Inferring Signal Transduction Pathways from Gene Expression Data using Prior Knowledge

Deepti Aggarwal

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE
in
Electrical Engineering

Lenwood S. Heath, Chair
Devi Parikh, Co-Chair
Ruth Grene
Guoqiang Yu

August 11, 2015
Blacksburg, Virginia

Keywords: Signal Transduction Pathways, Gene Expression, Inference Engine,Bioinformatics

Inferring Signal Transduction Pathways from Gene Expression Data using Prior Knowledge

ABSTRACT

Plants have developed specific responses to external stimuli such as drought, cold, high salinity in soil, and precipitation in addition to internal developmental stimuli. These stimuli trigger signal transduction pathways in plants, leading to cellular adaptation. A signal transduction pathway is a network of entities that interact with one another in response to given stimulus. Such participating entities control and affect gene expression in response to stimulus . For computational purposes, a signal transduction pathway is represented as a network where nodes are biological molecules. The interaction of two nodes is a directed edge.

A plethora of research has been conducted to understand signal transduction pathways. However, there are a limited number of approaches to explore and integrate signal transduction pathways. Therefore, we need a platform to integrate together and to expand the information of each signal transduction pathway. One of the major computational challenges in inferring signal transduction pathways is that the addition of new nodes and edges can affect the information flow between existing ones in an unknown manner.

Here, I develop the Beacon inference engine to address these computational challenges. This software engine employs a network inference approach to predict new edges. First, it uses mutual information and context likelihood relatedness to predict edges from gene expression time-series data. Subsequently, it incorporates prior knowledge to limit false-positive predictions. Finally, a naive Bayes classifier is used to predict new edges. The Beacon inference engine predicts new edges with a recall rate 77.6% and precision 81.4%. 24% of the total predicted edges are new i.e., they are not present in the prior knowledge.

ACKNOWLEDGMENTS

I would like to express my sincere thanks to my advisor, Dr. Lenwood S. Heath, for providing me an opportunity to work in Bioinformatics and for his patience, motivation and guidance. I thank him for patiently listening to my ideas and giving them direction. His guidance has helped me throughout my Masters research and the writing of my thesis. I am also thankful to the rest of my committee Dr. Devi Parikh, Dr. Ruth Grene, Dr. Guoqiang Yu. I would like to thank Dr. Grene who has patiently explained me the biological perspective of the work. Dr. Grene has provided her valuable feedback at the time of dilemmas throughout the research and has provided biological meaning to this research. Dr.Devi Parikh and Dr. Guoqiang Yu has been a valuable support from ECE department.

I would like to thank the ECE department, Virginia Tech for supporting me and guiding me throughout my first year of graduate study. My sincere thanks goes to Dr. Song Li for understanding my work in such a short time and providing me solutions for my research problems.

I thank my fellow lab mates Elijah Myers, Ying Ni, Doaa Altarawy and Delasa Aghamirzaie for discussions to solve my problems and providing me the support throughout my work. I would also like to thank my friends Vaishnavi, Dhaarna , Sarthak, Arnika and Tinny for motivating me to do my best and co-operating with me, whenever I had deadlines.

Last but not the least, I would like to thank my family: my parents Gajanand Aggarwal and Saroj Aggarwal and my brothers Varun Aggarwal and Dhruv Aggarwal, for having their faith and trust in me. I can never thank them enough for their unconditional support and love towards me.

TABLE OF CONTENTS

List of Figures

LIST OF TABLES

Chapter 1

INTRODUCTION

A signal transduction pathway depicts cellular responses to internal or external environmental changes by activation or suppression of genes. Plant cells produce various internal signals, such as hormones, in response to these stimuli. Hormones are molecules that trigger a series of responses called signal transduction pathways to respond to external or internal stimuli [54]. These are generally represented by networks, where the nodes are genes, transcripts, or proteins and where the edges are the interactions between pairs of nodes. It is important to understand signaling pathways and their regulation to gain a deep insight into plant responses to stimuli [14, 27, 42].

Recently, high-throughput gene expression data sets have offered insights into signal transduction pathways [18]. Bioinformaticians have proposed a large set of approaches, for example, TIGRESS, ARCANE, and others, to understand these pathways [33]. A total of 48,377 papers were published on signal transduction pathways in 2007 focusing on gene regulatory details in different signal transduction pathways [59]. However, the amount of information available on signal transduction pathways lags behind that available for gene expression data sets [30, 35]. Therefore, we need to develop computational methods that can expand signal transduction pathways with respect to the information present in gene expression data sets and also integrate prior knowledge. Furthermore, we should be able to predict the effect of addition of new nodes and edges on existing signal transduction pathways. One of the major challenges in developing these computational methods is to determine possible new edges within and among those signal transduction pathways [60, 26].

The Beacon inference engine proposed as a part of the Beacon project addresses these challenges.

The Beacon project develops software tools to provide computational support to the plant biologist. It helps the biologist to represent signal transduction pathway using standard graphical notations and archiving in a database. The software also provides a way of curating signal transduction pathways from experts for future use. The storage of signal transduction pathways in a database helps users to apply semantics to the databases. The Beacon simulation tool will help the biologist to produce results from the database that can be tested using the Beacon inference engine. The Beacon inference engine allows the user to construct and expand the present known signal transduction pathways with respect to the pathway components, inferring new edges. An added function is to integrate different signal transduction pathways.

In this thesis, I develop a network inference approach for the Beacon inference engine. Network inference is the reconstruction of a network from a high throughput gene expression time series data set [28]. In this method, prior knowledge will be used, along with high throughput gene expression time series data, to expand and integrate a known gene regulatory network. In the process of expansion, new edges will be predicted within the network. The first step, consist in the use of mutual information and context likelihood relatedness to predict edges from gene expression time-series data. Subsequently, it incorporates prior knowledge to limit the false-positive predictions from the first step. Finally, the Bayes classifier is used to predict new edges in the software engine. The novelty of the proposed approach is the implementation of a semi-supervised algorithm in addition to the mutual information measure to predict directed edges between the nodes.

The remaining contents of this thesis consists of 6 chapters. Chapter 2 describes the fundamental biological, computational, and statistical concepts employed in this research. It also defines new terms, concepts, and notations that are used in later chapters. In Chapter 3, the signal transduction pathways are explained in detail to formulate the actual problem and discussed with the description of existing tools. Chapter 4 describes the data and the preprocessing data models that were implemented. Chapter 5 elaborates the methodologies used in this research along with their implementation in the Beacon inference engine. It also describes the pipeline of algorithms used for the development of the Beacon inference engine. In Chapter 6, the results of the Beacon inference engine applied to a different data set are described. In Chapter 7, the potential drawbacks of the software are examined, and future directions of research are proposed.

Chapter 2

PRELIMINARIES

This chapter describes the terminology and notation used in the remainder of the thesis.

2.1 Biological Concepts

DNA or *deoxyribonucleic acid* is a molecule that contains the genetic material of an organism [51]. DNA has a double helix structure. A nucleic acid is a linear chain of nucleotides where each nucleotide comprises a nitrogenous base named adenine (A), thymine (T), guanine (G) or cytosine (C) along with a 5-carbon sugar and one phosphate molecule. The genetic information is the specific sequence of bases [8].

A *gene* is a unit of heredity that controls traits of a living organism [8]. It is a specific sequence of nucleotides present in DNA. The gene encodes the protein information which is copied into RNA for the manufacturing of the final protein product. Thus, genes encode information to build a cell and hence all its proteins including enzymes. A gene is composed of exons and introns as shown in Figure 2.1. During the process of RNA synthesis, the introns are spliced out, and the exons are transcribed to yield the mature messenger RNA. In this thesis, I study genes that are expressed during seed development in the model plant *Arabidopsis thaliana*.

The region of the gene upstream of the 5' untranslated region (5'UTR), is called the *regulatory region* of the gene [57]. The promoter is a part of the gene that controls transcription of the gene. It provides specific sites of attachment for transcription factors.

Transcription is the step in protein production in which the DNA message is copied to messenger

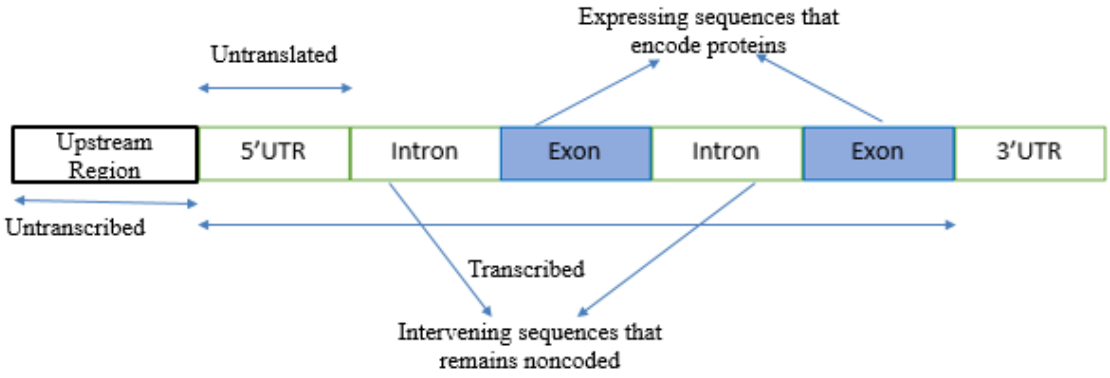

Figure 2.1: Gene model of a eukaryotic gene

RNA with the help of the enzyme RNA polymerase. Transcription plays an important role in plant response to stimuli. Genes are turned off and on in a coordinated manner. Transcription factors (TFs) are proteins that co-ordinate the function of groups of genes. As proteins or protein complexes, they bind to specific DNA sequences called *cis*-elements located in the gene promoter. Transcription factors are either general or gene-specific [57]. A gene-specific transcription factor targets a specific gene, or group of genes.

Gene expression provides one kind of information about the process used in the synthesis of proteins or functional product from genes under a given condition, or in a specific genotypes or types. Gene expression data sets provide information about genomic activities in response to external stimuli [1]. Therefore, their analysis can aid in comprehending the organism's complexity.

The processing of a gene expression data set results in a gene expression matrix [4]. A gene expression matrix provides information about the similarities and differences among the expression of genes. It contains a row for each gene and its expression at a time-point in a column. In a high-throughput gene expression data set, the number of genes is generally greater than the number of experiments performed to get the data set [63].

A signal transduction pathway is a type of regulatory network. The nodes in the network are the genes or transcripts, and the edges represent the interactions between them. The application of computational methods to the gene expression data and the subsequent increase in biological knowledge is leading to the development and understanding of large and complex networks [41, 61].

2.2 Computational Terms and Concepts

A *directed graph* $G = (V, E)$ is a graph in which the set of vertices are connected by directed edges. Mathematically, $G = (V, E)$, where $E \subseteq \{(u, v)|u, v \in V\}$ [40].

An *undirected graph* $G = (V, E)$ is a graph in which the set of vertices are connected by bidirectional edges and E is the set of edges. Formally, an edge (u, v) satisfies $(u, v) = (v, u)$ [38].

