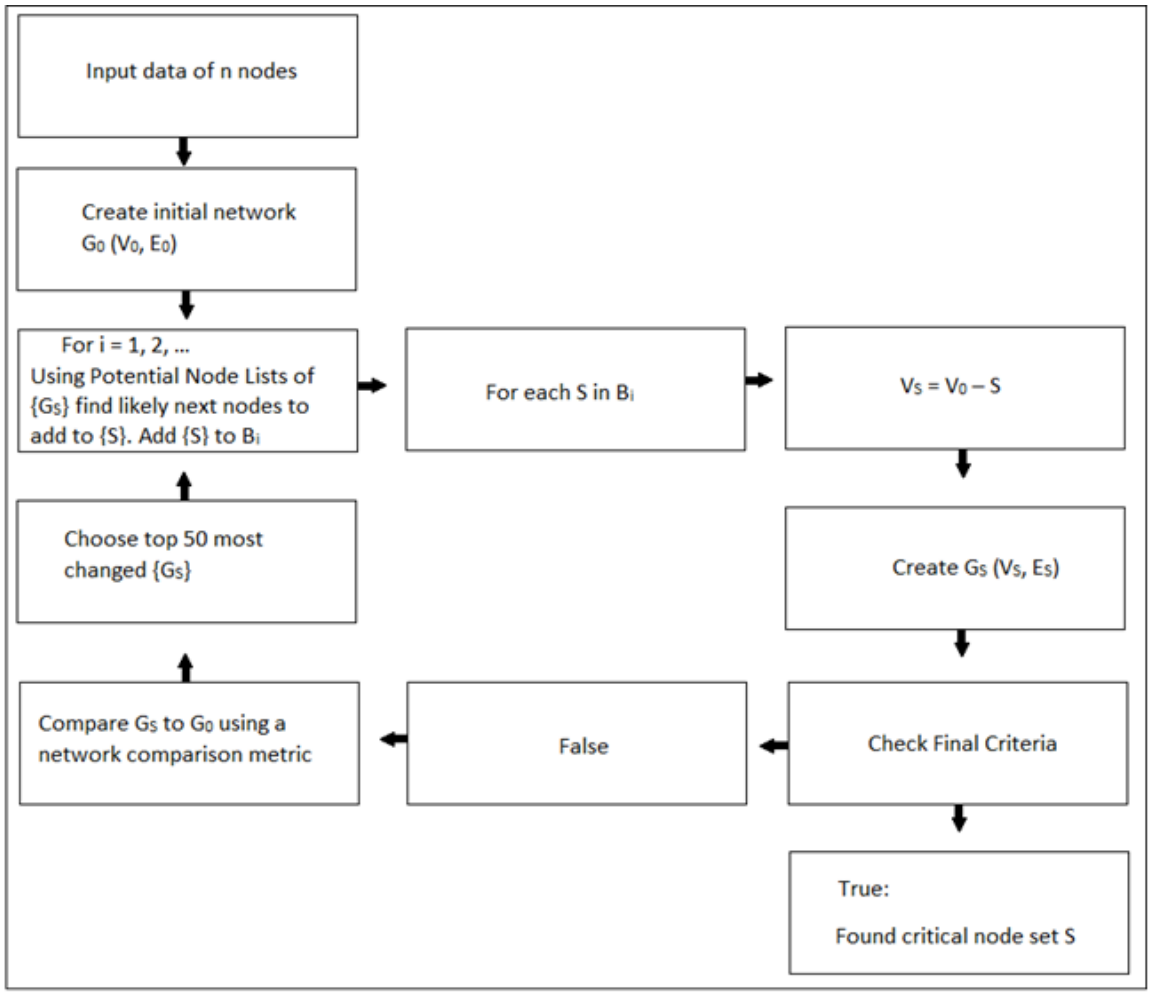

Figure 5.1: Impact node finder algorithm components and workflow

new network with the graph representation $G_S(V_S,\ E_S)$ is created by excluding those elements, which is represented by the subscript S. These elements in the potential node list grows by one at each iteration. The final outcome of the algorithm is chosen from these potential node list elements. Below, we review the components of the algorithm workflow, its steps, user-specific definitions and thresholds.

5.2.1 Input Data

The input data is an (n x p) data matrix with n number of unique nodes, and p number of samples. The n nodes can all connect to one another to form a network, or some might be disconnected. The disconnected subset might join the network structure in the event of an outside effect such as an introduction of a perturbation where a set of network nodes are lost. There might also be network nodes which would become disconnected as a result of this introduction. These node subsets may have sizes in the range of 0 to n. The list of n nodes is chosen such that the nodes show statistically significant changes measured over a set of samples. The samples are numerical values explaining a common trait, function, or state. According to the sample values, a subset or all of the nodes might result in a network that would explain a common outcome. In the scope of this thesis, we assume all the nodes are independent from one another, and all the samples are measured independently from one another.

5.2.2 Network Creation

The network creation step searches for statistically significant interactions between nodes according to their sample values. This step calculates pairwise partial correlations using the samples to find the interaction strengths between nodes. For the case of $n \gg p$, the partial correlations should be calculated by taking regularization into account. We used the lasso algorithm to create regularized partial correlation networks as explained in section 2.3.2. After the regularized partial correlation calculation, the network graph consists of a sparce number of edges between nodes that include only the direct interactions between the node pairs in the edge list.

Each network is defined as a graph $G(V, E)$ where G is weighted and undirected. V denotes the unique node list, a set that specifies node names that form the network. The set V might include the whole set of n nodes, if all the nodes interact with one another, or a subset of n. In this thesis, we used all-inclusive set V which contains all input node list regardless whether they form an edge or not in the network. This way, the network comparison step can be done considering the change in the node lists which we will discuss in section 5.2.4. Edge list E includes only unique node

pairs such that multiple edges and loops are not allowed. E is a two-column matrix with a list of two-node pair IDs as edge names in one column, and the partial correlations, which are numeric values that specify edge weights, in the second column. This list only contains nodes with edges.

The first step of the algorithm is to create a network using the full set of input data to form the initial structure, which is termed the initial network (G_0), with a weighted and undirected graph representation of $G_0(V_0, E_0)$. All the other networks created from here on out are compared to this original network. The new networks G_S of every iteration are represented as $G_S(V_S, E_S)$.

Mathematically, the above network graph with n nodes is represented as an (n x n) symmetric matrix, where rows and columns are identified as the nodes, and the matrix element a_{ij} specifies the edge weight. A value of zero in the matrix denotes no interaction between nodes. The diagonal elements are the interactions of nodes with themselves, and therefore, are excluded from the calculations below.

5.2.3 Potential Impact Node Lists

Potential impact node lists contain the node candidates to add to the next beam search iteration contenders. For each candidate subset of nodes in the beam, we look at potential additions to subset, based on node degree and node strength. The chosen list for every network is the union of the unique node set coming from the highest strength and highest degree node sets for a given threshold. This is the first level of search to choose the contenders with the highest potential of impact.

Figure 5.2 is an example of how our algorithm works according to potential node lists. It shows how the first two iteration potential node lists are created. At the first step, the initial network, G_0 is constructed. The first potential node list M_1 contains the four nodes with node degree or node strength values higher than the threshold according to G_0 network node characteristics. These are the first candidates of the beam search. The next step is new network creation. For each element in the potential node list, the element is eliminated from N_0 and network is created by the procedure explained in section 5.2.2. These four networks are compared to the original network G_0. The subscript numbers specify the iteration and the superscripts specify the set of nodes being eliminated. If, for a beam width of 2, the most changed networks are ${G_1}^{\{5\}}$ and ${G_1}^{\{89\}}$, then the second iteration only takes into account these two networks. The first potential nodes to take out are already decided. For second nodes, the potential node lists are coming from the chosen two most changed networks. The combined second iteration potential node list is a list of lists including node pairs coming from the first iteration and the new nodes chosen from the new networks. The list M_2 of step 7 in the figure contains the candidates for the second iteration of

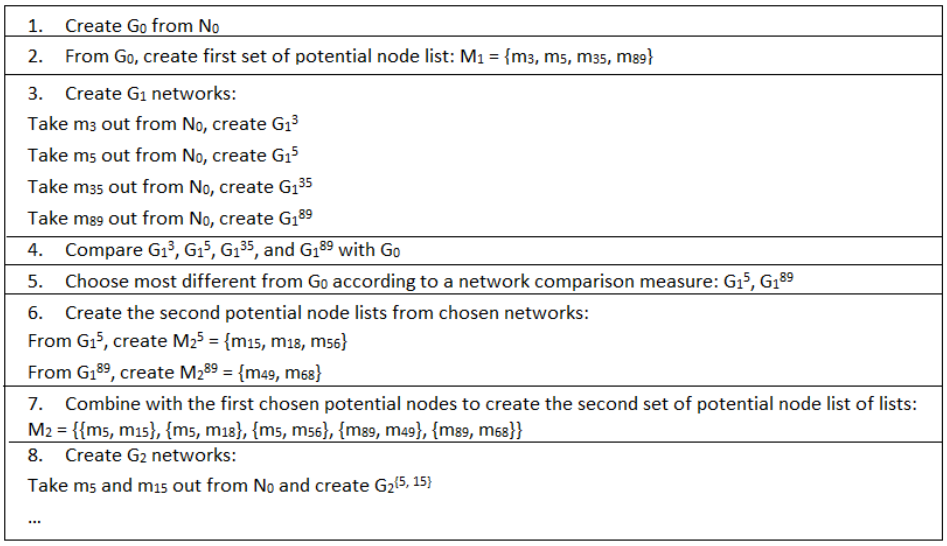

Figure 5.2: Example of potential node list of lists workflow

the beam search. The iterations continue in this way by adding one more node to lists chosen from the newly created networks. Since the order of nodes does not have significance, the lists with duplicate sets are eliminated.

Node degree, as mentioned in section 2.1.2, is the total number of edges a node has in the network. For a square partial correlation matrix A of size N, the degree k_i of node i given by equation 2.2 is as follows.

$$k_i = \sum_{j=1}^{N} c \;\; c = 1 \; if a_{ij} \neq 0, \;\; c = 0 \; otherwise \tag{5.1}$$

The degree of a node shows how connected a node in a network is, and therefore, the more edges a node has the greater the impact potential it will have when removed. However, not all edges are equally important. Even though the regularized partial correlation network gives a sparce network, already eliminating most of the edges if the interaction comes from an indirect source node, the edge weights still have a large range between their maximum and minimum values. Therefore, so as not to discard nodes with fewer edges that might have a high impact potential because they have a large correlation with their connected neighboring nodes, we also include node strength.

The partial correlation values may range between -1 and 1. Nodes connected by edges can be negatively or positively correlated to each other. There will also be nodes that have a positive

interaction with one node and a negative interaction with another. To ensure we do not decrease the impact potential of such nodes, we use the absolute value of their edge weights. By using equation 2.3 and the absolute value of weights, for the same matrix A, the strength s_i of node i is,

$$s_i = \sum_{j=1}^{N} |a_{ij}| \tag{5.2}$$

Here, we used a threshold of top 5^{th} percentile for both node degree and node strength to specify the node impact. The nodes with number of edges higher than the threshold form the potential node impact set by degree. The nodes with number of total absolute value weighted edges higher than the threshold form the potential node impact set by strength. The union of these two sets is the complete set of potential impact nodes selected to add to the beam.

5.2.4 Network Comparison

Network comparison step is the pruning step of the beam search. By using a network comparison metric, each created G_S is compared to the initial network. The contenders of the beam is ranked and according to a chosen beam width only the top contenders are chosen. Even from the first iteration where only one node is eliminated, the newly created networks can demonstrate differences compared to G_0. Some include new nodes that were not present in the original network, and some lose nodes. Some edges also differ. So, these differences, even if they start small, enable the networks to have different rankings.

Since we are interested in both the structure and the functionality of the network, to decide which networks changed the most, we explore three different metrics to compare G_S to G_0. These three metrics of network comparison step are Jaccard index, edges lost and nodes lost criteria. Below, we will go over each of these methods one by one.

Jaccard Index

Jaccard index is a measure of how two divisions of clustering of the same network resemble each other. To use Jaccard index network comparison metric as a pruning criteria, the first step is clustering nodes by using a clustering algorithm. We explained Jaccard index and different clustering algorithms in section 2.1.3. The clustering algorithms optimize different similarity or distance measures and needs to be chosen accordingly. The next step is to iteratively match up the cluster labels of G_0 and G_S. One can use a greedy algorithm, such as discussed in [132]. Here, by using the formulation of [132], the Jaccard index is calculated for two different sets of clustering labels for the same node set by

$$J = \frac{n_{11}}{n_{11} + n_{10} + n_{01}} \tag{5.3}$$

where, n_{11} denotes the nodes which cluster together in both divisions, and n_{01} and n_{10} denote the nodes which cluster together in only one division.

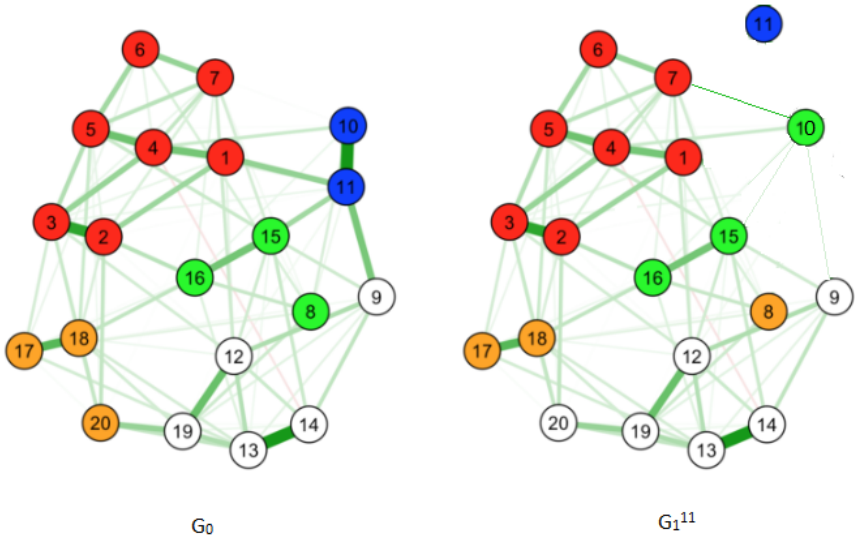

Figure 5.3: Example of Jaccard index network comparison measure. Modified from https://psych-networks.com/

Figure 5.3 shows an example of this clustering comparison. Here, in the first network, G_0, the nodes are numbered from 1 to 20. There are five clusters specified with colors red, orange, green, blue, and white. If, for a first iteration of 1, and for a potential node choice of 11, the second network G_1^{11} is created, and the clusters are formed, then the number of nodes which changed clusters would be 3. These nodes are 8, 10, and 20. So, n_{11} is 16, n_{01} and n_{10} is 3 without taking node 11 into account. As mentioned above, Jaccard index can only be calculated if the clusters of two divisions are matched. To match the labels, we start by finding the cluster in G_0 and the cluster in G_1^{11} that have the most nodes in common. Here, in the figure, first red clusters would match. The nodes in these clusters will have the same label. Then, a second step will continue the labeling to calculate the Jaccard index.

For Jaccard index network measure, at each iteration, the networks with the lowest Jaccard index values when compared to G_0 are the ones chosen as the most changed networks.

Edges Lost

The edges lost network comparison criterion compares the number of edges which are common for network G_0 and for a newly created network. For a list of nodes m chosen from a potential node list of lists B_i,

$E_{removed} = E\{S\}$

$E_{remaining} = E_0 - E\{S\}$

$E_{common} = E_0 \cap E_S$

$E_{lost} = E_{remaining} - E_{common}$

where $E\{S\}$ denotes the edge subset of S nodes. E_0 is the edge list of G_0. $E_{remaining}$ denotes the hypothetical edge set where only the edges which connects the nodes of S is lost. E_{common} is the set of edges which is common for both G_0 and G_S. The edges lost, E_{lost}, are the edges which were in the set $E_{remaining}$ but are not present in the new network G_S.

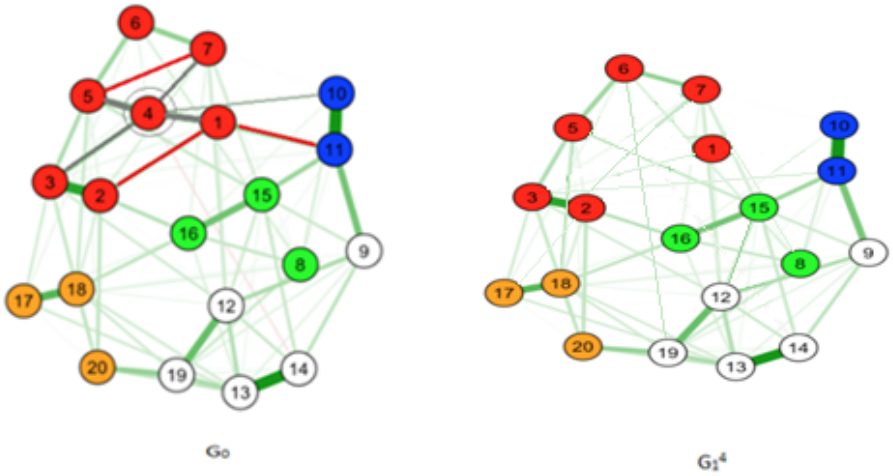

Figure 5.4: Example of edges lost network comparison measure. Modified from https://psych-networks.com/

Figure 5.4 shows an example of edges lost network comparison criterion. Here, the green lines depict edges. The thicker the line, the higher the edge weight. For a first iteration of a list of nodes $S = \{4\}$ is chosen to be eliminated. The edge list $E\{S\}$ includes the gray edges which connects node 4 to nodes 1, 3, 5, 7 and 11. The set $E_{remaining}$ includes all other edges of E_0 which are not in $E\{S\}$. So, after creating the network G_1^4, the edges are compared and the edges shown in red were not present. Then, the set E_{lost} would include those which are dropped, and the value for edges lost criterion would be 3. For this network comparison criterion, the networks with the

highest number of edges lost values are chosen to be the most changed network compared to G_0.

Nodes Lost

The nodes lost network comparison criterion compares the number of nodes which are common for network G_0 and for a newly created network. Similarly, for a list of node set S chosen from a potential node list of lists B_i,

$V_{remaining} = V_0$ - S

$V_{common} = V_0 \cap V_S$

$V_{lost} = V_{remaining} - V_{common}$

where $V_{remaining}$ is the node subset of V_0 without S. V_{common} is the node set common for both G_0 and G_S, and V_{lost} denotes the nodes which were present in $V_{remaining}$, but lost after the new network creation step.

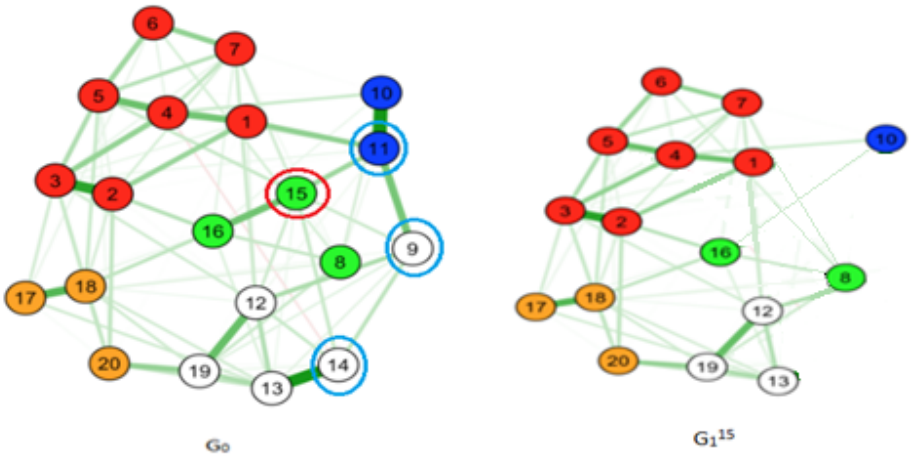

Figure 5.5: Example of nodes lost network comparison measure. Modified from https://psych-networks.com/

Figure 5.5 shows an example of nodes lost network comparison criterion. If for a first iteration and a chosen potential node 15, a new network G_1^{15} is created, and in this new network, nodes 9, 11, and 14 were absent, then the value of nodes lost criterion would be 3. For this network comparison criterion, the networks with the highest number of nodes lost values are chosen to be the most changed network compared to G_0.

5.2.5 Final Criteria

The final outcome of the algorithm is the sequences of nodes for which the algorithm stops when any one of the below final criteria is met. Further investigation by experimenents or literature review would identify those sequences with which the network would be disrupted or break down according to the definitions and set thresholds. The final criteria we check at every iteration are Jaccard index, nodes lost, edges lost, scale-free topology, and other user defined node ID list comparisons. The algorithm may also be stopped if it reaches a certain iteration number without hitting any final criteria thresholds on its way.

The final criteria is used to catch the cases where all of the critical nodes are removed, and the network is too different from G_0. After new networks $\{G_S\}$ are created, we compare each G_S with G_0 and decide whether G_S is sufficiently disrupted to be assumed not to have the same topology or function as the initial network. We check each of the final criteria for each network, and if the network fails for any one of them, it is assumed to be distrupted enough. The node subset S is added to the final outcome choices of impact node sets for further investigation. The algorithm stops when there is no other S in the beam.

Jaccard Index

One of the attributes we check at the final criteria step is the Jaccard index. As described earlier, the Jaccard index is the measure of to what extend the clusters of two networks with the same node list resemble each other. The greater the index is, the more similar the networks are in terms of clustering. The network function is related to this criterion. Therefore, if the Jaccard index between G_0 and any G_S is too low, the potential node impact sequence that leads to that result could be assumed sufficient to disrupt the network entirely when removed.

Nodes Lost and Edges Lost

When more than a certain number of nodes are lost, the possibility of the network functioning as before is low. Even if the nodes remain, if more than a certain number of interactions between nodes are lost, then those nodes become mostly disconnected and the continued functioning of the network is not possible. The nodes lost and edges lost criteria of the algorithm is calculated as follows given in section 5.2.4.

Scale-free Topology

Here, we investigated whether the newly created networks still have a structure which can be approximated as scale-free. As mentioned in section 2.2, most real-life complex networks have scale-free structures that allow them to have hub nodes and clusters formed mostly around these hub nodes that support efficient functionality. Mathematically, scale-free networks are represented as power laws. The normality tests can be used to determine to what extent an empirical result fits to a power-law distribution, which indicates whether a network can be modeled as scale-free. Here, we use the KS-test which measures the distance between an empirical data distribution and the power law distribution for all values above a certain threshold, such that only the data at the tail is included. It is possible to fit both the distance and the threshold value at the same time, although there is a risk of over fitting at high values. Therefore, we fitted only the distance by specifying the optimum threshold, the x_{min} value of the KS-test.

According to the above definitions, it is expected that a network with a KS-test p-value equal to or greater than a 0.05 statistical significance level agrees with the null hypothesis in this case, that it is a sampling from a power-law distribution. In such cases, the network has a high probability of showing scale-free characteristics. Here, we investigated the p-values of the node degree data of every G_S. We started with the highest node degree of the network as the first x_{min} value, and decreased it by increments of 1. As long as the p-value of the KS-test was equal to or greater than 0.05, we assumed that the network showed a scale-free structure.

