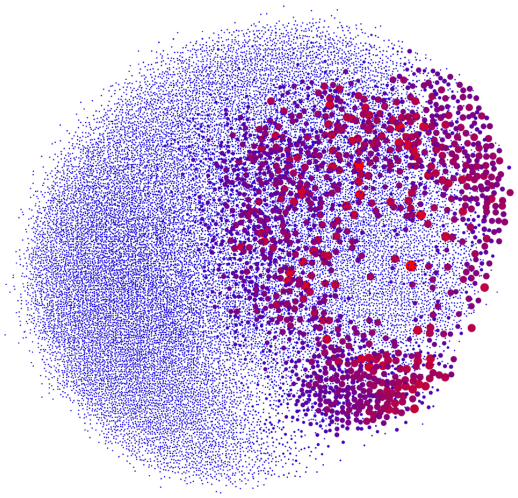
Regulatory Gene-Gene networks based on regression trees.

Methods for validating, comparing network evolution and estimating the graph properties of essential genes.

Author: Michael Forsmann



University Name: University of Copenhagen
Department Name: Niels Bohr institute
Submitted : 20-5-2020
Advisor : Ala Trusina

Degree : Master of physics

Abstract

We set out to make a method to capture the gene-gene regulation of single-celled RNA sequencing data by using a single regression tree. This data was taken from mice, respectively 6.5, 7.5, 8.5 and 9.5 days after fertilization. We called the method single tree network(STN). We compared our method to a well-known method in the field called GRNboost2. After comparing the methods, we used the inferred networks by STN and GRNboost2 to predict the underlying changes to the network structure and then related it to the underlying biology. Lastly, we observed if the methods captured the difference in between essential and non-essential genes for survival in the network properties. Based on the results, we could conclude that STN and GRNboost2 can not predict regulation, since they either predicted slightly better or worse than random. This performance could be because of how we validate the methods, but a deficient performance would happen either way. The tree does capture some biological events since the properties of essential and viable genes are different for both methods. This lead us to believe that essential genes are expressed in more celled than non-essential genes. We could also capture a change in the network structure between the days. GRNboost2 predicted that over the period, the networks gets less and less hierarchical, where STN predicted the opposite result.

Resume på dansk

Vores forskning startede ved at udvikle en metode til at udlede gen-gen regulering fra enkelt-cellede RNA sekventerings data ved at bruge et regressions træ. Dataen blev opsamlet fra mus henholdsvis 6.5, 7.5, 8.5 og 9.5 dage efter befrugtelse. Vi opkaldte vores metode single tree network(STN). Vi sammenlignede vores metode med en velkendt metode i feltet kaldt GRNboost2. Efter vi havde sammenlignet metoderne brugte vi STN og GRNboost2 til at forudse den uderliggende udvikling i netværks strukturerne og herefter sammenlignede vi den med underliggende biologi. Til sidst undersøgte vi om metoderne kunne opfange forkellene i netværk egenskaberne for essentielle og ikke essentielle-gener. Baseret på vores resultater kan vi konkludere at STN og GRNboost2 ikke kan forudse gen-regulering, da deres evne til at forudse gen-regulation enten er en smule bedre eller en smule værre end tilfældigt. Dette resultat kunne stamme fra vores måde at validere metoderne på, men metoderne ville stadig have lav præcision til forudselse af regulering. Metoderne opfangede stadig nogle mønstre i dataen, da de kunne opfange forskellen på essentielle gener og ikke essentielle gener på deres netværks egenskaber. Dette leder os til konklusionen at essentielle gener er udtrykt i flere celler end ikke essentielle gener. Vi kunne også opfange ændringer i netværks strukturerne over dagene. GRNboost2 forudså at netværkene fik mindre hierarkisk struktur over perioden. STN forudså det stik modsatte resultat.

Contents

Chapter 1

Introduction

During the last couple of years, the technology for collecting large samples of cells has been improving to a point where it is possible to collect between $10^4 - 10^7$ cells in animal experiments depending on the set-up. Inside the cells, we have genes that code for different proteins. With these large samples of cells and genes, we can try to predict how different genes regulate each other.
Predicting which genes regulate each other is a complicated task when working with more complex organisms. The mouse is one such organism and predicting the regulators is challenging for multiple reasons. The first reason is that the number of cells and genes are way higher than in simple organisms like yeast, so powerful computers are needed. The second reason is that single-celled data has a high noise level, which means we sometimes use noise to predict the regulators, resulting in awful predictions. The last thing is that mammals' genes do not have to be placed in the same cells to regulate each other, this is one of the primary reasons why modern methods do not perform very well on mammalian genetics.
We will try to predict regulator using methods based on regression trees. The methods are based on the assumption that genes are likely to regulate each other if genes are placed in the same cells. Regression trees also assume that genes that lie in the same cells are more likely to regulate each other. We will use GRNboost2, a more complex and well-tested method, and compare it to a single regression tree network(STN) that we have developed. It is a single tree with no restriction. After adding all the genes' predicted regulators, we can infer a network.
We will infer the networks from a mouse on the day 6.5 to 9.5 after fertilization. This period is where the shape of the mouse and the formation of organs occur. Therefore, we will look at how the inferred networks evolve during these days and try to estimate what makes essential genes for survival special based on the network properties. Lastly we will validate how much trust we can put on the predictions of the Regulators in the inferred networks by comparing them to networks where we know which genes are regulating each other.

Chapter 2

Theory

2.1 Introduction to genetics of Mammalian species

2.2 Research question section

Hypothesis
Can we infer a gene regulatory network based on regression trees that capture the underlying gene-gene regulation of a mouse's embryonic developmental stages?
To answer this question, we have four steps we need to go through

1) What data are we using, and where does it come from?

2) What methods are we using to infer the network, and what are the underlying assumptions of the method?

3)How does the network structure evolve over the days?

4) Can we find the profiles of essential genes' for survival?

5) How do we validate our network's performance?

2.2.1 Introduction Data-set

The data comes from an in vivo experiment. The experiment uses a method to extract cells and count the genes inside called single-celled RNA sequencing (scRNAseq). The experiment captures the mouse gene expression by performing scRNAseq on as many cells as possible on days 6.5, 7.5, 8.5, and 9.5. The experiment is close to the underlying biology since the embryos are grown inside a mouse. The data consists of 9000 cells and 23000 genes; figure 2.1 shows the

number of genes and cells for the different days in experiment. The describing the data-set is not assemble yest since the authors are not done with it yet but the data can be downloaded here here

Day	6.5	7.5	8.5	9.5
Cell count	745	2287	2986	849
Gene count	18315	20615	20829	17800

Figure 2.1: Overview of the experiment, cell count and gene count for the different days. Very low count of cells at day 9.5.

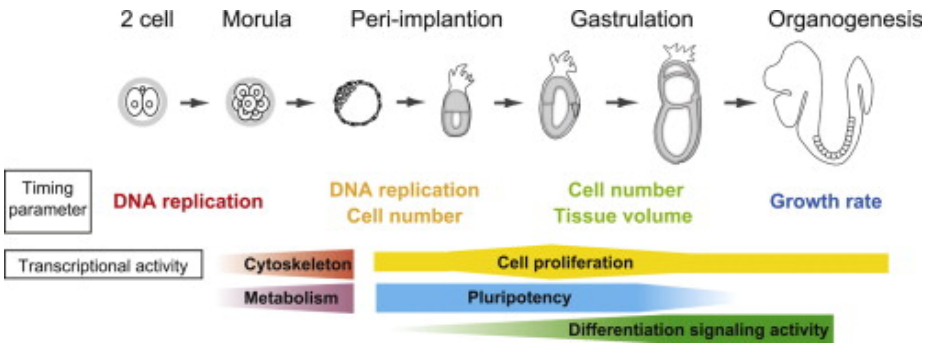


Figure 2.2: Important stages of early mouse development. Look at the changes from gastrulation to early Organogenesis these stages is a direct overlap with day 6.5 to 9.5 after fertilization [1]

The period between day six and day eight is called gastrulation; in this period the number of cells and the volume of the embryo drastically increases. In the late stages of gastrulation, the formation of crucial organs starts, and the number of stem cells decrease due to cell differentiation. The period from day 8.5 to the birth of a mouse is called organogenesis. In organogenesis, the rest of the stem cells differentiate into different types of cells, and the volume of the rest of the organs increase drastically.

2.3 How do we infer the Regulators?

After reading multiple articles on implementing the best ways of inferring regulatory gene-gene networks, we chose to work with models based on regression trees. Regression tree methods are well tested and have some of the best results in the field . The data they are using in the articles are from yeast data, which has less complex regulation than that of a mouse, so we expect the models in the articles to have better validation performance. To infer networks with regression trees we first need to understand the framework.

2.3.1 Introduction to regression trees

Regression trees use training data to predict future outcomes of samples. Samples are specific objects or events we want to know the outcome of, this could be the price of a car. To make the best prediction for the outcome, we need to give the regression tree multiple features, which could potentially influence the outcome. So if we are trying to predict the cost of a car, we could give it the features: How old is the car? What was the start price? Which brand is it?

2.3.2 How is a decision tree structured?

Figure 2.3 shows the structure of a decision tree, with a mother node at the top of the tree. The mother node is the first split and has the most impact on the regression.
Root nodes are all nodes that contain a split; the further away a root node is from the mother node, the less significant it is for the regression. The distance from the mother is called the depth.
Each root node contains predictions so the algorithm can estimate the performance of the split, these estimations are called Residuals. The algorithm needs a function to measure the performance, which is called a loss function. At the bottom of the tree, we have the predictions which are called the leaf nodes.

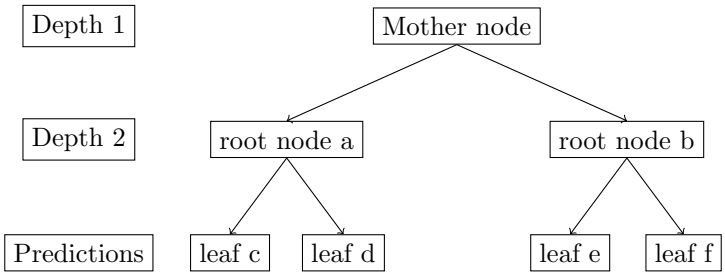


Figure 2.3: Figure of tree structure with depth 2

The standard loss function used for regression trees is the squared distance between the true value and the predicted value, also written as

$$R_{sum} = 1/N_m \sum_{y \in Q_m} (y - \bar{y}_m)^2 \tag{2.1}$$

R_{sum}= The sum over the Residual
$y_{i,m}$= True value of sample i
$y_{i,m}$ predicted value for split m.

2.3.3 How do we find the best feature and threshold for a split?

Given training data x and target data y, we want to find a threshold t and a feature j for a subset of the training data Q_m, which minimizes the total distance. First, we split the subset up into two nodes Q_m^{left} and a Q_m^{right}, and we define the parameter $\theta = (j_m, t_m)$ we can write this as:

$$Q_m^{left}(\theta) = \{(x, y)|x_j <= t_m\}, Q_m^{right}(\theta) = Q_m \notin Q_m^{left} \tag{2.2}$$

We can then describe the quality of the split by using the loss function H, which is a measure of distance between the predicted and true value of y, the total function for the split is then

$$G(Q_m, \theta) = \frac{N_m^{left}}{N_m} H(Q_m^{left}(\theta)) + \frac{N_m^{right}}{N_m} H(Q_m^{right}(\theta)) \tag{2.3}$$

where N_m is the number of samples in the split m.

The best parameters are found by minimizing the loss function H.

$$\theta^* = argmin_\theta(G(Q_m, \theta)) \tag{2.4}$$

2.3.4 Feature importance

Features are the different properties in our data x. If we want to predict the number of fires in a specific area, this could be: temperature, number of people grilling, and humidity in the air. The feature importance measures how efficient a feature is at predicting the outcome of the samples, and is formalized the following way:

$$FI_j = \sum_{m \in_j} R^j_{sum,m} \frac{\sum_{m \in_j} N_{m,j}}{N} \tag{2.5}$$

$R^j_{sum,i}$: The summed residual of split m feature j
$N_{m,j}$: Number of samples in split m feature j
FI_j: feature importance of feature j

2.4 iRGN with multiple regression trees

2.4.1 Stochastic Gradient Boosting

This method uses multiple trees in sequential order. Each tree is called an estimator, the first tree fits the residual between the true value and the mean of each sample, the next tree fits the residuals of the previous tree, etc. So in the case of three estimators, we would have three trees, and if we set the maximum depth to one, this is visualized in figure 2.4.

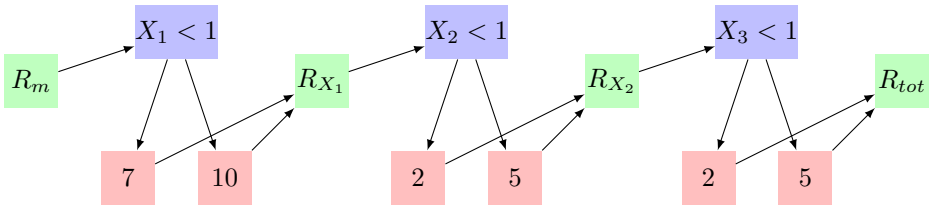


Figure 2.4: Example of gradient boosting algorithm with three estimators and a depth of one

Each tree Reduces the Residuals by a small amount. This reduction is called the learning rate or the gradient of each tree. This can be a fixed value or dependent on how well the tree reduces the total error, we will only be using a fixed value of the learning rate. To calculate the residual after each tree, we subtract the residual from the previous tree with the residual of the present tree multiplied by the learning rate written as

$$R_{i+1} = R_i - R_{i+1} * \epsilon \tag{2.6}$$

R_{i+1}: Residual after i trees
R_i: Residual after i-1 trees
ϵ: learning rate

To calculate the total Residual of the example in figure 2.4, we can use equation 2.6 and input the Residual for each tree.

$$R_{tot} = R_M - R_{X_1} * \epsilon - R_{X_2} * \epsilon - R_{X_3} * \epsilon \tag{2.7}$$

The main reason for having a learning rate is to make sure multiple samples can have an impact on the prediction of the outcome. This makes the algorithm more likely to predict the outcome using multiple features. This means the algorithm is more robust to changes in the data and also gives way more candidates for features. Stochastic Gradient Boosting uses the algorithm "Gradient descent" to compute the local minimums of a district function with low computational time. It is used to find the minimum of the loss functions and predict the best split of the regression trees. The Gradient descents and stopping after N tries, this represents the boosting part of the algorithm. Each tree a get fraction of the features and fraction of the samples to make sure each tree does not find the same features/samples, this is the stochastic part of the algorithm. All these changes make a more complex version of a single regression tree, but it has a lower variance between each run of sub-sampled data and is not biased towards specific features, which would normally dominate a standard regression tree.