A *complete graph* $G(V, E)$ is an undirected graph in which each pair of vertices is connected with an edge. A graph with n vertices is complete if it has $\frac{n(n-1)}{2}$ edges, and the number of edges incident to each vertex is $n - 1$ [58].

A *weighted graph* $G = (V, E)$ is one in which a numerical value (*weight*) is assigned to an edge via a weight function $W : E \to \mathbb{R}$.

A *curated graph* $G(V, E)$ is defined as a weighted graph that represents prior knowledge. In this thesis, prior knowledge refers to a known signal transduction pathway.

2.3 Statistical Terms and Concepts

A sample space S is the set of all possible outcomes of an experiment.

An event β is a set of possible outcomes of interest and is a subset of the sample space, that is, $\beta \subseteq S$.

Probability is the measure of likeliness of the occurrence of an event. It helps in assigning a value to the event occurrence between 0 and 1. The probability of an event A is $P(A)$ and the complement is $1 - P(A)$. S is the sample space of the event.

A *probability* $P(A)$ is a function $P : S \to \mathbb{R}$ such that

1. $P(A) \geq 0$ for all $A \in S$.
2. $P(A) \leq 1$.
3. Whenever $A_1, A_2, \ldots$ are pairwise disjoint sets in S then,

$$P\big(\cup_{n=1}^{\infty} A_n\big) = \sum_{n=1}^{\infty} P\big(A_n\big). \tag{2.1}$$

A random variable, X, is a function that assigns a real numbered value to every possible event in a sample space. $X : S \to \mathbb{R}$ where S is the sample space [2].

The *probability density function* $P(Y)$ for a random variable Y having density $P(y)$ is

$$P(y) = P(Y = y). \tag{2.2}$$

The probability density function [36] is such that

1. $P(y) \geq 0$ for all y;

The *joint probability function* for two random variables X and Y having joint density $P(x, y)$ is

$$P(X = x, Y = y) = P(x, y). \tag{2.3}$$

The *joint probability density function* [36] has the following property:

$p(x, y) \geq 0$ for all possible values $x,\ y$;

A pair of events is independent if the occurrence of one event does not affect the other event. A set of n events $\{A_1, A_2, \ldots, A_n\}$ is *independent* if

$$P(\bigcap_{i \in I} A_i) \ = \ \prod_{i \in I} P(A_i) \tag{2.4}$$

for every subset I of $\{1, \ldots, n\}$

The *conditional probability* is the probability of occurrence of an event A in the presence of a known event B It is defined as

$$P(A|B) = \frac{P(A \cap B)}{P(B)}, \tag{2.5}$$

assuming $P(B) > 0$.

The *expected value or mean* is the average of the values a random variable can take. It indicates the central point of the random distribution. The *mean* of a random variable X having m values $x_1, x_2, x_3 \ldots\ldots, x_m$ is:

$$\mu = E[X] = \sum_{j=1}^{m} x_j p(x_j) \tag{2.6}$$

where x_j represents the j^{th} value of the random variable X and has $p(x_j)$ probability distribution function.

The *standard deviation* measures the variability of data with respect to the mean μ. The *standard deviation* is generally denoted by σ and is calculated as follows:

$$\sigma^2 = E[(X - \mu)^2], \tag{2.7}$$

where μ is the mean of the random variable X.

Entropy is a measure of the randomness of a system [7]. The *entropy* for a random variable X with m finite states $x_1, x_2, \ldots, x_m$is

$$H(X) = -\sum_{i=1}^{m} p(x_i) \log(p(x_i)). \tag{2.8}$$

where $p(x_i)$ is the probability of x_i.

Joint entropy is a measure of randomness in the joint distribution of a pair of random variables. The *joint entropy* $H(X, Y)$ of two random variables X and Y with m and n finite states is defined as:

$$H(X, Y) = -\sum_{j=1}^{n} \sum_{i=1}^{m} p(x_i, y_j) \log(p(x_i, y_j)), \tag{2.9}$$

where $p(x_i, y_j)$ is the probability of x_i, y_j.

The mutual information is a measure of the dependence between two random variables. It tells about the relationship between two variables [9]. The higher the mutual information, the more it signifies the random variables are associated with each other [53]. The *mutual information* for two random variables X and Y is

$$I(X; Y) = H(X) + H(Y) - H(X, Y) \tag{2.10}$$

where $H(X), H(Y)$ is the entropy of random variables X and Y. $H(X, Y)$ is the joint entropy of the random variables X and Y.

The z-score for a random distributed variable X is the distance of X from the mean $\mu(X)$ measured in terms of standard deviation $\sigma(X)$ units. It is a by random variable Z [13], as follows:

$$Z = \frac{X - \mu(X)}{\sigma(X)}. \tag{2.11}$$

Chapter 3

PROBLEM DEFINITION

A signal transduction pathway plays an important role in understanding plant responses to changing conditions. It is usually represented as a network where the biological components represent the nodes and the interactions between them are the edges of the network. Generally, it is difficult to synthesize and compute signal transduction pathways. There are various tools to analyze signal transduction pathways. The existing tools limit the expansion of already known signal transduction pathways. In this thesis, an inference engine has been developed. This tool will assist biologists in exploring and understanding different signal transduction pathways.

3.1 Background

Plants regulate normal development growth processes by responding to various internal and external signals such as light, temperature, and water. They respond at the molecular and cellular levels to these stimuli via a network of events called a signal transduction pathway. Signal transduction pathways coordinate the expression of genes in response to the stimuli. Signal transduction pathways are generally divided into four phases as shown in Figure 1.1 [6].

The first phase involves receiving a stimulus and generating a response to it. Plants receive the stimulus via help of membrane associated proteins or receptors[43]. These proteins have both intracellular as well as extracellular binding sites. An extracellular binding site works as the receptor for the external signals and functions as the primary messenger for the cell.

The second phase is signal transduction, which is amplification of the received signal. The signal

amplification is achieved by generation of second messengers within the cell. For example, a single molecule can lead to the activation of an enzyme that produces many second messengers [49]. One of the mechanisms of signal transmission is protein phosphorylation [50]. Protein phosphorylation is the process of the attachment of a phosphate (PO4) group to a protein. The new phosphorus group alters the structure of the protein, thus affecting its function. The elevated amount of secondary messenger leads to the production of protein kinases in the cell, thus inducing protein phosphorylation.

The third phase is the binding of the proteins to the secondary messengers. These proteins are known as switch proteins. Switch proteins undergoes changes that cannot be reversed. They affect other proteins leading to gene activation or repression [6].

The last step is the signal mediation that is signal termination. The signal has to be terminated in order to let the plant respond to new signals.

3.2 Literature Review

Network inference is a method for reconstructing a biological network from a high-throughput gene expression data set, called reverse engineering biological networks in the DREAM (Dialogue on Reverse-Engineering Assessment and Methods) project [22]. The DREAM project proposes network inference methods for metabolic pathways, gene regulatory networks, molecular interaction networks, and signaling pathways. In recent years, various methods, such as correlation, regression, information theory, and algebraic methods, have been proposed to infer biological networks [41, 19, 23]. However, most of the above methods are not applicable to the problem, due to unidirectional inferring of the gene interactions.

Directed inferences [21, 46], that is, directed edges, narrows down the problem computationally, as it can take into account whether antecedent is likely to be the result of the precedent. Such directed relationships can be suggested by various types of semantics. Some prior approaches such as TIGRESS and ARCANE use direction to annotate the relationship between the nodes in the gene network [33]. However, the best performing methods in DREAM used a mixture of data sets, with perturbation, time series, and knock out.

TIGRESS (Trustful Inference of Gene Regulation with Stability Selection) studies gene regulatory networks using network inference on gene expression data sets [23] . It considers network inference

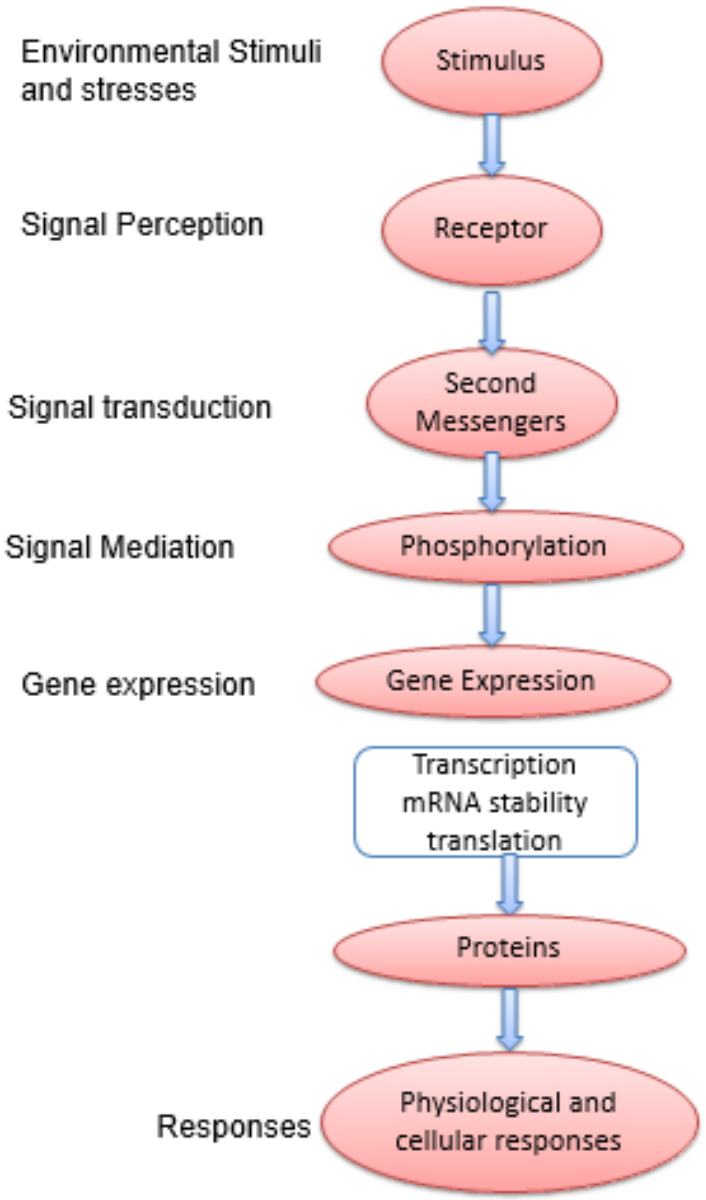

Figure 3.1: Description of Signal transduction phases in Plants, modified from [49].
Figure modified from K. Shinozaki and E. S. Dennis, Cell signalling and gene regulation: global analyses of signal transduction and gene expression profiles, Current Opinion in Plant Biology, 6 (2003), pp. 405-409

as a sparse regression problem and studies the performance of a popular feature selection method, least angle regression (LARS) combined with stability selection, for that purpose. However, the problem is that method has been applied to only steady state gene data sets and does not perform well with time series expression data sets.