We also checked the confidence levels of the model fits. We assumed that if the number of nodes equal to or greater than the threshold node degree was 50 or more, the fit and test results were reliable. Otherwise, we assumed the structure was unreliable with insufficient number of hub nodes in the network. The cases in which the KS-test p-value was less than 0.05 from the first threshold were assumed to result in networks that lacked scale-free characteristics.

Below is the procedure for the function IsScaleFree. The algorithm stops if IsScaleFree returns "FALSE". The user can also specify whether it also stops when the result is "UNRELIABLE". This must be determined according to the initial node size and whether or not it is safe to assume

having fewer than 50 hub nodes would distrupt the network.

Algorithm 1: Procedure for function isScaleFree

```
Function isScaleFree(G_S):
    Find nodeDegree of G_S
    initialize x_min as maximum of nodeDegree
    Fit nodeDegree and x_min to a power law distribution
    Find pValue of KS-test for the fit
    numberOfNodes = length of nodeDegree ≥ x_min
    if (pValue ≥ 0.05 and numberOfNodes ≥ 50) then
        return "True"
    if (pValue ≥ 0.05 and numberOfNodes <50) then
        while numberOfNodes < 50 do
            do xminLast = xmin
            numberOfNodesLast = numberOfNodes
            pValueLast = pValue
            xmin = xminLast - 1
            Fit nodeDegree and xmin to a power law distribution
            Find pValue of KS-test for the fit
            numberOfNodes = length of nodeDegree ≥ xmin
            if (pValue < 0.05) then
                break
        xmin = xminLast
        numberOfNodesLast = numberOfNodes
        pValue = pValueLast
        if (pValue ≥ 0.05 and numberOfNodes ≥) 50 then
            return "True"
        else
            (pValue ≥ 0.05 and numberOfNodes < 50)
            return "Unreliable"
    if (pValue < 0.05) then
        return "False"
End Function
```

Node ID Specific Final Criteria

The algorithm keeps track of nodes that are known to affect the network in real life if any. There may be groups of nodes known to be elements of the real network or known to have relations to the

nodes of the real network. The algorithm uses these node sets as one of the final criteria to check. It calculates how many of these known nodes are dropped at every iteration for each G_S. Calculation is done the same way as for nodes lost, and if the number of lost known network-related nodes goes above a certain threshold level, then the algorithm may be stopped for that network. The conclusion is that there are not enough nodes related to the real network to sustain functionality. The node subset, S, that is used to generate the network is added to the final outcome choices for further investigation. The algorithm continues with the next iteration for the rest of the networks which are not chosen.

The algorithm can also keep track of other network characteristics to investigate whether their trends change at each iteration. For every G_S edge density, network diameter, and modularity are calculated. We also calculate node betweenness of V_i which can be included in the potential node impact list choices. Node betweenness can indicate a node's potential of being a bridge between clusters.

Another criterion of the algorithm looks at the trend of how nodes with fewer edges behave during each iteration. Information in hierarchical networks travels from hub nodes to lower level nodes and no matter the path length, network response is possible as long as the information can reach these lower level nodes. However, if these lower level nodes are sufficiently changed and lost, functioning is disrupted. We defined fewer-edge-nodes as follows

$V_{fewerEdges}$ = V $\leq$ median(Node degree(G_0)) + IQR(Node degree(G_0)).

5.3 User Definitions and Thresholds

The whole set n of input nodes is defined as a general network node set at all times to make comparisons possible. When calculating the Jaccard index for network comparison, nodes with no edges are clustered separately. To exclude nodes that have been discarded or removed, we set a threshold to focus on clusters with large number of nodes. In chapter 6, for simulated data with an input data set of 100 nodes each, clusters that include 10 or more nodes are called modules and are the only ones that are investigated. In chapter 7, for biological data with an input data set of over 3000 nodes, G_0 clusters that include 100 or more nodes are called modules and are the only ones that are investigated. The networks with lowest Jaccard index compared to these modules are chosen as the list of G_S that are most affected. At this point, 50 networks with the lowest Jaccard index values are chosen for the beam width at the pruning step at every run.

It is also user defined how the final criteria is used. Final criteria thresholds change the algorithm's final outcome. Right now, for the Jaccard index, which takes values between 0 to 1,

the threshold is 0.01. If any G_S Jaccard index is lower than that, the algorithm returns the list of nodes taken out to reach of that specific network which is assumed to be broken down completely. As for nodes lost and edgesLost criteria, the threshold is 50%. If any network loses more than half of its nodes or edges compared to G_0, it will be assumed that it cannot function in the same way as G_0. For the user-defined comparisons, where a node list with known relationships to the network is introduced, our threshold is 100 %. For scale-free structure decisions, it might only include the networks where the function IsScaleFree gives "FALSE", or both "FALSE" and "UNRELIABLE". For the maximum number of iterations, the user-specific threshold depends on the size of the network and on how many impact nodes need to be investigated. For our simulation case with 100 nodes, the threshold chosen was 10. For larger networks where more nodes are expected to be taken out before networks show a change in functionality or structure this number should be greater.

According to these thresholds and definitions, the FinalCriteria function is calculated as in Algorithm 2.

Algorithm 2: Procedure for function finalCriteria

Function `finalCriteria`($G_S(V_S, E_S)$, $G_0(V_0, E_0)$)**:**

Calculate E_{lost}, N_{lost} by using the formulation of section 5.2.4

Check isScaleFree(G_S)

Calculate JaccardIndex(G_S)

Compare G_S to G_0:

if $N_{lost} \geq (V_0)/2$ **then**
  return "True"

if $E_{lost} \geq (E_0)/2$ **then**
  return "True"

if $V_{related}$ *of* $G_S == V_{related}$ *of* G_0 **then**
  return "True"

if $V_{fewerEdges}$ *of* $G_S \geq V_{fewerEdges}$ *of* G_0 **then**
  return "True"

if *isScaleFree*(G_S) $==$ *"FALSE"* **then**
  return "True"

if *JaccardIndex*(G_S) ≤ 0.01 **then**
  return "True"

else
  return "False"

End Function

5.4 Impact Node Finder Algorithm Procedure

Inpact node finder (INF) algorithm is a beam search algorithm that only keeps a set of contender impact node sets at each iteration. The number of contenders that are chosen at each iteration is predetermined with the beam width. Here, G_0 is the initial network created by including the n number of nodes in the node list V_0. S refers to a subset of nodes that is being considered as the most critical nodes. G_S is the network created by removing nodes S from the initial node set V_0 and re-computing the correlations for edge weights. The beam at each iteration i, B_i, includes all the chosen set of S's. These are the contenders being considered after the i^{th} step of the algorithm. Each S in B_i will have i nodes in it.

Algorithm 3: Procedure for function impactNodeFinder

Function `impactNodeFinder(`*dataMatrix*`):`
- foundNodeLists = {{}}
- Create $G_0(V_0, E_0)$ from dataMatrix(n x p)
- Initialize B_0 by using G_0 **for** $i = 1...$ **do**
  - B_i = {{}}
  - **for** *each* S *in* B_{i-1} **do**
    - Find likely next nodes to add to S by using the formulation in section 5.2.3
    - Add them into B_i (remove duplicates)
  - **for** *each* S *in* B_i **do**
    - set $V_S = V_0$ - S
    - Create $G_S(V_S, E_S)$
    - Compute network difference between G_S and G_0 by using the formulation in section 5.2.4
  - Prune B_i to keep the most different graphs from G_0
  - Check FinalCriteria
  - If any contenders in B_i meets the stopping criteria, add it to foundNodeLists.

End Function

The procedure, as in Algorithm 3, starts with an initial input data matrix of n columns, p rows. Each row corresponds to a unique node. The first regularized partial correlation network created without any change in input data is assumed to be the initial network with graph representation $G_0(V_0, E_0)$.

The first potential node set, B_1, is just one column vector that includes only the top 5^{th} percentile node degree and node strength value nodes of G_0. For each row in this column, the chosen node is eliminated from the input node list. A new network is created with graph representation $G_S(V_S, E_S)$. Network characteristics are calculated and final criteria are checked.

For the networks that are not finalized, the network comparison step according to a chosen measure is calculated for G_S and G_0. After all networks are run, those networks with the 50 most changed values are chosen for a beam width of 50. The specific nodes taken out to give these chosen most changed new networks are the first set of potential impact nodes of the potential impact node sequences. The procedure redefines the potential node list to include only these networks, and the corresponding rows in the potential node list are the only ones that remain. These are the candidates of the beam.

A new potential node list is created using these networks and their node lists and their input matrices. Thus, the second iteration of the list of potential impact node pairs contain the combination of nodes that show the highest potential of being critical among G_S and of nodes chosen from the most changed new networks based on degree and strength. The input node list decreases by two at this point. One is already chosen from the first iteration to decide the most changed new networks, and the other comes from the new potential impact node list. A new round of network creation and comparison steps will result in the possible third nodes of the selected sequences and the cycle repeats. Every iteration adds one new member to the node sets.

As a final outcome, the algorithm keeps track of the nodes taken out for every G_S. If at any point the FinalCriteria function gives "TRUE", the specific row from potential node set of beam B_i is added to the found node lists. Our algorithm gives a set of alternatives for impact node subsets instead of a single final node set. In a complex network, such as a gene regulatory network, there can be more than one way to impact the network. For a gene regulatory network, there can be gene groups with different functions that would disrupt the network if any of them are eliminated. Our algorithm searches these different node subsets as final result candidates for further investigation.

Figure 5.6 shows an instance of the INF algorithm, defined above in Algorithm 3, when the input data is from a biological source where the nodes are genes, and the samples show the gene expression values. The figure is an example of how we used the INF algorithm in chapters 6 and 7. The network is a square matrix with inputs of regularized partial correlations as weights. The first potential node list is chosen from G_0. The input node lists are modified by eliminating the corresponding chosen nodes. New networks are created by using these new input data. These networks are compared to the initial network G_0. The most different ones are chosen by using a network comparison metric. Then, a second list of potential node lists are created by using these most different networks. Since, every first iteration network was missing one gene from the input list, the newly chosen potential second genes form the second set. By combining with the corresponding first sets of nodes with respect to the specific network, the new potential node list is formed and the procedure repeats to find the third members.

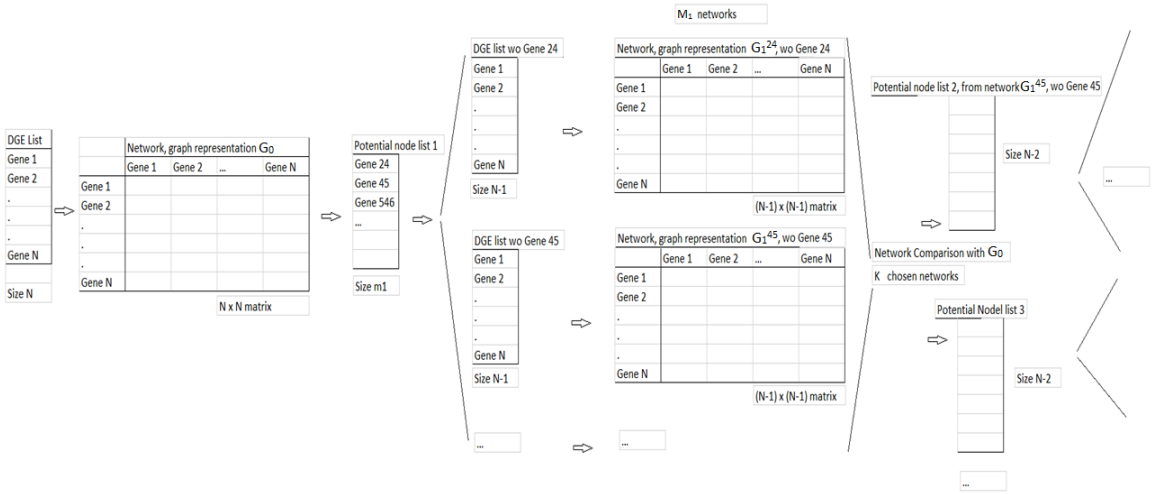

Figure 5.6: Impact node finder algorithm example workflow

5.5 Parallelizing the Algorithm

The most expensive part of the algorithm is the network creation step for each of the chosen S sets. For a set of n number of nodes, the pairwise edge weight calculations grow by the order of n^2. For a chosen network creation algorithm, each calculation may take minutes to hours long. So, for a biological data of thousands of genes, the time of a series calculation of chosen networks for each S in B_i is exponential. Since, the calculation of each network is independent from one another, this part is easily paralellized. The throughput is directly proportional to the number of nodes in a cluster used for parallelization. The only limit for this case is the temporary memory allocation for the network edge lists for each network created in a beam for a chosen iteration. The algorithm can also be paralelized to compute the impact node lists for different input data sets at the same time.

Chapter 6

Application to Simulated Data

For any estimation algorithm, it is of great value to demonstrate that the algorithm performs satisfactorily on simulated data. This allows us to do a very controlled experiment to test our algorithm. Simulated data are generated with a model of the phenomenon of interest with particular parameters. The estimation algorithm attempts to identify those parameters and its success is measured by how accurate the estimated parameters are with respect to the model parameters used in simulation. For impact node finding, our goal is to use an expression data generator where we implant a set of nodes that drive the functionality with much higher impact than the rest, which we will refer to in this chapter as the active nodes. When we apply our algorithm to the generated data, we expect to see a large intersection between the model impact nodes and the estimated set of nodes of high impact.

The aim of this chapter is to find whether our algorithm can detect transcriptionally active nodes in a network with a statistical significance higher than random choices. We tested three different network comparison measures we defined in chapter 5 and compare their effectiveness over the number of active nodes they found. We also examined the specific variables that final criteria choices set thresholds to. We investigated how these variables change and how close they get to the thresholds as the iterations increase to see the effectiveness of the chosen final criteria. In the sections below, we first review the generated data, then give the results of the initial network and then explain our findings with the algorithm runs.

6.1 Simulation Data and Analysis

We used the Java application, SynTReN [160], to create our simulated gene expression data. SynTReN application is described in section 3.5.4. This application is used for benchmarking new algorithms as done in Meyer et al [103] and Maier et al [95]. We generated 100 different datasets with the same initial parameters, but with different seed numbers. We generated seed numbers by

using R version 3.6.1, a random seed of 101 and a random sampling from 1 to 10,000. Figure 6.1

```
8009 2873 3281 5562 5471 7997 3004 8507 5920 4617
2531 2997 4667 9137 4653 1528 4213 1322  701 5437
9461 4693 1422 7048 5192 2215 1226 5786 4233 3936
7205 6111 1251 3610 8713 4372 6082 4673 1965 3100
1692 1694  668 4780 8815 1202 5666 8905 4517 4078
8194 1529 9298 7183 5859 5323 2441 1514 9607 2612
4305 9083 7912 8504  682 5530 7428 9668 9171 5184
3292  698 3902 2935 6041 9868 7765 3186 3879 1573
4149 8230 2259 9815 7790 6596 3606 1379 8588 7659
3951 3575 2836 2747 6457  141 1016 4623  806   14
```

Figure 6.1: Random seeds for simulated data generation

shows the random seed numbers generated. All our data is reproducible by using these numbers as the random seed parameters for the simulations.

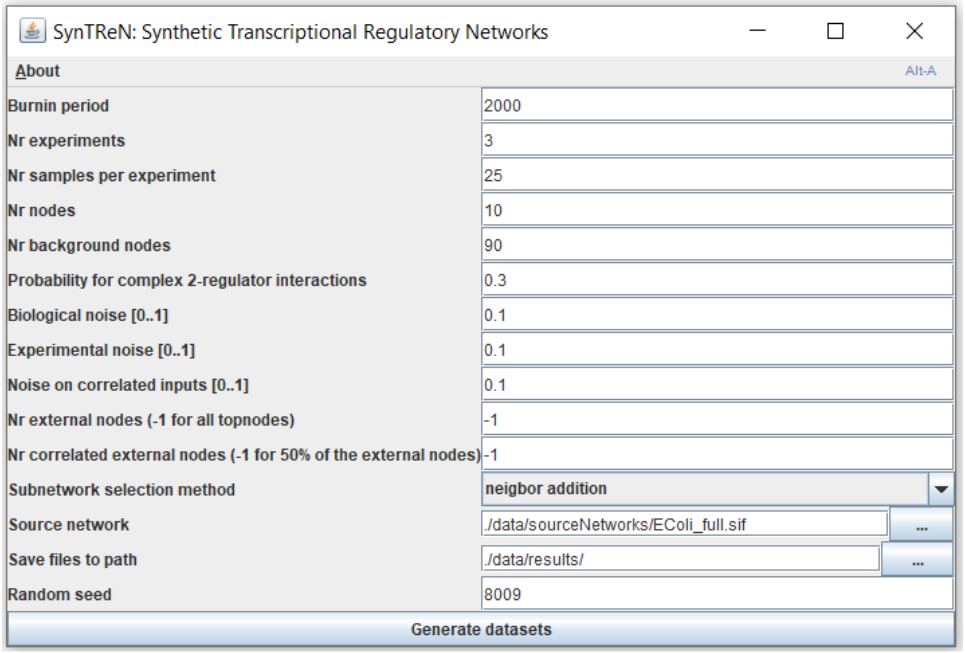

Figure 6.2: SynTReN application parameters

Figure 6.2 shows the initial parameters chosen for all of the data sets created. According to Maier et al [95], the low noise values were critical to test an algorithm's correctness, even though

the generated data will be less likely to show real life gene expression variability. We used default values for all the noise specifications. The total number of genes for each data set is 100. Each data set includes 10 transcritionally active nodes, and 90 background nodes. As a source network E.coli was the default, and subnetwork selection method was chosen as neighbor addition. The "Nr experiments" value in the figure specifies the number of stages in a time-dependent biological experiment. "Nr of samples per experiment" value specifies the number of replicates for each stage in an experiment. We chose 3 stages and 25 samples for each stage to mimic our biological data of initiation of puberty.

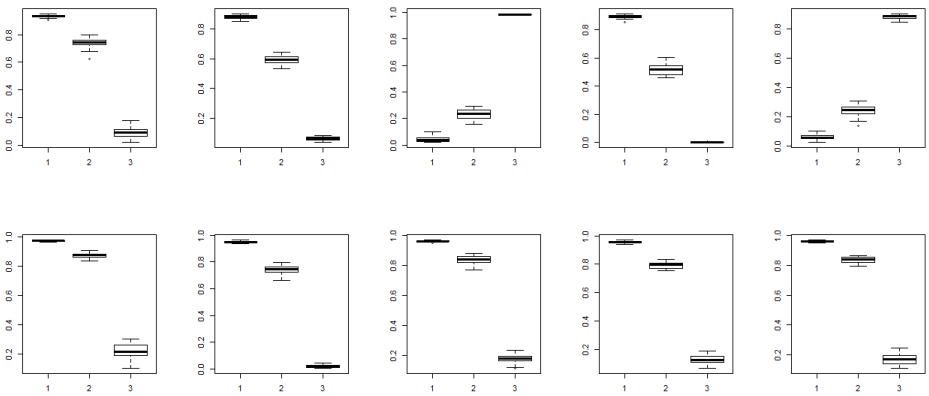

Figure 6.3: Transcriptionally active nodes (node 0-9)

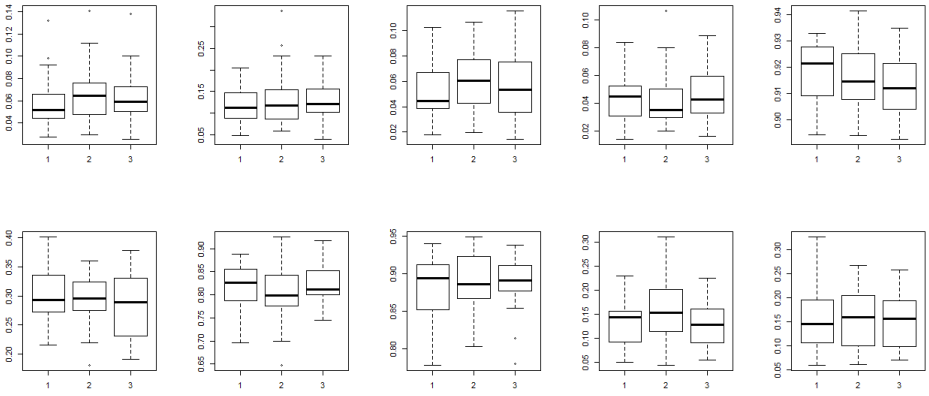

Figure 6.4: Background nodes (node 10-19)

Figures 6.3 and 6.4 show example boxplots generated from one of the data sets to show how the expression values change over the course of experiment stages. X-axis shows experimental

numbers and y-axis shows simulated gene expression levels. Each boxplot is the representation of the distribution over 25 samples for the same stage. Figure 6.3 shows the trend of transcriptionally active nodes. Figure 6.4 displays 10 of the background nodes to show the general trend of non active gene groups. The spread among the samples generated for the background nodes is much higher than the spread of the active nodes. Active nodes show significant changes over the course of the three stages, whereas the background nodes show little difference even though the expression values differ from gene to gene. In the rest of the analysis part, in section 6.1.1 we will review the results for initial 100 networks where we use all 100 genes and construct initial networks for all data sets without any node eliminations. We first will analyze the characteristics of these initial networks and show results of active node frequency before we use our algorithm. In section 6.1.2, we will review the specific parameters for the simulated data, and give examples of iteration results to show how the algorithm works and affects the initial networks.

6.1.1 Initial Networks

We used the parcor [80] package to create the initial networks. This package uses l_1-regularized partial correlations to calculate edge weights as explained in section 2.3.2. The optimum network is chosen by using 10 fold cross validation. The input data for each network includes 100 genes as nodes and 25 gene expressions for each of the 3 different stages to calculate edge weights. By using the same procedure, including all 100 genes for each, we generated 100 initial networks from 100 data sets.

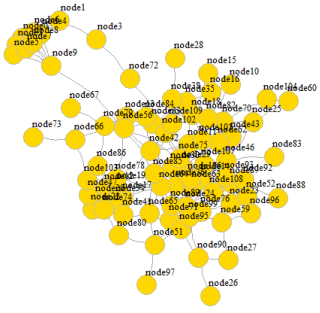

Figure 6.5: Simulated gene expression data regularized partial correlation network

Figure 6.5 shows an example of a network image. The network in Figure 6.5 includes 84 connected nodes according to their calculated partial correlation values with a lasso estimate. 16

genes did not get connected. In all our data sets, node numbers 0 to 9 represent the active nodes, whereas the others represent background nodes. As it is shown in the figure, all active nodes were connected to one another with edges. They clustered together at the top left corner of the image. Node 9 had the highest number of connections among all active nodes. Node 9 and node 3 also made connections to other background nodes which shows as two edges from the top left cluster to the rest of the network in the image. These connections make these nodes bridge nodes between active nodes and the background nodes. So, these nodes can be considered as candidates of active nodes according to some chosen metric.