2.4.2 Feature importance for Stochastic Gradient Boosting

each tree conri

$$FI_j = \sum_{m \in_j} R^j_{sum,m} \frac{\sum_{m \in_j} N_{m,j}}{N} \tag{2.8}$$

$R^j_{sum,i}$: The summed residual of split m feature j
$N_{m,j}$: Number of samples in split m feature j
FI_j: feature importance of feature j

2.4.3 Methods to infer a Gene-gene Regulatory Network(IGRN)

The true goal of standard machine learning is to generate a general model, which can find useful patterns in the data and predict future data points, classes, values, etc.
This means the models need to be robust, and the goal is not to be too specialized to a specific batch of data. This is not the case when we want to predict regulatory Genes; we want the model to fit specifically to the gene without including the underlying noise and indirect regulation. This is a difficult task since biological data has an extreme amount of noise and the regulation of a protein is sometimes a network of proteins. This makes it hard to predict the direct regulators instead of the indirect.

2.5 Inferring Regulatory Gene-gene networks

When we use a regression tree to infer a regulatory network, we assume that the gene's gene-expression is a combination of other genes' gene expressions multiplied by a weight written mathematically as

$$x_n = f(X_{\neq n}) = w_1 * x_1 + w_2 * x_2 + w_3 * x_3 + ...w_N * x_N = \sum_{i \neq n} w_i \cdot x_i \tag{2.9}$$

x_n: cell expression of gene n
X: the total matrix of genes
FI_n: the feature importants of gene n.

GRN approach using regression trees

When we are working with scRNAs data, the rows are cells and columns are genes, shown in figures 2.5 a, b, and c.

$C\|G$	g_1	g_2	g_3	g_4
c_1	10	90	0	0
c_2	40	80	80	0
c_3	0	0	200	200

(a) RNA seq data

$C\|G$	g_1	g_2	g_3	g_4
c_1	0.1	0.9	0	0
c_2	0.2	0.4	0.4	0
c_3	0	0	0.5	0.5

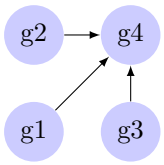
(b) Table a where each row is normalised by the sum of the cells

$C\|G$	g_1	g_2	g_3	g_4
c_1	0.1	0.9	0	0
c_2	0.2	0.4	0.4	0
c_3	0	0	0.5	0.5

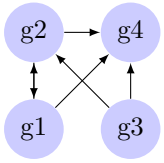
(c) We use the input green box to fit to the output y red box

$G_in\|G_out$	g_1	g_2	g_3	g_4
g_1	0	0	0	0.25
g_2	0	0	0	0.25
g_3	0	0	0	0.5
g_4	0	0	0	0

(d) We return the feature imports to the columns of a new matrix G



(e) Graph representation of table d



(f) Example of what the full network would look like

Figure 2.5: How to construct a network with a regression tree from single celled RNA sequencing data

Steps to create a GRN from Single celled data

- Normalise each row by its sum, see figure 2.5 b
- Remove cells and genes when the sum is equal to zero
- Choose a Regression tree method to infer the network regulation
- Use the method to fit the matrix X without feature j to the feature j, see figure 2.5 c
- Return the feature importance from the fit and store it in the column j of a m * m matrix which we call G, see figure 2.5 d
- Go back to step 4 until we reach $j = m$
- Set all values in G to one
- Convert the matrix G to a directed Graph, see figure 2.5 f

Figure 2.6

In figure 2.5, we have a step-by-step visual introduction to inferring a GRN from single-celled data with a regression tree method, and in figure 2.6, we have a more detailed text version of figure 2.5.

2.6 Single tree network(STN) and GRNboost2

Single tree network(STN) is our attempt to create a simplistic model to infer the gene-gene regulatory network. It is one regression tree that massively overfits, meaning it keeps grouping genes until there are no fits left or the loss function is zero. GRNboost2 is a well-tested method for inferring gene-gene regulatory networks from scRNAs data. It uses multiple trees with depth one and finds all likely regulators. To get a more detailed understanding of the models, we will go through the parameters of regression trees one by one. In generating the STN, we considered maximum depth, minimum decrease in the distance for a split, sub-sampling, maximum Numbers of features, number of estimators, and learning rate.

Maximum depth

The maximum depth is the longest path from the mother node to the leaf nodes. We choose not to use the maximum depth in STN since some genes' gene expressions are very complex, and other genes' gene expressions are very simple. This means that we would be limiting the number of inferred regulators for the more complex genes which is not preferred. STN first stops when no minimization of the loss function is possible. GRNboost2 has a maximum depth of one since there are multiple estimators.

Decrease in total distance.

A smaller decrease in the loss function means you are less likely to be a good fit for a regulator, but if a gene with almost the same gene expression is picked up, we will find one, maybe two genes. To make sure we find more than one regulator pr gene, we need to make sure it captures as many as possible for the limitations of STN we used the default which is zero. GRNboost2 has other parameters for regularisation of overfitting, so it is zero as well.

Sub-sampling

Sub-sampling is the number of samples each tree has access to, this can be a specific number or just a fraction of the samples. STN uses fraction one, and GRNboost2 uses 0.9.

Maximum number of features

Max features are the fraction of features each tree has access to; STN Uses one, and GRNboost2 uses 0.1

Number of estimators

The number of estimators is how many trees are used for making a prediction. STN uses one where GRNboost2 uses 5000

Learning rate

The learning rate is how much each tree contributes to reducing the Residuals, STNs learning rate is one, and since GRNboost2 has multiple trees, the learning rate is 0.01

2.6.1 Summing up the difference

STN is a single tree that is massively over-fitting with no constraints on parameters. Besides that, features should have a least 0.005 absolute Pearson correlation to be considered for the fit. GRNboost2 is 5000 trees with depth one, which has access to 90 % of the sources and 10 % of the features, and each tree contributes 1 % to the prediction, and the features can be picked multiple times. Therefore STN is one massive tree, versus GRNboost2 which is an extreme amount of small trees.

2.7 Graph theory

Graph theory is a computational/mathematical tool that looks at the relationship between objects or events. Graphs have a very simple framework but can describe very complex systems. This is one of the major reasons it is one of the most used methods for studying systems with a large number of events in relation to each other. A Graph consists of nodes symbolized as a circle, which is an event or an object, and the relation between them is an edge. Graphs with edges pointing both ways are undirected see figure 2.7a, and graph which are directed has edges which point from one node to another node, see figure 2.7 b.

(a) Undirected Graph (b) Directed Graph

Figure 2.7

Edges have a weight between 0 and 1 which tells us how related the two nodes are, we will use a binary version where it is either zero or one and only consider directional Graphs. In Graph theory, we have properties, the properties describe: the number of relations to a node, how central a node lies in the graph, which link has the most traffic and etc. We have chosen to focus on four properties in_degree, out_degree, betweenness, and flow see section 2.7.1.

2.7.1 Directional Graph properties

In_degree

In_degree is the number of edges pointing towards a specific node, this tells us how many nodes are related to the node. Looking at figure 2.8, we can see B has in_degree one with an edge pointing from A to B, and the node with the highest in_degree is G which has an in_degree of three.

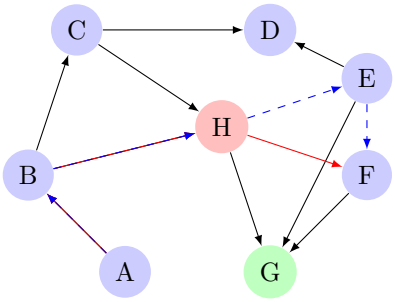


Figure 2.8: Directional Graph example with 8 nodes and 12 edges,the red node indicates the node with highest out_degree and the green node is the node with highest in_degree

Out_Degree

The out_degree is the number of edges pointing out from a specific node, this is how many nodes a specific node has a relation to. Looking at figure 2.8, we can see that B has an out_degree of two, one from B to C and one from B to H. The node with the highest out_degree is node H which has out_degree equal to three.

Betweenness

Betweenness is a way to find central nodes for the network structure. Betweenness is based on the assumption that the network uses the shortest path for transportation. The shortest path is the minimum number of edges between two nodes. When we want to calculate the shortest path between two nodes, we start at the source node and end at The target node. In figure 2.8 we use A as source node and F as target node. We start by moving from node A to B to H. From node H there are two paths: from H to E to F and from H to F. Since H to F is the shortest path, the shortest path is equal to three. To calculate the betweenness, we need to calculate the shortest path between all nodes as both target nodes and as source nodes. To calculate the betweenness, we then count the shortest going through a node divided by the total number of shortest paths in the network

$$C_B(n) = \sum_{s,t \in G} \frac{\sigma(s,t|n)}{\sigma(s,t)} \tag{2.10}$$

$C_B(n)$ betweenness of node n
$\sigma(s,t|v)$:number of shortest path from source s to target t going through node n
$\sigma(s,t)$: total number of shortest path from source s to target t
$\sum_{s,t\in G}$: sum over all nodes as sources and target in graph G

Directed Flow

The flow is an estimate of how many times a node is visited during a directed version of a random walk, called a random surf. In a direct network, a random walk would get stuck if a node has no out_degree, see figure 2.9 node D. A random surf fixes this by jumping to a random node if it gets stuck. It also fixes the problem if an area in the network only has out_degree to nodes inside the area. A Random surf does this by jumping to a random node at each jump with probability tau, see figure 2.9 red area. The number of visits for nodes in this area would dominate a random walk.

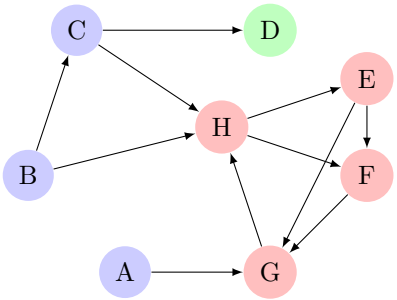


Figure 2.9: Directional Graph example with 8 nodes and 12 edges, the red nodes indicate an area where a random walk would get stuck since the area only has edges pointing to other nodes in the area, the green node indicates a node with no degree out

2.7.2 Directed Graph properties: Biological meaning.

Feature importance: How important is a Gene for another Gene's regulation.
Out_Degree: How many genes do a gene regulate?
In_Degree: How many Genes Regulate a specific gene?
Flow: How much is a gene regulated in the network?
Betweenness: How central is the gene to Regulation in the network?

2.8 Pearson correlation

The Pearson correlation coefficient is an estimate of how likely two samples x and y are shifted higher or lower than the mean at the same time. The Pearson correlation is a normalized property, it is normalized by the standard deviation of each sample. Because of this, The correlation will always lay between minus

one and plus one. When one value is below its own mean, and the other is above its mean, we get a negative value, and if both samples are above or below their means at the same time, we get a positive value. When we sum over all values in the sample, we get the most common case. The Pearson correlation coefficient can be calculated the following way:

$$\rho_{x,y} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{2.11}$$

$\bar{x}$ mean of sample x
$\bar{y}$ mean of sample y
x_i value i in sample x
y_i value i in sample y
n is the number of data points in the samples.
$\rho_{x,y}$ is the Pearson correlation coefficient of sample x, y.

Chapter 3

Results

To better understand how the network's structure evolves, we will first look at the ratio between the number of edges and nodes; this allows us to know the average number of connections pr. node in the network. After this, we will visualize the networks to get a feeling for the placement of high degree nodes. Lastly, we will look at the distribution of the graph properties and the correlation between them. These distributions will give us a view of the hierarchy of genes for different properties.
In the section "essential genes for survival", we will find the graph properties that separate essential and non-essential genes for survival. The correlation between the essential genes' properties for survival and non-essential genes. This will show the tendencies between the properties, so we compare the two kinds of genes' tendencies in the network. In the last section, "Wnt-signalling network", we will look at the validation performance of the methods for inferring the regulators and how robust the methods are for changes in the input. All results are assembled in code in results.zip.

3.1 Is the Network structure evolving?

To understand how the network structure is evolving. we will first introduce the networks and look at the ratio between the number of edges and nodes. Then, we visualize all the networks, and comment on our observations. We will look for patterns when plotting in_degree versus out_degree. At last, we look at the distribution of network properties to understand the network's hierarchy of genes based on properties.

3.1.1 Stats and visual view of GRNboost2 and STN's networks.

We used GRNboost2 and STN to infer eight networks for the days: 6.5, 7.5, 8.5, and 9.5 for both methods. Figure 3.1 shows the number of edges and nodes

in the network and their ratio.

STN day:	6.5	7.5	8.5	9.5
Number of edges(N_{ES})	67407	238333	456025	36134
Number of Nodes(N_{NS})	12212	18356	19305	11276
Ratio between N_{ES} and N_{NS}	5.5	13	24	3.2
GRNboost2 day:	6.5	7.5	8.5	9.5
Number of edges (N_{EG})	2844589	2143939	2407523	1686469
Number of Nodes (N_{NG})	18315	20615	20829	17800
Ratio between N_{EG} and N_{NG}	155.3	104	115.6	94.7

Figure 3.1: Number of nodes and edges for all networks
STN and GRNboost2's networks have very different ratios. GRNboost2's ratios are between 94.57 and 155.3, and STN is between 3.2 and 24. These ratios indicate that GRNboost2's networks are more connected than STN's networks. This difference also shows the algorithms are very different in the way they infer Regulators, STN predicts fewer regulators than GRNboost2.
STN's ratios increase from day 6.5 to 8.5 and collapses on itself at day 9.5 with a lower ratio than the rest of the days, where GRNboost2's ratios decrease steadily from day to day.

Now we have a feeling for the average number of edges pr. node and how differently GRNboost2 and STN predicts regulators. To better observe how the nodes are placed in the networks and especially the nodes with high in_degree and out_degree. We have visualised the networks and highlighted the in and out_degree in the plots. This is shown visually in figure 3.2 and 3.3.

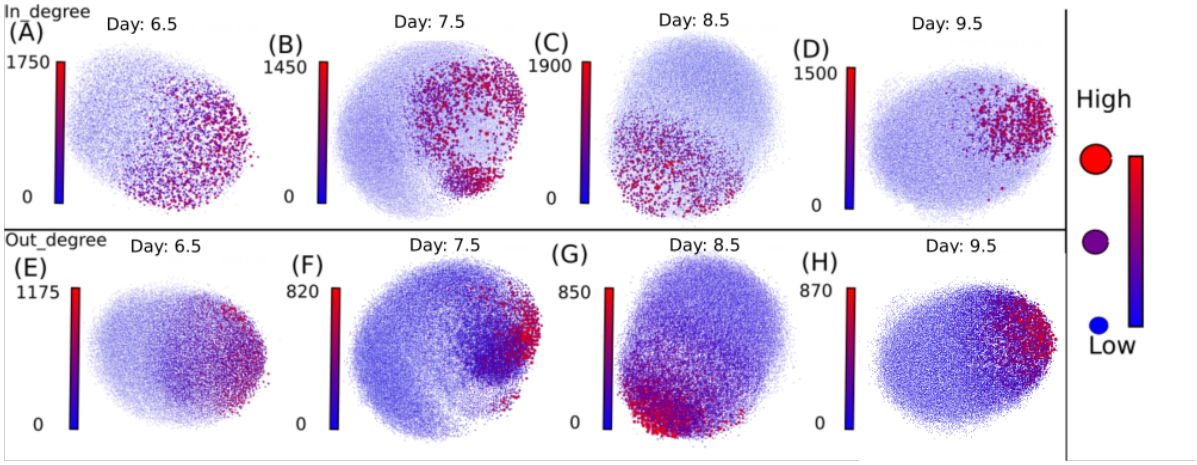


Figure 3.2: Visualization of GRNboost2's networks.
Figure A to D: networks highlighting in_degree, and figures E to H networks highlighting out_degree.
Figure I: Indicates how to analyze the plots; it shows that a small blue node means a low degree and the redder and larger the nodes get, the higher the degree.
Figure A to H: the nodes with high in and out_degree get more and more clustered towards day 9.5.
We have one end where all genes have both low in_degree and out_degree and another end with high in and out_degree.
figures E to H the genes with high out_degree are also centralized in a way smaller area than the in_degree.