Time series data sets have been used in methods such as MRNET, CLR, Arcane, and many structural methods for the reconstruction of gene regulatory networks [32]. The analysis of the above methods based on similarity measures shows that Spearman Rank correlation with AWE scoring scheme performs better in the case of time series data with Gaussian noise in it [61]. The ANOVA method [55] described in the DREAM5 challenge uses time series data.

Pearl [37] used different graph models and suitable algorithms to infer the presence or absence of directed links. The implementation of Bayesian network, [62, 21, 17] based on directed acyclic graph, and structural equation modeling [48, 29, 15], is done to understand different signaling pathways. The above methods are based on unsupervised learning, which means that the network is predicted without using prior knowledge [11].

The SIRENE method [34] uses supervised learning, i.e., support vector machines, to infer gene regulatory networks on the basis of prior known gene pairs. It uses a steady state gene expression data set and an array of known regulation relationships between gene and target as an input. In the case of unsupervised learning, clustering of genes with similar expression is done, however, in the case of supervised learning the similarity between gene expression helps in clustering of genes that may have the same regulator. The disadvantage of this method is that it does not predict the new interaction in the gene it does not have any prior knowledge about them.

Most of the above methods are not applicable due to the use of undirected edges to represent an interaction. Directed edges can narrow down the problem, as it can take into account whether the antecedent is the result of the precedent through some unknown mechanism. The Beacon inference engine uses this approach to predict new connections in signal transduction pathways. In addition, the above mentioned methods that annotate information flow direction do not use any kind of prior knowledge to predict and expand the known signal transduction pathways. In the proposed inference engine, prior knowledge is used along with the directed inference to predict new connections in the signal transduction pathways.

3.3 Mathematical Definitions

Let Y be a gene set of q elements $Y = \{g_1, g_2, ..., g_q\}$ and let T be a set of n time points $T = \{t_1, t_2, ..., t_n\}$. The *expression value* for gene g_i at time t_j is u_{ij} and the gene expression matrix is $U = (u_{ij})$, where $i \in Y$ and $j \in T$. In general, gene expression values $u_{i,j} \in \mathbb{R}$ and are ≥ 0. Let $H = (V, E)$, be a directed curated graph, where $V \subseteq Y$.

3.4 Problem Definition

Given time series data of a gene data set having q genes as gene expression matrix $U = (u_{ij})$, where $i \in Y$ and $j \in T$. Firstly, we want to predict new edges (u, v) between the nodes in the curated graph $H = (V, E)$ where $u \in V$ and $v \in V$. Also, we want to predict new edges in the curated graph $H = (V, E)$ with respect to the genes present in the gene set Y other than V in $H = (V, E)$.

Chapter 4

DATA MODEL

This chapter describes the data model used in this thesis.

4.1 Gene Expression Data

In the Beacon inference engine, network reconstruction has been done with data from model organism *Arabidopsis thaliana*. The gene name data set and time expression data set used for the Beacon inference engine analysis were provided by Dr. Eva Collakova and Dr. Ruth Grene, Professors at Virginia Tech. The gene name data sets consist of all the known genes of *Arabidopsis thaliana* that are expressed in developing seeds. It contains fields tcon_id gene_id, gene_short_name, AGI index, tss_id, locus and description. The gene_id is the unique identifier of the gene and relates to the gene identifier at NCBI. The AGI index represent the gene id as per the *Arabidopsis thaliana* initiative, and tss_id represents the transcription site of the gene. The locus field indicates the locus of the gene on the chromosome on which it is located. The last field describes the function of the gene. The gene name data set Table 4.2 is related to the time expression data set through the unique identifier that is gene_id or the transcript name, i.e,TCON_id . The time series data sets have been generated by an NSF funded project on the molecular regulation of seed development. The gene expression data represent the intensity values and are stored in a gene expression matrix. The transcripts whose expression were detectable at one or more time points during time course of embryo development, formed the Data Set1. Subsequently, a gene data set containing time course expression values of all splice variants in two genotypes were generated at 6 time points ($7d, 8d, 10d, 12d, 13d, 15d$). 8000

transcripts were identified out of 40,000 as being differentially expressed between the two genotypes.

gene_id	gene_short_name	AGI	TAIR10_Functional_Computatio	tss_id	locus
XLOC_000001	ANAC001	AT1G01010	NAC domain containing protein	TSS1	1:3630-5899
XLOC_000002	DCL1,MIR838A	AT1G01046;AT1G01040	;dicer-like 1 (DCL1); CONTAINS	TSS2,TSS3,TSS4	1:23145-33153
XLOC_000003	AT1G01073	AT1G01073	unknown protein; FUNCTIONS IN	TSS5	1:44676-44787
XLOC_000004	IQD18	AT1G01110	IQ-domain 18 (IQD18); FUNCTIC	TSS6,TSS7,TSS8	1:52047-54692
XLOC_000005	AT1G01115	AT1G01115	unknown protein; FUNCTIONS IN	TSS9	1:56623-56740
XLOC_000006	GIF2	AT1G01160	GRF1-interacting factor 2 (GIF2)	TSS10	1:72338-74967
XLOC_000007	AT1G01180	AT1G01180	S-adenosyl-L-methionine-depen	TSS11	1:75582-76758
XLOC_000008	MIR165A			TSS12	1:77599-80484
XLOC_000009	AT1G01210	AT1G01210	DNA-directed RNA polymerase,	TSS13,TSS14	1:88677-90767
XLOC_000010	FKGP	AT1G01220	L-fucokinase/GDP-L-fucose pyrc	TSS15	1:91375-95748
XLOC_000011	AT1G01225,AT1G012:	AT1G01230;AT1G01225	ORMDL family protein; FUNCTIC	TSS16,TSS17,TSS18	1:95851-99240
XLOC_000012	AT1G01240	AT1G01240	unknown protein; BEST Arabido	TSS19,TSS20	1:99893-101834
XLOC_000013	AT1G01260	AT1G01260	basic helix-loop-helix (bHLH) D	TSS21,TSS22	1:108945-111609
XLOC_000014	CYP703A2	AT1G01280	cytochrome P450, family 703, s	TSS23	1:112262-113947
XLOC_000015	CNX3	AT1G01290	cofactor of nitrate reductase an	TSS24	1:114273-116311
XLOC_000016	AT1G01300	AT1G01300	Eukaryotic aspartyl protease far	TSS25	1:116942-118764
XLOC_000017	AT1G01305	AT1G01305	unknown protein; FUNCTIONS IN	TSS26	1:119396-119997
XLOC_000018	AT1G01310	AT1G01310	CAP (Cysteine-rich secretory pro	TSS27	1:120153-130577
XLOC_000019	AT1G01350	AT1G01350	Zinc finger (CCCH type) family	TSS28	1:135133-139169

Table 4.1: Sample of Data Set1 gene name *Arabidopsis thaliana* data set

4.1.1 Data Set1 (Wild type embryos time course)

The Data Set1 (Wild type embryos time course) represents the behavior of all the detectable known transcripts in the *Arabidopsis thaliana* embryo over a time course of seed development. The Beacon inference engine divides the Data Set1 into two data sets, that is, Data Set1 gene name *Arabidopsis thaliana* data set and Data Set1 *Arabidopsis thaliana* time series data set. The gene name data set relates to the time series data set through a unique identifier gene name. The Data Set1 *Arabidopsis thaliana* gene name data set follows the structure as represented in Table 4.1. The gene name data set consist of 10,000 genes and contains fields gene_id, gene_short_name, AGI index, tss_id, locus and description. The Data Set1 *Arabidopsis thaliana* time series data set contains field gene_id along with its expression values at 7 time points $(7d, 8d, 10d, 12d, 13d, 15d, 17d)$; see Table 4.2

4.1.2 Data Set2 (Differential Expression transcripts)

The Data Set2 (Differential Expression transcripts) represents the differential behavior of 8000 detectable transcripts in two *Arabidopsis thaliana* embryos over a time course of seed development. The Beacon inference engine divides the Data Set2 in two data sets, that is, Data Set2 transcripts name *Arabidopsis thaliana* data set and Data Set2 *Arabidopsis thaliana* time series data set. The Data Set2 transcripts name *Arabidopsis thaliana* data set relates to the time series data set through a unique identifier transcript name i.e., TCON_id. The Data Set2 transcripts name *Arabidopsis thaliana* data set follows the structure shown in Table 4.3. It contains fields *gene_id*, *transcript_name*, AGI index, *class_code*, nearest ref, locus, description. The Data Set2 *Arabidopsis thaliana* time series data set contains field TCON_id along with its expression values at 6 time points $(7d, 8d, 10d, 12d, 13d, 15d)$; see Table 4.4.

gene_id	d7_WT_FPKM	d8_WT_FPKM	d10_WT_FPKM	d12_WT_FPKM	d13_WT_FPKM	d15_WT_FPKM	d17_WT_FPKM
XLOC_000001	0.876846	0.555093	0.133433	0.418929	0.504827	1.13617	1.34929
XLOC_000002	1.62084	2.02802	5.26349	6.25829	4.89964	3.41851	3.59797
XLOC_000003	0	0	0	0	0	0	0
XLOC_000004	5.94789	3.87047	2.2977	2.63623	1.36597	0.434363	0.179354
XLOC_000005	0	0	0	0	0	0	0
XLOC_000006	33.0851	30.3209	27.2214	30.3298	37.1312	61.1768	64.1005
XLOC_000007	0.0447507	0.0552952	0.193438	0.814342	1.62088	14.3186	15.4317
XLOC_000008	0.0266753	0.00949203	0.0953671	0.249898	0.562627	1.02909	1.20143
XLOC_000009	17.5279	10.5394	4.28049	4.30843	10.4455	27.8088	15.6147
XLOC_000010	2.11129	2.14301	2.10665	1.58969	1.88851	1.19488	1.12845
XLOC_000011	22.8851	35.8957	88.1796	93.8142	90.5739	99.297	92.6698
XLOC_000012	1.03475	2.11061	10.8882	12.3799	21.407	80.9719	211.773
XLOC_000013	6.9393	8.16527	12.7493	17.4822	21.2145	14.2379	6.20847
XLOC_000014	0	0	0	0.0105578	0.00989481	0	0
XLOC_000015	10.889	9.49646	9.79181	10.5481	10.3534	9.2577	7.55219
XLOC_000016	47.6721	62.7126	81.5441	102.571	81.7031	88.1307	39.7768
XLOC_000017	0.111183	0.0730221	0.47233	3.46273	9.21936	14.5205	1.54422
XLOC_000018	1.23757	0.625007	0.238245	0.371281	0.673185	0.748135	0.622384