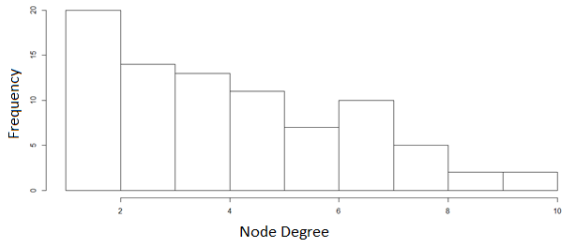

Figure 6.6: Node degree histogram of the simulated gene network

Figure 6.6 displays the histogram of node degree for the same example network. X-axis shows the number of edges each node has, and y-axis shows the number of nodes for each node degree. There are 20 genes with node degree 2 or less. The last two bins corresponds to node degree values 9 and 10. There were each 2 nodes that have 9 or 10 edges. The highest number of edges which corresponds to the last bin of the histogram belongs to two background nodes, node 14, node 17. For a threshold of top 5^{th} percentile node degree values, the above network included these two nodes with 10 edges, two background nodes with 9 edges, two active nodes, node 9 and node 6, with 8 and 7 edges, and two other background nodes. If we were trying to find active nodes by considering the top 10 highest node degree values, we would find 2 active nodes. So, for this network we compare our results to two active nodes found.

The node degree figure is also useful for estimating what the distribution looks like. The higher frequency of lower number edge counts and the lower frequencies of highly connected nodes as edge counts increase is what would be expected in a clustered scale-free network histogram. We fitted the node degree data using a power-law model, and used KS-test to check the scale-free condition of the network. For a scale-free network, the KS-test statistic should be greater than 0.05. The closer this value approaches to 1, the better the fit to a power-law. The KS-test distance should be small. The fit values for this simulated gene network were as follows. The KS-statistic was 0.99, the distance to the fit value was 0.06 and power-law exponent was 7.23 with a minimum degree

of fit value of 7. The KS-test always checks the fit on the tail of the distribution. As long as the trend is correct at the tail, the network is assumed to be scale-free. In this case, the distribution is fitted to the power-law for nodes with 7 edges or more. There were a total of 19 nodes that satisfy this criterion, and these were taken into account to calculate the test results.

The edge density of this sample network was 0.03. This lower edge density is also a sign of a scale-free network, indicating there were many nodes with few edges and very few with more edges.

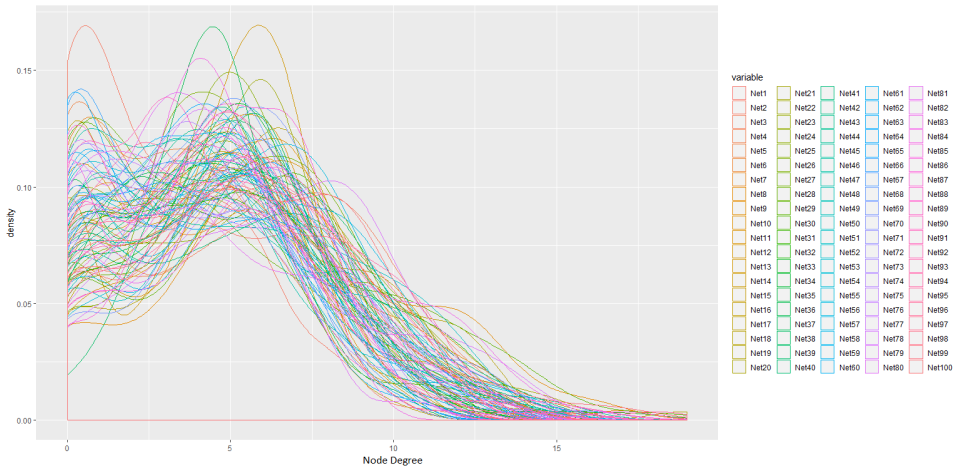

Figure 6.7: Node degree density plot of 100 first networks

Figure 6.7 shows the density plot for each of 100 created initial networks. Each color represents a different initial network coming from a different data set. The x-axis represents node degree and the y-axis represents the corresponding density values. These networks are the original networks that we compare our results to. The input data for these 100 initial networks included 100 genes each without any node eliminations. All networks showed scale-free characteristics with a KS-test statistic of 0.05 and above. Even though the curves did not show a perfect fit to a power law for nodes with lower node degree values, the drop at the higher node degree values for all the networks were suffcient to accept scale-free topology.

Figure 6.8 shows the results of how many active nodes are chosen by using the initial networks. We are using the results of choosing active nodes from the initial networks as our baseline to compare to our algorithm results. Here, as a simpler criterion for active node selection, we used the node degree values to rank which gene would be most likely chosen as a potential active gene to take out from the initial network as explained in the example network above. The violin plot shows the distribution over 100 initial networks with specified seed numbers above. We first ranked

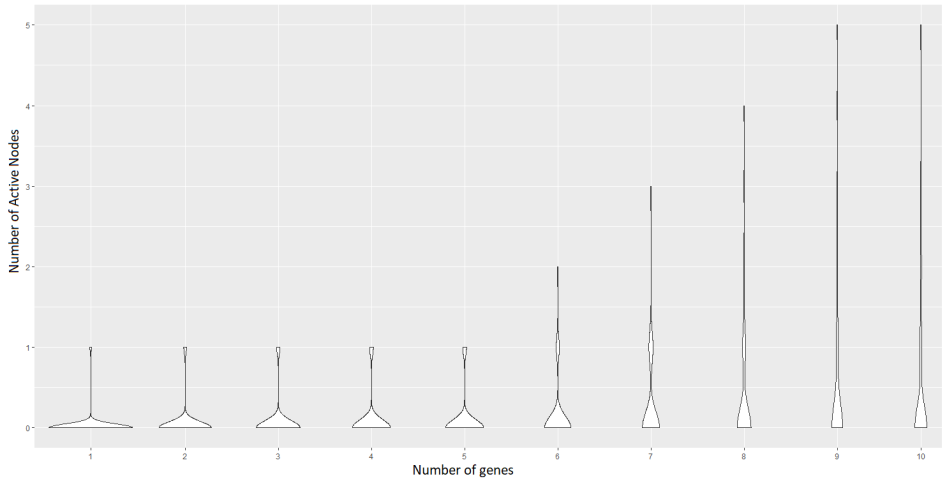

Figure 6.8: Violin plot of number of active nodes in first networks

all genes according to their node degree values, and then reordered the genes with the same node degree values alphabetically so that nodes 0 to 9, which corresponds to the active nodes, so that they would be chosen first. The violin plot shows that for genes chosen as active nodes from 1 to 10, the node degree values for each network usually gives 0 and very rarely 1 for the first 5 genes out. For a set of genes chosen to be active nodes with a size of 5 to 10 genes from the networks, the majority still found 0 active nodes. Some found 1 or more active nodes. There were also a couple of networks with 5 active nodes found in a set of 10 chosen genes.

6.1.2 Algorithm Parameters

Here, we will go over the parameters we used for our INF algorithm, and give examples of the iteration results to explain how the algorithm works and how the initial networks get affected. We used the top 5% of total unweighted edges and total weighted edges together to include both the node degree and node strength. The first iteration potential node lists S included nodes with approximately eight or more edges. A total of 7 to 11 nodes were in the first potential active node lists. We used clustering values of 10 nodes and up as a network similarity comparison measure for the Jaccard index calculation. We used a beam width of 50. The lowest 50 networks after each iteration is chosen as the potentially most disruptive networks for the Jaccard index. For the edges lost and nodes lost comparison measures, we used networks with the highest 50 values. The total number of networks created for each iteration for each of 100 data sets was around 500. The candidates of the next iteration to add to the beam of gene subsets to be excluded come from these networks.

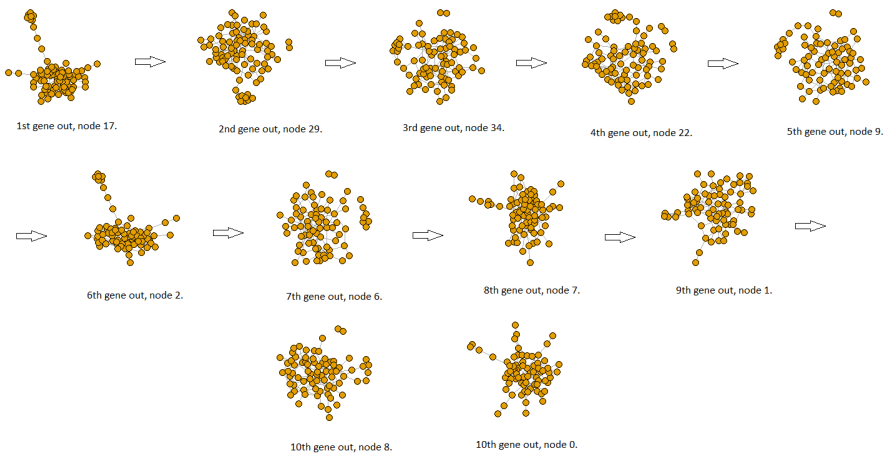

Figure 6.9: Network image samples for different iterations

Figure 6.9 shows how a representation of the sample network described above changes as genes are eliminated from the input list one by one. This is one sequence of many due in the search tree of beam search. In Figure 6.9, first image, the top cluster shows the transcriptionally active genes. Even when only one gene is eliminated, there are changes in the network structure. At first, both node 9 and node 3 were connected to the rest of the network genes. When node 17 is out, node 9 loses its connection to the background genes, even though node 17 was not originally connected to node 9. When node 29 is taken out, node 2 edges increase by one, and it becomes a bridge to the background nodes. There are also disconnected pairs which are dropped, or reconnected as the iterations increase. At iteration 10, the two choices of eliminating node 0 or 8 gives completely different connections between active nodes and background nodes. The median number of nodes dropped and nodes gained for each iteration was 5.

Figure 6.10 shows an example of how node degree and node strength changes in four iterations for active node 2 from the same network. The first plot shows the node degree, and the second is the node strength. The red line shows the iterations and which nodes are out until node 2 is chosen as its node degree becomes greater than the threshold. The green line shows another iteration and gene list where node 2 is chosen when its node strength becomes greater than the threshold. The node degree and node strength thresholds change as the iterations increase. The active nodes are the ones affected the most when nodes are eliminated. So, an active node may not be the first one to be chosen, but as the number of iterations increase, the active nodes change their node degree and node strength values and become a potential impact node to be eliminated.

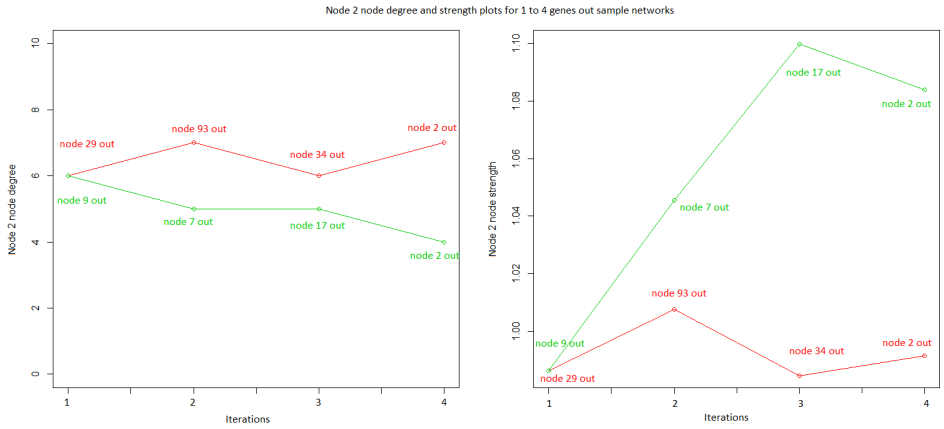

Figure 6.10: Node degree and strength changes vs iterations

6.2 Algorithm Results

We used a parallelized version of our algorithm to run 100 input data sets at the same time and to create around 500 networks for each iteration faster than a series run. Here, the results include 100 data sets for each of Jaccard index, edges lost and nodes lost network comparison measures. The results are gathered until 10 genes out iteration. Below we will discuss our results for different network comparison measures and for different final criteria selections.

6.2.1 Network Comparison Measure Results

We used three different network comparison measures to choose our 50 most changed networks at each iteration as mentioned before. Jaccard index is calculated to measure how the clustering changes for the new networks at each iteration. Edges lost and nodes lost measures compare the initial network edges or initial network nodes missing in the new networks. Here, we are going to go over the distribution results for each of these measures among the input data sets used.

Jaccard Index Network Comparison Measure Results

The Jaccard index measure was the most successful among the three measures for identifying transcriptionally active nodes. Figure 6.11 shows the median number of active nodes taken out per iteration. For each of the 10 iterations, and across the 100 input data sets, the median value of 50 most changed network according to their Jaccard index results are taken. This violin plot shows the distribution of these median values of active nodes found. The widest part of each violin element shows the largest number of medians for that iteration. As seen in the figure, the

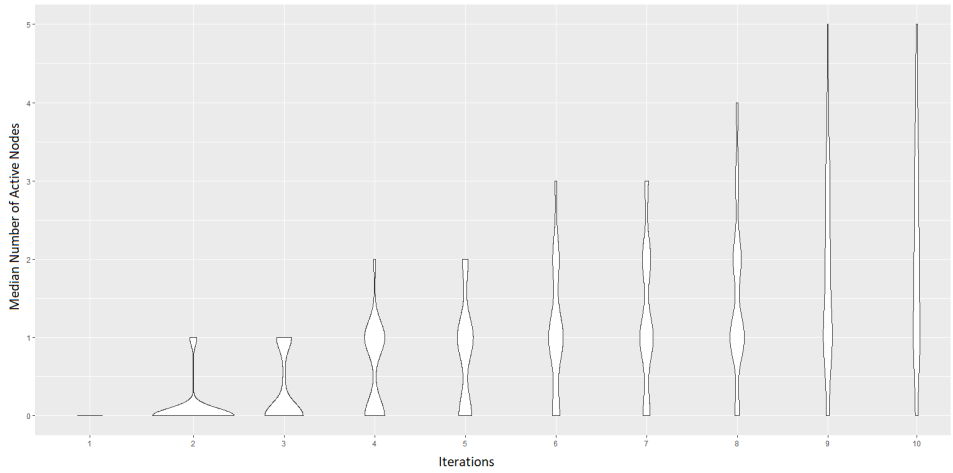

Figure 6.11: Jaccard index measure, median number of active nodes taken out in every iteration

number of most median active nodes found starts at 0 at iteration 1 gene out. For the second gene out iteration, most of the median numbers are still 0, but some input data included median active nodes found to be 1. As the iterations increase, the width of median active nodes found 0 decreases and the width of 1 and 2 increase. At the iteration of 10 genes out, the median value is around 2, with also some input data that found 3 median active nodes. This result shows a better performance for choosing active nodes when compared to the initial network active node distribution shown in figure 6.8.

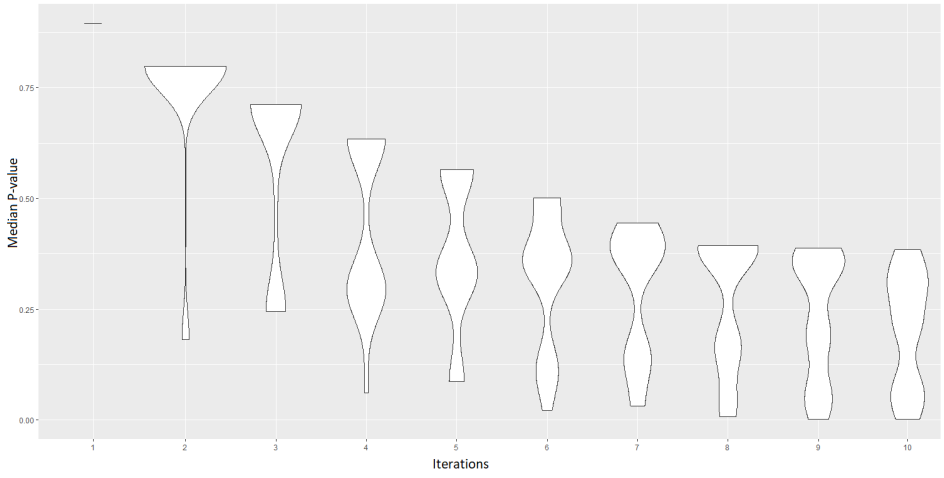

Figure 6.12: Jaccard index measure. Fisher's test p-value for number of active nodes taken out per iteration

In Figure 6.12 the median p-value spread for jaccard index median node calculation starts

at around 0.8. The smaller the p-value gets the more statistically significant the active node calculation gets. Any value below a p-value of 0.05 is assumed to be statistically significant. As the figure depicts, the decrease in p-values suggests that as the number of iterations increase the median number of active nodes found by using the algorithm have a higher probability to be significant compared to a random choice. The width of the violin element below a p-value of 0.05 is increasing, but still most of the input data sets stay above the line. This result suggests that an increased number of iterations might find results with more median values below the threshold of significance.

Edges Lost Network Comparison Measure Results

The next two measures, edges lost and nodes lost, were not as successful as the Jaccard index measure. Again here, the total number of data sets taken into account when generating the results is 100. The maximum iteration of this measure is also 10.

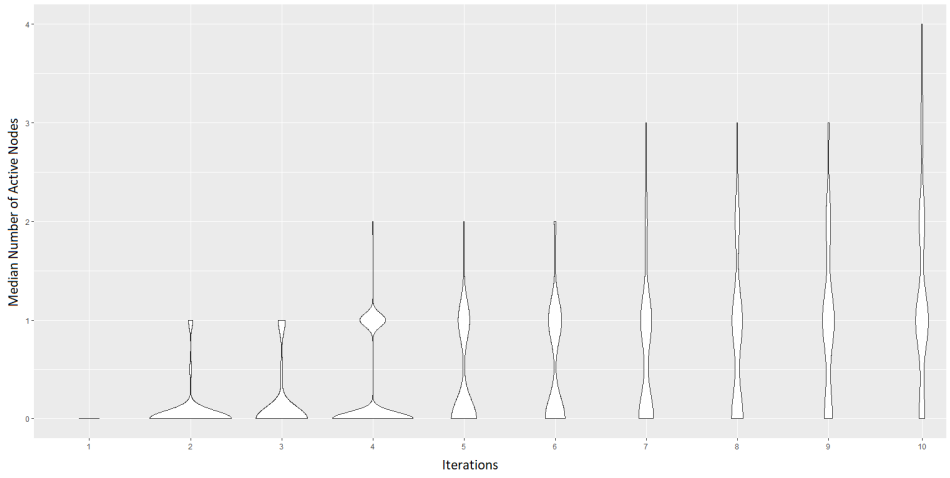

Figure 6.13: Edges lost measure, median number of active nodes taken out in every iteration

Figure 6.13 shows the median number of active nodes taken out per iteration. For each of 10 iterations, and across 100 data sets, the median value of 50 most changed network according to their edges lost results are taken. This violin plot shows the distribution of these median values of active nodes found. The widest part of each violin element shows the largest number of medians for that iteration as mentioned above. As the figure shows the number of most median active nodes found starts at 0 at iteration 1 gene out. For the second gene out iteration, most of the median numbers are still 0, but some input data included median active nodes found to be 1. As the iterations increase, the width of median active nodes found 0 decreases and the width of 1

increases. The edges lost network comparison measure did not result in median active nodes found as well as Jaccard index measure. It still had a better performance for choosing active nodes when compared to the initial network active node distribution shown in figure 6.8.

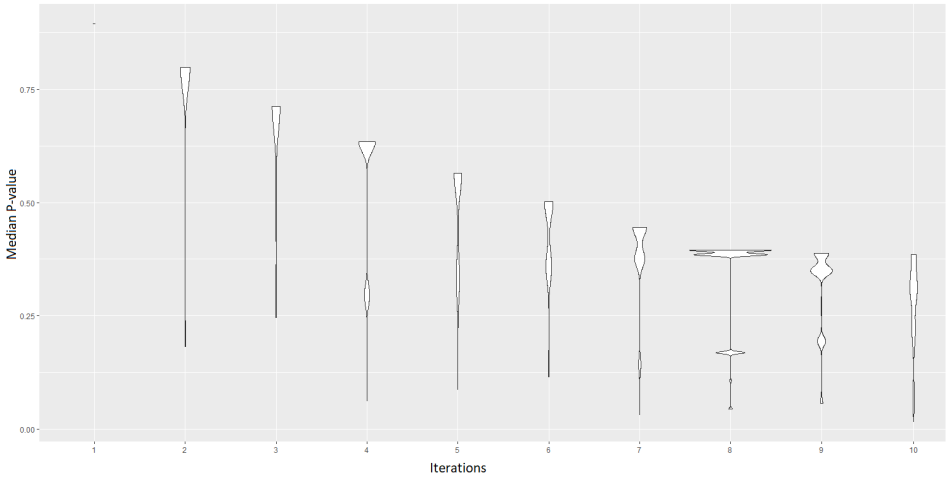

Figure 6.14: Edges lost measure. Fisher's test p-value for number of active nodes taken out per iteration

Figure 6.14 shows the median p-value for edges lost median active node calculation. There were very little median p-values lower than 0.05. But, the decrease shows that for iterations above 9 genes out, the p-value has the potential to be statistically significant.

Nodes Lost Network Comparison Measure Results

The nodes lost measure was the least successful one among all three network comparison measures. Here also, the total number of input data taken into account when generating the results is 100, and the maximum iteration is 10.

Figure 6.15 shows the median number of active nodes taken out per iteration. For each of 10 iterations, and across 100 data sets, the median value of 50 most changed network according to their nodes lost results are taken. This violin plot shows the distribution of these median values of active nodes found as explained above. As the figure shows the number of most median active nodes found starts at 0 at iteration 1 gene out. For the second gene out iteration, most of the median numbers are still 0, but some input data included median active nodes found to be 1. As the iterations increase, the width of median active nodes found 0 decreases and the width of 1 increases. The largest width for 10 genes out iteration was at 1. The nodes lost network comparison measure had a little better performance for choosing active nodes when compared to the initial network active node distribution shown in figure 6.8.