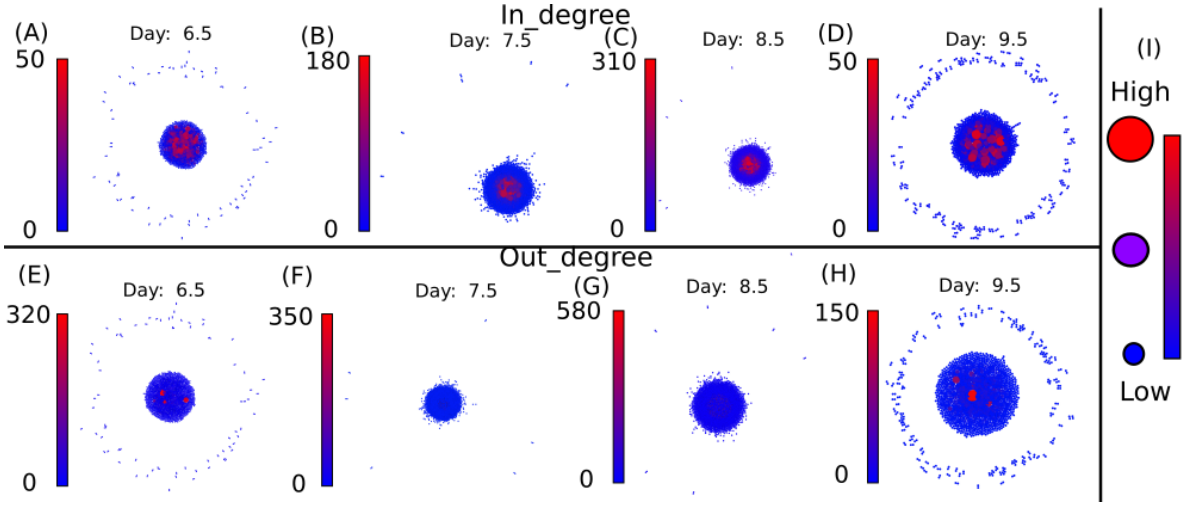


Figure 3.3: Visualization of STN's networks.
Figure A to D: in_degree and **Figure E to H**: out_degree
Figure I: Indicates how to analyze the plots; it shows that a small blue node means a low degree, the redder and larger the nodes get, the higher the degree.
Figure A to D: the networks have a central cluster of genes in the middle with an outer ring that only connects to the closest gene.
The nodes with the highest in_degree are in the center, with smaller nodes connected to them. The structure does not seem to change, but the number of genes in the outer ring varies.
The maximum in_degree increases between day 6.5 and 8.5. Afterwards, the maximum in_degree goes back to a similar range of day 6.5 at day 9.5.
Figure E to H: The out_degree is high for a fewer genes in the center, whereas the in_degree networks have more genes with lower values. A gene underneath the central cluster has a very high out_degree, but the cluster blocks its view in the plots.

We have now observed the difference between GRNboost2 and STN visually. We now want to observe if there are any patterns in the plot of in_degree versus out_degree for both STN and GRNboost2.

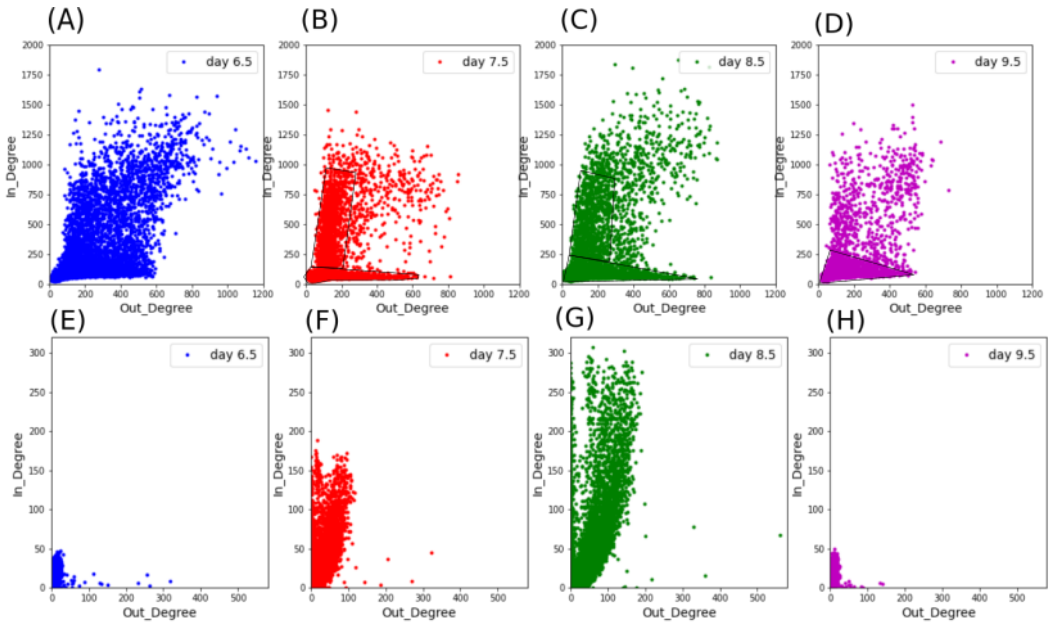


Figure 3.4: In_degree versus out_degree patterns.
Figure A to D: GRNboost2 between day 6.5 and 9.5, and figures E to H are STN between day 6.5 and 9.5.
The color indicates the days after fertilization: Day 6.5 is blue, day 7.5 is red, day 8.5 is green, and day 9.5 is magenta.
Figure A: The in and out_degree of the genes are spread over a large area, with no clear pattern.
Figures B -C: has a similar pattern; their dense areas can be approximated to a triangle with a rectangle on top. If we compare it to figure D, we can see that the triangle is still there, but the rectangle is missing on top. These observations indicate a massive change in the network structure over the days, especially from day 6.5 to 7.5 and day 8.5 to 9.5.
Figures E to H: The in_degree and out_degree are correlated, which means that we are over-fitting massively and capturing indirect regulation.
We can also see that the in and out_degree increase from E to G and then collapse and go back to a similar structure in figure H.

We now observed how the in and out_degree are correlated and the visible in_degree versus out_degree patterns for the networks. We now want to understand which genes are important for the network structure based on how the properties are distributed on the network.

3.1.2 Distribution of properties

The distribution of different properties shows different aspects of the network structure. The distribution will show the importance of different genes based

on their different properties. In the case of the distribution of In_degree, it will show if genes are regulated by the same numbers of genes or if all genes are regulated by a different number of genes. In general, it highlights if all genes are essential for maintaining the structure or if only a few genes are essential for maintaining the network's structure.

3.1.3 Who contributed the most?

To analyse which genes contributed the most to the network structure, we have plotted the cumulative function of the sorted list each property divided by the total sum of the property. This will clearly show, how much different intervals of the distributions contributes to the total network structure. This way of plotting is useful, since we want describe the network's evolution over the time period.

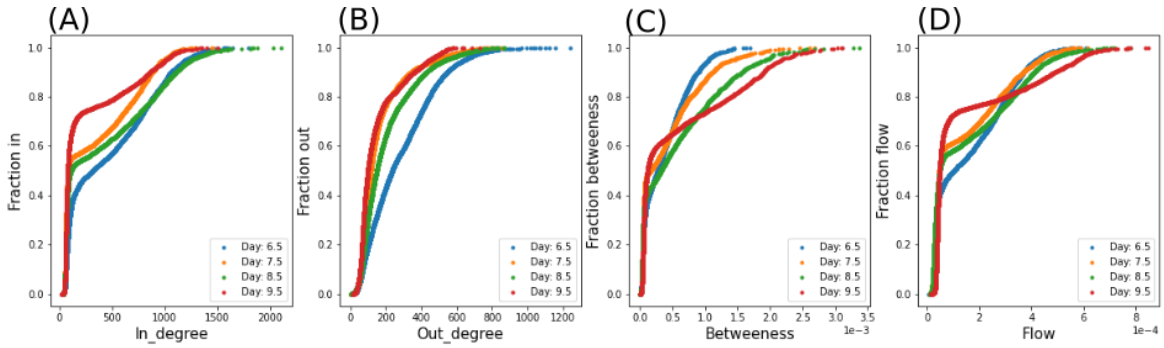


Figure 3.5: Property hierarchy of nodes
inferred by GRNboost2
The color indicates the number of days after fertilization, day 6.5 is blue, 7.5 yellow, 8.5 green, 9.5 is red.
The steeper the function's slope is in the intervals, the more impact the interval has on the network structure.
Low range properties are between zero and 20 % of the maximal x axis length, medium range is between 40 and 70% and high range is between 70 and 100 %
Figure A: day 9.5 relies heavily on the nodes with a small in_degree. After we have 7.5 and 8.5, which have the same steepness of their functions between 0.4 and 0.8. Day 8.5 is more reliant on the nodes with medium-ranged in_degree than 7.5. At last, we have day 9.5, which is not that reliant on low in_degree nodes and relies primarily on nodes with medium to high in_degree.
Figure B: the out_degree structure is very similar for all days, and they mostly rely on low to medium ranged nodes.
Figure C: The days which rely most on nodes with low betweenness are day 9.5,7.5,8.5 then 6.5
Figure D: The days which rely most on nodes with low flow are 9.5, 7.5,8.5 than 6.5
GRNboost2: predicts that besides out_degree, that day 6.5 relies more on nodes with medium to high ranged properties to maintain the structure compared to the other days. Day 7.5 and 8.5 rely almost equally on nodes with medium to high properties, and lastly, we have day 9.5 which relies the least on nodes with medium to high properties.

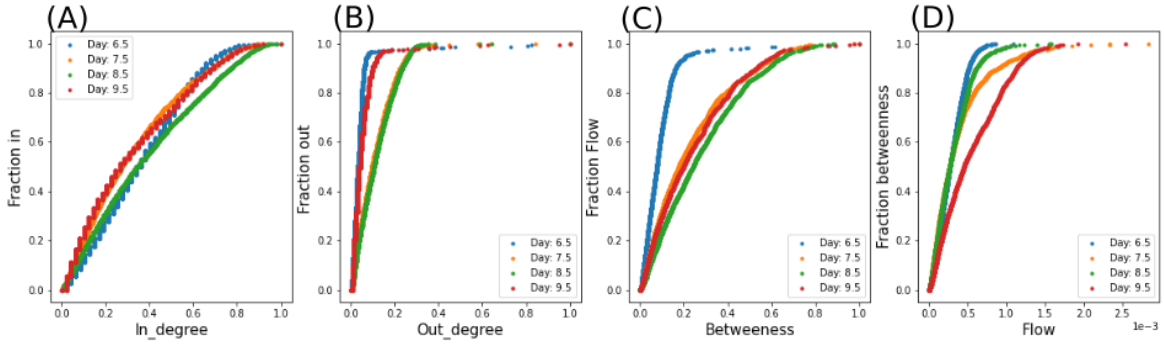


Figure 3.6: Property hierarchy of nodes inferred by STN
Low range properties are between zero and 20 % of the maximal x axis length, medium range is between 40 and 70% and high range is between 70 and 100 %
Figure A: all the days are very similar, and they rely on all nodes except nodes with really high in_degree.
Figure B: day 6.5 is reliant on nodes with really low in_degree, and the ranking from reliance on nodes with low out_degree to high out_degree is day 6.5 and 7.5, Day 8.5 has the same reliance as 9.5.
Figure C: Day 6.5 is heavily reliant on the nodes with low betweenness compared to days 7.5,8.5, and 9.5, which rely equally on all ranges.
Figure D: day 6.5,7.5 and 8,5 relied on nodes with low flow. Their rank order is 6.5,7.5,8.5, and 9.5, whereas 9.5 relies way more on nodes with the medium-ranged flow.
STN: predicts that day 6.5 relay more on nodes with low properties than the other days, without considering the in_degree.

We have now compared the different days structure and what we are missing is observing the original distribution.

3.1.4 How does the in_degree distribution evolve?

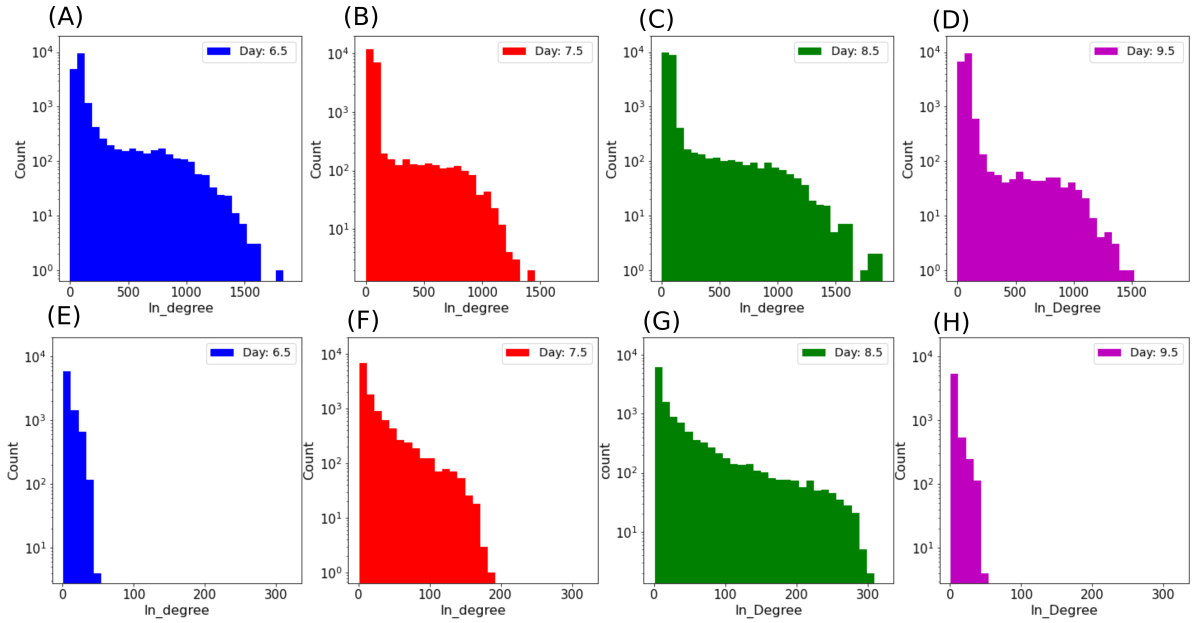


Figure 3.7: In_degree distributions inferred by STN and GRNboost2.
Figure A to D: are inferred by GRNboost2, and **Figure E to H**: are inferred by STN.
Figure A to D : when we look at the distributions, we can observe that there are an extreme amount of genes at the start of the distributions. The difference in the distribution lay in-between 20 to 40 % of the max in_degree, so around 300-700 in_degrees(See figure 3.5). Besides the difference, the shape of the distributions look similar.
Figure E to H: The shape of the distribution looks similar, but the tail of days 7.5 and 8.5 is way wider than 6.5 and 9.5. If we look at figure 3.6, we can see the count decreases very similarly towards the maximum in_degree, when the distributions are normalised. The in_degree expands to day 8.5 and collapses between days 8.5 and 9.5.