Table 4.2: Sample of Data Set1 *Arabidopsis thaliana* time series data set

tcons_id	locus id	AGI	gene name	anscript nam	nearest ref	class code	ucntional description
TCONS_00012743	XLOC_00764	AT1G68825	DVL5	AT1G68825.1	AT1G68825.1	=	ROTUNDIFOLIA like 15 (RTFL15); INVC
TCONS_00012744	XLOC_00764	AT1G68825	DVL5	AT1G68825.2	AT1G68825.2	=	ROTUNDIFOLIA like 15 (RTFL15); INVC
TCONS_00021158	XLOC_01286	AT2G31980	AtCYS2	AT2G31980.1	AT2G31980.1	=	PHYTOCYSTATIN 2 (CYS2); FUNCTION:
TCONS_00043789	XLOC_02670	AT5G25210	AT5G25210	AT5G25210.1	AT5G25210.1	=	unknown protein; BEST Arabidopsis t
TCONS_00050009	XLOC_03046	AT5G25190	AT5G25190	AT5G25190.1	AT5G25190.1	=	Integrase-type DNA-binding superfa
TCONS_00007207	XLOC_00428	AT1G02180	AT1G02180	AT1G02180.1	AT1G02180.1	=	ferredoxin-related; FUNCTIONS IN: n
TCONS_00019813	XLOC_01205	AT2G18120	SRS4	AT2G18120.1	AT2G18120.1	=	SHI-related sequence 4 (SRS4); CONT
TCONS_00022225	XLOC_01350	AT2G42885	AT2G42885	AT2G42885.1	AT2G42885.1	=	Defensin-like (DEFL) family protein;
TCONS_00017869	XLOC_01076	AT2G42840	PDF1	CUFF.2976.3	AT2G42840.1	x	protodermal factor 1 (PDF1); Has 746
TCONS_00017868	XLOC_01076	AT2G42840	PDF1	CUFF.2976.2	AT2G42840.1	x	protodermal factor 1 (PDF1); Has 746
TCONS_00049311	XLOC_03006	AT5G17810	WOX12	AT5G17810.2	AT5G17810.2	=	WUSCHEL related homeobox 12 (WO
TCONS_00043153	XLOC_02634	AT5G17800	AtMYB56	AT5G17800.1	AT5G17800.1	=	myb domain protein 56 (MYB56); CO
TCONS_00032465	XLOC_01987	AT3G54910	AT3G54910	AT3G54910.2	AT3G54910.2	=	RNI-like superfamily protein; FUNCTI
TCONS_00051236	XLOC_03139	AT5G41610	ATCHX18	AT5G41610.1	AT5G41610.1	=	ARABIDOPSIS THALIANA CATION/H+
TCONS_00051235	XLOC_03139	AT5G41610	ATCHX18	AT5G41610.2	AT5G41610.2	=	ARABIDOPSIS THALIANA CATION/H+
TCONS_00006631	XLOC_00401	AT1G76892	AT1G76892	AT1G76892.1	AT1G76892.1	=	
TCONS_00021567	XLOC_01311	AT2G36270	ABI5	CUFF.2749.1	AT2G36270.1	j	ABA INSENSITIVE 5 (ABI5); CONTAINS
TCONS_00037209	XLOC_02273	AT4G36870	BLH2	AT4G36870.2	AT4G36870.2	=	BEL1-like homeodomain 2 (BLH2); CC
TCONS_00026241	XLOC_01601	AT3G44880	LLS1	CUFF.4097.2	AT3G44880.1	j	ACCELERATED CELL DEATH 1 (ACD1); F

Table 4.3: Sample of Data Set2 transcripts name *Arabidopsis thaliana* data set

tcons_id	WT_7	WT_8	WT_10	WT_12	WT_13	WT_15
TCONS_00012743	41.25	52.8333333	26.3611111	17.25	5.77777778	0
TCONS_00012744	18.8333333	21.4166667	9.58333333	7.41666667	3.13888889	0
TCONS_00021158	141.75	2775.91667	4035.69444	1209.16667	816.333333	115.388889
TCONS_00043789	54.6666667	119.222222	100.305556	64.5833333	67.8888889	89.0833333
TCONS_00050009	18.0833333	36.1666667	12	8.19444444	3.75	3.97222222
TCONS_00007207	284.972222	217.555556	72.25	68.8888889	52.6666667	17.0555556
TCONS_00019813	33.9166667	34.5833333	4.77777778	5.25	2.55555556	0.75
TCONS_00022225	7.22222222	45.3055556	48	23.0555556	10.8333333	2.08333333
TCONS_00017869	44.5	69.4166667	111.166667	87.25	136.527778	119.305556
TCONS_00017868	41.6111111	66	108.222222	84.75	130.75	116.305556
TCONS_00049311	126.833333	218.694444	291.138889	271.166667	181.611111	41.3333333
TCONS_00043153	57.4166667	38.75	27.3333333	38.3055556	42.0277778	11.4166667
TCONS_00032465	3.83333333	7.36111111	5.5	5.36111111	2.86111111	0.5
TCONS_00051236	69.1666667	149.083333	218.416667	202	138.083333	52.3333333
TCONS_00051235	99.75	250.583333	356.083333	312.638889	222.25	103.972222
TCONS_00006631	4.19444444	1.08333333	0.44444444	0	0.66666667	1.16666667
TCONS_00021567	50.9166667	136.777778	222.111111	317	760.777778	4264.97222
TCONS_00037209	262.5	340.833333	173.277778	213.166667	231.416667	84.0833333
TCONS_00026241	630.083333	1330.44444	1138.02778	1312.44444	2220.75	4445.77778

Table 4.4: Sample of Data Set2 *Arabidopsis thaliana* time series data set

Chapter 5

METHODOLOGY

For decades, biologists have been trying to understand and explore different signal transduction pathways. Numerous network inference methods have been proposed to expand and detect cross-signaling between different signal transduction pathways [45]. It remains unclear which method is more effective and is more efficient in exploring signal transduction pathways. This thesis focuses on network inference methods to explore new connections in signal transduction pathways. A network inference method generally requires a measure to predict edges between gene pairs and a threshold to limit the false positive predictions and an algorithm to integrate the prior knowledge. The Beacon inference engine follows a three step network inference pipeline to predict new edges between gene pairs; see Figure 5.1.

The first step in the pipeline is prediction, which is to predict connections using a time series

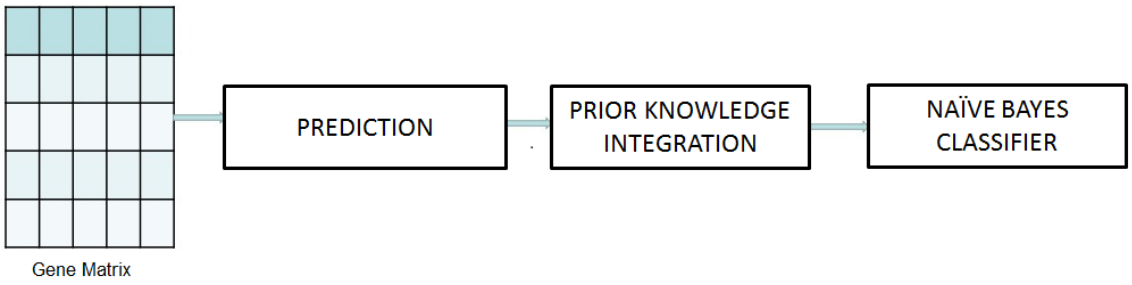

Figure 5.1: Beacon Inference Engine Basic Pipeline

gene expression data set. One of the challenges in analyzing a time-series gene expression data set is the noise present in it [5]. The noise may be due to biological variability of the gene expression in gene experiments. In this work, a B-spline estimator binning approach [16] has been implemented to compensate for the noise present in the data. The simplest binning approach assumes that a data point can be present in only one bin. However the data points at the border of the bins can be represented in two bins with a little bit of uncertainty. The B-spline binning approach assumes a possibility of a data point lying in three different bins, thereby compensating for the noise present in the data set. Another challenge is to exploit the time-series data set to predict the directed connections between the genes. In this research, directed mutual information is used along with context likelihood relatedness to predict the information flow between gene pairs [25]. The directed mutual information measure shifts the time-series of a gene one step into the future compared to the other gene to predict the directed relationship between them. The context likelihood relatedness provides a significance to the directed mutual information values for the prediction of edges.

A second important step is prior knowledge integration, as shown in the Beacon inference engine of Figure 5.2, which is to set a threshold to limit the false positives. In general, the threshold is a value above which all the predicted edges are considered. However, in our research, we have used a numerical range as a threshold. The values above and below the numerical range are ignored. The context likelihood relatedness values corresponding to the prior knowledge are interpolated in a histogram and a numerical range is defined to limit the false positives. The numerical range is the histogram interval that corresponds to 75-85% of the prior information. The choice of numerical range as a threshold has reduced the false positives to a great extent and has led to more accurate results.

The last step in the pipeline is naive Bayes classification, as shown in Figure 5.2, which is the implementation of a naive Bayes classifier to reduce the false positive prediction from the second step and predict the new edges. The prior knowledge that satisfies the threshold criteria is considered as a positive sample for the training data set. However, the negative sample is the subset of edges that do not exist in the prior knowledge within the threshold range. A naive Bayes classifier classifies the output of step 2 of the Beacon inference engine into the present edges and the absent edges after training. Present edges are the final predicted edges, and absent edges are the ignored ones. The naive Bayes classifier has helped greatly in reducing the number of false positive edges.

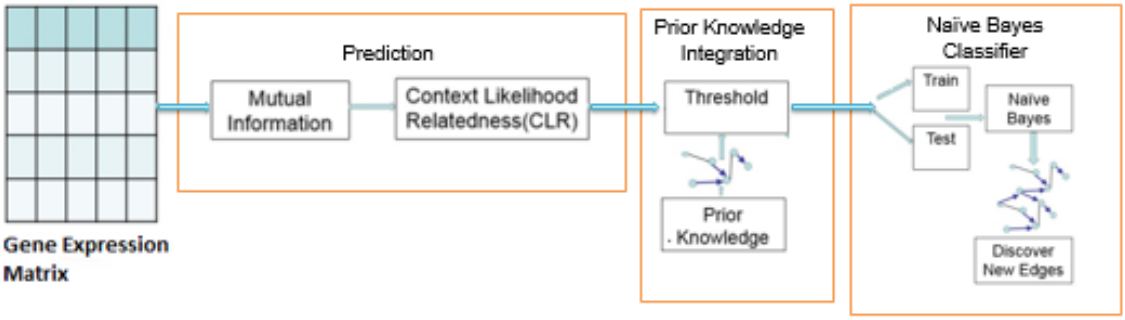

Figure 5.2: Beacon Inference Engine Detailed Pipeline

5.1 Mutual Information

Mutual information is a measure of inter-dependence between random variables. The implementation of mutual information for gene expression analysis requires a binning approach that will be discussed later in the thesis [56]. Mutual information uses entropy. The entropy tells how evenly the states of a random variable are distributed. Entropy $H(X)$ for a random variable X with m finite states $\{x_1,x_2,...,x_m\}$ is given as

$$H(X) = -\sum_{i=1}^{m} p(x_i)\log(p(x_i)), \tag{5.1}$$

where $p(x_i)$ is the probability of state x_i.