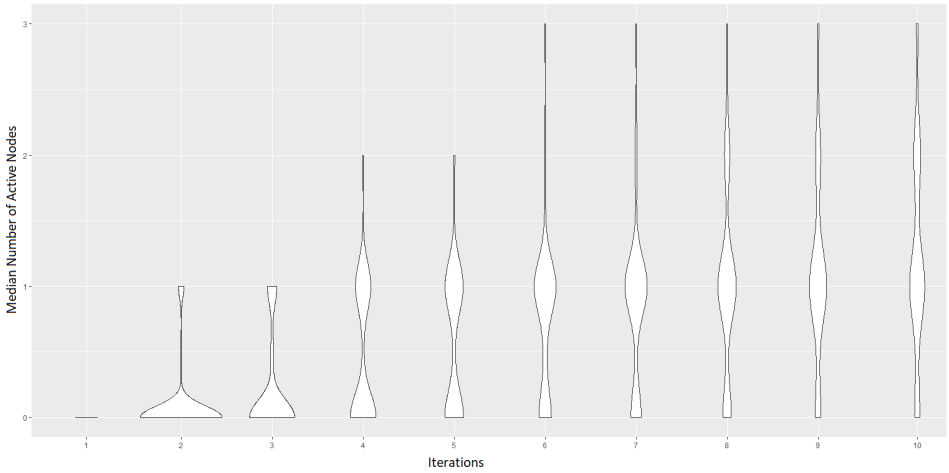

Figure 6.15: Nodes lost measure, median number of active nodes taken out in every iteration

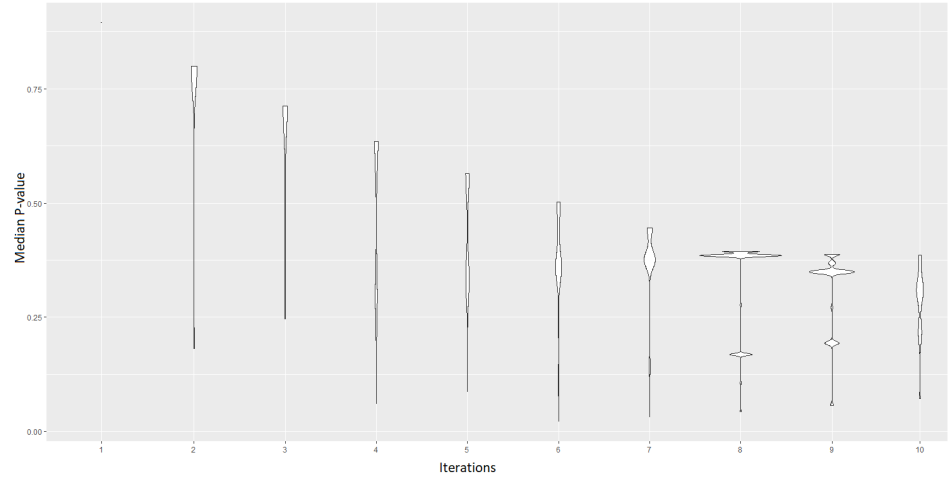

Figure 6.16: Nodes lost measure. Fisher's test p-value for number of active nodes taken out per iteration

Figure 6.16 shows the median p-value for nodes lost median node calculation. There were very little median p-values lower than 0.05.

6.2.2 Final Criteria Comparison

There were no networks in any of the input data sets that reached the final criteria thresholds when we stopped our algorithm at iteration 10. We will discuss the impact of stopping earlier that hitting a final criteria in section 6.3. Here, we investigate the trends of four different variables used in final criteria. Edges lost and nodes lost thresholds were 50% of the original network in comparison. The jaccard index final criterion required a value of 0.01 or lower. The scale-free criterion checks the KS-test statistic for a power-law model fit and the network node degree data for every network created. Below are our findings. Here, to generate the plots we used our results from Jaccard index network comparison measure of 1 to 10 genes out iterations.

Edges Lost Criterion

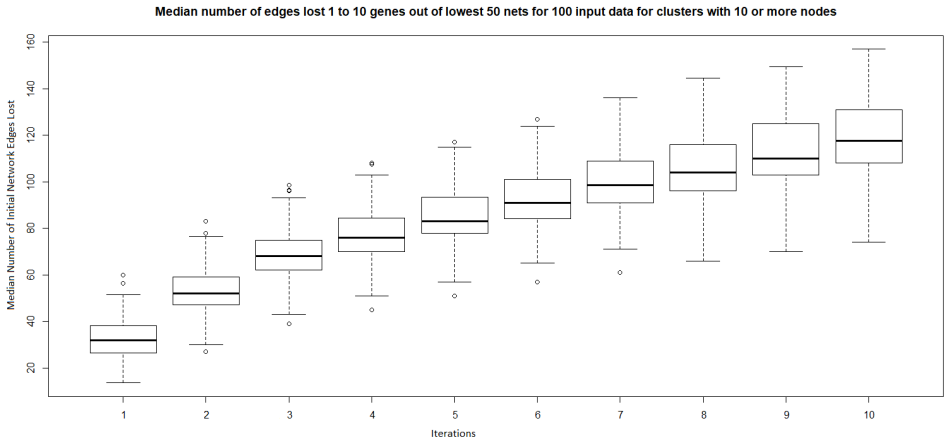

Figure 6.17: Edges lost results with jaccard index network comparison measure

Figure 6.17 shows how the initial network edges are lost as the iteration increases from 1 to 10. Here, the edges lost criterion only includes the edges of nodes which were not included in the potential node list to take out. The change in the median number of edges lost per each iteration changes more in the first iterations compared to the iterations 6 to 10. There is an overall increase. This final criterion shows that our algorithm affects the network structure when the nodes chosen by the jaccard index network comparison measure is taken out at each iteration. The average total number of edges across 100 initial networks is 458.58. The range is from 270 to 656 total edges. So, the threshold of total initial network edges ranges from 135 to 328, with an average of 229.29.

Here, the figure shows that the median of the last boxplot element is close to 120. So, an increased number of iterations might have found networks that stopped by this criterion.

Nodes Lost Criterion

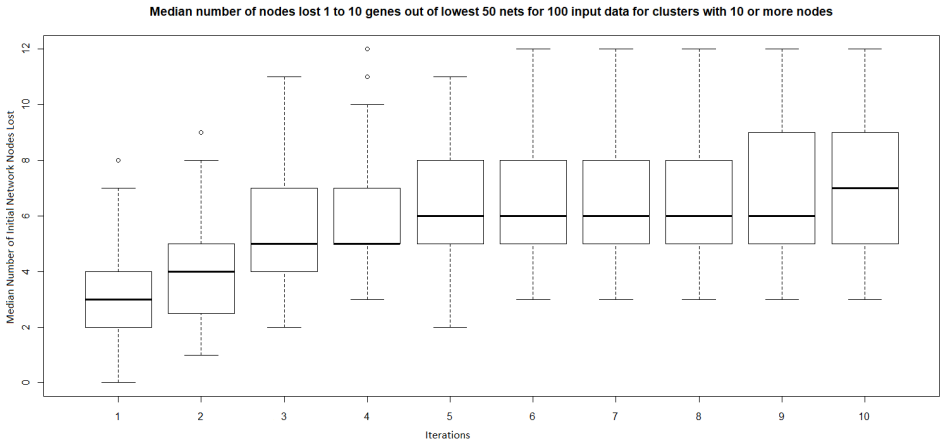

Figure 6.18: Nodes lost results with jaccard index network comparison measure

Figure 6.18 shows how the initial network nodes are lost as the iteration increases from 1 to 10. Here, the nodes lost criterion only includes the nodes which were not included in the potential node list to take out. The change in the number of nodes lost per each iteration changes more in the first iterations. Then, for the iterations 5 to 9 genes out the boxplots are almost the same. This means that the number of original nodes lost and new original nodes gained for these iterations are almost the same. This small amount of change may also be the reason of having little number of nodes lost at each iteration for each network. So, the median of 50 networks, and the boxplot of the median number of nodes does not show an effect which would probably work on a larger network. There is still an overall increase, although not as good as edges lost criterion. So, this final criterion does not show the affect to the network structure as well as edges lost.

Jaccard Index Criterion

Figure 6.19 shows the boxplot of the median jaccard index calculated for 50 lowest jaccard index networks across 100 input data sets. The values show an overall decrease. The range of values also decrease as the iteration continues. The jaccard index values were calculated by taking account the clusters including 10 nodes or more. Here, as the number of genes selected to eliminate gets larger the nodes originally in the same cluster change clusters. This change results in lower jaccard

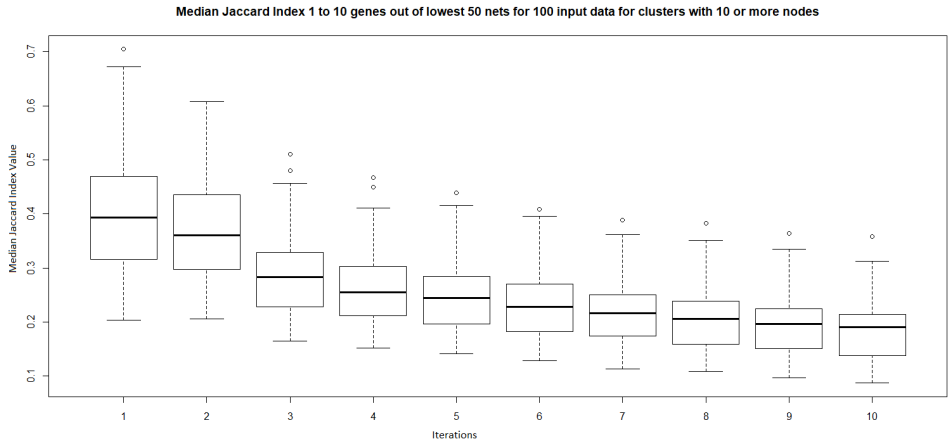

Figure 6.19: Jaccard index results with jaccard index network comparison measure

index values as the iteration proceeds. Again, the nodes in comparison do not include the chosen potential node list to be taken out.

Scale-free Criterion

Our results show that the scale-free characteristics of a network is not easily broken. If there is no limit on how many nodes should be accepted as hubs, then the power-law with an exponent α and KS-test fit to a power-law with the same exponent gives passing results for p-values. The reason lies within the mathematical representation of the power-law. Even if there are three nodes in a network, if only one connects to the other two, then the preferential attachment criterion is sufficed. All three nodes will be in one cluster which also satisfies the high clustering coefficient criterion. So, for these three nodes, the node degree diagram would be a straight line with a decrease of inclination of 45 degrees. And, this line also satisfies the power-law equation with an exponent of $log(2)$. This means, even if the drop from low-degree nodes to high-degree nodes is not as large as a power-law equation with an exponent higher than the inclination of a straight line, the mathematical test still gives a p-value of 0.05 or larger. But, the difference in numbers between low-degree, median-degree, and high-degree nodes would be small. Below figures show this effect in our calculations. Figure 6.20 shows the result of the median active nodes found for iterations 2 to 10 with jaccard index network comparison above, and the corresponding first network node degree plots below. For these two example networks, the algorithm was not able to find more than 1 active node at the end of 10 genes out iteration. The first node degree diagram shows two peaks at different values, and not a smooth decrease. The second scale-free diagram shows node degree values almost staying the same in the middle. The KS-test still passes since the tails of the

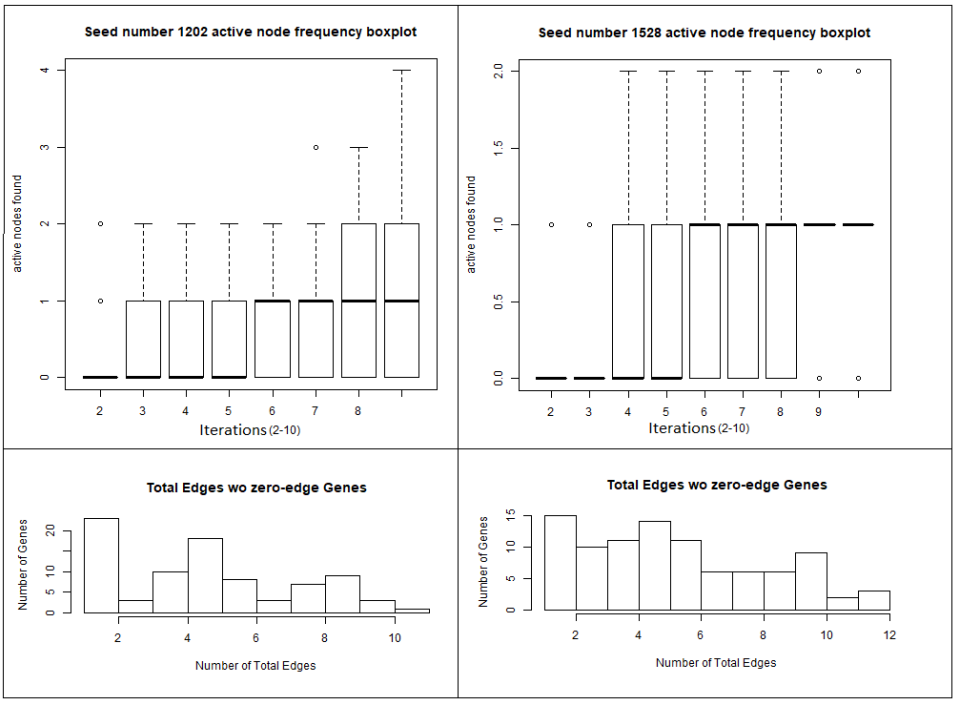

Figure 6.20: Simulation results for 300 gene expression values.

diagrams are scale-free. But, the algorithm is not able to differentiate the nodes enough to choose the active nodes by taking node degree values into account. In figure 6.21 both input data sets ended with median active nodes found results of 4 to 5 at the end of iteration 10. The node degree diagrams below depict a smooth decrease, and even the first one with a peak in the middle gives good results, since overall decrease from low-degree to high-degree nodes is good enough.

The final criteria discussed here showed an overall change across 100 data sets for iterations up to 10. Even though the thresholds were not reached at the end of 10 iterations, the plots showed that for more iterations there is a potential that some networks would stop by hitting a final criteria, especially for edges lost and Jaccard index. More work is needed to find out how many iteration it takes to reach final criteria thresholds.

6.3 Discussion

This thesis introduces a new method for identifying the most critical nodes of a scale-free network according to specified scoring metrics. Our results depend on the network model fit to the data, the beam width, and the thresholds for choosing potential impact gene lists and for network comparison steps. In chapter 7, we will use our algorithm for a real and complex data set, the

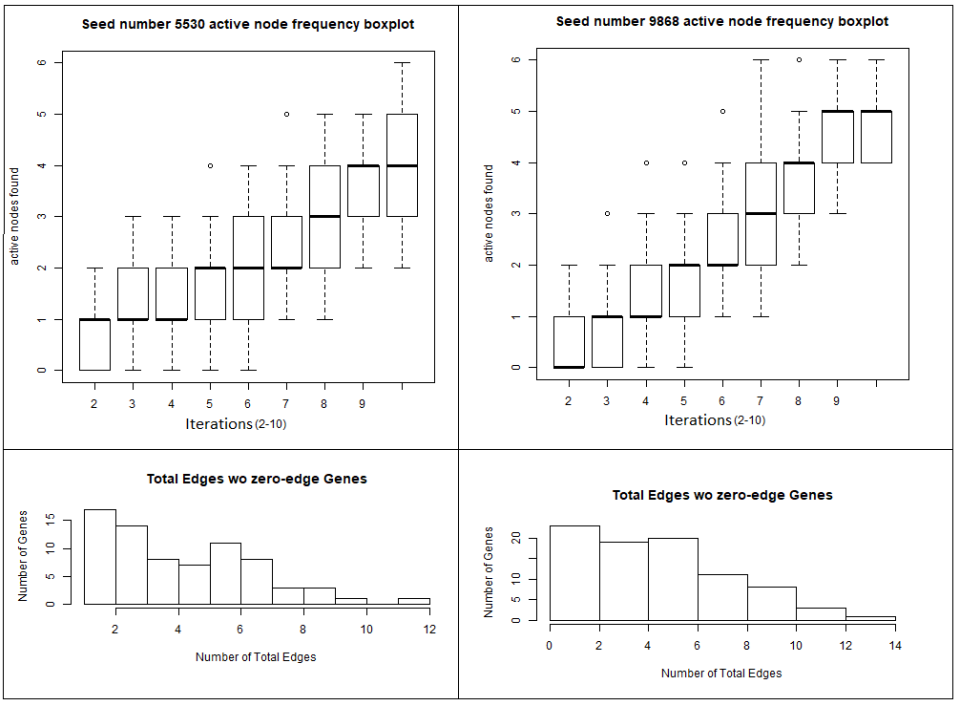

Figure 6.21: Simulation results for 300 gene expression values.

initiation of puberty data. We tried to mimic this data as much as possible when testing our algorithm. Since biological data is noisier and the beam ranking is very close to one another, we followed the same steps and instead of choosing a best contender, we assumed all the contenders are alternative results and looked at the median value of the beam at each iteration. Although the results were dampened because we took the median of 50 networks each for each iteration and showed the distribution across 100 data sets, our simulated gene expression data results still showed that our method is able to identify transcriptionally active nodes from among background nodes better than the initial network. For the biological data set, there is also the randomness of the beam candidates that we need to consider about. This is the reason that we stop our algorithm at a certain iteration instead of waiting to hit a final criterion for the biological data case. To see the same effect of stopping early, we also stopped our algorithm early for the simulated data case. We will discuss the randomness effect in chapter 7 again.

The network comparison measure Jaccard index gave the best results compared to edges lost and nodes lost measures. The p-value plots and also the plots for variables used in final criteria showed that more iterations are needed. The scale-free criterion showed that the initial networks affect the final results. The initial networks can be chosen with a restricted power-law α exponent range to mimic real-life complex networks better.

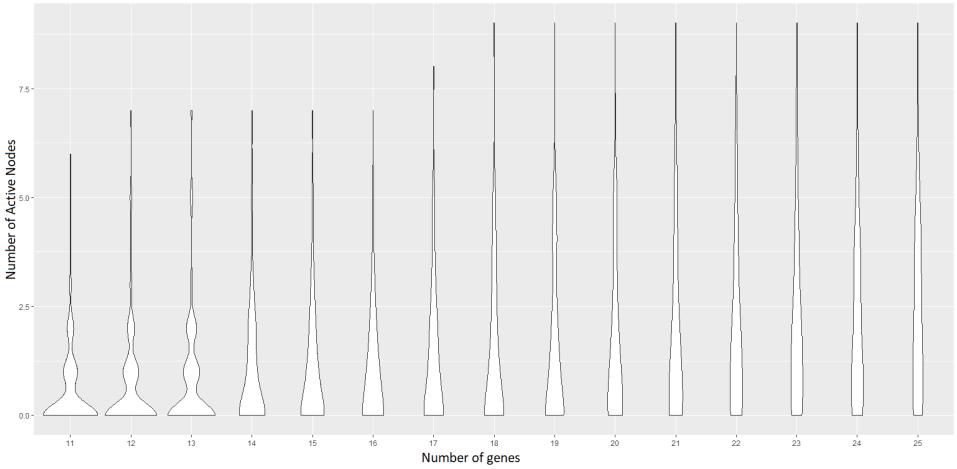

Figure 6.22: Violin plot of number of active nodes in first networks for the genes selected between 11 to 25.

Figure 6.22 shows the initial network median number of active nodes found results for the iterations from 11 to 25. As a future work, the next step for our simulation data is to include all 100 data sets for all network comparison measures, and do the iteration till 25 number of genes out. We will compare our results with the first network found active nodes in figure 6.22.

Chapter 7

Application to Initiation of Puberty

In the previous chapter, we demonstrated how our algorithm successfully identified the set of nodes of greatest impact on simulated expression data. The motivating domain that led to the development of our algorithm is the identification of genes that have the biggest influence on the regulation of the onset of puberty. In this chapter, we apply our algorithm to real world temporal rat expression data. The initiation of puberty is a well researched subject as we covered in chapter 4. Although researchers have already identified more than 300 genes related to puberty, the genetic network behind the hormonal changes as a whole is still largely unknown, that is, there could still be many factors yet to be identified that have a large impact on the onset and development of menarche. The complexity of this system also hinders progress. In this chapter, we describe our analysis of puberty data and the identified gene sets related to puberty with our algorithm, impact node finder (INF).

7.1 Network Gene Set Preparation

We used female rat (Rattus norvegicus) medial basal hypothalamus (MBH) tissue data collected by our collaborators, Dr. Lomniczi and colleagues [158] for their research about the initiation of puberty. Tissue samples were collected at four different postnatal development (PND) stages on days of 14 (PND14), 21 (PND21), 28 (PND28) and 29 to 35 (PND29-35) after the rats' births. From here on, we specify these PND days with their corresponding developmental stage names. PND14 is when the rats are at their infantile (INF) stage. PND21 is called early juvenile (EJ). PND28 is late juvenile (LJ), and PND29-35 is where rats are at late puberty (LP).

For every stage, there were four animal samples, for a total of 16 samples. The RNA-seq data for these samples was collected to investigate how genes change values as the rats develop from the infantile to puberty periods. There were 32,662 rat genes/transcripts in the file, so, our initial data matrix included 32,662 rows and 16 columns. The RNA-seq feature counts were converted

into expression values, and the data was corrected for batch effects to compensate for technical differences among samples using the R Bioconductor package edgeR [5]. Gene expressions were in the form of log-cpm values.

In sections 7.1.1 and 7.1.2 we will describe the general procedure of gene selection from RNA-seq data as covered in [93, 85, 78, 163]. For the rest of this paragraph, we will preview sections 7.1.1 and 7.1.2. To choose the most variable differential gene expression (DGE), to construct a network, and to study their correlations and connections to one another, we followed the workflow for RNA-seq data described in Law et al [85]. First, we found human ortholog genes that correspond to rat genes in our data. We excluded rat genes without human orthologs. Then, we eliminated unexpressed genes to decrease the gene pool from which the most variable genes were selected. We checked sample quality and the distributions and relations among samples. Last, we used the limma [135] package to identify the genes with the greatest change in expression according to developmental stages. Below we go over these steps.

7.1.1 Data Preprocessing

The preprocessing procedure includes various steps to shrink the gene pool from the initial high throughput data and also check the sample and gene expression qualities. It is important to eliminate genes of no iterest or no differential expression changes across samples since the p-value estimates are not reliable for large data sets with small sample sizes. Also, there could be experimental or instrumental errors which would effect the sample quality and expression values.

Human Orthologs

To find the counterpart human genes that correspond to the rat genes in our data we used the Ensembl genome databases, www.ensembl.org, and its dataset export tool, the biomaRt package [39]. Ensembl datasets are continuously updated and changed. Our latest download was the Ensembl 95 annotation dataset. We kept genes only if matching human homologue genes existed. The human genes dataset for Ensembl 95 was “GRCh38.p12”. Out of 32,662 genes, we found 20,452 matches.

Eliminating Low Count Genes

We are interested in genes with highest variations among samples of different developmental stages. Therefore, we searched for the lowest p-value genes. P-values with an FDR lower than 0.05 are defined as statistically significant. Since choosing genes from a smaller pool of input data gives greater accuracy, it is important to eliminate genes of no interest from the beginning.