3.1.5 How does the out_degree distributions evolve?

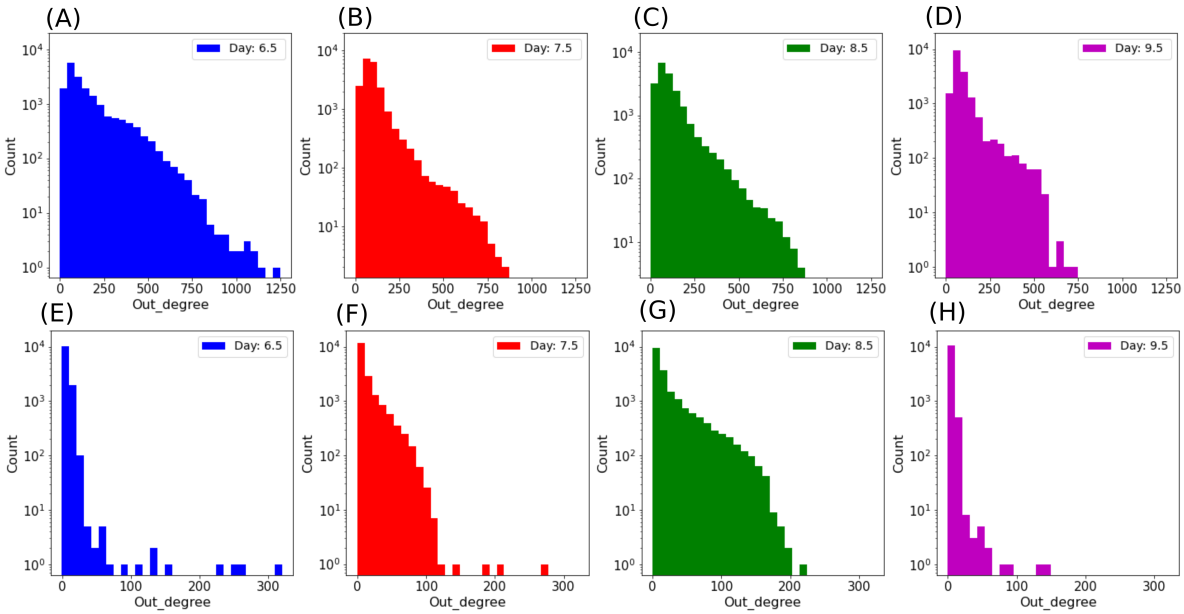


Figure 3.8: Distributions of out_degrees
Figure A to D: is inferred by GRNboost2, and **figure E to H**: is inferred by STN.
The color indicates the days after fertilization: Day 6.5 is blue, day 7.5 is red, day 8.5 is green, and day 9.5 is magenta.
A to D: when we look at day 6.5, we can see it has a way longer tail than the rest of the days and also fewer genes with low out_degree. By looking at the plots, we can see that the in_degree gets pushed towards zero over the days. The distribution shapes look similar. **E to H**: day 6.5 and 9.5 has a similar distribution, with day 6.5 have several genes with out_degree above 100 compared to day 9.5. We can also see that days 6.5 and 9.5 have similar out_degree scaling, and 7.5 and 8.5 are similar and have a similar out_degree scaling. It looks like the network expands and collapses back on day 9.5.

3.1.6 How does the distribution of betweenness evolve ?

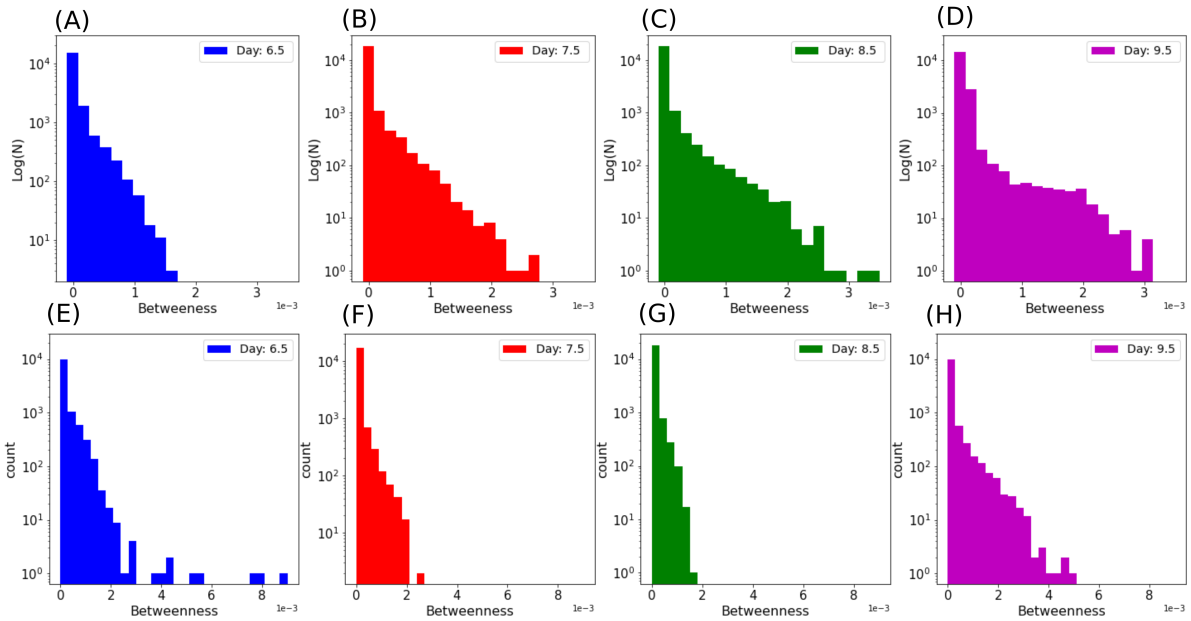


Figure 3.9: Distribution of Betweenness.
Figure A to D: is inferred by GRNboost2, and **figure E to H**: is inferred by STN.
The color indicates the days after fertilization: Day 6.5 is blue, day 7.5 is red, day 8.5 is green, and day 9.5 is magenta.
A to D: when we observe the distribution, we can see that the tail of the distribution expands towards day 9.5. The days 6.5 to 8.5 have similar distributions. Day 9.5 is different because it has way less nodes in the interval between 1 to 2 10^{-3}. betweenness
E to H: These distributions have similar shapes. Day 6.5 has a few genes with massive betweenness, and again day 6.5 and 9.5 are in a similar range, and 7.5 and 8.5 are in a similar range. Looking at figure 3.6, we can see that day 6.5 scales very differently compared to the other days.

3.1.7 How does the distribution of *Flow* evolve ?

Flow is highly correlated with the in_degree since the in_degree measures how many genes are regulating a gene. The flow measures how often or how much a gene is getting regulated compared to other genes. This also means their distributions will be closely related.

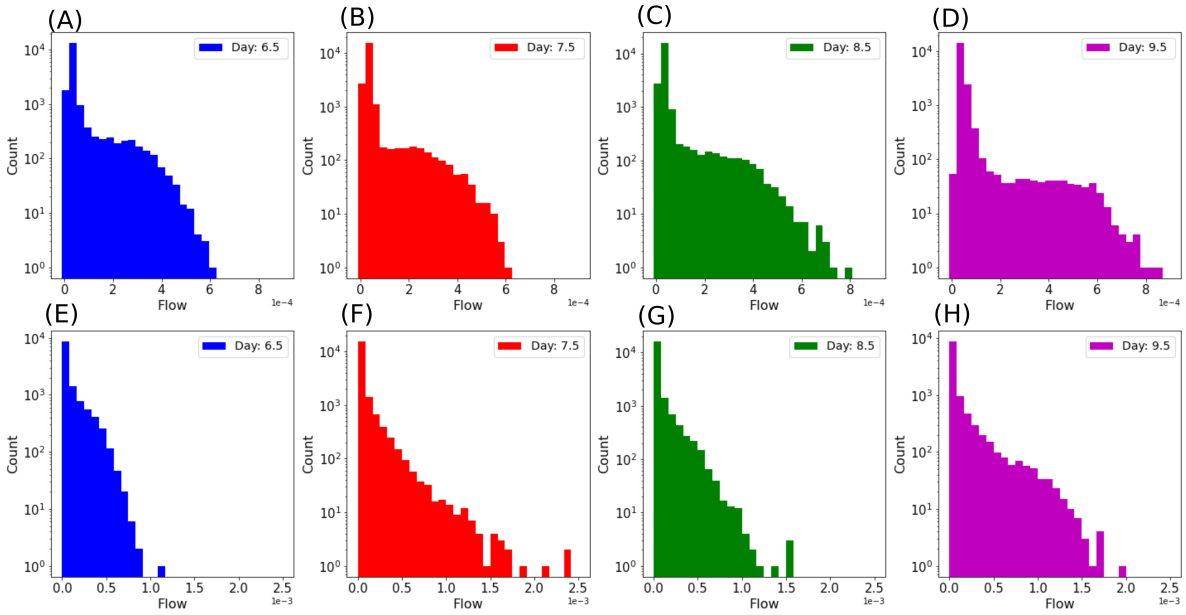


Figure 3.10: Distribution of Flow.
Figure A to D: is inferred by GRNboost2, and **Figure E to H**: is inferred by STN.
The color indicates the days after fertilization: Day 6.5 is blue, day 7.5 is red, day 8.5 is green, and day 9.5 is magenta.
A to D : The shapes are very similar, day 6.5 has a very narrow distribution tail, towards day 9.5 the distribution tails get wider. Day 9.5 is different because it has way less nodes in the interval between 2 to 5 10^{-4} Flow .
E to H : The shape of day 6.5 looks different from the rest days. Day 9.5 looks like it has more genes in the medium between 0.5 and 1 10^{-3} flow. This can also be confirmed in figure 3.6, the tail changes width multiple times between day 7.5 and 9.5

3.1.8 how does the correlation between properties change?

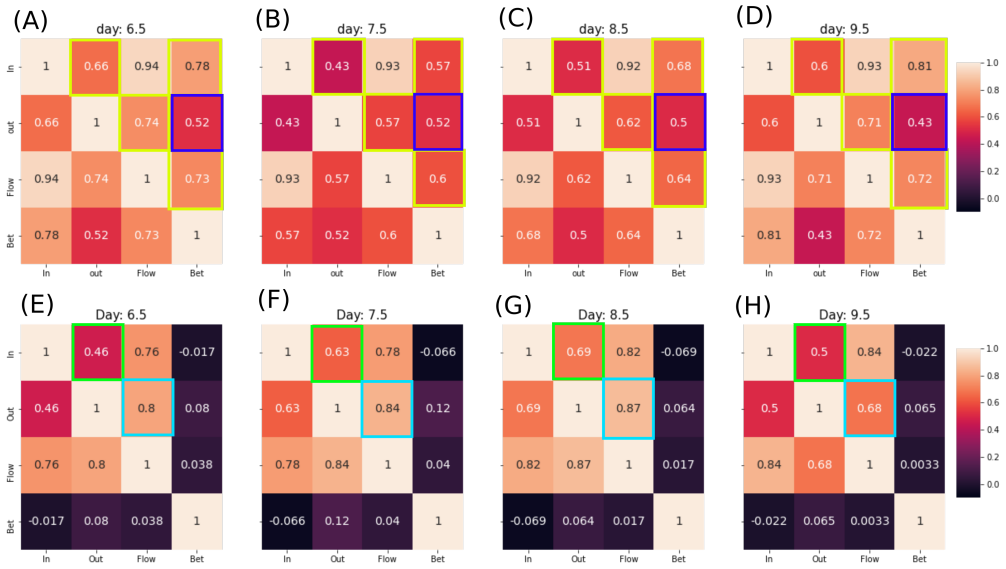


Figure 3.11: Correlation matrix for STN and GRNboost2
Figure A to D is inferred by GRNboost2, and **figure E to H** is inferred by STN.
The x and y-axis are labels for properties. In, out and bet is the in_degree, out_degree, and betweenness.
Figure A-D Yellow frames highlight the first pattern. The yellow pattern show that day 6.5 and 9.5 are in a similar range and Day 7.5 and 8.5 are in a similar range. The yellow pattern does not include correlations between in_degree and flow and between out_degree and betweenness.
The blue frames show that the correlation between betweenness and out_degree stays in a similar range until day 9.5 which indicates a structural change in the network.

3.2 What are the profile of essential genes for survival?

In the following section, we estimate the profile of essential genes for survival. An essential gene is a gene which when removed causes The mouse to die before birth. A Non-essential gene which when removed allow the mouse to survive after birth. Figure 12 shows the distribution of essential and viable genes' properties for day 8.5 inferred by GRNboost2.

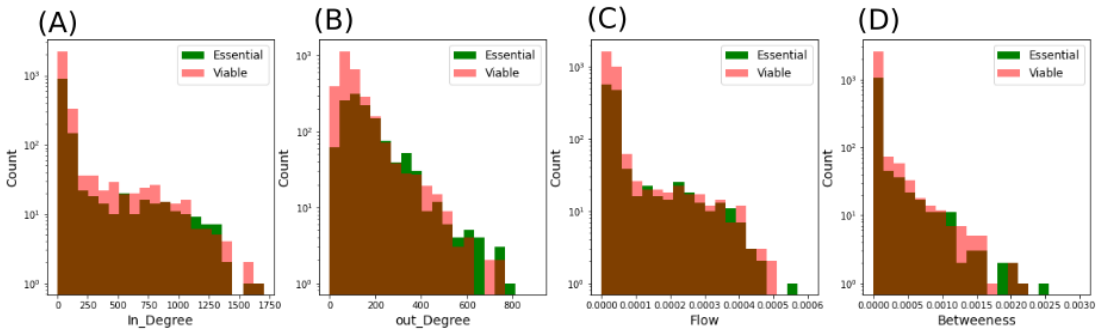


Figure 3.12: **Distributions of essential and viable genes' properties**. Distributions of essential genes' properties are green, distributions of viable genes' properties are pink and the overlap between them is brown

Since the distribution of essential and viable genes' properties did not show a clear tendency for any days or methods, we decided to use "Bootstrap with replacement" see section 3.2.1 to estimate the underlying distribution of the means.

3.2.1 Bootstrap with replacement

Bootstrap with replacement is a method for simulating data when the sample size is too small to estimate the underlying distribution.
Bootstrap with replacement generates a new data-point by taking N data points with replacement from the sample x and take the mean of that sample, this returns a value between the minimum and maximum of x. If we do this enough times and store the values in an array, we have generated a new and larger sample with the mean and standard deviation of the original sample. As the number of generated samples goes towards infinity, the distribution of the sample converges towards a Gaussian distribution due to the central limit theorem.