The *joint entropy* of two random variables X and Y with m and n finite states is given as

$$H(Y,X) = -\sum_{j=1}^{m}\sum_{i=1}^{n} p(y_i,x_j)\log(p(y_i,x_j)), \tag{5.2}$$

where m denotes the finite possible states of Y and X random variable and $p(y_i,x_j)$ is the joint probability of y_i and x_j [56].

The *conditional entropy* is the uncertainty of one variable with respect to the given variable. The *conditional entropy* $H(Y|X=x_j)$ of random variable Y given $X=x_j$ is

$$H(Y|X=x_j) = -\sum_{i=1}^{m} p(y_i|x_j)\log(p(y_i|x_j)), \tag{5.3}$$

where $m=n$ denotes the finite states of Y and X random variable and $p(y_i|x_j)$ is the conditional probability of the y_i given x_j [25].

The mutual information is defined as

$$I(Y;X) = H(X) + H(Y) - H(Y,X) \tag{5.4}$$

In this thesis, the directed mutual information measure is implemented between the expression time-series of the two genes [25]. The *directed mutual information* between two genes X and Y having time series (x_1,x_2,...,x_n), (y_1,y_2,...,y_n) is

$$I_d(Y;X) = -\sum_{i=2}^{n} I((Y_i;X_i)|X_{i-1}) \tag{5.5}$$

where X_{i-1} is (0,x_1,x_2,...,x_{i-1}) and Y_i is (y_1,y_2,...,y_i).

The directed mutual information accounts every time point by taking the sum of all of them (Y_2, X_1),(Y_3, X_2)...(Y_n, X_{n-1}). The above equations can be simplified using conditional entropy, mutual information

$$I_d(Y;X) = -\sum_{i=2}^{n} I(Y_i;X_i) - I(Y_i;0X_{i-1}) \tag{5.6}$$

where $0X_{i-1}$ (0,x_1,x_2,...,x_{i-1}) and Y_i is (y_1,y_2,...,y_i).

The above equation denotes that we have shifted the time series for X by one step. The mutual information is directed as the Y time series has been shifted by one time point into the future compared to X. A high value of mutual information indicates that the genes are dependent on each other while the independent genes have mutual information close to 0. An example to calculate directed mutual information between two genes is shown in Figure 5.3.

For a random variable X with m finite states, the simple binning method is to divide the data into k discrete bins. The indicator function $\theta_i(x_j)$ counts the number of measurements that lie in each bin a_i where $i = 1, 2, ..., k$ [47]. The indicator function θ_i can be 0 or 1 depending upon whether the value lies in the bin or not. Mathematically

$$\theta_i(x_j) = \begin{cases} 1 & : \quad x_j \in a_i \\ 0 & : \quad otherwise \end{cases}$$

.

The probability of each bin is the number of states of random variable X in the bin divided by the total number of finite states of X. [56].

$$p(a_i) = \frac{1}{m}\sum_{j=1}^{m} \theta_i(x_j) \tag{5.7}$$

Figure 5.3: Example to calculate directed mutual information between two genes X and Y

Directed mutual information example:

Suppose two genes X and Y have time series (0.6,0.2,0.4), (0.3,0.4,0.5). Calculate the directed mutual information

Using Equation 5.7, directed mutual information for X,Y is

$I_d(Y;X) = (I(Y_2;X_2) - I(Y_2;0X_1)) + (I(Y_3;X_3) - I(Y_3;0X_2))$

where Y_2 = (0.3,0.4) , X_2 = (0.6,0.2), $0X_1$ = (0,0.6)

and Y_3 = (0.3,0.4,0.5) , X_2 = (0.6,0.2,0.4), $0X_2$ = (0,0.6,0.2)

$I(Y_2;X_2)$, $I(Y_2;0X_1)$, $I(Y_3;X_3)$, $I(Y_3;0X_2)$ is calculated using Equation 5.4.

The *joint probability* of two random variables X and Y with m finite states in bins a_i, b_j is defined as

$$p(a_i, b_j) = \frac{1}{m}\sum_{j=1}^{m}(\theta_i(x_j)\theta_k(y_j)). \tag{5.8}$$

After calculating the above probabilities, the mutual information for a random variable can be computed. The standard binning method assumes that every value can lie in one bin only. However the values at the margins of bins can lie in two bins with small fluctuations. This introduces Gaussian noise in the mutual information. Therefore, in addition to mutual information, we use B-spline interpolation to compensate for the noise [56].

5.2 B-spline

A B-spline is a piecewise polynomial representation that approximates the data points, also known as control points. The algorithm used for the approximation should not be overly sensitive to the data points. The piecewise polynomial representation avoids the problem of over fitting and numeric instability [12]. We use a cubic spline to define curve segments over the control points. The cubic spline ensures continuity at the points where segments join and matches their curvature [39].

Suppose we have a random variable X with m values $x_1, x_2, ..., x_m$. The m values are distributed into the j bins. The parameters $(r_0, r_1, r_2, ..., r_j)$ represent knot vectors in each bin [47]. A knot vector represents the weight of each point in each bin.

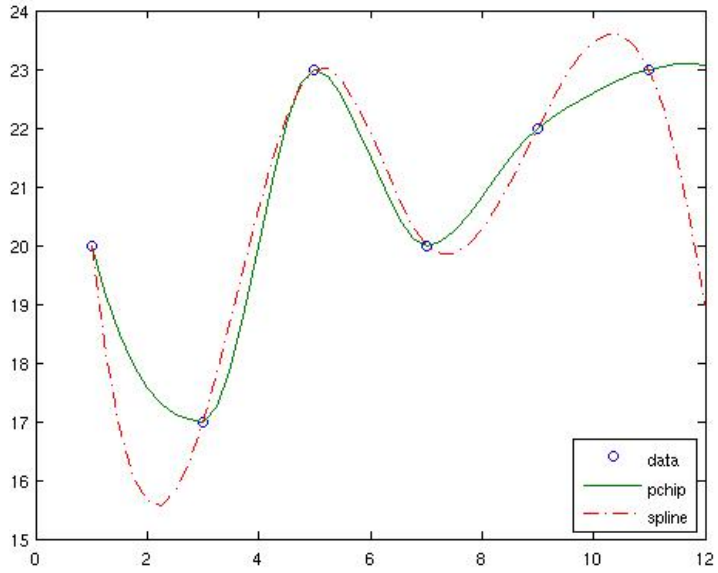

Figure 5.4: Interpolation of cubic B-spline and spline at data points. pchip is the cubic B-spline at knots (r=1,3,5,7,9,11).

In general a knot vector r_i with j bins and the spline order k is defined as [47]

$$r_i = \begin{cases} 0 & : \quad i < k \\ i - k + 1 & : \quad k \leq i \leq j - 1 \\ j - k + 1 & : \quad i > j - 1 \end{cases} \tag{5.9}$$

The m values of a random variable X are normalized to fit in the numerical range, given by $j-k+1$ on the number line. The normalization is carried out in two steps. First is to find the minimum (x_{min}) and the maximum value (x_{max}) among m values of X [44]. Then, using the equation below, calculate the normalized values z_i $(i = 1, 2..m)$

$$z_i = \frac{(x_i - x_{min})(j - k + 1)}{x_{max} - x_{min}}. \tag{5.10}$$

After computing the normalized values for a random variable X the B spline function will be used to calculate the weighting coefficients in the j bins. The B-spline function can be defined using the

- Spline order ($k = d + 1$), d is degree of piecewise polynomial.
- knots vector($r_i \leq r < r_{i+1}$), i(1..j)
- Control points (x_1,x_2,...,x_n).

The B-spline basis function is given by the Cox-de Borr [44] recursion formula

$$B_{i,1}(z) = \begin{cases} 1 & : \quad r_i \leq z \ \leq r_{i+1} \\ 0 & : \quad otherwise \end{cases}$$

$$B_{i,k}(z) = \frac{z - r_i}{r_{i+k+1} - r_i} B_{i,k-1}(z) + \frac{r_{i+k} - z}{r_{i+k} - r_{i+1}} B_{i+1,k-1}(z). \tag{5.11}$$

In general, a B-spline is the function that approximates the data points with the help of a weight matrix. In this research, the B-spline is implemented to smooth the time series gene expression data. The gene expression time series for gene g_l is u_{lj} where j is $(1, 2, ..., n)$ at time points $(t_1, t_2, ..., t_n)$ is divided into n number of bins. The gene expression are normalized z_i $(i = 1, 2..n)$ in range $(n - k + 1)$ where k is the spline order using Equation 5.10. The border value of bins are defined using knot vector r_i using equation 5.9. The B-spline weight matrix $W = (w_{ij})_{i \in n, j \in n}$ represents weighing coefficient of value z_i in bin a_n that is $w_{ij} = B_{i,k}(z_i)$ is calculated using equation 5.11. The implementation of the algorithm is explained later in the thesis.

Figure 5.5: Example to calculate B-spline weight coefficient matrix between for a gene X

B-spline weight coefficient matrix example:

Suppose we have a X having time expression at 3 time points (0.2,0.4,0.6). Calculate the B-spline weight matrix $W = (w_{ij})_{i\leq 3, j\leq 3}$, given spline order $k = 2$ and number of bins $r = 3$, $1 \leq j \leq r$

Solution:

- Firstly the normalized value is calculated at each time point i.e., $z_i (i = 1, 2, 3)$
  Using Equation 5.11 $z_i = \frac{(x_i - x_{min})X(r-k+1)}{x_{max} - x_{min}}$, $x_{max} = 0.6$, $x_{min} = 0.2$
  gives $z_1 = 0$, $z_2 = 1$,$z_3 = 2$

- For value $z_1 = 0$ weight coefficient w_{1j} are calculated for each bin following below steps
  Knot vector is determined using Equation 5.10. Knot vector defines the interval range for values of X. Weight coefficient w_{1j} defines the weight of point $z_1 = 0$ on a B-spline curve in the interval j defined by knot vector
  For k=1 ,knot vector is, $r_i = (1, 2, 3)$
  For k=2 ,knot vector is, $r_i = (0, 1, 2)$

- Using equation 5.11 The weight coefficient matrix for each $w_{ij} = B_{j,k}(z_i)$
  For point $z_1 = 0$ weight coefficients are calculated in 3 bins i.e. $1 \leq j \leq r$
  w_{11} =0, w_{12} =0, w_{13} =0 Similarly calculation of $z_2 = 1$ and $z_3 = 2$ we get weight coefficient matrix as:

$$W = \begin{pmatrix} 0.06 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \tag{5.12}$$

- The probability of each bin is calculated $1 \leq j \leq r$
  $p(a_j) = \frac{1}{n} \sum_{i=1}^{n} B_{j,k}(z_i)$;
  where z_i is the normalized value for each value of X. For $z_1 = 0$, $z_2 = 1$,$z_3 = 2$ the probability for each bin is $p(a_1) = 0.02$, $p(a_2) = 0.32$ and $p(a_3) = 0.66$

- The entropy for variable X is $H(X) = -\sum_{j=1}^{r} p(a_j) \log(p(a_j))$.