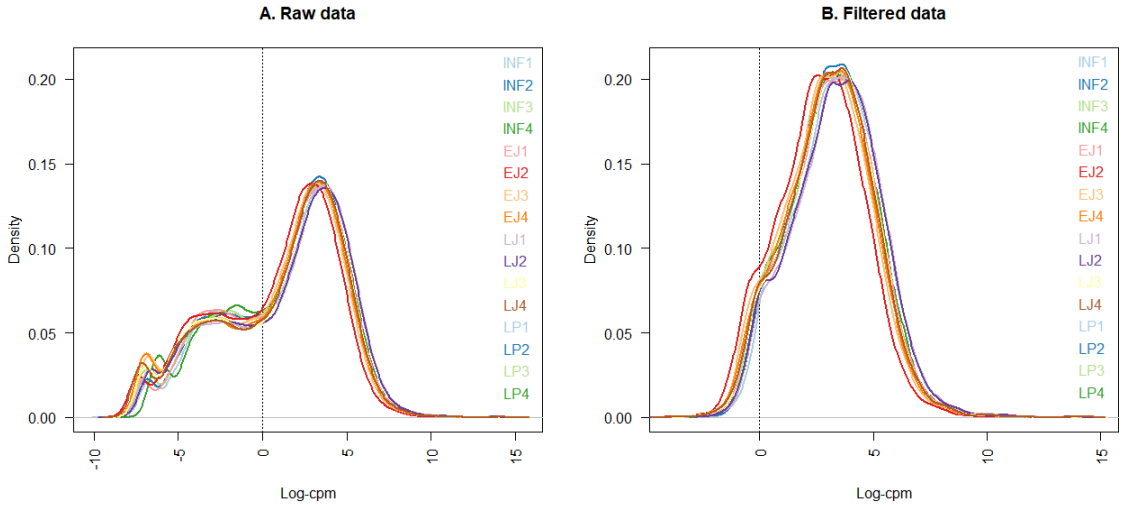

Figure 7.1: The densities of samples versus gene level log-cpm values

Figure 7.1 shows the densities of samples versus gene-level log-cpm values. On a logarithmic scale, a value less than zero means a feature count of less than one, so, these values would belong to genes with very low expression. We discarded those genes whose log-cpm values across all samples were negative. Out of 20,452 gene expressions, 6,351 genes had negative expression values across all 16 samples which corresponded to 31%. Panel A in Figure 7.1 corresponds to the samples before filtering with 20,452 genes, and panel B corresponds to the same samples after filtering. The total number of genes remaining after this step was 14,101.

Sample Quality

To determine whether the above filtering decision was appropriate and whether the resulting sample distributions would need further corrections, we looked at sample boxplots and normalization factors.

Figure 7.2 boxplots show the sample distributions before (panel A) and after (panel B) the filtering. Every boxplot represents a specific sample. As the figure shows, after the filtering, the samples are more evenly distributed relative to their median values. The sample spreads are also smaller. Both sets of boxplots show very similar data distributions, with median, highest and lowest quadrant values close between the two panels. Still, to decide whether the samples would need to be normalized further or not, we also looked at their normalization factors.

Figure 7.3 shows the normalization factors of samples before (green dots) and after (red dots) the filtering. All the normalization factors after the filtering are very close to 1, and very close in

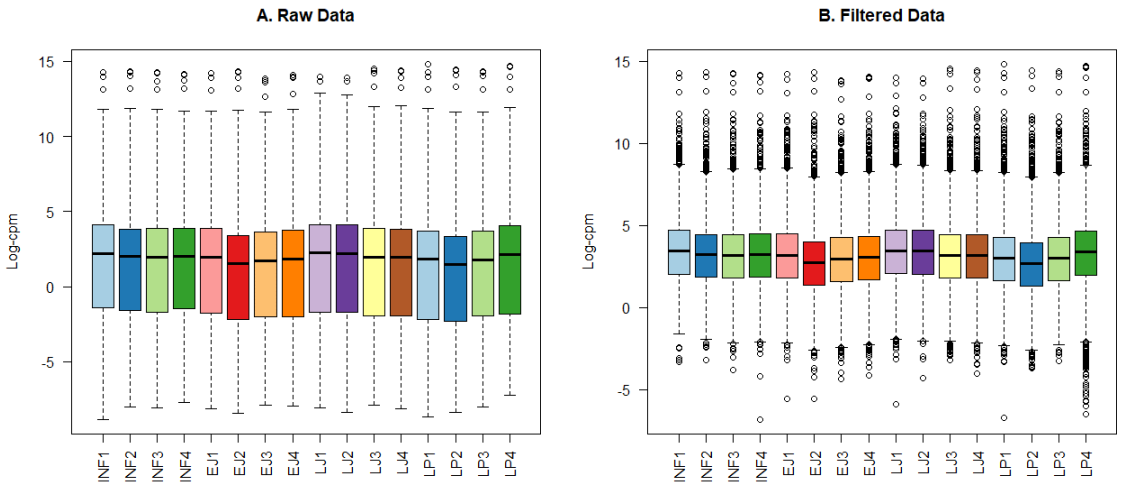

Figure 7.2: Boxplots show the sample distributions, before (panel A) and after (panel B) the filtering.

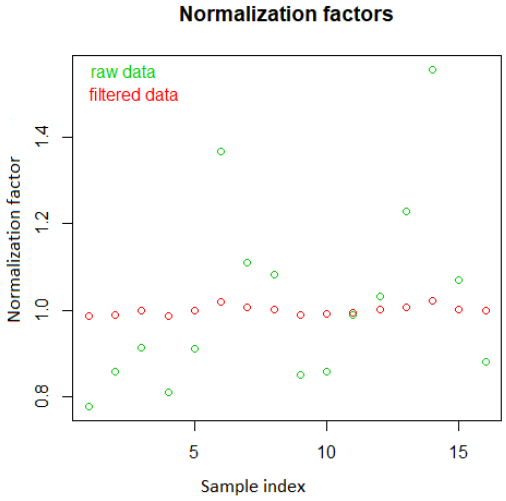

Figure 7.3: The normalization factors of samples

value, so, we decided no further normalization or correction steps were needed.

Sample Groups

Before going any further into gene-level data manipulations, we looked at the sample groups to determine how they were related. According to the experiment design, we expect to have differences among the different developmental stages but similarities between same-stage samples. We looked at both the multi dimensional scaling (MDS) plot and the hierarchical clustering of samples.

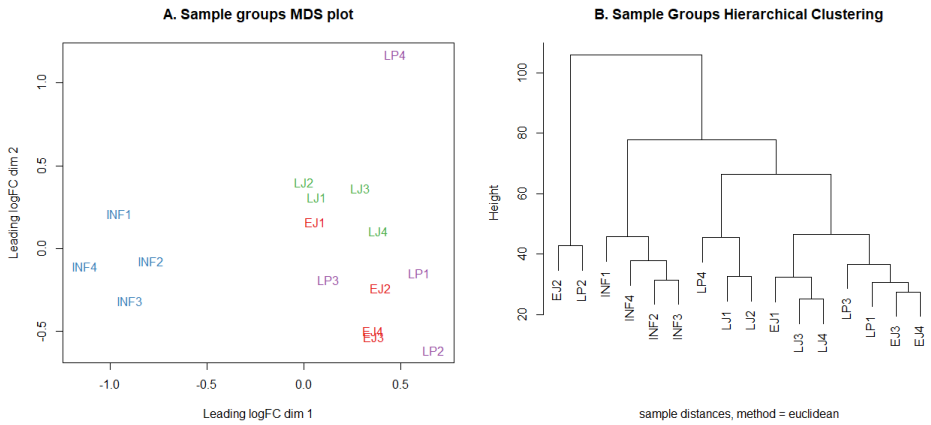

Figure 7.4: Development stages

In figure 7.4, panel A, the different colors represent different developmental stages. The X and Y axes are the leading log-fold change (logFC) values of the first two principal component dimensions which explain the largest proportions of the variation in the data. In hierarchical clustering, as illustrated in panel B, the height shows the distance between samples. Here, we used Euclidean distance to calculate the similarity between groups. As the height increases, the differences between samples also increase. If the distance between two samples is small, they are grouped together first, and then the distance is calculated again to find the next closest sample to group together.

In both the MDS plot and the hierarchical clustering graph, all four samples of the INF group showed similar variations. They clustered close to each other, and were separated from the rest of the samples. On the other hand, the other three groups were not separated from one another for both dimensions in the MDS plot. Although all LJ samples and three samples of the LP group were separated from each other in the second dimension, EJ samples had logFC values very similar to those. The LP4 sample was far away from the rest of the LP samples in the MDS plot, but its dimension 1 value was still close to those of the other LP samples, and since it was not separated

from the other samples in the hierarchical clustering with a larger height value than the rest, we did not treat sample LP4 separately.

7.1.2 Most Variable Gene Selection

After the preprocessing steps, we used 14,101 gene expressions covering 16 samples in four groups, and used the R package limma [135] to find the top variable gene expression levels over all samples.

Design and Contrasts

Design

The first step in the DGE analysis is specifying a numerical design for the experiment. This step shows the experiment's initial design aim of reaching a final target result. Figure 7.5 displays the matrix for our design. We grouped our samples into corresponding developmental stages.

	INF	EJ	LJ	LP
INF1	1	0	0	0
INF2	1	0	0	0
INF3	1	0	0	0
INF4	1	0	0	0
EJ1	0	1	0	0
EJ2	0	1	0	0
EJ3	0	1	0	0
EJ4	0	1	0	0
LJ1	0	0	1	0
LJ2	0	0	1	0
LJ3	0	0	1	0
LJ4	0	0	1	0
LP1	0	0	0	1
LP2	0	0	0	1
LP3	0	0	0	1
LP4	0	0	0	1

Figure 7.5: Matrix for design

Figure 7.5 shows the design matrix which was generated by assuming the intercept of the linear model was zero and all the samples were treated equally. There are also other ways to represent the same experiment that would result in equivalent outcomes. We could have used fewer or more groups according to sample comparison results given in the previous section. We could also have specified a linear model with a reference group and designed the matrix that way.

Contrasts

As mentioned in section 3.3.4, contrast matrices are used for pairwise comparison of design groups. There is no limit to how many comparisons may be performed on the data. Pairwise comparison separates genes with expression values showing high variances in the chosen two sample groups. From Figure 7.4 MDS plot and hierarchical clustering, it can be seen that the infantile group comparison would separate the most of the high variance genes. Thus, we made our contrasts taking the infantile group as the first one to separate. We also included the rest of the group comparisons to determine whether any genes show variances only in those groups. Below are our contrasts formulas, and corresponding contrasts matrix.

$INFvsEJ = INF - EJ$

$INFvsLJ = INF - LJ$

$INFvsLP = INF - LP$

$EJvsLJ = EJ - LJ$

$EJvsLP = EJ - LP$

$LJvsLP = LJ - LP$

	INF vs EJ	INF vs LJ	INF vs LP	EJ vs LJ	EJ vs LP	LJ vs LP
INF	1	1	1	0	0	0
EJ	-1	0	0	1	1	0
LJ	0	-1	0	-1	0	1
LP	0	0	-1	0	-1	-1

Figure 7.6: Contrasts matrix

The choice of which group is subtracted from which only changes the genes groups that are assumed to have a positive or negative differential variation according to the selected pairwise comparison. It does not have an effect on the complete set of chosen differentially expressed genes at the end of the analysis.

Linear Model Fit

By using the above design and contrasts matrices, a linear model was fitted for every gene expression separately by assuming gene expressions were independent from another. All contrasts were used separately when pairwise comparison was done. Next, by using the output of the linear model, moderated empirical Bayes formulation was performed to determine residual variances. Figure 7.7 represents the two plots of the residual variances of each gene before and after fitting the model.

In figure 7.7, both X and Y axes are rescaled. Each dot in the plot corresponds to a gene. Before

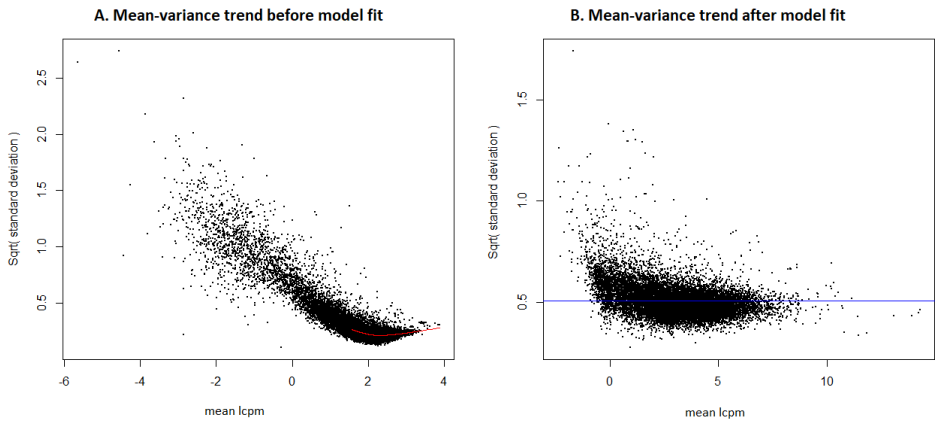

Figure 7.7: Variances versus mean plots

the fit, the variances and means were correlated. The figure on the left shows this trend. After the fit, the model removed this correlation while calculating residual variances for each gene. The blue line corresponds to the mean model estimate of the variance by the Bayes algorithm.

Differential Gene Expression

After calculating the residuals, the algorithm looks at the p-values to find the DGE. P-values are adjusted using Benjamini-Hochberg (BH) correction. The p-value cutoff for DGE was 0.05. Figure 7.8 lists the results according to the contrasts. As predicted, all DGEs except one showed large variations when the pairwise comparison was done using INF as a reference.

	INF vs EJ	INF vs LJ	INF vs LP	EJ vs LJ	EJ vs LP	LJ vs LP
Down	199	641	371	1	1	0
NotSig	10738	12887	10586	14100	14100	14100
Up	3164	573	3144	0	0	1

Figure 7.8: Differential gene expressions

The genes in the first row are called down. These genes are the ones with a negative value when the second group expression values were subtracted from the first group values in the model contrast. Later on, down is also referred to as -1. The second row shows all the genes which show no significant variation relative to the contrasts chosen. The last row corresponds to the genes that are up, or +1, with first group values higher than second group values. Figure 7.9 illustrates

the relationship among the first three contrasts according to the above results.

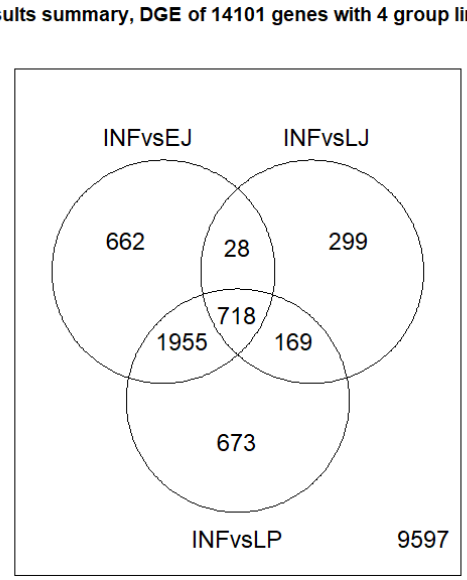

Figure 7.9: Venn diagram

Out of 14,101 genes, 9,597 showed no significance according to the chosen cutoff values. A total of 4,504 genes were significant with an adjusted p-value lower than the cutoff. Out of these 4,504 genes 3,167 genes were selected as DGE with an FDR lower than 5%.

We also looked at the logFC of the model-fitted residuals to find which genes show variation according to their relative fold changes using the same contrasts. Again, genes were only divided when the INF group was considered as one of the pairs for contrasts, and the same 3,167 genes were selected with the above criteria of adjusted p-value and FDR cutoffs. Figure 7.10 plots show the logFCs according to the same first three reference groups. Again, every dot corresponds to one gene. The green and red dots are DGEs relative to specific contrasts, and the black ones are nonsignificant gene expressions. The plots show logFC values versus mean lcpm for each gene. From these results, we chose 3,167 genes as the most variable genes across puberty developmental stages.

When matching rat genes to their human orthologs, in some cases multiple rat genes correspond to the same human genes. To only use one gene expression set for each unique gene name, we eliminated genes when there were duplicates in the final data set. When there was more than one set of expression values for the same gene name, we included only the gene with the highest F-statistics score calculated by the model. This way, the most variable DGE was selected from among a set of expressions. Our final set included 3,045 unique genes.

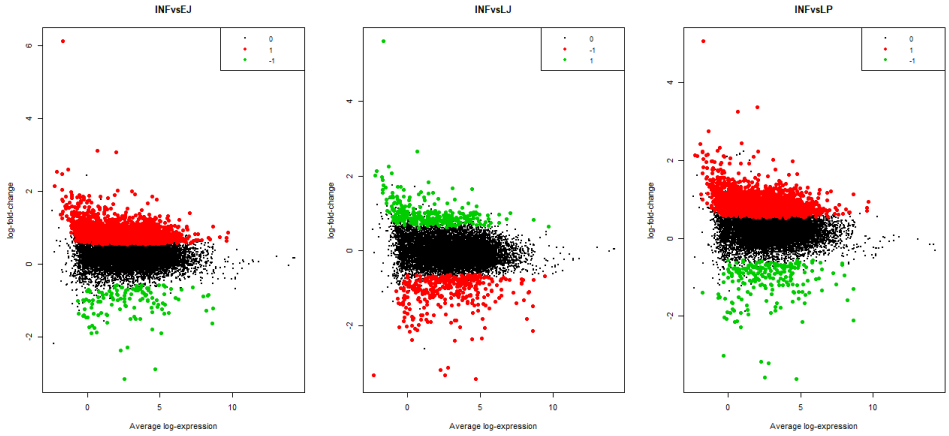

Figure 7.10: Log-fold changes

7.2 Initial Network

We used the R parcor package [80] to construct a network using the 3,045 gene expressions. The parcor package uses a regularized lasso algorithm with a 10-fold cross validation to find the optimum network result. Here, we consider weighted undirected graphs to represent the constructed networks. We used the same procedure to construct all networks including the initial network and all subsequent networks after genes are eliminated by using INF algorithm. For the initial network, a total of 2,928 genes were connected with at least one edge in the network. A total of 5,266 edges were created with partial correlation values different than zero. In section 7.2.1 we will give the mathematical properties of the initial network where all nodes are used as the input data set. In section 7.2.2 we will go over the network clustering. Since we use Jaccard index network criterion metric, the initial network clustering gives us an idea from where our algorithm starts.

7.2.1 Network characteristics

Networks are specified by their nodes and edges. The node characteristic we considered here is the node degree. Below is the node degree distribution of the network. The analysis covered in this section is to verify that the created network meets the expectation of being scale-free. Since we also needed to compare different networks with different node numbers, we always included the nodes with no edges in our calculation. This choice does not change the results of the scale-free fit calculation below, and presents ignorable differences when calculating density and modularity values. The first point in both graphs in Figure 7.11 corresponds to the nodes without edges, and should be ignored when considering the scale-free fit curve.

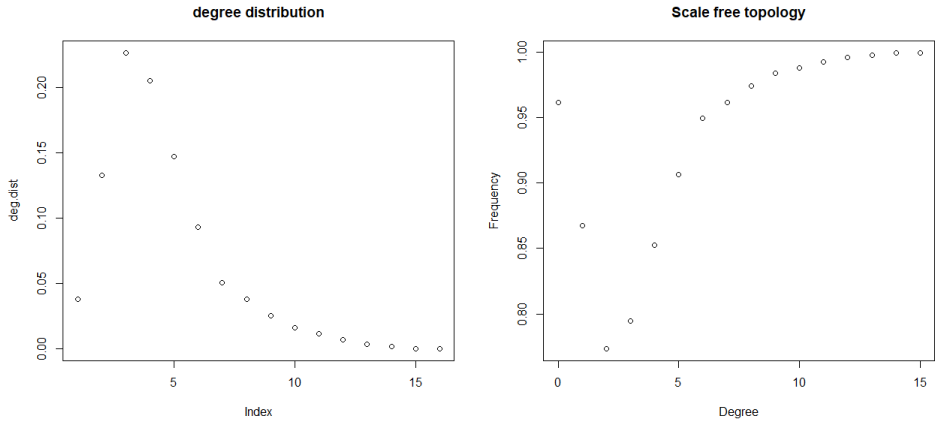

Figure 7.11: Degree distributions

The first panel in Figure 7.11 indicates that the degree distribution follows a power-law curve. The second panel shows the node degree frequency versus node degree. We used power-law fit function of the igraph package to fit our node degree data into a power-law curve. This function uses the KS-test to calculate the probability that the fitted curve estimates the real data correctly. A test result with a p-value of 0.05 or greater suggests that the fit is correct. However, the reliability of the test results should also be considered. The test requires a minimum value of node degree selection. The power law is fitted only to the data above that value. To be reliable, a large network should have at least 50 or more nodes with node degree values equal to or higher than the chosen minimum value.

For all the minimum node degree values equal to or above 8, the network showed scale-free characteristics. For the values of 8 to 10, the network had more than 50 nodes with degrees equal to or higher than the minimum value. The second plot in Figure 7.11 shows this result. The curve reaches to the frequency value of 1 at the point where the degree reaches 10, approximately, and it stays more or less constant afterward, even though the degree increases. The scale-free network does not change scale and stays the same even after additions of new nodes. Here, scale corresponds to the over-all image of the network. New nodes are more likely to connect to already structured clusters and to the hubs with large number of edges rather than starting a new structure.

The p-value of the lowest minimum node degree fit was 0.15 where the KS-test indicated the power-law model fit the data. The corresponding power-law fit exponent was 5.95. Out of 3,045 genes, 2,928 became nodes of the first network. There were a total of 5,266 edges with partial correlation values as edge weights ranging between [-0.48, 0.97]. The edge density of the network was close to zero (0.00057). The median of node degrees was 3, and the interquartile range (IQR)

was 2. We set those genes with node degree values one IQR or less away from the median as part of the genes with fewer edges group. The number of genes with five or fewer edges in the first network was 2,453. The diameter of the network was 12.

We also compared the gene names that were chosen as DGE and gene names that became initial network nodes with two known gene sets: a gene set of 370 unique names from the GWAS study of AAM measurements and a putative puberty-related gene set of 224 gene names compiled in Dr. Lomniczi's laboratory. There were 94 out of 370 and 63 out of 224 matching names in the list of 3,045 DGE. After the network construction step, the intersection counts were 59 and 90, respectively. These numbers were high enough to show us that the DGE choice procedure and the network construction steps are working well. So, the original network to start our algorithm iterations have potential to find new puberty related genes.

7.2.2 Network Clustering

We used the igraph [27] package's fast-greedy clustering algorithm. Table 7.1 shows how the nodes with edges clustered into groups. We will compare the clustering results after INF algorithm iterations to see how the initial network is affected by the beam search node lists in section 7.4.1.

Cluster	1	2	3	4	5	6	7	8	9	10	11	12	13
Size	208	119	58	95	492	106	185	75	70	46	38	65	64
Cluster	14	15	16	17	18	19	20	21	22	23	24	25	26
Size	95	97	236	37	101	136	217	18	105	215	5	5	3
Cluster	27	28	29	30	31	32	33	34	35	36	37		
Size	3	2	2	2	2	2	2	2	2	2	2		

Table 7.1: First network cluster sizes

The modularity value for the above clustering was 0.64. The largest cluster was cluster 5. It was also the most connected cluster and the one with the most nodes with the highest node degree. We denoted the clusters with sizes of 100 genes and up as modules. We compared only those modules when calculating the Jaccard index for newly constructed networks. Below are the module means (Figure 7.12) and module eigengene (ME; Figure 7.13) boxplots. Module means boxplots show the trend of the genes in the same modules. The distribution range is wider and the trend change across experimental stages is less clear compared to ME boxplots. However, the module means is a first measure to see how the genes in the same module behave and is a step to understand whether the genes and ME values behave in the same way. ME is the first principal component representation of the gene group in one module. The x-axis corresponds to the four stages of puberty data. Each graph corresponds to a module. The boxplots show the distribution

of genes at each stage.