3.2.2 What are the tendencies of Essential genes?

To determine if the models could capture the patterns of essential genes in the single-celled data, we found two papers that listed essential and viable genes based on experiments. The first list is from university of Manchester's bio-science of the future group which has made a database for essential and viable genes [2]. The second paper made the list by experimentally removing the genes from mice [3]. We added the two lists together to get 1306 essential genes and 3460 viable genes, adding up to 4766 genes. We made eight samples for each network, four samples containing the in_degree, out_degree, betweenness, and flow of essential genes. We also made Four samples for the properties of the viable genes.
We performed the Kolmogorov-Smirnov two-sample test,(see appendix A1) on the original samples to test if the essential and viable genes came from the same

underlying distribution. The Kolmogorov-Smirnov two-sample test returns the likelihood for two samples to come from the same underlying distribution. The likelihood is estimated by comparing the largest distance between the distributions of the samples versus the number of points of in the samples, for a more detailed description see appendix A1.
To visualize the difference in Graph properties between the viable and essential genes, we performed bootstrap with a replacement on each sample. Figures 3.13 to figure 3.16 show each property plotted separately in different figures with the bootstrapped distribution of the means of the essential and viable samples.

In_degree

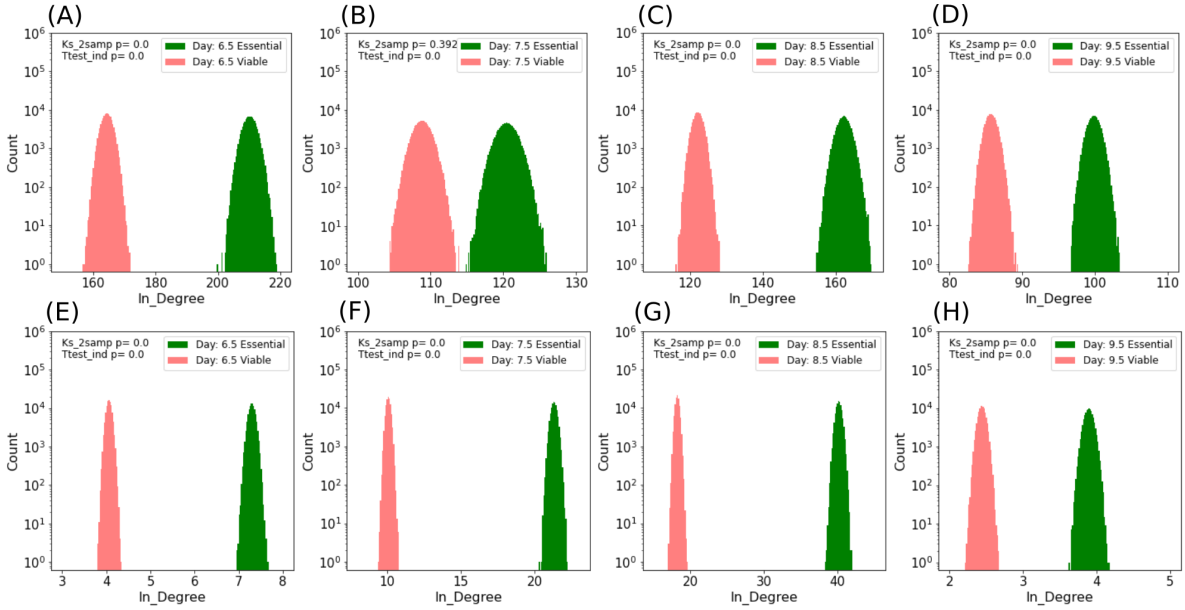


Figure 3.13: Prediction of essential and viable genes' in_degree tendencies.
Figures A-D are networks inferred by GRNboost2 and **Figures E-H** are networks inferred by STN.
Essential genes are plotted in green, viable genes are plotted in pink.
All figures show that essential and viable genes' means and underlying distributions are different since their Kolmogorov-Smirnov two-sample test, and t-test P values are zero.
Essential genes have a tendency to have higher in_degree, thus GRNboost2 and STN predicts essential genes are regulated by more genes than viable.
Be aware the x axis limits are different from figure to figure.

Out_degree

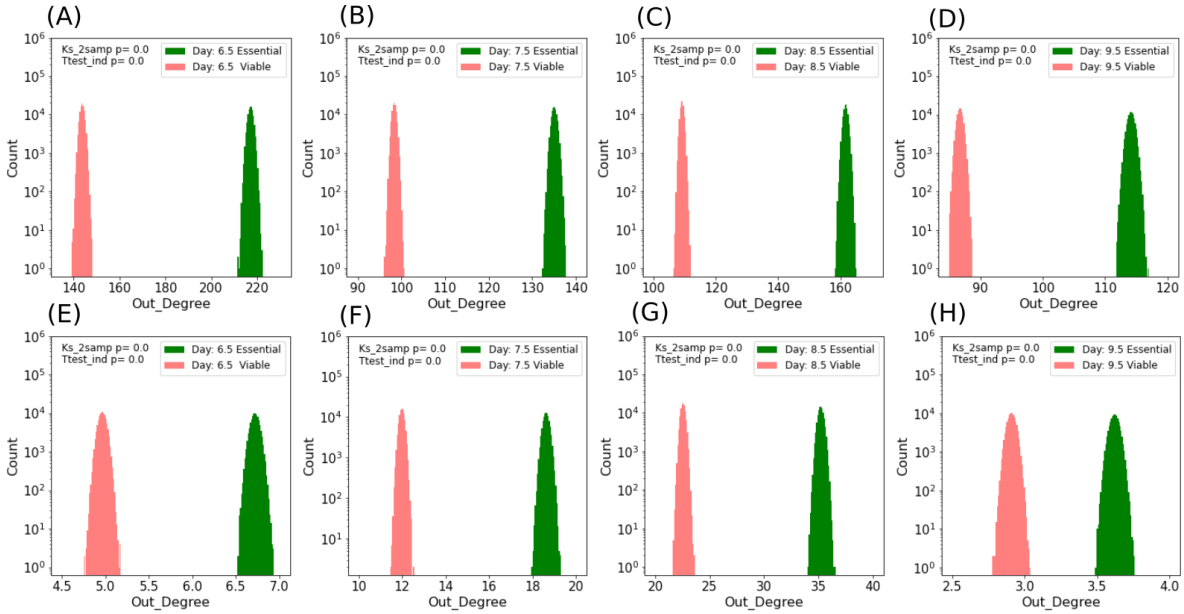


Figure 3.14: Prediction of essential and viable genes' out_degree tendencies.
Figures A-D are networks inferred by GRNboost2 and **Figures E-H** are networks inferred by STN.
Essential genes are plotted in green, viable genes are plotted in pink.
All figures show that essential and viable genes' means and underlying distributions are different since their Kolmogorov-Smirnov two-sample test, and t-test P values are zero.
All figures show Essential genes tend to have higher out_degree, thus GRNboost2 and STN predicts that essential genes tend to regulate more genes than viable. textbfBe aware the x axis limits are different from figure to figure.

Betweenness

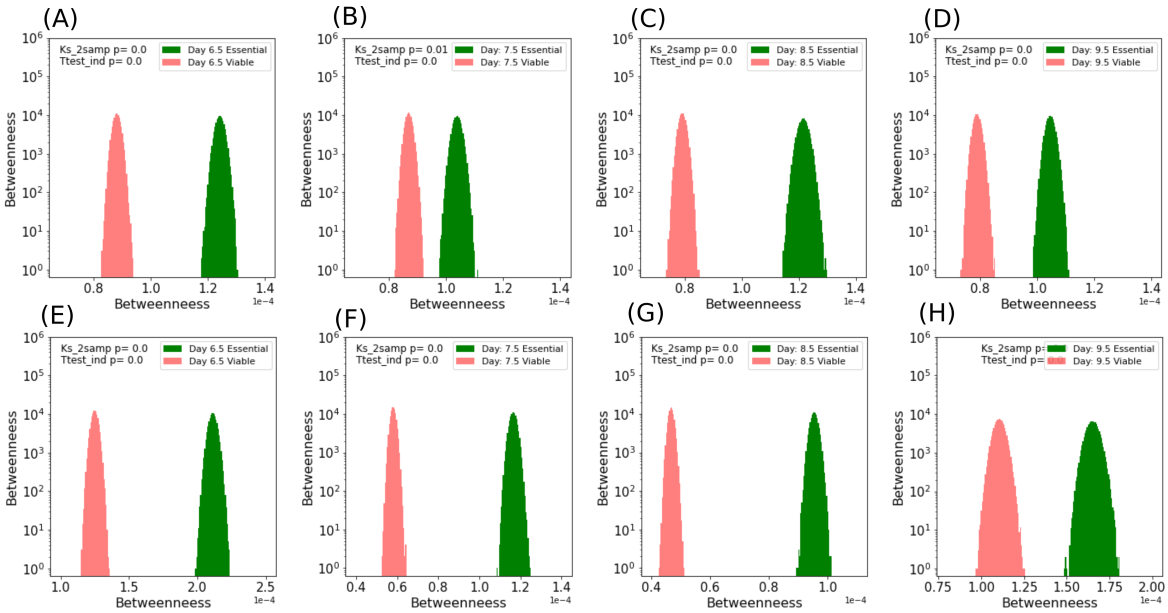


Figure 3.15: Prediction of essential and viable genes' betweenness tendencies.
Figures A-D are networks inferred by GRNboost2 and **Figures E-H** are networks inferred by STN.
Essential genes are plotted in green, viable genes are plotted in pink.
All figures show that essential and viable genes' means and underlying distributions are different since their Kolmogorov-Smirnov two-sample test, and t-test P values are zero.
Essential genes tend to have higher betweenness, thus GRNboost2 and STN predict that essential genes have a higher tendency to be more central for regulation in the networks than viable genes.
Be aware the x axis limits are different from figure to figure.

Flow

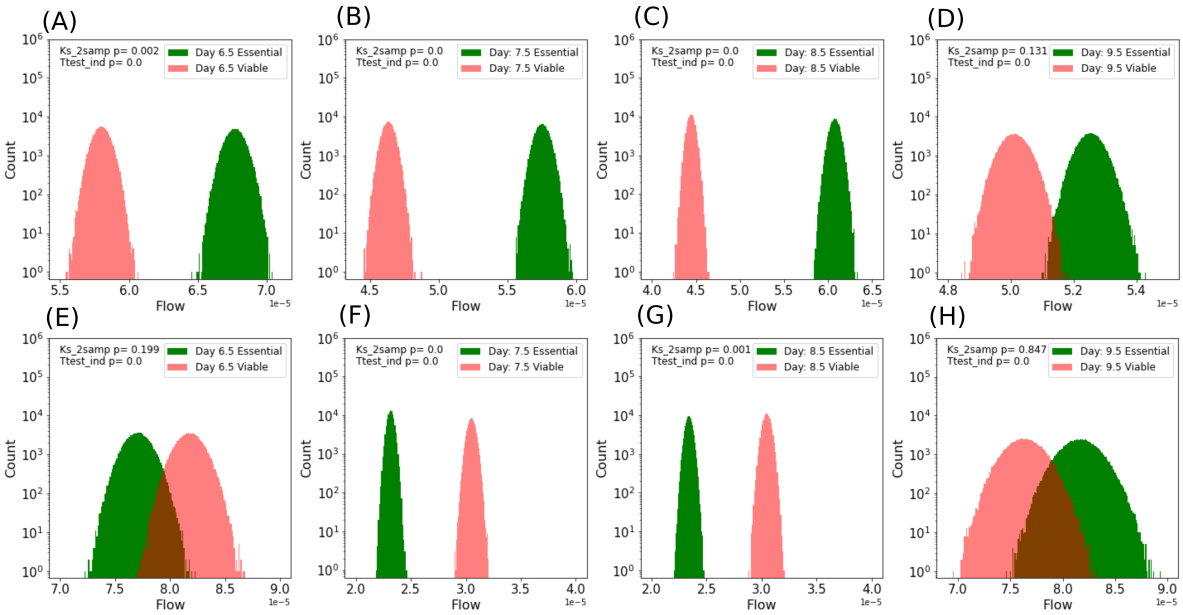


Figure 3.16: Prediction of essential and viable genes' flow tendencies

Figures A-D are networks inferred by GRNboost2 and **Figures E-H** are networks inferred by STN.

Essential genes are plotted in green, viable genes are plotted in pink their overlap is brown.

Figures A,B,C,F and G Show that essential genes and viable genes has different means and distribution, since t-test and Kolmogorov-Smirnov two-sample test P-value are equal to zero. This means that viable and essential genes have different means and come from different underlying distributions.

Figure D day 9.5 t-test P-value is zero so the flow of essential and viable genes are separate enough to have different means. The Kolmogorov-Smirnov two-sample test is 0.131, which is above 0.05, therefore we can not reject that the two original samples come from the same underlying distribution.

Figures E and H shows that The Kolmogorov-Smirnov two-sample is way above 0.05. This indicates that essential and viable genes for days 6.5 and 9.5 come from the same distribution. The t-test shows a clear mean separation since they are both zero.

Figures F and G have either a close to zero or zero p-value for both the t-test and the Kolmogorov-Smirnov two-sample test, which indicate that essential and viable genes for days 7.5 and 8.5 come from different distributions and have different means.

GRNboost2 predict that essential genes tend to be be more regulated than viable genes.

STN predict that essential genes have a tendency to be less regulated than viable, but E and H are close to the same means and have similar original distribution so should therefore be taken with a grain of salt for those days.

Be aware the x axis limits are different from to figure.

3.2.3 Correlation between Essential properties

To see how essential genes' properties are correlated and how viable genes' properties are correlated, we have made two correlation matrices, one for essential properties and one for viable. We have made these matrices for both models and all the days. These matrices will show if essential and viable genes' properties have the same patterns between them.