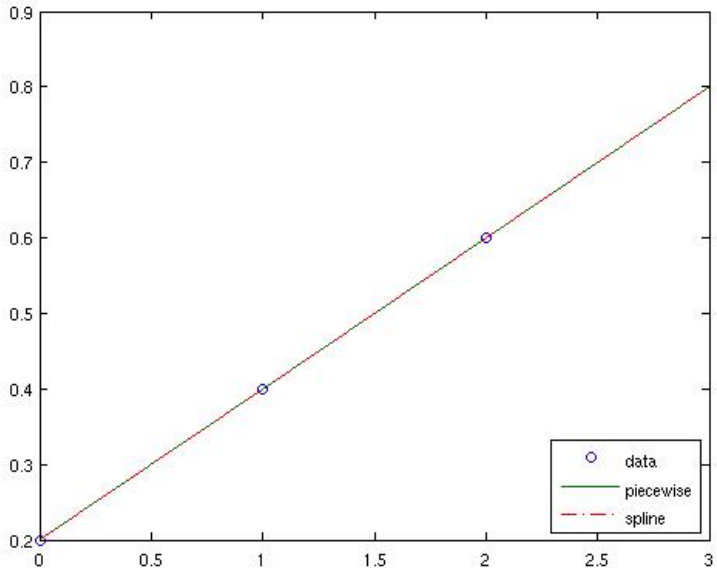

Figure 5.6: Cubic B-spline(r=0,1,2)

5.3 Context Likelihood Relatedness

Context likelihood relatedness (CLR) is used to compute the significance of the mutual information values between gene pairs. The CLR algorithm is based on a null distribution approach [31]. The null distribution approach assumes that there is no interaction between the gene pairs. It extends the relevance network approach [3]. The relevance network algorithm approach is based on threshold clustering algorithms [10]. The context likelihood relatedness values above the threshold are considered to cluster, and the values below the threshold are ignored. The values in the cluster are considered to be true positives and the other values represent false positives. In general, it is a trade-off between false positive and true positive prediction. However, CLR has an automatic correction approach. This automatic approach determines the relative position of the gene pair g_l, g_s with respect to the gene g_l and gene g_s. Hence, two z scores are computed. The row z score determines the relative position of the gene with respect to the row values and the column z score determines the position with respect to column values. A final Z score is calculated by taking the root mean of the two scores.

Suppose M represents the mutual information matrix with each value M_{ij} representing the directed mutual information score between each pair. Let S represent the CLR scoring matrix with each value $S(g_i, g_j)$.

A_i^2 represent the z score of the gene pair(g_i, g_j) with respect to gene g_l

$$A_i^2 = \max\left(0, \frac{1}{\sigma_i} - \frac{\bar{M_i}}{M_{ij}\sigma_i}\right), \tag{5.13}$$

where σ_i is the standard deviation and mean $\bar{M_i}$ of DTI in row i of M that is M_i The standard deviation σ_i in row M_i is

$$\sigma^2 = E[(M_i - \bar{M_i})^2] \tag{5.14}$$

and the mean $\bar{M_i}$ is defined as

$$\bar{M_i} = \frac{\sum_{j=1}^{q} M_{ij}}{q} \tag{5.15}$$

where q is the number of genes.

A_j^2 represent the z score of the gene pair(g_i, g_j) with respect to gene g_s

$$A_j^2 = \max\left(0, \frac{1}{\sigma_j} - \frac{\bar{M_j}}{M_{ij}\sigma_j}\right), \tag{5.16}$$

where, σ_j represent the std deviation and mean $\bar{M_j}$ of DTI in column j of M that is M_j

The standard deviation σ_i in column M_j is

$$\sigma^2 = E[(M_j - \bar{M_j})^2] \tag{5.17}$$

and the mean $\bar{M_j}$ is defined as

$$\bar{M_j} = \frac{\sum_{i=1}^{q} M_{ij}}{q} \tag{5.18}$$

where q is the number of genes.

$S(g_i, g_j)$ represent the CLR score between gene g_i and gene g_j and can be computed as follows

$$s(g_i, g_j) = \sqrt{(A_i^2) + (A_j^2)} \tag{5.19}$$

The CLR scoring matrix is:

$$S = \begin{pmatrix} S(g_1; g_1) & S(g_1; g_2) & . & . & . & S(g_1, g_q) \\ S(g_2; g_1) & S(g_2; g_2) & . & . & . & S(g_2, g_q) \\ . & . & . & . & . & . \\ S(g_n; g_1) & S(g_n; g_2) & . & . & . & S(g_n, g_q) \end{pmatrix}. \tag{5.20}$$

5.4 Naive Bayes Classifier

For classification of gene pairs, we chose the naive Bayes classifier because of its ease of use, simple design, and good classification performance [24]. The naive Bayes classifier is based on the Bayesian inference method. It focuses on the probability that a gene pair belongs to a particular category C $= \{c_1, c_2, ..., c_K\}$. Here we use two categories (present edges and absent edges). Each gene pair is considered as an event and will be labeled as a present edge or an absent edge. We calculate the posterior probability that the gene pair belongs to either of the categories and chose the category with the highest probability.

The posterior probability of that a gene pair g i.e. $p(g|c_g)$ belongs to a category c_g can be calculated in formula:

$$p(c_g \mid g) = \frac{p(c_g)p(g|c_g)}{\sum_{j=1}^{k} p(c_j)p(g)} \tag{5.21}$$

In order to evaluate the above, we need to first calculate the probability of a category c_g, i.e., $p(c_g)$ and the probability of the gene pair g, i.e. $p(g)$. $p(c_g)$ is given as the ratio of the number of gene pair that fall into the category c_g and the total number of gene pairs.

$$p(c_g) = \frac{n(c_g)}{\sum_{g=1}^{n} n\,(c_g)} \tag{5.22}$$

Here, $n(c_g)$ is the total number of gene pair which belong to category c_g as per the prior knowledge. $p(g)$ can be evaluated based on all the gene pair.

$p(g)$ is calculated the using Gaussian distribution. The mean and variance of g is calculated in each class, using training set. Let μ_{c_g} be the mean of the CLR scores corresponding to gene pairs g associated with class c, and $\sigma^2_{c_g}$ is the variance. The probability distribution of a gene pair score g_i given a class c_g, p($g|c_g$) is,

$$p(g|c_g) = \frac{1}{\sqrt{2\pi\sigma^2_{c_g}}}, e^{-\frac{(g_i-\mu_{c_g})^2}{2\sigma^2_{c_g}}}. \tag{5.23}$$

5.5 Implementation

The above methodology is implemented on time series gene data set to predict new edges in between genes. The time series gene data set consist of q genes $Y = \{g_1, g_2, ..., g_q\}$. For a gene g_l there are n measurements at n time points $T = \{t_1, t_2, ..., t_n\}$. The first step in the process is to predict all the possible edges between the genes. The prediction of all edges is a complex process as shown in

Figure 5.7. The time series gene data set is processed to calculate entropy for every gene by dividing the time series data of each gene into j intervals, i.e., j bins. The time series expression of each gene g_l where $l \in \{1, 2, ..., q\}$ consist of n finite states is represented as u_{lj} where $j \in \{1, 2, ..., n\}$. The normalized value z_i, $(i = 1, 2..n)$ for each gene g_l is calculated for a cubic spline, $degree = 3$, as shown in step 3 of Figure 5.7. The weight coefficient matrix is calculated for every gene at each normalized value z_i is $W = (w_{ij})_{i \in n, j \in n} = B_{j,k}(z_i)$. The weighing coefficient calculates the probability of a normalized value z_i in bin j where j is $(1, 2, ..., n)$. The probability of each bin is calculated by using the Equation 5.8 as shown in Figure 5.7. The entropy of gene is calculated over j bins using Equation 5.1.

After the entropy calculation of genes, the joint probability is calculated between the gene pairs. The time series of a gene is shifted into the future with respect to the other gene to infer the direct edges for every gene pair g_l, g_s where $l \in \{1, 2.., q\}$, $s \in \{2..q\}$. The time series of g_s is shifted one step into the future with respect to time series of g_l. Hence the time series of g_s is u_{si} where $i \in \{0, 1.., (n-1)\}$ and for gene g_l is u_{li} where $l \in \{1, .., n\}$. We calculate the joint probability gene g_l with respect to gene g_s by calculating the probability for each bin of g_l with respect to each bin of G_s as shown in Step 2 of Figure 5.8. Once the probability is calculated for each bin the joint entropy is calculated for gene pair g_l, g_s using Equation 5.9.

Using the joint entropy for gene pair (g_l, g_s) mutual information is calculated using the algorithm shown in Figure 5.9. The mutual information is calculated using Equation 5.6 and 5.7. The equation adds the entropy of the gene independently and then subtracts the conditional entropy between them. The mutual information between gene pairs is represented using the mutual information matrix $M = (M_{ij})$ where $i \in \{1, 2.., q\}$, $j \in \{1, 2.., q\}$. The mutual information between a gene pair measures the dependency between them. A higher directed mutual information between two genes means that one gene is associated with the other, and it is more likely that they have a biological relationship. The function to calculate the directed mutual information, joint probability, and entropy is implemented in C++. A part of the code is taken from 6.10.

The context likelihood relatedness algorithm is used to assign a significant meaning to the mutual information values between gene pairs. It calculates the cumulative z score between the row z scores and column z scores for a gene pair g_l, g_s as shown in Equation 5.28. The row z score of gene pair g_l, g_s is calculated with respect to gene g_l while the column z score of gene pair g_l, g_s is with respect to gene g_s as shown in Equation 5.26 and 5.27. The cumulative z score represents the

Figure 5.7: Algorithm to calculate entropy for a gene [47]

INPUTS: Gene expression data for a gene g_l at n time points is $U_l = (u_{l1}, u_{l2}, \ldots, u_{ln})$, number of bins j, spline order k

OUTPUT: Entropy of a gene $H(U_l)$

ALGORITHM: Gene Entropy Calculation

- For genes g_l
- For each time expression u_{li} $1 \leq i \leq n$

  Do
- Calculate the normalized variable z_i

  $z_i = \frac{((u_{li})-(u_{li})_{min})(j-k+1)}{((u_{li})_{max}-(u_{li})_{min})}$

  where $(u_{li})_{max}$= maximum expression value of gene g_l among n time points and $(u_{li})_{min}$= minimum expression value of gene g_l among n time points
- For each bin r, $1 \leq r \leq j$

  Do
- Determine the weight coefficient matrix using Equation 5.11 for a gene at each normalized value (z_i) $W = (w_{ir})_{i \in n, r \in j} = B_{r,k}(z_i)$

  end

  end
- For each bin r, $1 \leq r \leq j$

  Do
- Calculate the probabilities for each bin p(a_r),r=(1..j)
- For each r $p(a_r) = \frac{1}{n} \sum_{i=1}^{n} B_{r,k}(z_i)$

  end
- Determine entropy using Equation $H(U_l) = -\sum_{r=1}^{j} p(a_r) \log(p(a_r))$.