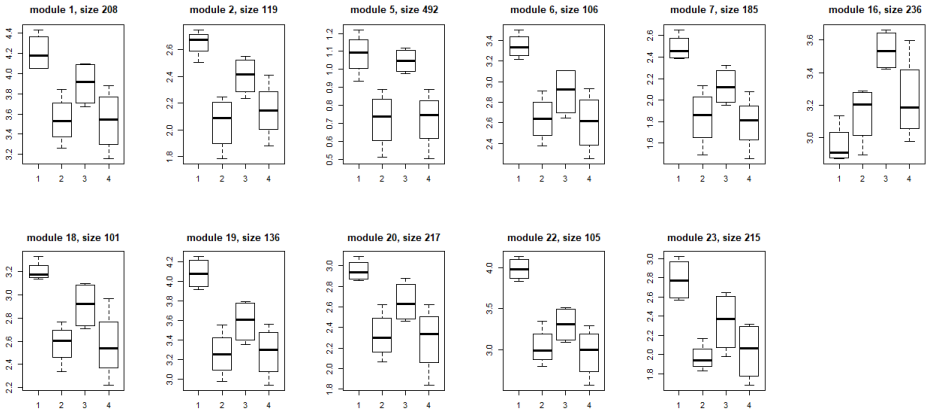

Figure 7.12: Rat data, clusters with 100 or more genes module means

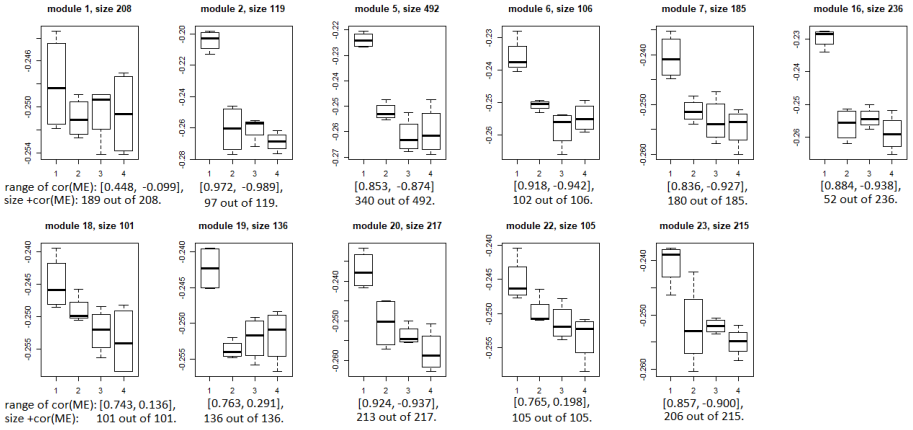

Figure 7.13: Rat data, clusters with 100 or more genes MEs.

In Figure 7.13, cor(ME) shows the correlation to ME. When most of the genes in one module show approximately the same trend, the cor(ME) values are positive and high, but this correlation does not apply to all genes in the group. There can also be genes in the same group that show different behavior than the ME trend. As seen in the boxplot for module 16, the trend between module means and ME is different. The number of positively correlated genes to ME is really low for this module, 52 out of 236 total genes in the module. The rest of the modules show more similar trends when cor(ME) sizes and module mean boxplot comparisons are considered. The bi-plots in Figure 7.14 also indicate how the module genes relate to one another in the same group.

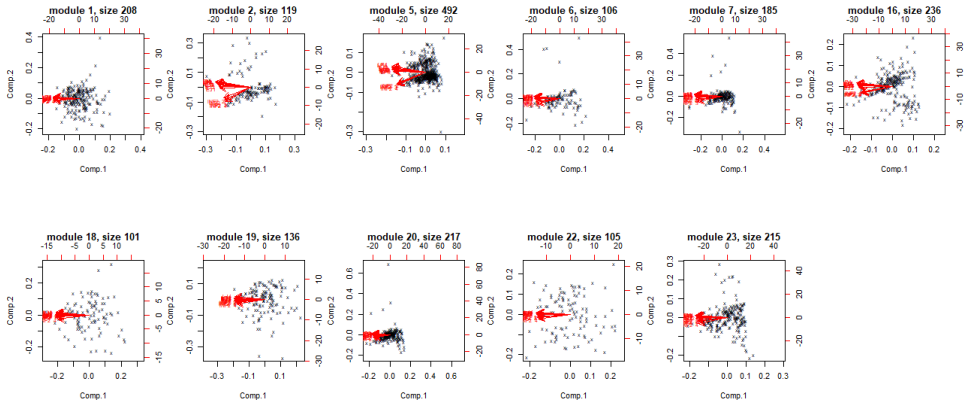

Figure 7.14: Rat data, clusters with 100 genes and more, module biplots.

7.3 Running the Algorithm

For the biological data, we used only the Jaccard index network comparison measure. Every network module with 100 genes or more was taken into account. The histograms in figures 7.15 and 7.16 are the node degree and node strength values of the initial network. The x-axis is the node degree in figure 7.15 and node strength, the absolute value of total edge weights, in figure 7.16. The y-axis shows the number of genes with the same node degree and strength values. The highest bin in figure 7.15 shows that there are a lot of genes with only 1 or 2 total edges. As the number of edges per gene increases, the number of genes with the same total edge decreases which depicts a power-law distribution. The heavy tail of the node strength plot also fits to a power-law distribution. The first potential node list of the beam search uses the values corresponding to the last bins of these histograms.

The top 5% for both node degree and node strength indicated the potential list of active genes. A total of 245 genes were selected for the first iteration. Of these, 205 genes had 8 edges or more, which was the cutoff for edge count. A total of 153 genes were selected as having absolute values of their total edge weights of 0.976 and higher. We compared the gene names with the lists introduced in section 7.2.1. From the 370-gene list, there were three matches on this first potential node list: BARHL2, SEC16B and VEGFC. There were nine matching gene names from the 224-gene list: ALDH1A1, PTGER4, TNFSF10, NPY, DNMT3B, KISS1R, APOA4, SEC16B and MC4R. 11 of these genes (note that SEC16B appears on both lists) were clustered in module 5. BARHL2 was in module 7 and MC4R was in module 3; both satisfied only the node degree criterion. VEGFC also satisfied only the node degree criterion. ALDH1A1, TNFSF10 and DNMT3B were on the list

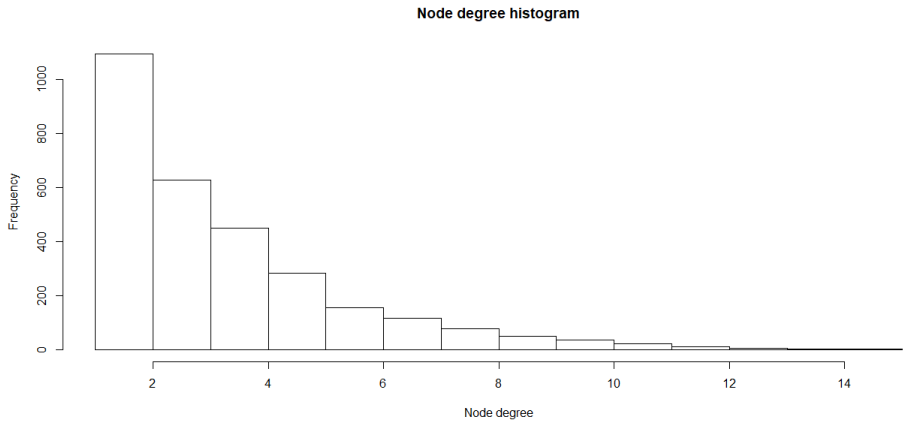

Figure 7.15: First network, node degree histogram

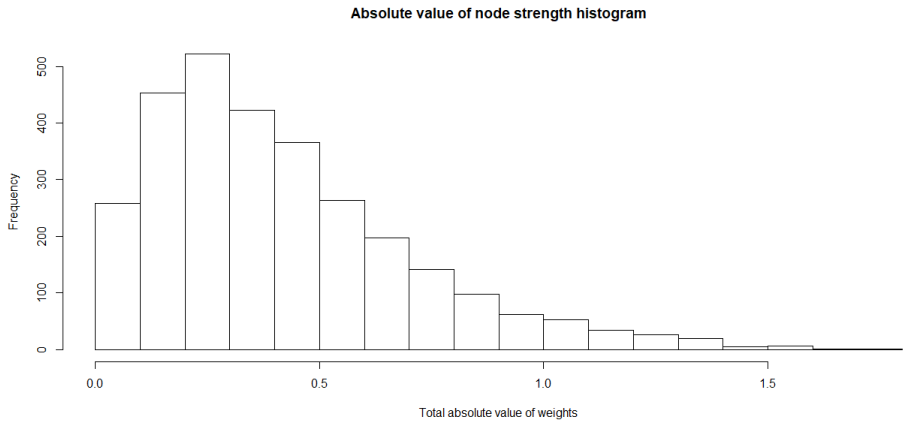

Figure 7.16: First network, node strength absolute value

because of their edge weights. The remaining genes were included in both the highest node degree and node strength lists.

For each of the 245 genes selected for the first run, the selected gene was taken out of the 3,045 DGE list, and a regularized partial correlation network was constructed. The Jaccard index values calculated by comparing the clusters of each of these new networks and the clusters of the initial network to determine the most changed networks. A beam width of 50 was used to select the most changed networks that constituted the second set to calculate node degree and node strength sets. Each iteration compares approximately 12,000 networks at the ranking and pruning step of the beam search.

7.4 Results

In this section, we will go over the results of our INF algorithm. In section 7.4.1, we will give examples of how INF algorithm iterations resulted in biologically interesting genes and how the initial network clustering is affected which shows indications of network function changes. In section 7.4.2, we will discuss our results in gene ontology and transcription factor enrichment analysis related to the initiation of puberty.

7.4.1 Gene and Cluster Result Examples

KISS1R was on the first potential gene set, and after taking it out, the resulting network displayed one of the lowest Jaccard index values. The genes that followed KISS1R were also on the lowest Jaccard index networks for the next nine genes taken out. At 12 genes out, the algorithm chose DNMT3B as one of the lowest Jaccard index networks, and the sequence including DNMT3B stayed as one of the lowest Jaccard index values until the 22 genes out iteration. We investigated results for iterations 1 to 30.

Figure 7.17 shows an example of how clusters of initial network is affected as the iterations increase. The left two module plots show the trends of modules 1 and 19 of the initial network. The right five and six total module plots come from the highest ranked network of the beam of the iteration 27 genes out with a Jaccard index value of 0.11. As the figure shows the initial network genes of each module got distributed over different new modules. Some even ended up modules with different trends. This change in clusters shows an indication that the network function is also affected by the change in the network. The new groupings of genes in different clusters might have different functions in the cell. So, even eliminating 27 genes out of 3,000 affects the network clusters, and probably the funtioning if the selected genes are at critical positions.

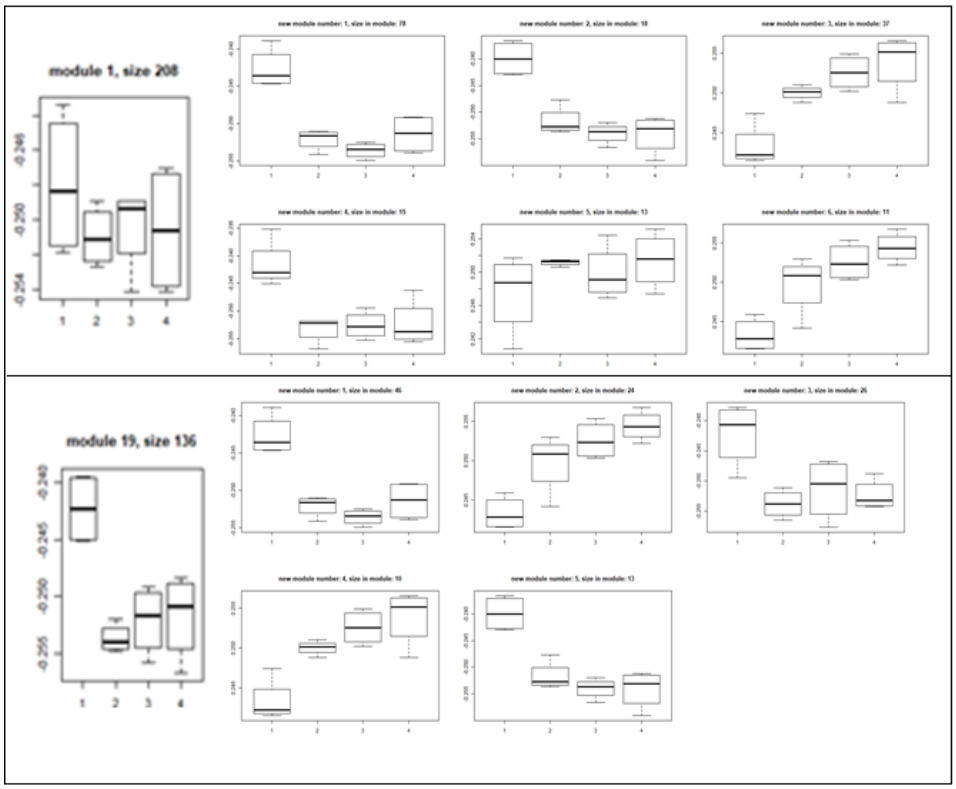

Figure 7.17: Cluster changes example

7.4.2 Gene Ontology Results

After gathering the eliminated gene subsets of the beam for iterations 1 to 30, we further investigated the gene subsets by using enrichment analysis as explained in section 3.5.5. When compared to the initial network first potential node list of 245 genes, our algorithm confirmed 164 genes as potential impact genes. There were also 33 new genes found which were not included in the first potential node list. These 197 genes were the ones found using enrichment analysis of adjusted p-values of equal to or lower than 0.05. Figures 7.18 and 7.19 show the enrichment results for TFs and gene ontology (GO) terms with adjusted p-values less than 0.05 for genes starting from 1 to 30 genes out based on the Jaccard index-measured lowest 50 networks.

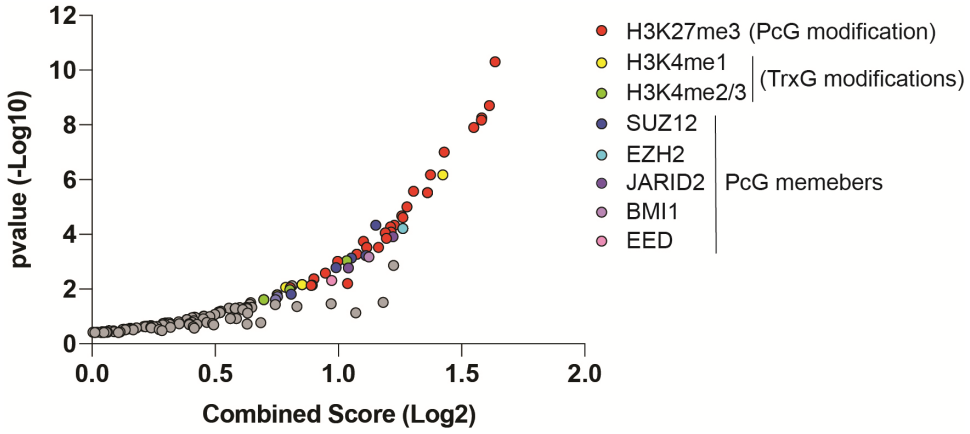

Figure 7.18: Enrichment ENCODE database TFs

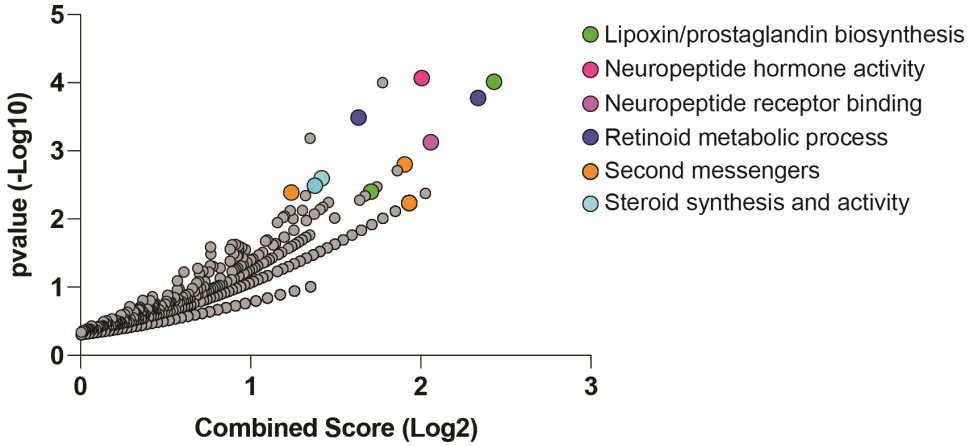

Figure 7.19: Enrichment GO database

Gene ontology (GO) analysis using the Enrichr database [83] determined that our gene set is

enriched in the following functional gene sets:

1) Lipoxin/prostaglandin biosynthesis. As mentioned before, astrocytes and ependymoglial cells lining the third ventricle enhance GnRH secretion, and thus control reproductive development, by releasing growth factors and prostaglandins, especially PGE2 [90, 23, 130]. Our analysis potentially discovers a subset of genes central in the control of prostaglandin secretion.

2) Neuropeptide activity. Neuropeptides and their receptors are the most effective and stronger controllers of hypothalamic GnRH release. Neurons releasing kisspeptin and neurokinin B as well as neuropeptide Y stimulate pubertal development, while neurons that release gonadotropin inhibitory hormone, proopiomelanocortin and prodynorphin mainly restrain sexual development by inhibiting GnRH release. [154, 77, 37, 116, 109, 65, 159]

3) Retinoid metabolism. Retinoic acid functions as a differentiation factor of the nervous system [100]. Genome wide association studies of age at menarche identified loci highly enriched in/near genes that encode nuclear hormone receptors, co-activators or co-repressors, including those receptors involved in retinoic acid signaling. They bind to retinoic acid response elements, activate gene transcription and have diverse functions across embryonic development (including GnRH neuron migration) [108], cell differentiation and homeostasis (including GnRH expression and secretion) [21].

4) Second messengers. All neurotransmitters and neuropeptides signal through a diversity of receptors coupled to the production of intracellular second messengers like cAMP, cGMP, IP3 and Ca2+ . This demonstrates that our gene set not only controls reproductive development at the neuro-glial interface, but also intracellularly at the level of signal transduction.

5) Steroid synthesis. Reproductive development is highly dependent on steroid receptor and steroid synthesis. GnRH release from the hypothalamus increases luteinizing hormone release from the pituitary gland that ultimately induces increased steroid synthesis from the gonads. These gonadal steroids are responsible for the negative feedback loop of control of hypothalamic drive [127].

This analysis demonstrates that our algorithm is not only capturing a subset of genes of high network connectivity but also with a wide variety of demonstrated cellular functions associated with reproductive development.

Enrichment for transcription factor binding and histone modifications, using ChIP-seq studies from the ENCODE project, demonstrate that our gene list is highly controlled by several members of the Polycomb Group (PcG) of transcriptional repressors like SUZ12, EZH2, JARID2, BMI1 and EED. As a result of this targeting, the upstream regulatory regions of these genes are found to be highly controlled by H3K27me3, a repressive histone modification. This result very well aligns

with previous findings from the Lomniczi lab where they demonstrated the central role of two families of epigenetic modifiers, the PcG and the Trithorax group of epigenetic activators in the control of reproductive development [89, 158].

Our analysis sheds new light into further putative target genes of the PcG family of epigenetic repressors. Opening new venues for the study of developmentally regulated gene networks controlled by the epigenetic machinery throughout the neuroendocrine hypothalamus.

7.5 Discussion

The rat RNA-seq data we used as biological data is a more challenging data set with a larger list of variables and higher levels of noise list than the simulated data we discussed in the last chapter. The gene sets that the algorithm identified, including genes taken out ranging from 1 to 30 genes, confirmed 164 genes chosen from the initial network and found an additional 33 genes which match gene lists with enrichment adjusted p-values less than or equal to 0.05 from databases such as ENCODE, GO, REACTOME, KEGG, ChEA and Wikipathways.

Our optimization process step of choosing the most-changed networks at each iteration limited the pool of available networks and was very stringent when it came to real data of 3,045 genes. More experiments are needed to find the optimum thresholds for our process and to reach more biologically functional gene lists.

Chapter 8

Discussion and Future Work

We have developed an iterative graph perturbation algorithm for identifying nodes with the greatest impact on a given graph/network model. There are many techniques for generating the initial network model depending on how the correlation across node variables are captured. Our method enables identification of sets of nodes that influence the output and/or functionality represented by the network. The specific novel contributions of this work are:

1. Proposed method for sorting networks using graph theoretic distances: We propose employing graph theoretic distances to measure the impact of the perturbation relating to the removal of a given node.

2. Selecting the perturbation with the greatest impact using the proposed graph sorting method: Network sorting with respect to graph distances allows us to identify perturbations with the greatest impact at a given step.

3. Algorithm for iterative network perturbations to identify the list of most impactful nodes: We propose an iterative heuristic beam search algorithm that employs the perturbation sorting to estimate a sequence of nodes of highest impact. The algorithm is highly parallel in that many perturbations can be measured and compared in a simple MapReduce algorithm.

4. Demonstration of the algorithm on simulated gene expression data: We simulated gene expression data where a set of nodes were implanted with much higher impact than those of the background nodes. Our algorithm successfully identified the original set based on simulated expression data only.

5. Application of the algorithm to real world data on regulation of initiation of puberty: We applied our algorithm to a temporal gene expression dataset on the onset of puberty and identified a set of biologically plausible and interesting genes that are enriched in pathways of interest in the hormonal control of onset of puberty.

8.1 Further Work on the Algorithm

8.1.1 Additional Choices for Potential Impact Node List

We have shown that our choices for the potential impact node list worked well with both the simulated and biological data of our choice. We would also like to add to our search the nodes that have high node betweenness and high correlation to module eigengene values. The betweenness measure, as mentioned before, gives high values for the nodes where there is significant information traffic. Thus, bridge nodes that enable two or more clusters to communicate with each other might have higher betweenness values than hubs, or nodes with high strengths, which we would miss with our calculation. These nodes are also important in biology, and tracking them mathematically would help. High correlation to module eigengene would give greater weight to belonging to a module and would also show that those genes follow the same trend as the eigengene of the given module. Since in puberty research we are interested in specific genes that show specific trends over developmental stages, keeping track of these genes also would help to identify some that we might have missed before.

Another interesting score to investigate is hub score, which is the same as authority score for undirected networks. Kleinberg [75] defines it as the principle eigenvector of the matrix multiplication of an adjacency matrix and its transpose. If the node is connected to hub nodes in a network, this score is high, showing that the information distribution from hub nodes to other nodes depends greatly on this connected node.