GRN2boost

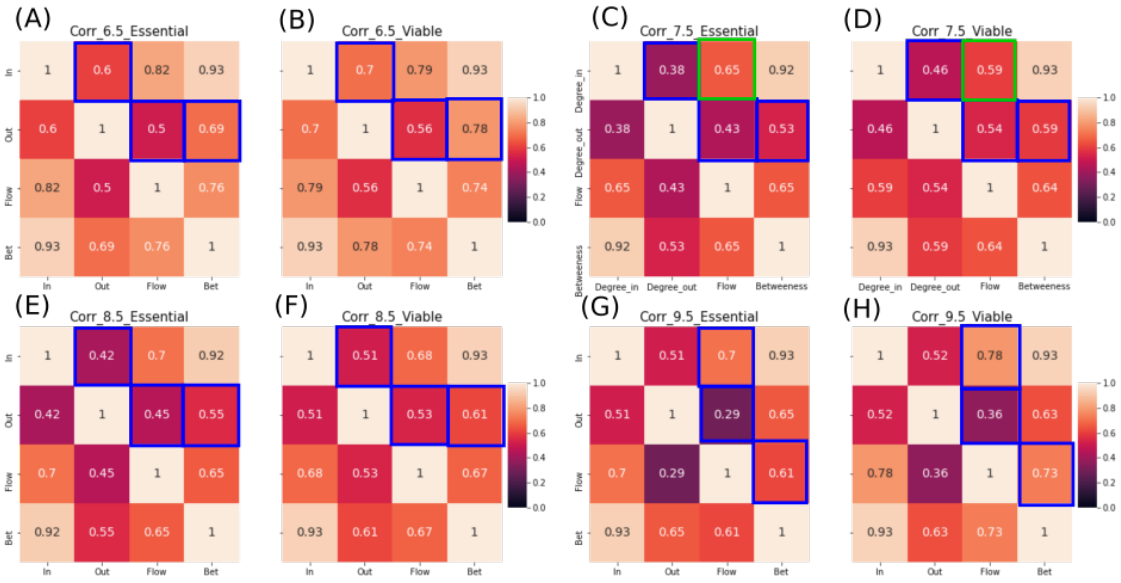


Figure 3.17: Tendencies between properties for viable and essential genes.
Networks inferred by GRNboost2. **figures: A, C, E and G**: are matrices of correlation between essential genes' properties, and **figures: B, D, F, and G** :are matrices of correlation between viable genes' properties.
The x and y-axis are labels for properties. In, out and bet is the in_degree, out_degree, and betweenness.
The x and y-axis are the labels of the properties. The labels in, out, and bet are the in_degree, out_degree, and betweenness.
To highlight where essential genes have a higher difference than 0.06 Pearson correlation than viable, we have made blue frames for spaces where the correlation of essential genes' properties are lower and green frames where essential genes' properties have a higher correlation.
Figures A to F: have a lower correlation between in and out_degree, between flow and out_degree, and between betweenness and out_degree.
Figures C and D: show that essential genes have a higher correlation between flow and in_degree at day 7.5
Figures G and H: correlation matrices are very different from the other days; the essential properties' correlation between in_degree and flow and between out_degree and flow is lower than the viable genes and the only overlap is between flow and out_degree with the other figures.
figures A,B, G and H: Day 6.5 has a similar range of correlation for the properties as 9.5 for both the essential and viable genes, which indicates that they have a similar network structure.
Day 7.5 and 8.5 have correlations in a similar range, indicating a similar network structure. GRNboost2 predicts that essential genes are more likely to either be to regulated by a few genes while regulating many other genes or to be regulated by many genes while regulating few other genes than viable ones.
From Day 6.5 to 8.5: the chance of being a more central gene to gene regulation while regulating many genes is lower for essential genes than viable.
Day 9.5 essential genes are less likely to be regulated a lot while being regulated by many genes simultaneously. It is also less likely to be regulated a lot and to be a more central gene for regulation.

STN

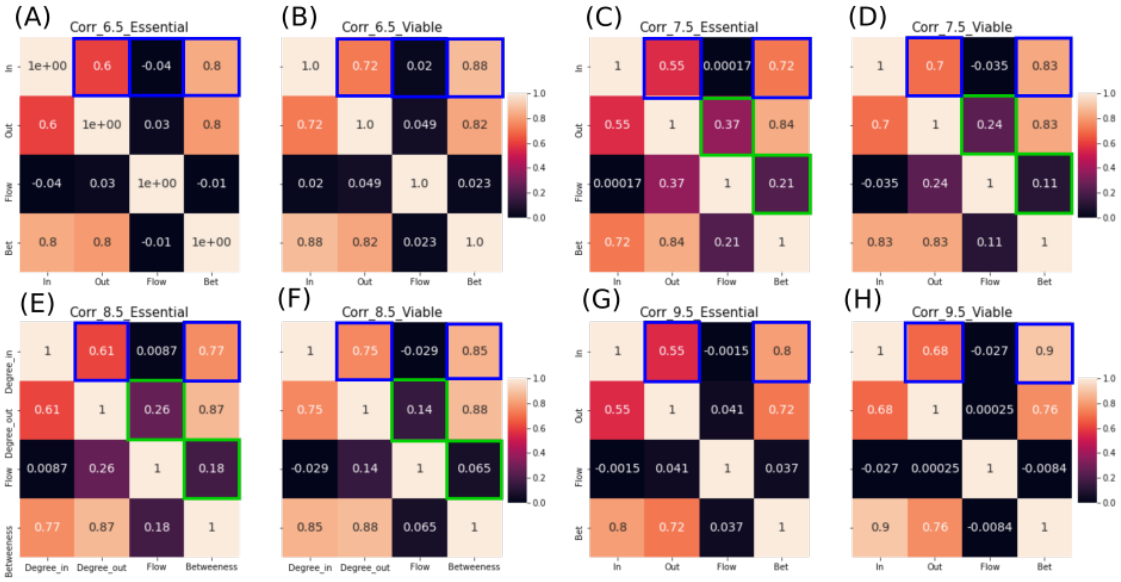


Figure 3.18: Tendencies between properties for viable and essential genes. Networks are inferred by STN.
Figures A, C, E, and G: are matrices for essential properties. **Figures B, D, F and G** :are matrices of viable properties. The x and y-axis are labels for properties. In, out and bet is the in_degree, out_degree, and betweenness.
To highlight where essential genes have a higher difference than 0.06 Pearson correlation than viable, we have made blue frames for spaces where the correlation of essential genes' properties is lower and made green frames where essential genes' properties have a higher correlation.
The correlation between in and out_degree and between in_degree and betweenness is lower for all days.
Figures A and B: show that day 6.5 essential genes have a lower correlation between in_degree and flow. **Figures C, D, E and F**: shows that essential genes have higher correlation between Flow and out_degree and between flow and betweenness. Viable genes have similar correlation values and essential genes have similar correlation values between the days.
STN: predicts that essential genes are less likely to be regulated by many genes at the same time as regulating many other genes compared to viable genes. It also predicts that essential genes are less likely to be central to regulation in the network at the same time as getting regulated by many genes compared to viable genes.

3.3 Wnt-signalling network

Introduction to the Wnt-network

The Wnt-signalling network is an essential subset of genes that control cell division and cell differentiation cite. This network is universal for mammals, but the network's complexity is different from species to species. Multiple kinds of cancer and Alzheimer's are directly linked to Mutation and other damage in this system. We want to investigate this system because the Wnt signaling network is highly active during gastrulation days 6 to 8.5 and that it is one of the most essential networks for survival in early stages of development in mammals [4].

Wnt-validation data

We will be focusing on two networks: the mouse and human Wnt networks. We originally only wanted to use the mouse Wnt network, but there were no studies that had an easy way to validate the network, and the Wnt genes are well conserved between humans and mice. Figure 3.19 shows the four sub-Wnt networks which are present in humans. Figure 3.20 shows a simplistic version of the mouse Wnt-network.

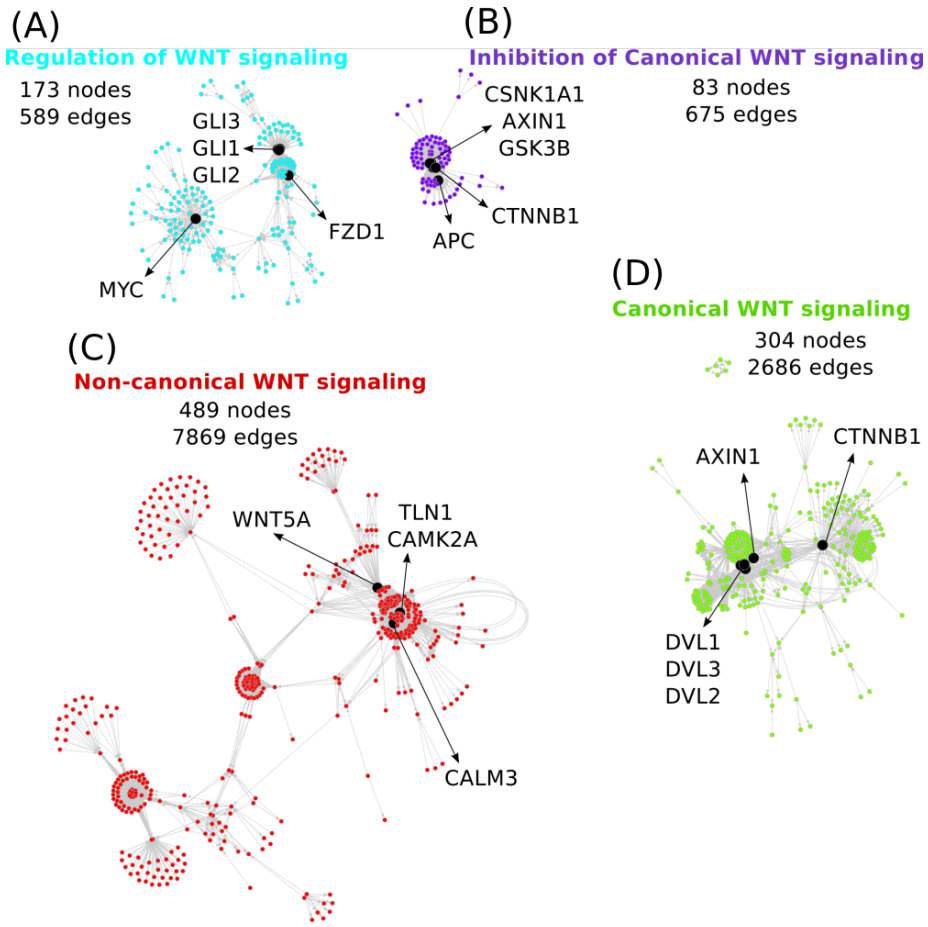


Figure 3.19: The four subnetwork of the human Wnt network
Figure A: Canonical Wnt-signaling
Figure B: The inhibition of canonical-Wnt signaling
Figure C: Non-canonical Wnt-signaling
Figure D: The regulation of Wnt-signaling pathways. [5]

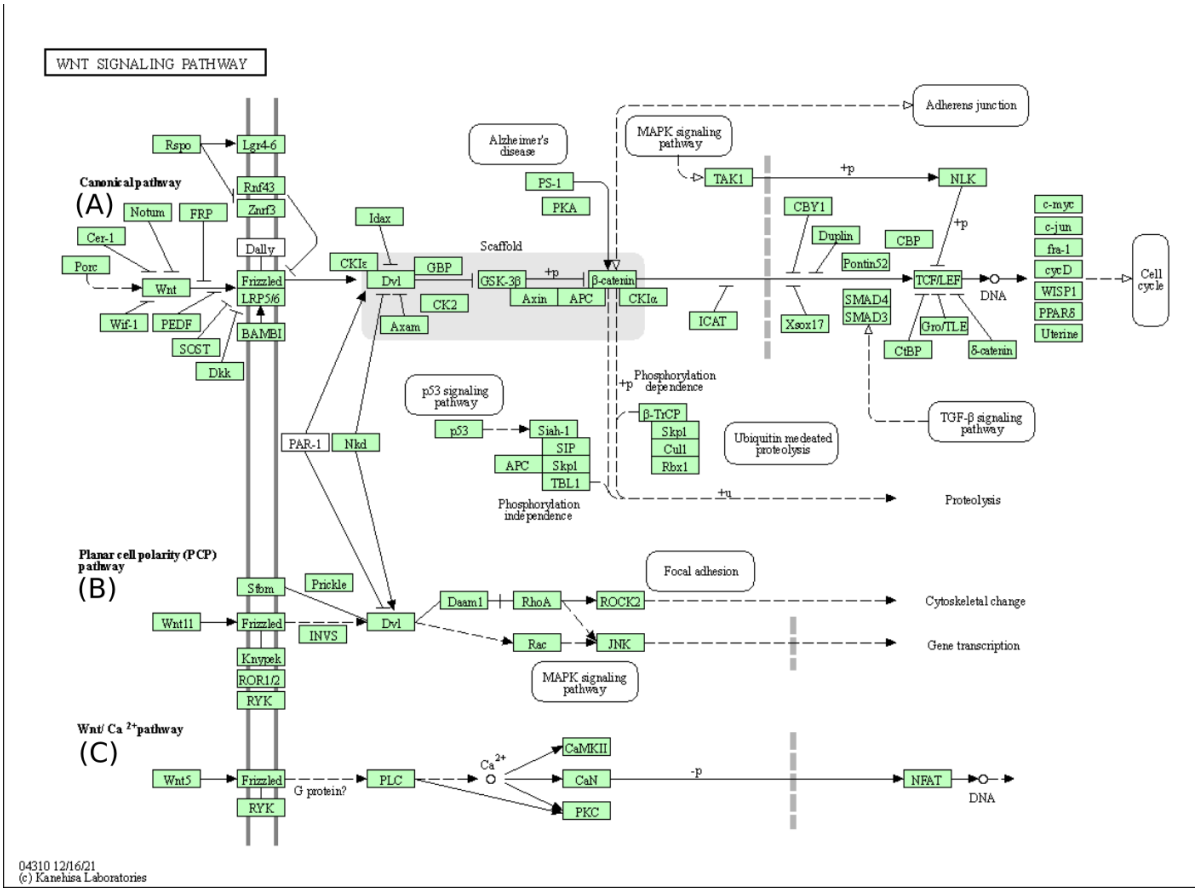


Figure 3.20: **Figure A-C**: A simple version of the mouse Wnt network. Each label with without numbers are multiple genes.
Figure A: Canonical Wnt-signaling
Figure B: Planar cell polarity pathway
Figure C: WNT path way
Notice some of the pathways lead to cancer and Alzheimer's diseases. [6]

3.3.1 How validate methods for inferring the regulators

To estimate how well we are predicting the regulators, we first need to introduce four terms: the true positive(TP), the false positive(FP), the false negative(FN), and lastly, the true negative (TN). See equations 3.1-3.4. In IGRN models, positive means regulating, and negative means non-regulating. So TP is the number of correctly predicted regulators, and FP is number of predicted regulators which are not regulators in reality. FN is the number of non-regulators falsely predicted . TN number of correctly predicted non-regulators. After introducing the basics, we can calculate the true positive rate and the false positive rate,

see equations 3.5 and 3.6. The prediction of regulators is stored as a binary matrix, each row and column is a gene and the values inside the matrix show if the gene are regulating each other. If the gene in row four has a 1 in column 3 gene 1 is regulating gene 3. To validate networks wee flatten the matrix G for each network and flatten the validation data-set and used the formulas below.

$$N_{TP} = \sum_{i=0}^{N} P_i \cdot T_i \tag{3.1}$$

P: Array with prediction
T: Array with true values.
N_{TP}: Number of correctly predicted regulators.
N: The number of elements in the arrays P and T.

$$N_{FP} = \sum_{i=0}^{N} P_i \neq T_i \tag{3.2}$$

N_{FP}: Number of falsely predicted regulators

$$N_{TN} = \sum_{i=0}^{N} (P_i - 1) \cdot (T_i - 1) \tag{3.3}$$

N_{TN}: Number of true non-regulators

$$N_{FN} = \sum_{i=0}^{N} P_i \neq T_i \tag{3.4}$$

F_{FP}: Number of falsely predicted non-regulator
With these four values we can calculate The true positive rate(TPR) and the False Positive rate(FPR):

$$TPR = \frac{N_{TP}}{N_{TP} + N_{FN}} \tag{3.5}$$

$$FPR = \frac{N_{FN}}{N_{FN} + N_{TN}} \tag{3.6}$$

We use TPR and FPR to estimate how well a given predictor predicts the regulators correctly in the networks. TPR is the normalized value for how well we are predicting the regulators. If it is a perfect estimator, it will return the value one. FPR is the normalized value for how well we are predicting non-regulators, we have the N_{FN} in the nominator; this means the closer to zero we get, the better the predictor is at estimating non-regulators, see Figure 3.19 A-D. The ratio between TPR and FPR shows if the prediction is better than random. If it is above one it is better than random and if it is below one it is worse than random.