Figure 5.8: Joint Entropy Calculation between two genes [47]

INPUTS: Gene expression data for a gene g_l at n time points is is $U_l = (u_{l1}, u_{l2}, \ldots, u_{ln})$, number of bins j, Gene expression data for a gene g_s at n time points is is $U_s = (u_{s1}, u_{s2}, \ldots, u_{sn})$

OUTPUT: Determine joint entropy $H(U_l, U_s)$

ALGORITHM: Calculate Joint Entropy for Gene pair (g_l, g_s)

- For genes g_l where l =$(1, 2.., q)$,the expression value at n time points is $U_l = (u_{l1}, u_{l2}, \ldots, u_{ln})$.
- For each gene G_s where $s = (2..q)$,the expression value at n time points is $U_s = (u_{s1}, u_{s2}, \ldots, u_{s(n-1)})$.
- Calculate the entropy $H(U_l), H(U_s)$.
- For each bin e, $1 \leq e \leq j$ **do**
- For each bin d, $1 \leq d \leq j$ **do**
- Calculate the joint probability p$(p(a_e, b_d))$ for all j bins
  $p(a_e, b_d) = \frac{1}{n} \sum_{i=1}^{n} \sum_{i=1}^{n} B_{i,e}(z_i) B_{i,d}(z_i)$.
  end
  end
- Determine joint entropy
  $H(U_l, U_s) = -\sum_{d=1}^{j} \sum_{e=1}^{j} p(a_e, b_d) \log(p(a_e, b_d))$.

Figure 5.9: Mutual Information matrix between the genes (g_l, g_s) [47]

INPUTS: Joint Entropy $H(U_l, U_s)$, Entropy $H(U_l),\ H(U_s)$

OUTPUT: Mutual Information matrix$M = (M_{ij})$ where $i \in Y,\ j \in Y$

ALGORITHM: Calculation Mutual Information

Begin For genes g_l where l =$(1, 2.., q)$,the expression value at n time points is $U_l = (u_{l1}, u_{l2}, \ldots, u_{ln})$

- For each gene $g_s = u_{si}$ where $s = (2..q)$, $U_s = (u_{s1}, u_{s2}, \ldots, u_{s(i-1)})$
- Compute the mutual information U_{li} is a random variable representing gene expression of gene g_l upto i time points. U_{si} is a random variable representing gene expression of gene g_s upto s time points.

  $I(U_l, U_s) = H(U_l) + H(U_s) + H(U_l, U_s)$

  $I(U_l, U_s) = -\sum_{i=2}^{n} I(U_{li}, U_{si}) - I(U_{li}; 0U_{s(i-1)})$

End

Figure 5.10: Scoring Matrix for all the gene pairs as per gene expression matrix [19]

INPUTS: Mutual Information matrix$M = (M_{ij})$ where $i \in Y$, $j \in Y$

OUTPUT: Scoring Matrix $S = (s_{ij})$ where $i \in Y$, $j \in Y$

ALGORITHM: Calculate Scoring Matrix

Begin For $M = (M_{ij})$ where $Y = (g_1, g_2.., g_q)$

$$A_i^2 = \max\left(0, \frac{1}{\sigma_i} - \frac{M_i}{M_{ij}\sigma_i}\right),\ A_j^2 = \max\left(0, \frac{1}{\sigma_j} - \frac{M_j}{M_{ij}\sigma_j}\right),\ s_{ij} = \sqrt{(A_i^2) + (A_j^2)}$$

End $S = \begin{pmatrix} s(g_1;g_1) & s(g_1;g_2) & . & . & . & s(g_1,g_q) \\ s(g_2;g_1) & s(g_2;g_2) & . & . & . & s(g_2,g_q) \\ . & . & . & . & . & . \\ s(g_q;g_1) & s(g_q;g_2) & . & . & . & s(g_q,g_q) \end{pmatrix}$.

context likelihood relatedness scores of a gene pair g_l, g_s. The context likelihood relatedness scores are represented using scoring matrix $S = (s_{ij})$ where $i \in Y$, $j \in Y$. The scoring matrix represents the likeliness of all the possible gene pair between the genes in the time series data set and each gene pair represents an edge. Therefore the scoring matrix represents the value of all possible edges between the genes. The CLR scoring scheme is implemented in C++ using boost libraries.

After scoring all the possible edges using the above algorithms, the Beacon inference engine integrates the prior knowledge to eliminate the false positive edges. It plots a histogram of the scores corresponding to the prior known edges represented as a curated network and chooses a numerical range which contains the maximum number of edges. The edges outside these range are eliminated and are not considered for further evaluation. It is observed that a more rigorous approach is required to reduce the number of false positive edges in the predicted edges. A regression classifier, i.e., a naive Bayes classifier, is used to predict the true positive edges. The positive sample is the prior known edges and the negative sample is the sample of edges that do not exist in the prior knowledge. The naive Bayes classifier classifies the edges into present and absent. The present are the true positive edges, and the absent are the edges that do not exist. The Bayes classifier calculates the posterior probability of a gene pair belonging to a category present or absent. The edges which

are predicted as present edges are considered as predicted edges and plotted using Cytoscape in the results section. The implementation of the naive Bayes classifier is done in Python using the sklearn package. The summarized algorithm used in Beacon inference engine is shown in Figure 5.5.

Figure 5.11: Algorithm used in Beacon Inference Tool to predict edges in the curated network i.e. prior knowledge

INPUTS:Time series gene data set consist of q genes $Y= g_1, g_2, ..., g_q$,*expression value* for gene g_l at time t_j is u_{ij}, number of bins j, spline order k, curated graph $H = (V, E)$ where cordinality of $V \leq q$
OUTPUT: Predicted edges $E \in (u, v)$ in the curated graph $H = (V, E)$ where $u \in Y$ and $v \in Y$

- For genes g_l where $l =(1, 2.., q)$,the expression value at n time points is $U_l = (u_{l1}, u_{l2}, \ldots, u_{ln})$ **Do**
- For each gene g_s where $s = (2..q)$,the expression value at n time points is $U_s = (u_{s1}, u_{s2}, \ldots, u_{sn})$**Do**
- Determine entropy $H(U_l), H(U_s)$ using algorithm 5.7
- Determine joint entropy $H(U_l, U_s)$ using algorithm 5.8
- Compute the directed mutual information $I(U_l, U_s)$ using algorithm 5.9
  end
  end
- For gene g_l where $l =(1, 2.., q)$, $U_l = (u_{l1}, u_{l2}, \ldots, u_{ln}$ **Do**
- For each gene g_s where $s = (2..q)$, $U_s = (u_{s1}, u_{s2}, \ldots, u_{s(n-1)}$**Do**
- Compute the score s_{ij} for gene pair (g_l, g_s) giving scoring matrix s_{ij}
  end
  end
- Scoring matrix containing score for every predicted edge.
- Integrate prior known edges that is curated network $H = (V, E)$ to set the threshold $\alpha 1 \leq \ edge \leq \alpha 2$
- Naive Bayes Classifier predicts edges $E \in (u, v)$ in the curated graph $H = (V, E)$ where $u \in Y$ and $v \in Y$

Chapter 6

RESULTS

Signal transduction plays a role in different developmental stages throughout the life of cells and organisms. A gene can be expressed in different stages of the seed development process. The analysis of different signal transduction pathways can help in understanding how the action of a given gene influences a given signal transduction pathway. In this thesis, the Beacon inference engine is developed to understand seed development by inferring new edges in prior known seed development networks using time-series gene expression data. The Beacon inference engine uses three prior known different seed development networks. The two seed development networks posited by Dr. Ruth Finkelstein [20] consist of the similar genes. In this thesis , these networks are named as Seed Development Network1 Figure 6.1 as Seed Development Network2 Figure 6.2. The third network posited by Dr. Sreenivasulu Nese, Dr. Wobus Ulrich is named as Seed Development Network3 Figure 6.3 and excludes 20 genes which are present in Seed Development Network1 and Seed Development Network2. Hence, Seed Development Network3 is used to explore new edges with respect to the genes that are exclusive to Seed Development Network1. The time-series gene expression data use to infer new edges is taken from the data sets described in Chapter 4.

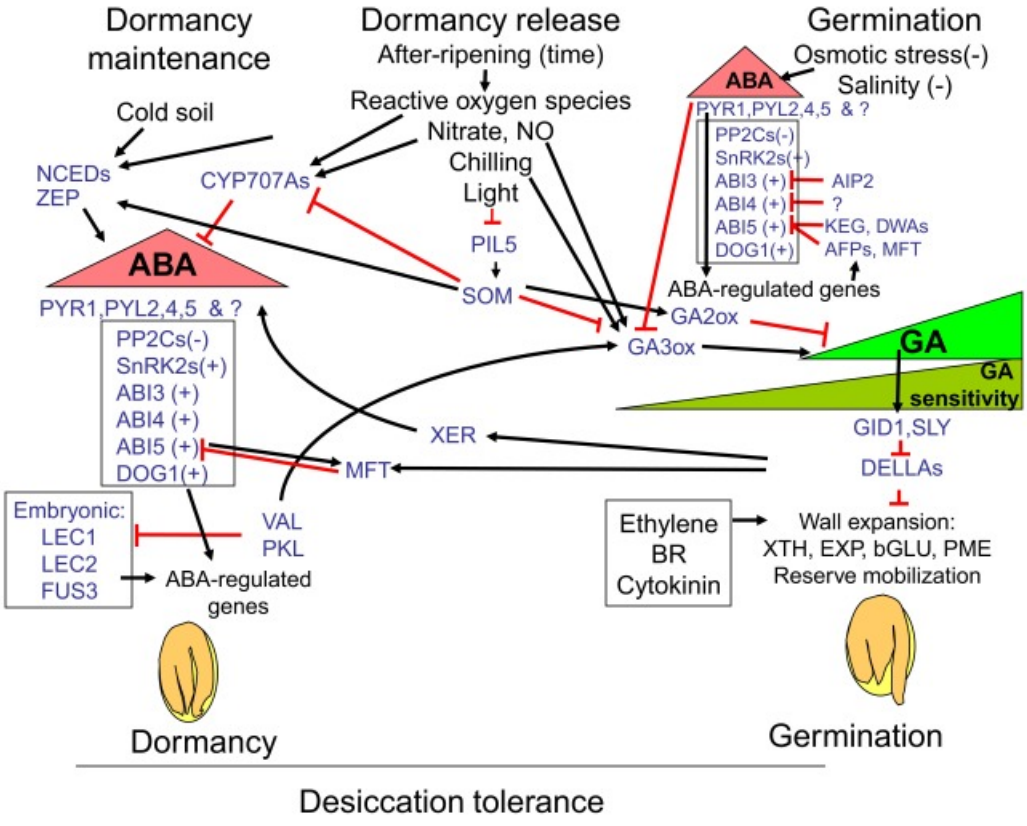

Figure 6.1: Prior known Seed Development Network1 [20].