8.1.2 Additional Choices for the Network and Cluster Construction Steps

We used partial correlation with lasso regularization as our calculation to find edges and their strengths. This method results only in a sparse matrix that shows direct edges. We also plan to use different algorithms, such as mutual information, graphical lasso, and adaptive lasso, discussed in chapter 2, to calculate network edges to compare with our findings. We used the package parcor [80] for our calculations. With a simulated data set of 50 genes with both generated expression values and network representation, we compared other packages that use mutual information (minet) [103], graphical lasso (Glasso), and adaptive lasso (parcor) [80]. These tests showed that the specificity of the Glasso and minet packages is better than edges found with lasso. However, the sensitivity was lower, and minet and Glasso generated many more edges than lasso. Adaptive lasso, by contrast, was much sparser than lasso. Adaptive lasso is a two-step process that first uses lasso and then uses mutual information to find edges. To find other associations that a partial

correlation network would miss, we plan on including algorithms that use mutual information and Pearson correlation for the network construction step. We would take additional steps when comparing the networks since these associations include both direct and indirect effects in edge values.

The package parcor [80] uses cross validation to choose the optimum network. We could also use other optimization processes such as BIC or EBIC criteria. The R package WGCNA (weighted gene coexpression network analysis) uses Pearson correlation coefficient moments to find edge strength. This way, all the nodes become connected to one another with different weights. To eliminate low-strength ones, the package uses the scale-free topology plot and chooses the power-law exponent where the curve becomes a straight line, and the network is closest to its scale-free optimum fit. Because of the Pearson correlation, WGCNA includes both direct and indirect edges between nodes, but the optimization process can be used for different network construction steps, such as partial correlation.

As for clustering, we used a fast-greedy method, and our simulation showed that it was successful for clustering genes with similar expressions. There are also other clustering algorithms that we will try to determine the clustering among genes with different expression trends to find gene clusters with other nonlinear interactions that a fast-greedy algorithm would miss. The clustering algorithms were discussed in chapter 2.

8.2 Work on Biological Data

We used rat data collected and processed with RNA-seq. We also have microarray data generated from samples taken from male and female monkeys and rats. Although noisier, microarray data is still useful to identify genes that are variable during developmental changes. We plan to use our algorithm with these samples and to compare our findings. A comparison of male and female monkey data would shed light on the differences of puberty-specific genes according to gender. We also plan to calculate our data with species other than rats and monkeys, such as mice. Data from different species would show gene orthologs with the same functions, which would help us to understand human genes. We are also interested in differences of genes between glial and neuronal cells. These types of data sets would enable us to see the different genes that interact with different parts of the brain at puberty. Another research interest is comparing samples taken from obese and normally fed animals to see how the metabolism and puberty initiation gene sets are related.

Our simulated data results, discussed in chapter 6 show that Jaccard index comparison metric is successful to find impact nodes. The metric we use so far only compares general network

characteristics. To include more genes related to the initiation of puberty in our selection step at each iteration, we plan to include biological phenotype, pathway and known function relations in our network comparisons.

8.3 Work on Other Network Applications

Our INF algorithm is a general algorithm to search impactful node subsets in a complex network. We are interested in using our algorithm in real-life networks seen in different areas. We are interested in searching for impactful people in a social network and impactful papers in a citation network. These networks also show complex network characteristics with cluster and hub formations. We plan to test the efficiency and success rate of our algorithm in different network structures.

Bibliography

[1] A. P. Abreu, A. Dauber, D. B. Macedo, S. D. Noel, V. N. Brito, and J. C. e. al. Gill. Central precocious puberty caused by mutations in the imprinted gene MKRN3. *N. Engl. J. Med.*, 368:2467 – 2475, 2013.

[2] R. S. Ahima, C. B. Saper, J. S. Flier, and J. K. Elmquist. Leptin regulation of neuroendocrine systems. *Front. Neuroendocrinol.*, 21:263 – 307, 2000.

[3] R. Albert. Scale- free networks in cell biology. *J Cell Sci*, 10 2005.

[4] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. *Molecular Biology of the Cell, 4th edition*. Garland Science, New York, 2002.

[5] R. C. AU Gentleman, V. J. Carey, D. M. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, J. Gentry, K. Hornik, T. Hothorn, W. Huber, S. Iacus, R. Irizarry, F. Leisch, C. Li, M. Maechler, A. J. Rossini, G. Sawitzki, C. Smith, G. Smyth, L. Tierney, J. Y. H. Yang, and J. Zhang. Bioconductor: open software development for computational biology and bioinformatics. *Genome Biology*, 5(10), 2004.

[6] A. D. Ay A. Mathematical modeling of gene expression: a guide for the perplexed biologist. . *Crit Rev Biochem Mol Biol.*, 46(2):137–151, 2011.

[7] P. Baldi and A. Long, D. A bayesian framework for the analysis of microarray expression data: regularized t -test and statistical inferences of gene changes. *Bioinformatics*, 17(6):509–519, June 2001.

[8] A. J. Bannister and T. Kouzarides. Regulation of chromatin by histone modifications. *Cell Res.*, 21:381 – 395, 2011.

[9] A.-L. Barabasi and E. Bonabeau. Scale-free networks. *Scientific American*, pages 50–58, 2003.

[10] A. Barrat, M. Barthélemy, R. Pastor-Satorras, and A. Vespignani. The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences*, 101(11):3747–3752, 2004.

[11] P. J. Batista and H. Y. Chang. Long noncoding RNAs: Cellular address codes in development and disease. *Cell*, 152:1298 – 1307, 2013.

[12] K. E. Beale, J. S. Kinsey-Jones, J. V. Gardiner, E. K. Harrison, E. L. Thompson, and M. H. e. al. Hu. The physiological role of arcuate kisspeptin neurons in the control of reproductive function in female rats. *Endocrinology*, 155:1091 – 1098, 2014.

[13] G. Y. Bedecarrats and U. B. Kaiser. Mutations in the human gonadotropin-releasing hormone receptor: Insights into receptor biology and function. *Semin. Reprod. Med.*, 25:368 – 378, 2007.

[14] B. E. Bernstein, T. S. Mikkelsen, X. Xie, M. Kamal, D. J. Huebert, J. Cuff, B. Fry, A. Meissner, M. Wernig, K. Plath, R. Jaenisch, A. Wagschal, R. Feil, S. L. Schreiber, and E. S. Lander. A bivalent chromatin structure marks key developmental genes in embryonic stem cells. *Cell*, 125:315 – 326, 2006.

[15] Z. W. Birnbaum and F. H. Tingey. One-sided confidence contours for probability distribution functions. *The Annals of Mathematical Statistics*, 22(4):592–596, 1951.

[16] H. T. Bjornsson, M. D. Fallin, and A. P. Feinberg. An integrated epigenetic and genetic approach to common human disease. *Trends Genet.*, 20:350 – 358, 2004.

[17] B. Bollobas. *Graph Theory:An introductory course*. Dover Publications, New York, 1984.

[18] S. G. Bourdakou MM, Athanasiadis EI. Discovering gene re-ranking efficiency and conserved gene-gene relationships derived from gene co-expression network analysis on breast cancer data. *Sci Rep.*, 6:20518, 2016.

[19] T. R. Cech and J. A. Steitz. The noncoding RNA revolution-trashing old rules to forge new ones. *Cell*, 157:77 – 94, 2014.

[20] G. Chartrand. *Introductory Graph Theory*. Springer Verlag, 1984.

[21] S. Cho, H. Cho, D. Geum, and K. Kim. Retinoic acid regulates gonadotropin-releasing hormone (GnRH) release and gene expression in the rat hypothalamic fragments and GT1-1 neuronal cells in vitro. *Brain Res Mol Brain Res*, 54:74 – 84, 1998.

[22] F. Chung. The heat kernel as the pagerank of a graph. *Proceedings of the National Academy of Sciences*, 104, 12 2007.

[23] J. Clasadonte, P. Poulain, N. K. Hanchate, G. Corfas, S. R. Ojeda, and V. Prevot. Prostaglandin E2 release from astrocytes triggers gonadotropin-releasing hormone (GnRH) neuron firing via EP2 receptor activation. *Proceedings of the National Academy of Science U.S.A*, 108:16104 – 16109, 2011.

[24] A. Clauset, C. R. Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *Society for Industrial and Applied Mathematics*, 51(4):661–703, 2009.

[25] R. P. Corley, A. M. Beltz, S. J. Wadsworth, and S. A. Berenbaum. Genetic influences on pubertal development and links to behavior problems. *Behavioral Genetics*, 45(3):294 – 312, 2015.

[26] D. L. Cousminer, D. J. Berry, N. J. Timpson, W. Ang, E. Thiering, and E. M. e. al. Byrne. Genome-wide association and longitudinal analyses reveal genetic loci linking pubertal height growth, pubertal timing and childhood adiposity. *Hum. Mol. Genet.*, 22:2735 – 2747, 2013.

[27] G. Csardi and T. Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006.

[28] P. Cukier, H. Wright, T. Rulfs, L. F. Silveira, M. G. Teles, B. B. Mendonca, I. J. Arnhold, S. Heger, A. C. Latronico, S. R. Ojeda, and V. N. Brito. Molecular and gene network analysis of thyroid transcription factor 1 (TTF1) and enhanced at puberty (EAP1) genes in patients with GnRH-dependent pubertal disorders. *Horm. Res. Paediatr.*, 80:257 – 266, 2013.

[29] X. d'Anglemont de Tassigny and W. H. Colledge. The role of kisspeptin signaling in reproduction. *Physiology*, 25(4):207–217, 2010. PMID: 20699467.

[30] F. R. Day, D. J. Thompson, H. Helgason, D. I. Chasman, H. Finucane, and P. e. al. Sulem. Genomic analyses identify hundreds of variants associated with age at menarche and support a role for puberty timing in cancer risk. *Nat. Genet.*, 49:834 – 841, 2017.

[31] A. de la Fuente, N. Bing, I. Hoeschele, and P. Mendes. Discovery of meaningful associations in genomic data using partial correlation coefficients. *Bioinformatics*, 20(18):3565–3574, 07 2004.

[32] N. de Roux, E. Genin, J.-C. Carel, F. Matsuda, J.-L. Chaussain, and E. Milgrom. Hypogonadotropic hypogonadism due to loss of function of the kiss1-derived peptide receptor gpr54. *Proceedings of the National Academy of Sciences*, 100(19):10972–10976, 2003.

[33] J. Deardorff, M. Fyfe, J. P. Ekwaru, L. H. Kushi, L. C. Greenspan, and I. H. Yen. Does neighborhood environment influence girls' pubertal onset? Findings from a cohort study. *BMC Pediatr.*, 12, 2012.

[34] S. Draghici. *Data Analysis Tools for DNA Microarrays*. Chapman and Hall/CRC, 2003.

[35] H. Drees, A. Janßen, S. I. Resnick, and T. Wang. On a minimum distance procedure for threshold selection in tail analysis. *arXiv.org*, 2018.

[36] E. Ducret, G. M. Anderson, and A. E. Herbison. RFamide-related peptide-3, a mammalian gonadotropin-inhibitory hormone ortholog, regulates gonadotropin-releasing hormone neuron firing in the mouse. *Endocrinology*, 150:2799 – 2804, 2009.

[37] B. Dudas and I. Merchenthaler. Journal of Neuroendocrinology. *J. Neuroendocrinol.*, 18:79 – 95, 2006.

[38] S. Durinck, Y. Moreau, A. Kasprzyk, S. Davis, B. De Moor, A. Brazma, and W. Huber. Biomart and bioconductor: a powerful link between biological databases and microarray data analysis. *Bioinformatics*, 21:3439–3440, 2005.

[39] S. Durinck, P. Spellman, E. Birney, and W. Huber. Mapping identifiers for the integration of genomic datasets with the r/bioconductor package biomart. *Nature Protocols*, 4:1184–1191, 2009.

[40] C. F. Elias. Leptin action in pubertal development: Recent advances and unanswered questions. *Trends Endocrinol. Metab.*, 23:9 – 15, 2012.

[41] C. E. Elks, P. Jr, P. Sulem, D. I. Chasman, N. Franceschini, C. He, K. L. Lunetta, J. A. Visser, E. M. Byrne, D. L. Cousminer, D. F. Gudbjartsson, T. Esko, B. Feenstra, J. J. Hottenga, D. L. Koller, Z. Kutalik, P. Lin, M. Mangino, M. Marongiu, P. F. McArdle, A. V. Smith, L. Stolk, S. H. van Wingerden, J. H. Zhao, E. Albrecht, T. Corre, E. Ingelsson, C. Hayward, P. K. Magnusson, E. N. Smith, S. Ulivi, N. M. Warrington, L. Zgaga, H. Alavere, N. Amin, T. Aspelund, S. Bandinelli, I. Barroso, G. S. Berenson, S. Bergmann, H. Blackburn, E. Boerwinkle, J. E. Buring, F. Busonero, H. Campbell, S. J. Chanock, W. Chen, M. C. Cornelis, D. Couper, A. D. Coviello, P. d'Adamo, de Fu, E. J. de Geus, P. Deloukas, A. Doring, G. D. Smith, D. F. Easton, G. Eiriksdottir, V. Emilsson, J. Eriksson, L. Ferrucci, A. R. Folsom, T. Foroud, M. Garcia, P. Gasparini, F. Geller, C. Gieger, V. Gudnason, P. Hall, S. E. Hankinson, L. Ferreli, A. C. Heath, D. G. Hernandez, A. Hofman, F. B. Hu, T. Illig, M. R.

Jarvelin, A. D. Johnson, D. Karasik, K. T. Khaw, D. P. Kiel, T. O. Kilpelainen, I. Kolcic, P. Kraft, L. J. Launer, J. S. Laven, S. Li, J. Liu, D. Levy, N. G. Martin, W. L. McArdle, M. Melbye, V. Mooser, J. C. Murray, S. S. Murray, M. A. Nalls, P. Navarro, M. Nelis, A. R. Ness, K. Northstone, B. A. Oostra, M. Peacock, L. J. Palmer, A. Palotie, G. Pare, A. N. Parker, N. L. Pedersen, L. Peltonen, C. E. Pennell, P. Pharoah, O. Polasek, A. S. Plump, A. Pouta, E. Porcu, T. Rafnar, J. P. Rice, S. M. Ring, F. Rivadeneira, I. Rudan, C. Sala, V. Salomaa, S. Sanna, D. Schlessinger, N. J. Schork, A. Scuteri, A. V. Segre, A. R. Shuldiner, N. Soranzo, U. Sovio, S. R. Srinivasan, D. P. Strachan, M. L. Tammesoo, E. Tikkanen, D. Toniolo, K. Tsui, L. Tryggvadottir, J. Tyrer, M. Uda, R. M. van Dam, J. B. van Meurs, P. Vollenweider, G. Waeber, N. J. Wareham, D. M. Waterworth, M. N. Weedon, H. E. Wichmann, G. Willemsen, J. F. Wilson, A. F. Wright, L. Young, G. Zhai, W. V. Zhuang, L. J. Bierut, D. I. Boomsma, H. A. Boyd, L. Crisponi, E. W. Demerath, C. M. van Duijn, M. J. Econs, T. B. Harris, D. J. Hunter, R. J. Loos, A. Metspalu, G. W. Montgomery, P. M. Ridker, T. D. Spector, E. A. Streeten, K. Stefansson, U. Thorsteinsdottir, A. G. Uitterlinden, E. Widen, J. M. Murabito, K. K. Ong, and A. Murray. Thirty new loci for age at menarche identified by a meta-analysis of genome-wide association studies. *Nat. Genet.*, 42:1077 – 1085, 2010.

[42] B. J. Ellis. Timing of pubertal maturation in girls: An integrated life history approach. *Psychol. Bull.*, 130(6):920 – 958, 2004.

[43] S. Epskamp and E. Fried. A tutorial on regularized partial correlation networks. *Psychological Methods*, 01 2017.

[44] J. J. e. a. Faith. Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles. *PLoS Biol*, 5, 2007.

[45] K. J. Falkenberg and R. W. Johnstone. Histone deacetylases and their inhibitors in cancer, neurological diseases and immune disorders. *Nat. Rev. Drug Discov.*, 13:673 – 691, 2014.

[46] J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *JASA*, 96:1348–1360, 2001.

[47] G. Ficz, M. R. Branco, S. Seisenberger, F. Santos, F. Krueger, T. A. Hore, C. J. Marques, S. Andrews, and W. Reik. Dynamic regulation of 5-hydroxymethylcytosine in mouse ES cells and during differentiation. *Nature*, 473:398 – 402, 2011.

[48] W. Filipowicz, S. N. Bhattacharyya, and N. Sonenberg. Mechanisms of post-transcriptional regulation by microRNAs: Are the answers in sight? *Nat. Rev. Genet.*, 9:102 – 114, 2008.

[49] S. Fletcher and M. Z. Islam. Comparing sets of patterns with the jaccard index. *Australasian Journal of Information Systems*, 22, 2018.

[50] M. R. Fortes, A. Reverter, S. H. Nagaraj, Y. Zhang, N. N. Jonsson, W. Barris, S. Lehnert, G. B. Boe-Hansen, and R. J. Hawken. A single nucleotide polymorphism-derived regulatory gene network underlying puberty in 2 tropical breeds of beef cattle. *J. Anim. Sci.*, 89:1669 – 1683, 2011.

[51] S. Fukusumi, R. Fujii, and S. Hinuma. Recent advances in mammalian RFamide peptides: The discovery and functional analyses of PrRP, RFRPs and QRFP. *Peptides*, 27:1073 – 1086, 2006.

[52] A. Girard, R. Sachidanandam, G. J. Hannon, and M. A. Carmell. A germline-specific class of small RNAs binds mammalian Piwi proteins. *Nature*, 442:199 – 202, 2006.

[53] A. J. Gruber and M. Zavolan. Modulation of epigenetic regulators and cell fate decisions by miRNAs. *Epigenomics*, 5:671 – 683, 2013.

[54] J. U. Guo, Y. Su, C. Zhong, G. L. Ming, and H. Song. Emerging roles of TET proteins and 5-hydroxymethylcytosines in active DNA demethylation and beyond. *Cell Cycle*, 10:2662 – 2668, 2011.

[55] M. Guttman, I. Amit, M. Garber, C. French, M. F. Lin, D. Feldser, M. Huarte, O. Zuk, B. W. Carey, J. P. Cassady, M. N. Cabili, R. Jaenisch, T. S. Mikkelsen, T. Jacks, N. Hacohen, B. E. Bernstein, M. Kellis, A. Regev, J. L. Rinn, and E. S. Lander. Chromatin signature reveals over a thousand highly conserved large non-coding RNAs in mammals. *Nature*, 458:223 – 227, 2009.

[56] K. P. Harden, Mendle, and J. Gene-environment interplay in the association between pubertal timing and delinquency in adolescent girls. *J. Abnorm. Psychol.*, 121(1):73 – 87, 2012.

[57] K. P. Harden, J. Mendle, and N. Kretsch. Environmental and genetic pathways between early pubertal timing and dieting in adolescence: Distinguishing between objective and subjective timing. *Psychol. Med.*, 42(1):183 – 193, 2012.

[58] T. Hastie, , R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2009.

[59] T. Hastie, , R. Tibshirani, and M. Wainwright. *Statistical Learning with Sparsity, The Lasso and Generalizations*. CRC Press, 2015.

[60] C. He, P. Kraft, C. Chen, J. E. Buring, G. Pare, and S. E. e. al. Hankinson. Genome-wide association studies identify loci associated with age at menarche and age at natural menopause. *Nat. Genet.*, 41:724 – 728, 2009.

[61] C. He, P. Kraft, C. Chen, J. E. Buring, G. Pare, S. E. Hankinson, S. J. Chanock, P. M. Ridker, D. J. Hunter, and D. I. Chasman. Genome-wide association studies identify loci associated with age at menarche and age at natural menopause. *Nat. Genet.*, 41:724 – 728, 2009.

[62] S. Heger, C. Mastronardi, G. A. Dissen, A. Lomniczi, R. Cabrera, C. L. Roth, H. Jung, F. Galimi, W. Sippell, and S. R. Ojeda. Enhanced at puberty 1 (EAP1) is a new transcriptional regulator of the female neuroendocrine reproductive axis. *J. Clin. Invest.*, 117:2145 – 2154, 2007.

[63] A. E. Herbison and S. M. Moenter. Depolarising and hyperpolarising actions of GABA(A) receptor activation on gonadotropin-releasing hormone neurons: Towards an emerging consensus. *J. Neuroendocrinol.*, 23:557 – 569, 2011.

[64] H. M. Herz, M. Mohan, A. S. Garruss, K. Liang, Y. H. Takahashi, K. Mickey, O. Voets, C. P. Verrijzer, and A. Shilatifard. Enhancer-associated H3K4 monomethylation by Trithorax-related, the Drosophila homolog of mammalian Mll3/Mll4. *Genes Dev.*, 26:2604 – 2620, 2012.

[65] S. Hinuma, Y. Shintani, S. Fukusumi, N. Iijima, Y. Matsumoto, and M. e. al. Hosoya. New neuropeptides containing carboxy-terminal RFamide and their receptor in mammals. *Nat. Cell Biol.*, 2:703 – 708, 2000.

[66] B. Huang, C. Jiang, and R. Zhang. Epigenetics: The language of the cell? *Epigenomics*, 6:73 – 88, 2014.

[67] R. Jaenisch and A. Bird. Epigenetic regulation of gene expression: How the genome integrates intrinsic and environmental signals. *Nat. Genet.*, 33:Suppl – 245 – 254, 2003.

[68] T. Jenuwein and C. D. Allis. *Science*, 293:1074 – 1080, 2001.

[69] K. K. L. K. Jr. Epigenetic changes coincide with in vitro primate GnRH neuronal maturation. *Endocrinology*, 151:5359 – 5368, 2010.

[70] S. G. K. Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, 3(1), 2004.

[71] W.-H. N. Kadarmideen HN. Building gene co-expression networks using transcriptomics data for systems biology investigations: Comparison of methods using microarray data. *Bioinformation*, 8(18):855–861, 2012.

[72] J. Y. Kim, K. B. Kim, G. H. Eom, N. Choe, H. J. Kee, H. J. Son, S. T. Oh, D. W. Kim, J. H. Pak, H. J. Baek, H. Kook, Y. Hahn, H. Kook, D. Chakravarti, and S. B. Seo. KDM3B is the H3K9 demethylase involved in transcriptional activation of lmo2 in leukemia. *Mol. Cell. Biol.*, 32:2917 – 2933, 2012.

[73] V. N. Kim. Small RNAs just got bigger: Piwi-interacting RNAs (piRNAs) in mammalian testes. *Genes Dev.*, 20:1993 – 1997, 2006.

[74] P. Kiviranta, T. Kuiri-Hänninen, A. Saari, M. Lamidi, L. Dunkel, and U. Sankilampi. Transient postnatal gonadal activation and growth velocity in infancy. *Pediatrics*, 138:1–8, 2016.

[75] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46, 1999.

[76] K. P. Koh, A. Yabuuchi, S. Rao, Y. Huang, K. Cunniff, J. Nardone, A. Laiho, M. Tahiliani, C. A. Sommer, G. Mostoslavsky, R. Lahesmaa, S. H. Orkin, S. J. Rodig, G. Q. Daley, and A. Rao. Tet1 and Tet2 regulate 5-hydroxymethylcytosine production and cell lineage specification in mouse embryonic stem cells. *Cell Stem Cell*, 8:200 – 213, 2011.