3.3.2 Wnt-human TPR and FPR

The data used in this section is from the Wnt validation data and comes from the paper [5]. We have tried to predict the regulators in the Wnt-signalling networks using GRN2boost and STN for each method and day. We calculated the true positive and the false positive rate, see equations 3.5 and 3.6. Each network can be represented as a heatmap; the yellow spaces are regulation, and the blue spaces are non-regulation. Figure 3.21 Shows an example of this.

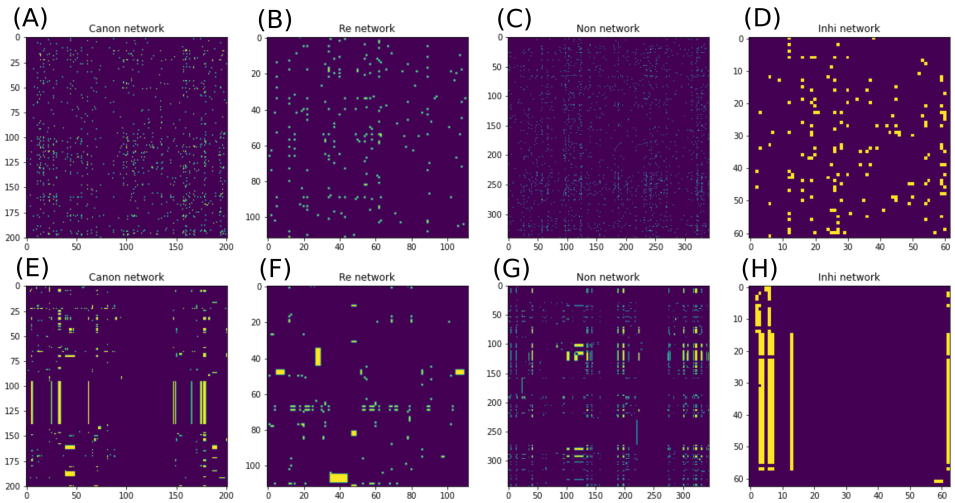


Figure 3.21: Heatmaps of Wnt networks inferred by GRNboost2 day 6.5
Figure A-D: The Wnt subnetwork of day 6.5 inferred by GRNboost2
Figure 3.19 E-H: true regulators for day 6.5.
Comparing Figure A-D to Figure E-H we can easily see that GRNboost2 does not predict many values correctly. Here we can see how different the matrix of the validation-set is compared to GRNboost2's predictions.

Before we go to validating the methods for inferring the network, we want to introduce some earlier results of GRNboost2. It has been used to infer networks from scRNA-seq data from humans. The networks they infered had the sizes from 7000 to 11214 nodes. They predicted 4.1 % of the regulators correctly, their validation data is from a human and is based on experimental data which is preferred [7]. Similar studies have been done on yeast with a performance of 0.69 AUROC [8]. This means 69% of all prediction are correct, not just the predicted regulators. The performance is due to regulation in yeast being less complicated. The assumption is that when genes lie together, they are more likely to regulate each other. This is a very good approximation for bacteria, now want to confirm if this is a good approximation for mice. In general there is no perfect way to validate the regulation. Regulation is very messy, so one should not expect a perfect performance for any method or validation-set. To

quantify the quality of the predictions like figure 3.21, we calculated the TPR and FPR for all methods and sub-networks and shown in figure 3.22.

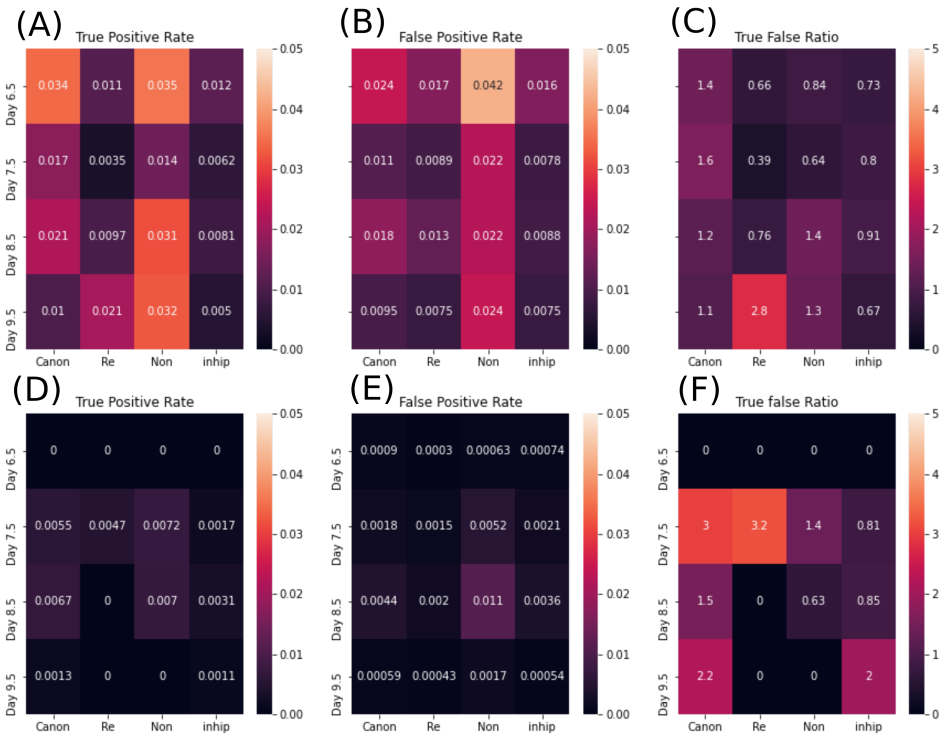


Figure 3.22: Validating the performance of predicting the regulators.
Figures A to C are inferred by GRNboost2 , and figure D to F are inferred by STN.
The x-axis of all the figures is the labels of the Wnt networks and the y-axis is the different days. The colorbar in figures A and B indicates the True positive rate(TPR), the colorbar in B and E indicates the False positive rate(FPR) and the last colorbar in figures C and F indicates the ratio between TPR and FPR . Figure A: How well GRNboost2 is predicting the regulators. The TPR in the networks are between 0.005 to 0.034. This means in the best case scenario, when we are trying to predict 100 regulators only 3.4 are truly regulators.
Figure B shows the performance of predicting non regulation, FPR of the network is between 0.0075 to 0.024 (remember that we want this number close to zero). Figure C shows the ratio between TPR and FPR if the number is above one it is predicting better than random, the ratio is between 0.39 and 2.4, which shows we are predicting better than random for some of the networks. This is mostly due to the fact that GRNboost does not predict many regulators. Figures D to F Show the performance of STN. Here we see many days have a TPR of zero, and the days which are not zero have a performance in a similar range to GRNboost2. Here the number of predictions is even lower, which makes some of the days look better than GRNboost2.

The performance shown in figure 3.22 can be explained in two ways: The validation network of the mouse is not very close to that of a human even in the Canonical network or the assumption of similar gene-expression between genes is a bad estimate for predicting regulators in mice. It could also be mixture of the two explanations.

3.3.3 Robustness of STN and GRNboost2

To see how robust the results in figure 3.22 are, we have run five simulations of STN and GRNboost2 with sub-sampling of half the cells for day 6.5 and 7.5. We have calculated the means and standard deviance of the FPR and TPR for both the network inferred by GRNboost2 and STN. Every data-point is a different Wnt-subnetwork's performance. Before we start on the analysis, we need to make some small comments about the plot in figure 3.23 It is a method normally use in machine learning to showcase the performance of a predictor. The good thing about this kind of plot is prediction performance above the red line is better than a random estimator and the prediction performance below is worse than random, since TPR needs to be higher than TFP.

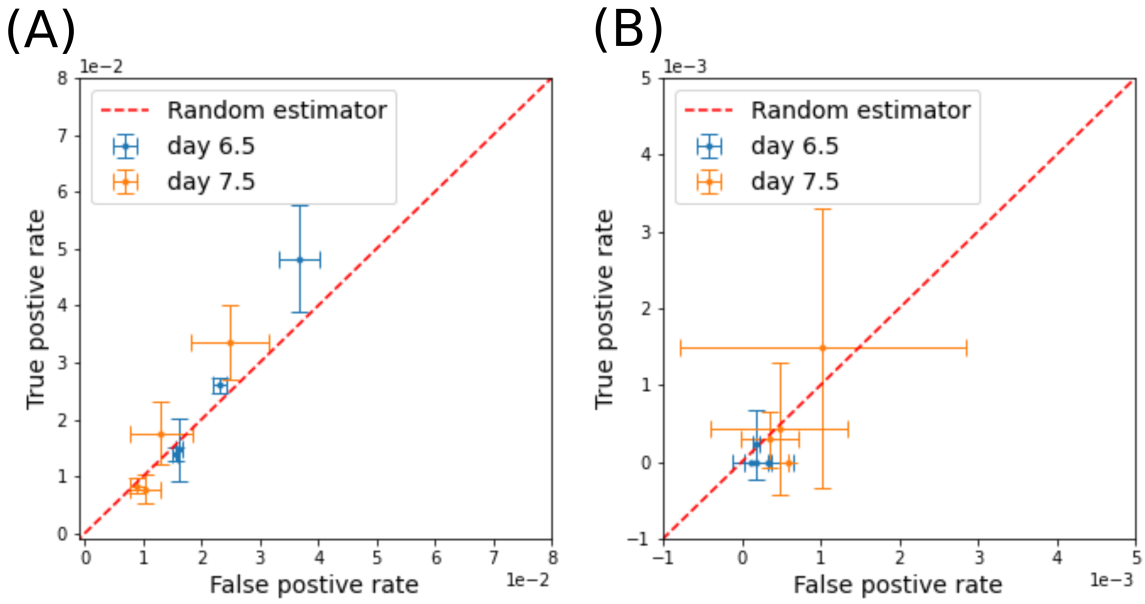


Figure 3.23: Robustness of methods for inferring regulatory networks.
Figure A shows GRNboost2 robustness, notice the large variance in standard deviations between different networks and days. Only four out of eight networks have a performance better than random.
Figure B: The robustness of STN, notice how many networks have a TPR of zero and a standard deviation of zero, and a minimal variance between the false-positive rates. In contrast to other networks, which have a sizeable standard deviation compared to the predictor performance. Overall we can confirm that both STN and GRNboost2 by themselves are not suitable methods for inferring gene-gene regulations for the mouse. This is because almost none of the predictions of the regulators are correct.

Chapter 4

Discussion

Our main question was:
Hypothesis:
Can we infer a gene regulatory network based on regression trees that capture the underlying gene-gene regulation of a mouse's embryonic developmental stages?
To answer this question, we have four steps we need to go through

1) What data are we using, and where does it come from?

2) What methods are we using to infer the network, and what are the underlying assumptions of the method?

3)How does the network structure evolve over the days?

4) Can we find the profiles of essential genes' for survival?

5) How do we validate our network's performance?

4.1 What data are we using, and where does it come from?

In the introduction to the theory section, we commented on the lack of cells for all the days. Day 9.5 lacked the most cells. The lack of cells could contribute to a different network since a small number of nodes can significantly impact the network structure, but we have a high number of nodes meaning above 10^5, which means the distributions would not change that much. The validation of the methods for inferring the regulating would not change either, since most of the genes from the Wnt network are already present in the current networks. There are two ways to improve the data to make more conclusive statements. One could start with taking cells on days 5.5 and 6, so we can get the transi-

tion from pre-gastrulation to gastrulation. This would highlight the transition which happens between 120 cells to 1000, which indicates a significant structural change. The second way, is to get more timestamps between the days 6.5 to 9.5, since one day is too much time. We can not see how the networks transition into each other. So, in conclusion, the dataset is a good fit for the question asked. A dataset with more cells could improve a small aspect of the results but nothing significant. A better time resolution would allow us to see the transition between the stages.

4.2 Underlying assumptions of the method used to infer the networks?

We used a regression tree-based method called GRNboost2 based on earlier results on single-celled yeast data, with a high ratio between TPR and FPR, around 0.86 AUROC, which is way better than random. The performance for GRNboost2 on our single-celled mouse data was only slightly better than random, so around 0.5 AUROC. The validation set is not optimal since it is from a human. However, there are overlapping regulators, so some of the days should have decent performance for the canonical Wnt network, which both the mouse and the human have in common, but that was not seen in the results. Because of this, we have concluded that the underlying assumption that genes need to be placed in the same cell is a bad way to estimate regulators based on single celled RNAsec data from mice. In future studies, clustering the genes before using the methods based on regression trees could potentially increase the performance. We have not commented on STN since its performance was terrible, with zero corrected predictions for eight networks. The rest of the networks still have a slightly better performance than random, due to the lack of predictions. If methods for clustering are used before inferring the regulation, a single tree should not be used to infer the networks. A single regression tree puts too much importance on a few genes, since it has a learning rate of 1. The methods which use multiple trees do not necessarily have this problem, if the learning rate is low like stochastic gradient boosting.

4.3 How does the network structure evolve across the period?

GRNboost2 networks showed that nodes with high degrees would cluster together over the days. The ratio between the number of edges and nodes decrease over the period as well. The network ranking from least dependent to most dependent on nodes with low valued properties: day 9.5,8.5,7.5 and 6.5. Day 9.5 clearly gets impacted more by nodes with high properties compared to the other days. If we compare theses results to the biology it predicts that early gastrulation is more hierarchical than late gastrulation and early organogene-

sis. This does make some sense since most of the restructuring of the embryo and cell differentiation happens before organogenesis. STN predicts the total opposite, that organogensis has a more hierarchical structure than gastrulation. One comment to be made is that the stages happen differently from mouse to mouse, so our prediction is that day 6.5 is gastrulation, day 7.5 and 8.5 are the transition between the stages and at day 9.5 we reach organogensis. This could be confirmed by having a day 10.5.

4.4 Can we find the profiles of essential genes for survival?

Both GRN2boost and STN predict that essential genes regulate a larger number of genes, that they get regulated by more genes and that they tend to be more central for regulation on the network. For GRNboost2, the flow was conclusive; essential genes are regulated more. For STN, the essential genes' tendency shifted during the days. The central tendency was that essential genes were less likely to be regulated a lot. Day 6.5 and 9.5 were inconclusive, since there was statistical evidence of essential genes and viable genes coming from the same distribution, but the means were different. These results show that the regression tree does capture some of the underlying biology but does not capture the regulation. Our guess is that essential genes are expressed in more cells than viable genes, and that this is what the regression tree are picking up on due the genes finding essential genes to be more similar to them self.

4.5 How do we validate our network's performance?

We chose to validate our method for inferring gene- regulatory networks by comparing the network to the human wnt network. We did this because of the lack of studies with specific pathways for the regulation in mice. This has influenced our validation since we do not have a precise estimate of the overlap of genes and how they regulate each other. This gives our results from the validation section less significance, but we know some genes overlap with humans, so we expect some percentiles of the predicted regulators to be correct. This leads us to conclude that the underlying assumption of modern methods that genes placed in the same cell are more likely to regulate each other is not a good fit for single-celled data from a mouse in this period.

4.6 Did the methods capture the underlying gene-gene regulation

The answer is clearly no it does not capture the regulation. However, it does capture the similarity between genes, which is why the methods are still very good at separating the essential and non-essential genes. This also means that changes in the network structure are based on the similarity between the genes and not the regulation.

Bibliography

[1] Yoji Kojima, Oliver H. Tam, and Patrick P. L. Tam. Timing of developmental events in the early mouse embryo. *Seminars in Cell & Developmental Biology*, 34:65–75, October 2014.

[2]

[3] Mitra Kabir, Ana Barradas, George T. Tzotzos, Kathryn E. Hentges, and Andrew J. Doig. Properties of genes essential for mouse development. *PLOS ONE*, 12(5):e0178273, May 2017.

[4] Catriona Y. Logan and Roel Nusse. THE WNT SIGNALING PATHWAY IN DEVELOPMENT AND DISEASE. *Annual Review of Cell and Developmental Biology*, 20(1):781–810, November 2004.

[5] Michaela Bayerlová, Florian Klemm, Frank Kramer, Tobias Pukrop, Tim Beißbarth, and Annalen Bleckmann. Newly Constructed Network Models of Different WNT Signaling Cascades Applied to Breast Cancer Expression Data. *PloS one*, 10:e0144014, December 2015.

[6] KEGG PATHWAY: Wnt signaling pathway - Mus musculus (house mouse).

[7] Yoonjee Kang, Denis Thieffry, and Laura Cantini. Evaluating the Reproducibility of Single-Cell Gene Regulatory Network Inference Algorithms. *Frontiers in Genetics*, 12, 2021.

[8] Luis F. Iglesias-Martinez, Barbara De Kegel, and Walter Kolch. KBoost: a new method to infer gene regulatory networks from gene expression data. *Scientific Reports*, 11(1):15461, July 2021.

Appendix A

Statistical test

A.1 Kolmogorov-Smirnov two sample test

Estimates a probability for two samples coming from the same underlying distribution. The test does this by comparing the larges distance between the distribution of two samples, which returns a statistical threshold based on the significance level α you choose. The threshold and the statistical difference can be calculated the following way

$$D_{n,m} = sup_x |F_{1,n}(x) - F_{2,m}(x)| \tag{A.1}$$

$D_{n,m}$ is the statistical difference
$F_{1,n}$ is distribution of sample n at all point along x
$F_{2,m}$ is distribution of sample m at all point along x
sup_x is the maximal bound of the distribution

we can now calculate the threshold for $D_{n,m}$ by giving in our confidence level α

$$D_{n,m} > \sqrt{-ln(\frac{\alpha}{2}) \cdot \frac{1}{2}} \cdot \sqrt{\frac{n+m}{m \cdot n}} \tag{A.2}$$

α is the minimal confidence level we accept
n number of points in sample n
m number of points in sample m
if $D_{n,m}$ is above the threshold, we can reject the hypotheses that both sample come form the distribution.

A.2 Welch's t-test

The test estimates the probability of how likely two samples have the same mean. It does this by comparing the distance between means, the degree of freedom, and the variance of the samples in a combined distance t and a combined degree

of freedom ν. It assumes the distribution of the samples follows a Gaussian distribution but does not assume the two samples have the same variance as the standard t-student-test do. To calculate the statistics distance t for the Welsh's t-test, we need to use the following equation:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{\sigma_{X1}}{\sqrt{N_1}}^2 + \frac{\sigma_{X2}}{\sqrt{N_2}}^2}} \tag{A.3}$$

$\bar{X}_i$ mean of sample i
σ_{X_i} is the standard deviance of sample i
N_i is the number of points in sample i
t statistical difference

we can now calculate the combined degree of freedom the following way:

$$\nu = \frac{(\frac{\sigma_1^2}{N_1} + \frac{\sigma_2^2}{N_2})^2}{\frac{\sigma_1^4}{N_1 * \nu_1} + \frac{\sigma_2^4}{N_2 * \nu_2}} \tag{A.4}$$

ν_i is the degree of freedom for the sample i which is $N_i - 1$

we can now calculate the p-value by giving ν and t as input. If p is below the significance level, we can reject the hypotheses.

Appendix B

Flow conservation

B.0.1 Is flow preserved form day to day ?

Flow measures how often a gene gets a visit during a random surf on the network. Flow is highly correlated with in_degree and is a mixture of in_degree, a stochastic element. Flow also captures important network patterns like loops and bottlenecks.

B.0.2 Modular Structure

The Direct Flow in section 2.7.1 can also be used to estimate areas that have more common relations than others. These areas are called clusters or modules. Modules are based on how long a Random surf is in a specific area. Modules are a construct, meaning the user of the algorithm chooses how detailed the modules should be. The framework can be seen as the view in google maps; when we zoom in, we get more detailed information about an area like specific countries, buildings, lakes, etc., but less overview of the earth. If we zoom out, we have less detailed information about the area, but more overview of more general structure like oceans, continent and land borders. The parameter which controls the detail level is "Markov time", which sets the bar for how long time a specific area of the graph should be visited to become a module. Low values of Markov time meaning less than one, is detailed modules. High Markov time meaning above one has less detailed modules but captures more overall structure. The mathematical formula can be described in the following way.

$$modules \tag{B.1}$$

STN

Looking at the STN in figure 3.7 b, we can see that days 7.5 and 8.5 are closely related in flow; they only have a few genes indifference. Day 6.5 does not correlate with anything. Day 9.5 has a flow closer to 7.5 and 8.5 but only with half of the genes in common.

GRN2boost

Looking at figure 3.7, we see that all the days look similar except for days 6.5 and 9.5 because they have fewer genes than 7.5 and day 8.5. Day 7.5 and 8.5 have a specific module, which is the same for both days. (Looking the biology in section 2.1.1, the most similar stages of the mouse are late gastrulation day 7.5 and early organogenesis 8.5.) These results could indicate that a subset of genes regulates the early formation of organs in this period. These results also agree with the correlation between properties and out_degree versus in_degree section.

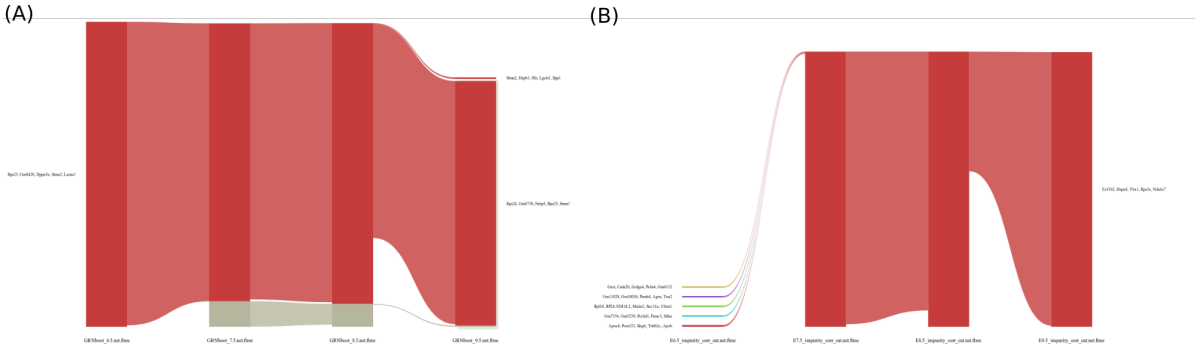


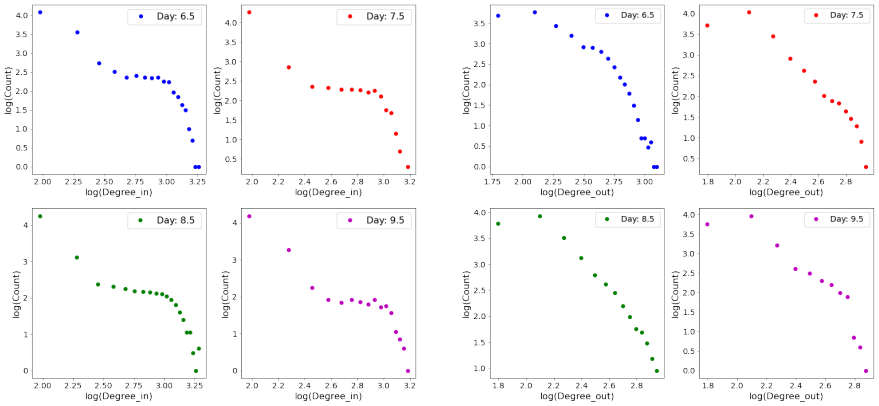
Figure B.1: Alluvial Diagram from day 6.5 to 9.5 for both STN and GRNboost2

Appendix C

out and in_degree distribution on log log plots

log log can easily showcase when things are a distributed as a power law if we do a first order polynomial fit between multiple point 3 + and they are on the that line.

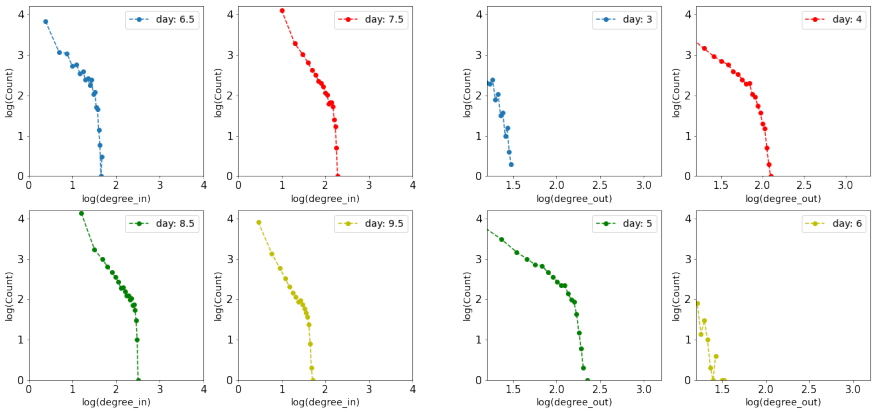
C.1 GRNboost2



(a) (b)

Figure C.1: GRNboost2 degree in and out distribution on log log plot

C.2 STN

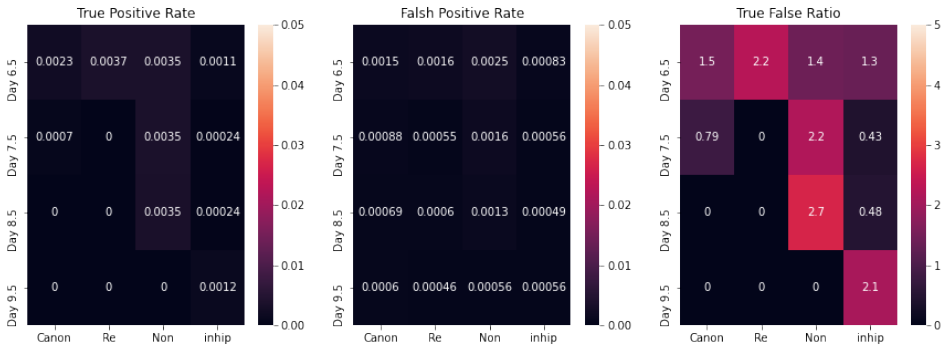


(a) (b)

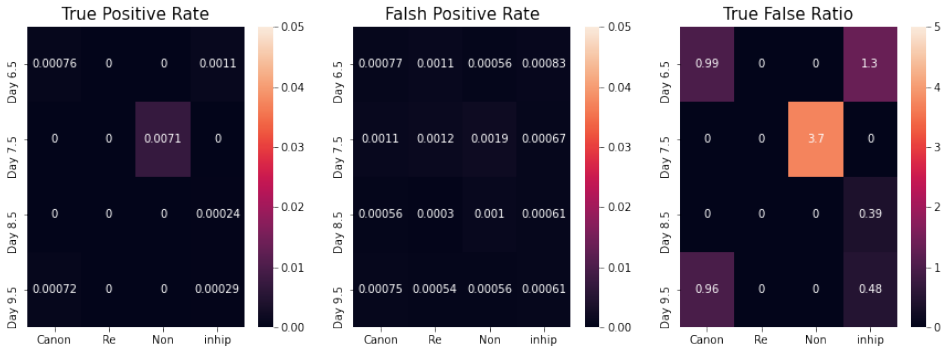
Figure C.2: GRNboost2 degree in and out distribution on log log plot

Appendix D

FPR and TPR for depth 5 and 10



(a) Single tree with max depth 5



(b) Single tree with max depth 10.

Figure D.1

Appendix E

Wnt-correlation

E.0.1 Wnt-mice-correlation different days

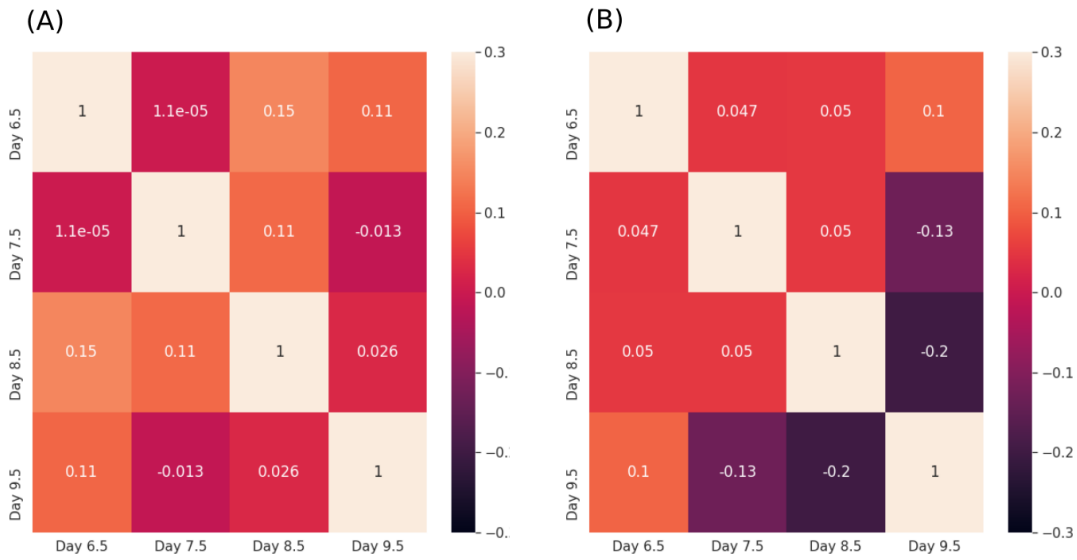


Figure E.1: In figure 3.4 a GRNboost2s Wnt correlation matrix, we can see that day 6.5,8.5 and 9.5 has a Pearson correlation between them of 0.15 and 0.26 seems these days has a overlap of a decent amount of gene expression. Day 6.5 as the only day has a correlation of 0.11 to day 7.5 in the higher range than correlation of day 8.5 and 9.5 to day 7.5 of only 0.09 and 0.06, this lead to a small hit that day 7.5 is different from the other days. When we compare with figure 3.5b STN, we can see day 6.5 to 8.5 has a low correlation between them. We can see that day 6.5 and 9.5 is positively correlated with 0.1 and that 9.5 has a negative correlation to day 7.5 and 9.5 between -0.13 and -0.2. This lead us to the conclusion that small change is happening from day 6.5 to 7.5 and from day 7.5 to 8.5 and between 8.5 and 9.5 a drastic change happens which has a small similarity to the gene expression of day 6.5.