[77] C. Kordon, S. V. Drouva, G. M. de la Escalera, and R. I. Weiner. Role of classic and peptide neuromediators in the neuroendocrine regulation of luteinizing hormone and prolactin. *Reproduction*, 1:1621 – 1681, 1994.

[78] E. Korpelainen, J. Tuimala, P. Somervuo, M. Huss, and G. Wong. *RNA-seq Data Analysis: A Practical Approach*. Chapman and Hall/CRC, 2014.

[79] T. Kouzarides. Chromatin modifications and their function. *Cell*, 128:693 – 705, 2007.

[80] N. Kraemer, J. Schaefer, and A.-L. Boulesteix. Regularized estimation of large-scale gene regulatory networks using gaussian graphical models. *BMC Bioinformatics*, 10(384), 2009.

[81] S. Krishnan, S. Horowitz, and R. C. Trievel. Structure and function of histone H3 lysine 9 methyltransferases and demethylases. *Chembiochem*, 12:254 – 263, 2011.

[82] N. Krämer, J. Schäfer, and A.-L. Boulesteix. Regularized estimation of large-scale gene association networks using graphical gaussian models. *BMC Bioinformatics*, 10(1), 2009.

[83] M. V. Kuleshov, M. R. Jones, A. D. Rouillard, N. F. Fernandez, Q. Duan, Z. Wang, S. Koplev, S. L. Jenkins, K. M. Jagodnik, A. Lachmann, M. G. McDermott, C. D. Monteiro, G. W. Gundersen, and A. Ma'ayan. Enrichr: A comprehensive gene set enrichment analysis web server 2016 update. *Nucleic Acids Res.*, 44:90 – 97, 2016.

[84] P. Langfelder and S. Horvath. Wgcna: an r package for weighted correlation network analysis. *BMC Bioinformatics*, 9(1), 2008.

[85] C. Law, M. Alhamdoosh, and S. S. et al. RNA-seq analysis is easy as 1-2-3 with limma, Glimma and edgeR [version 3; peer review: 3 approved]. *F1000Research*, 5:1408, 2018.

[86] M. N. Lehman, L. M. Coolen, and R. L. Goodman. Minireview: Kisspeptin/neurokinin B/dynorphin (KNDy) cells of the arcuate nucleus: A central node in the control of gonadotropin-releasing hormone secretion. *Endocrinology*, 151:3479 – 3489, 2010.

[87] R. Li, L. J., and L. L. Variable selection via partial correlation. *Statistica Sinica*, 27:983–996, 2017.

[88] Liu and A. E. Herbison. Estrous cycle- and sex-dependent changes in pre- and postsynaptic GABAB control of GnRH neuron excitability. *Endocrinology*, 152:4856 – 4864, 2011.

[89] A. Lomniczi, A. Loche, J. M. Castellano, O. K. Ronnekleiv, M. Bosh, G. Kaidar, J. G. Knoll, H. Wright, G. P. Pfeifer, and S. R. Ojeda. Epigenetic control of female puberty. *Nat. Neurosci.*, 16:281 – 289, 2013.

[90] A. Lomniczi and S. R. Ojeda. A role for glial cells of the neuroendocrine brain in the central control of female sexual development. *Astrocytes in (Patho)Physiology of the Nervous System*, pages 487 – 511, 2009.

[91] A. Lomniczi, H. Wright, J. M. Castellano, V. Matagne, C. A. Toro, S. Ramaswamy, T. M. Plant, and S. R. Ojeda. Epigenetic regulation of puberty via Zinc finger protein-mediated transcriptional repression. *Nat. Commun.*, 6:10195, 2015.

[92] A. Lomniczi, H. Wright, J. M. Castellano, K. Sonmez, and S. R. Ojeda. A system biology approach to identify regulatory pathways underlying the neuroendocrine control of female puberty in rats and nonhuman primates. *Horm. Behav.*, 64:175 – 186, 2013.

[93] M. Love, S. Anders, V. Kim, and W. Huber. RNA-Seq workflow: gene-level exploratory analysis and differential expression. *F1000Research*, 4:1070, 2015.

[94] D. B. Macedo, L. F. G. Silveira, D. S. Bessa, V. N. Brito, and A. C. Latronico. Sexual Precocity – Genetic Bases of Central Precocious Puberty and Autonomous Gonadal Activation. *Endocr. Dev.*, 29:50 – 71, 2016.

[95] K. R. Maier R, Zimmer R. A Turing test for artificial expression data. *Bioinformatics*, 29:2603–2609, 2013.

[96] P. C. Maisonpierre, M. M. L. Beau, R. . I. I. I. Espinosa, N. Y. Ip, L. Belluscio, S. M. de la Monte, S. Squinto, M. E. Furth, and G. D. Yancopoulos. Human and rat brain-derived neurotrophic factor and neurotrophin-3: Gene structures, distributions, and chromosomal localizations. *Genomics*, 10:558 – 568, 1991.

[97] A. A. e. a. Margolin. Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, 7 Suppl 1, 2006.

[98] C. Mastronardi, G. G. Smiley, J. Raber, T. Kusakabe, A. Kawaguchi, V. Matagne, A. Dietzel, S. Heger, A. E. Mungenast, R. Cabrera, S. Kimura, and S. R. Ojeda. Deletion of the Ttf1 gene in differentiated neurons disrupts female reproduction without impairing basal ganglia function. *J. Neurosci.*, 26:13167 – 13179, 2006.

[99] V. e. a. Matagne. Thyroid transcription factor 1, a homeodomain containing transcription factor, contributes to regulating periodic oscillations in GnRH gene expression. *Journal of neuroendocrinology*, 24:916–929, 2012.

[100] P. McCaffrey and U. C. Drager. Regulation of retinoic acid signaling in the embryonic nervous system: a master differentiation factor. *Cytokine & Growth Factor Reviews*, 11:233 – 249, 2000.

[101] M. M. McCarthy, A. P. Auger, T. L. Bale, G. J. D. Vries, G. A. Dunn, and N. G. e. al. Forger. The epigenetics of sex differences in the brain. *J. Neurosci.*, 29:12815 – 12823, 2009.

[102] A. Messina, F. Langlet, K. Chachlaki, J. Roa, S. Rasika, N. Jouy, S. Gallet, F. Gaytan, J. Parkash, M. Tena-Sempere, P. Giacobini, and V. Prevot. A microRNA switch regulates

the rise in hypothalamic GnRH production before puberty. *Nat. Neurosci.*, 19:835 – 844, 2016.

[103] P. E. Meyer, F. Lafitte, and G. Bontempi. minet: A r/bioconductor package for inferring large transcriptional networks using mutual information. *BMC Bioinformatics*, 9(1), 2008.

[104] P. E. e. a. Meyer. Information-theoretic inference of large transcriptional regulatory networks. *EURASIP J Bioinform Syst Biol*, 79879, 2007.

[105] S. Mostafavi, D. Ray, D. Warde-Farley, C. Grouios, and Q. Morris. GeneMANIA: A real-time multiple association network integration algorithm for predicting gene function. *Genome biology*, 9 Suppl 1:S4, 02 2008.

[106] J. K. Mueller, A. Dietzel, A. Lomniczi, A. Loche, K. Tefs, W. Kiess, T. Danne, S. R. Ojeda, and S. Heger. Transcriptional regulation of the human KiSS1 gene. *Mol. Cell. Endocrinol.*, 342:8 – 19, 2011.

[107] J. K. Mueller, I. Koch, A. Lomniczi, A. Loche, T. Rulfs, J. M. Castellano, W. Kiess, S. Ojeda, and S. Heger. Transcription of the human EAP1 gene is regulated by upstream components of a puberty-controlling Tumor Suppressor Gene network. *Mol. Cell. Endocrinol.*, 351:184 – 198, 2012.

[108] F. Nagl and e. al. Retinoic acid-induced nNOS expression depends on a novel PI3K/Akt/DAX1 pathway in human TGW-nu-I neuroblastoma cells. *Am. J. Physiol. Cell Physiol.*, 297:1146 – 1156, 2009.

[109] V. M. Navarro, M. L. Gottsch, C. Chavkin, H. Okamura, D. K. Clifton, and R. A. Steiner. Regulation of gonadotropin-releasing hormone secretion by kisspeptin/dynorphin/neurokinin B neurons in the arcuate nucleus of the mouse. *J. Neurosci.*, 29:11859 – 11866, 2009.

[110] V. M. Navarro, M. L. Gottsch, M. Wu, D. García-Galiano, S. J. Hobbs, M. A. Bosch, L. Pinilla, D. K. Clifton, A. Dearth, O. K. Ronnekleiv, R. E. Braun, R. D. Palmiter, M. Tena-Sempere, M. Alreja, and R. A. Steiner. Regulation of NKB Pathways and Their Roles in the Control of Kiss1 Neurons in the Arcuate Nucleus of the Male Mouse. *Endocrinology*, 152(11):4265–4275, 11 2011.

[111] M. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 69, 2004.

[112] M. E. J. Newman. Power laws, pareto distributions and zipf's law. *Contemporary Physics*, 46(5):323–351, 2005.

[113] M. E. J. Newman. Modularity and community structure in networks. *PNAS*, 103(23):8577–8582, 2006.

[114] A. E. Oakley, D. K. Clifton, and R. A. Steiner. Kisspeptin Signaling in the Brain. *Endocrine Reviews*, 30(6):713–743, 10 2009.

[115] S. R. Ojeda, J. Hill, D. F. Hill, M. E. Costa, V. Tapia, A. Cornea, and Y. J. Ma. The Oct-2 POU-domain gene in the neuroendocrine brain: A transcriptional regulator of mammalian puberty. *Endocrinology*, 140:3774 – 3789, 1999.

[116] S. R. Ojeda and M. K. Skinner. San Diego: Academic Press/Elsevier, 3rd edition edition, 2006.

[117] S. R. Ojeda and E. Terasawa. Neuroendocrine regulation of puberty. *Brain Behav.*, 4:589 – 659, 2002.

[118] K. Okamura, W. J. Chung, J. G. Ruby, H. Guo, D. P. Bartel, and E. C. Lai. The Drosophila hairpin RNA pathway generates endogenous short interfering RNAs. *Nature*, 453:803 – 806, 2008.

[119] K. K. Ong, C. E. Elks, S. Li, J. H. Zhao, J. Luan, and L. B. e. al. Andersen. Genetic variation in LIN28B is associated with the timing of puberty. *Nat. Genet.*, 41:729 – 733, 2009.

[120] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. (1999-66), November 1999. Previous number = SIDL-WP-1999-0120.

[121] A. S. Parent, A. E. Mungenast, A. Lomniczi, U. S. Sandau, E. Peles, and M. A. e. al. Bosch. A contactin-receptor-like protein tyrosine phosphatase beta complex mediates adhesive communication between astroglial cells and gonadotrophin-releasing hormone neurones. *J. Neuroendocrinol.*, 19:847 – 859, 2007.

[122] J. Parkash and G. Kaur. Neuronal-glial plasticity in gonadotropin-releasing hormone release in adult female rats: Role of the polysialylated form of the neural cell adhesion molecule. *J. Endocrinol.*, 186:397 – 409, 2005.

[123] A. Pekowska, T. Benoukraf, J. Zacarias-Cabeza, M. Belhocine, F. Koch, H. Holota, J. Imbert, J. C. Andrau, P. Ferrier, and S. Spicuglia. H3K4 tri-methylation provides an epigenetic signature of active enhancers. *EMBO J.*, 30:4198 – 4210, 2011.

[124] A. D. Perera, C. F. Lagenaur, and T. M. Plant. Postnatal expression of polysialic acid-neural cell adhesion molecule in the hypothalamus of the male rhesus monkey (Macaca mulatta). *Endocrinology*, 133:2729 – 2735, 1993.

[125] J. R. Perry, L. Stolk, N. Franceschini, K. L. Lunetta, G. Zhai, and P. F. e. al. McArdle. Meta-analysis of genome-wide association data identifies two loci influencing age at menarche. *Nat. Genet.*, 41:648 – 650, 2009.

[126] T. Plant and S. Witchel. *Puberty in Nonhuman Primates and Humans*, volume 2, pages 2177–2230. 12 2006.

[127] T. M. Plant. 60 YEARS OF NEUROENDOCRINOLOGY: The hypothalamo-pituitary-gonadal axis. *J. Endocrinol.*, 226:41 – 54, 2015.

[128] C. L. Poirel, A. Rahman, R. R. Rodrigues, A. Krishnan, J. R. Addesa, and T. M. Murali. Reconciling differential gene expression data with molecular interaction networks. *Bioinformatics*, 29(5):622–629, 01 2013.

[129] M. C. Poling, J. Kim, S. Dhamija, and A. S. Kauffman. Development, Sex Steroid Regulation, and Phenotypic Characterization of RFamide-Related Peptide (Rfrp) Gene Expression and RFamide Receptors in the Mouse Hypothalamus. *Endocrinology*, 153:1827 – 1840, 2012.

[130] V. Prevot. Glial-neuronal-endothelial interactions are involved in the control of GnRH secretion. *J. Neuroendocrinol.*, 14:247 – 255, 2002.

[131] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017.

[132] J. A. Ramey. *clusteval: Evaluation of Clustering Algorithms*, 2012. R package version 0.1.

[133] M. Rees. The age of menarche. *ORGYN.*, 4:2–4, 1995.

[134] A. Reverter and E. K. F. Chan. Combining partial correlation and an information theory approach to the reversed engineering of gene co-expression networks. *Bioinformatics*, 24(21):2491–2497, 09 2008.

[135] M. E. Ritchie, B. Phipson, D. Wu, Y. Hu, C. W. Law, W. Shi, and G. K. Smyth. limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Research*, 43(7), 2015.

[136] A. J. Ruthenburg, H. Li, D. J. Patel, and C. D. Allis. Multivalent engagement of chromatin modifications by linked binding modules. *Nat. Rev. Mol. Cell Biol.*, 8:983 – 994, 2007.

[137] U. S. Sandau, A. E. Mungenast, Z. Alderman, S. P. Sardi, A. I. Fogel, and B. e. al. Taylor. SynCAM1, a Synaptic Adhesion Molecule, is Expressed in Astrocytes and Contributes to erbB4 Receptor-Mediated Control of Female Sexual Development. *Endocrinology*, 152:2364 – 2376, 2011.

[138] U. S. Sandau, A. E. Mungenast, J. McCarthy, T. Biederer, G. Corfas, and S. R. Ojeda. The Synaptic Cell Adhesion Molecule, SynCAM1, Mediates Astrocyte-to-Astrocyte and Astrocyte-to-GnRH Neuron Adhesiveness in the Mouse Hypothalamus. *Endocrinology*, 152:2353 – 2363, 2011.

[139] M. Schena. *Microarray Analysis*. Wiley-Liss, 2002.

[140] B. Schuettengruber, A. M. Martinez, N. Iovino, and G. Cavalli. Trithorax group proteins: Switching genes on and keeping them active. *Nat. Rev. Mol. Cell Biol.*, 12:799 – 814, 2011.

[141] S. B. Seminara, S. Messager, E. E. Chatzidaki, R. R. Thresher, J. S. Acierno, J. K. Shagoury, Y. Bo-Abbas, W. Kuohung, K. M. Schwinof, A. G. Hendrick, D. Zahn, J. Dixon, U. B. Kaiser, S. A. Slaugenhaupt, J. F. Gusella, S. O'Rahilly, M. B. Carlton, W. F. Crowley, S. A. Aparicio, and W. H. Colledge. The gpr54 gene as a regulator of puberty. *New England Journal of Medicine*, 349(17):1614–1627, 2003. PMID: 14573733.

[142] M. Shahab, C. Mastronardi, S. B. Seminara, W. F. Crowley, S. R. Ojeda, and T. M. Plant. Increased hypothalamic gpr54 signaling: A potential mechanism for initiation of puberty in primates. *Proceedings of the National Academy of Sciences*, 102(6):2129–2134, 2005.

[143] Y. Shi, F. Lan, C. Matson, P. Mulligan, W. Jr, P. A. Cole, R. A. Casero, and Y. Shi. Histone demethylation mediated by the nuclear amine oxidase homolog LSD1. *Cell*, 119:941 – 953, 2004.

[144] S. A. Simon, J. Zhai, R. S. Nandety, K. P. McCormick, J. Zeng, D. Mejia, and B. C. s. Short-read sequencing technologies for transcriptional analyses. *Annual Review of Plant Biology*, 60(1):305–333, 2009. PMID: 19575585.

[145] P. S. Steeg, T. Ouatas, D. Halverson, D. Palmieri, and M. Salerno. Metastasis suppressor genes: Basic biology and potential clinical use. *Clin. Breast Cancer*, 4:51 – 62, 2003.

[146] A. Strobel, T. Issad, L. Camoin, M. Ozata, and A. D. Strosberg. A leptin missense mutation associated with hypogonadism and morbid obesity. *Nat. Genet.*, 18:213 – 215, 1998.

[147] S. H. Strogatz. Exploring complex networks. *Nature*, 410(6825), 2001.

[148] J. Stuart, E. Segal, D. Koller, and S. Kim. A gene-coexpression network for global discovery of conserved genetic modules. *Science*, 302:249–255, 2003.

[149] A. e. a. Subramanian. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America*, 102:15545–15550, 2005.

[150] P. Sulem, D. F. Gudbjartsson, T. Rafnar, H. Holm, E. J. Olafsdottir, and G. H. e. al. Olafsdottir. Genome-wide association study identifies sequence variants on 6q21 associated with age at menarche. *Nat. Genet.*, 41:734 – 738, 2009.

[151] M. Tahiliani, K. P. Koh, Y. Shen, W. A. Pastor, H. Bandukwala, Y. Brudno, S. Agarwal, L. M. Iyer, D. R. Liu, L. Aravind, and A. Rao. Conversion of 5-methylcytosine to 5-hydroxymethylcytosine in mammalian DNA by MLL partner TET1. *Science*, 324:930 – 935, 2009.

[152] C. Tanikawa, Y. Okada, A. Takahashi, K. Oda, N. Kamatani, and M. e. al. Kubo. Genome wide association study of age at menarche in the Japanese population. *PLoS ONE*, 8, 2013.

[153] M. G. Teles, S. D. Bianco, V. N. Brito, E. B. Trarbach, W. Kuohung, and S. e. al. Xu. A GPR54-activating mutation in a patient with central precocious puberty. *N. Engl. J. Med.*, 358:709 – 715, 2008.

[154] E. Terasawa and D. L. Fernandez. Neurobiological Mechanisms of the Onset of Puberty in Primates*. *Endocrine Reviews*, 22(1):111–151, 02 2001.

[155] A. K. Topaloglu, F. Reimann, M. Guclu, A. S. Yalin, L. D. Kotan, and K. M. e. al. Porter. TAC3 and TACR3 mutations in familial hypogonadotropic hypogonadism reveal a key role for Neurokinin B in the central control of reproduction. *Nat. Genet.*, 41:354 – 358, 2008.

[156] A. K. Topaloglu, J. A. Tello, L. D. Kotan, M. N. Ozbek, M. B. Yilmaz, and S. e. al. Erdogan. Inactivating KISS1 mutation and hypogonadotropic hypogonadism. *N. Engl. J. Med.*, 366:629 – 635, 2012.

[157] K. A. Topaloglu and L. D. Kotan. Genetics of Hypogonadotropic Hypogonadism. *Endocr. Dev.*, 29:36 – 49, 2016.

[158] C. A. Toro, H. Wright, C. F. Aylwin, S. R. Ojeda, and A. Lomniczi. Trithorax dependent changes in chromatin landscape at enhancer and promoter regions drive female puberty. *Nature Communications*, 9, 2018.

[159] K. Tsutsui, G. E. Bentley, G. Bedecarrats, T. Osugi, T. Ubuka, and L. J. Kriegsfeld. Gonadotropin-inhibitory hormone (GnIH) and its control of central and peripheral reproductive function. *Front. Neuroendocrinol.*, 31:284 – 295, 2010.

[160] T. van den Bulcke, K. van Leemput, B. Naudts, P. van Remortel, H. Ma, A. Verschoren, B. De Moor, and K. Marchal. Syntren: a generator of synthetic gene expression data for design and analysis of structure learning algorithms. *BMC Bioinformatics*, 43(7), 2006.

[161] Y. Wakabayashi, T. Nakada, K. Murata, S. Ohkura, K. Mogi, and V. M. N. et. al. Neurokinin B and dynorphin A in kisspeptin neurons of the arcuate nucleus participate in generation of periodic oscillation of neural activity driving pulsatile gonadotropin-releasing hormone secretion in the goat. *J. Neurosci.*, 30:3124 – 3132, 2010.

[162] A. Wald and J. Wolfowitz. Confidence limits for continuous distribution functions. *The Annals of Mathematical Statistics*, 10(2):105–118, 1939.

[163] X. Wang. *Next-Generation Sequencing Data Analysis*. CRC Press, Boca Raton, FL, 2016.

[164] D. Warde-Farley, S. L. Donaldson, O. Comes, K. Zuberi, R. Badrawi, P. Chao, M. Franz, C. Grouios, F. Kazi, C. T. Lopes, A. Maitland, S. Mostafavi, J. Montojo, Q. Shao, G. Wright, G. D. Bader, and Q. Morris. The GeneMANIA prediction server: Biological network integration for gene prioritization and predicting gene function. *Nucleic Acids Res.*, 38:214 – 220, 2010.

[165] S. Wasserman and K. Faust. *Social network analysis : methods and applications*. Cambridge, New York, 1994.

[166] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684), 1998.

[167] J. Whittaker. *Graphical Models in Applied Multivariate Statistics*. John Wiley & Sons, 1990.

[168] C. Wiwie, J. Baumbach, and R. Rottger. Comparing the performance of biomedical clustering methods. *Nature Methods*, 12:1033–1038, 2015.

[169] S. Wray. . *Journal of neuroendocrinology*, 22:743–753, 2010.

[170] R. Xu and D. C. Wunsch. Clustering algorithms in biomedical research: A review. *IEEE Reviews in Biomedical Engineering*, 3:120–154, 2010.

[171] C. Yang, J. Ye, X. Li, X. Gao, K. Zhang, L. Luo, J. Ding, Y. Zhang, Y. Li, H. Cao, Y. Ling, X. Zhang, Y. Liu, and F. Fang. DNA Methylation Patterns in the Hypothalamus of Female Pubertal Goats. *PLoS ONE*, 11:e0165327, 2016.

[172] K. T. Yeung, S. Das, J. Zhang, A. Lomniczi, S. Ojeda, C. F. Xu, T. A. Neubert, and H. H. Samuels. A novel transcription complex that selectively modulates apoptosis of breast cancer cells through regulation of FASTKD2. *Mol. Cell. Biol.*, 31:2287 – 2298, 2011.

[173] M. Yun, J. Wu, J. L. Workman, and B. Li. Readers of histone modifications. *Cell Res.*, 21:564 – 578, 2011.

[174] D. e. a. Zhou. Learning with local and glocal consistency. *Adv. Neural Inf. Process. Syst.*, 16:321–328, 2004.

[175] X. e. a. Zhu. Semi-supervised learning using Gaussian fields and harmonic functions. *The Twientieth International Conference on Machine Learning*, pages 912–919, 2003.

[176] H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.