Figure taken from Ruth Finkelstein, *Abscisic acid synthesis and response*, The Arabiposis Book,(2013), pe0166

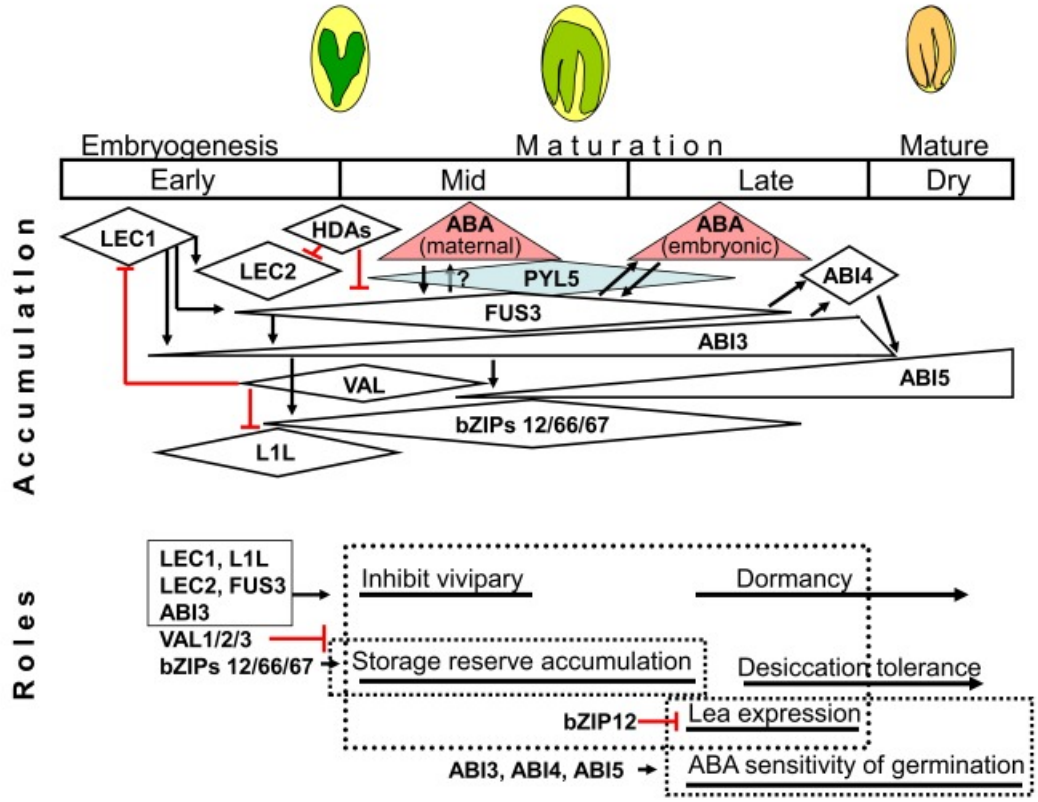

Figure 6.2: Prior known Seed Development Network2 [20].

Figure taken from Ruth Finkelstein, *Abscisic acid synthesis and response*, The Arabiposis Book,(2013), pe0166

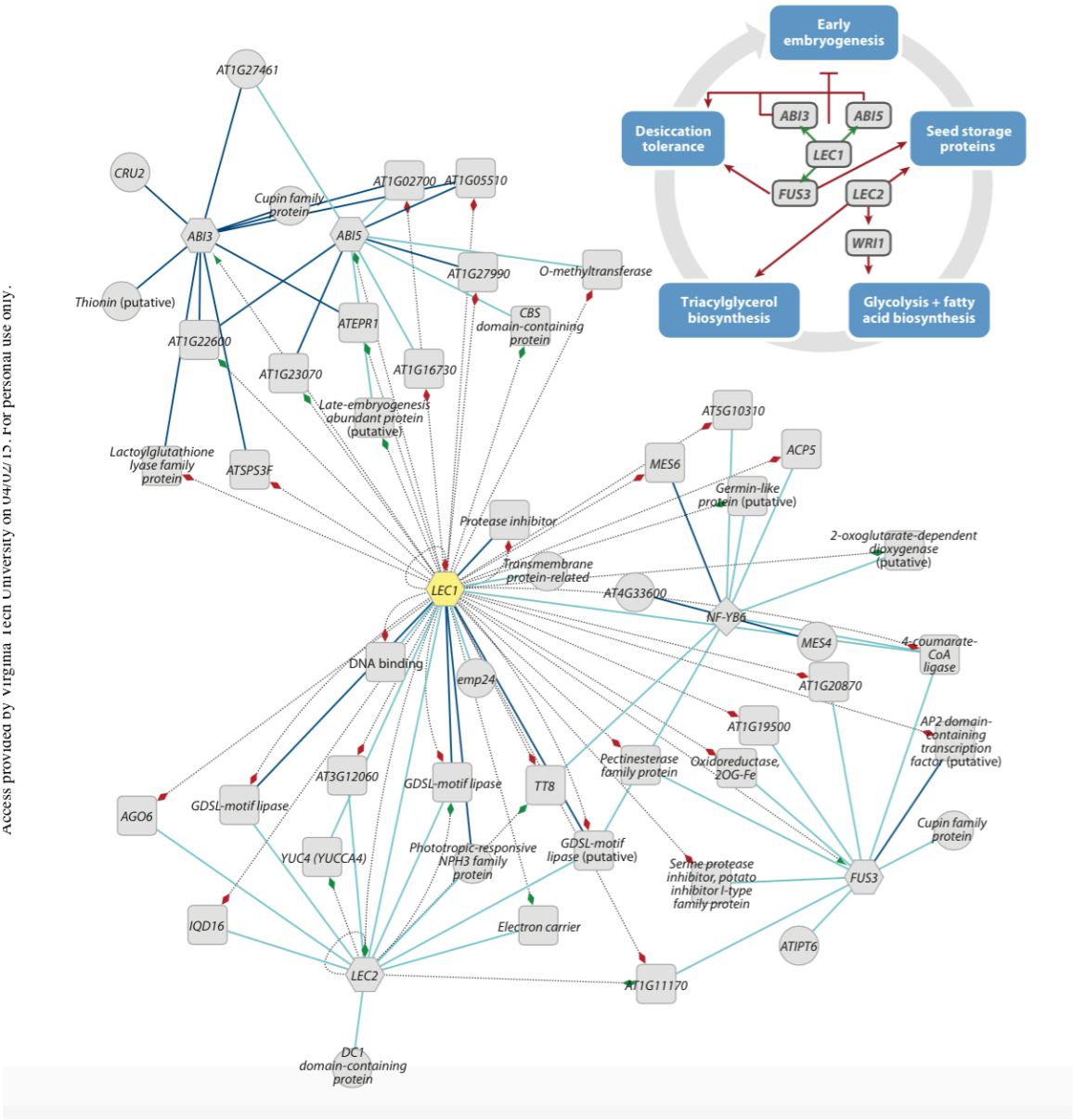

Figure 6.3: Prior known Seed Development Network3 [52].

Figure taken from Sreenivasulu Nese and Wobus Ulrich, *Seed-development programs: A systems Biology-based comparison between dicots and monocots*, Annual Review of Plant Biology,64(2013), pp. 189-217

6.1 Beacon inference engine validation to infer new edges in Seed Development Network1

In this thesis, we have explored new edges in Seed Development Network1 by using the Beacon inference engine; see Figure 6.4. A part of Seed Development Network1 is considered as the prior knowledge and is termed curated graph [20]; see Figure 6.10. The time series data corresponding to the 33 genes in Seed Development Network1, is taken from the Data Set1;refer Chapter4.1. The Beacon inference engine uses this 33-gene time series data set of plant *Arabidopsis thaliana* to predict new edges in Seed Development Network1. The prediction step of the Beacon inference engine predicts all the possible edges between the 33 genes from Seed Development Network1, present in time series data set and scores them using mutual information and the CLR algorithm. A total of 1081 edges were predicted using the *Arabidopsis thaliana* time series data set; see Figure6.5.

The prior integration knowledge step in the beacon inference engine uses the prior knowledge to limit the number of false positives in the predicted edges. The prior knowledge consist of 7 known edges in the Seed Development Network1; see Figure 6.1.

The prior knowledge integration step of the Beacon inference engine uses the prior knowledge to define the bounds of the scores. The CLR scores corresponding to the prior knowledge in Figure 6.7, defines the upper bound and the lower bound score for the predicted edges that is 0.55-0.8; see Figure 6.7. The predicted edges outside these bounds are ignored which reduces the number of predicted edges to 191. The third step i.e. naive Bayes classification plays a major role in reducing the false positive edges. The naive Bayes classifier considers the prior known edges as its positive sample. The negative sample consist of the edges provided by user, that do not exist in prior knowledge. The naive Bayes classifier is trained over these set of samples and is used to predict the positive edges in 191 predicted edges. The new edges are the one that are not present in the Seed Development Network1 and are verified through Seed Development Network2 [20]. The Beacon inference engine predicts a total of 47 edges; see Fig 6.8 and Table 6.2. The 40 edges are the connections present in the Seed Development Network1 that were not considered in the curated graph [20]. The remaining 7 edges are considered as new edges and are highlighted in red in Figure 6.8; see Table 6.1.

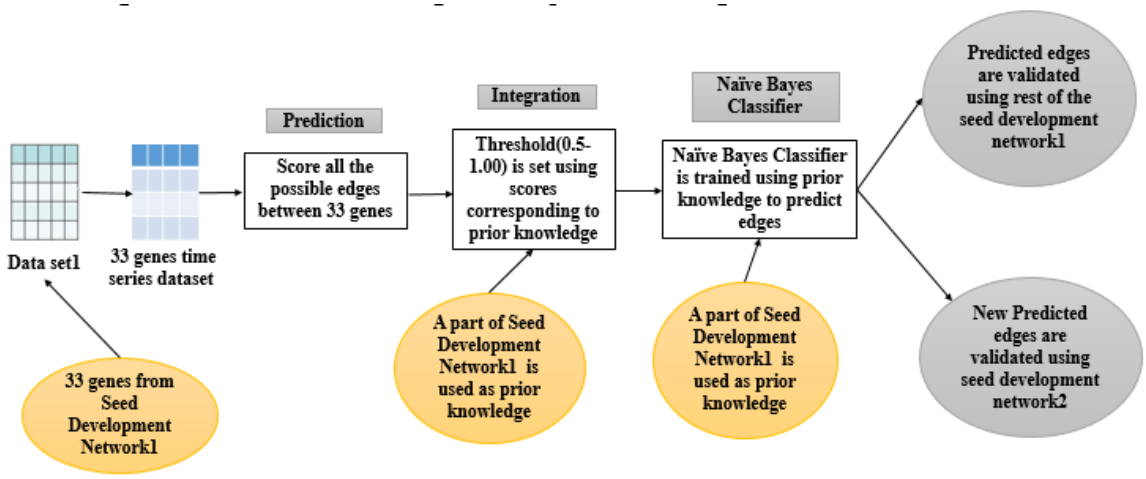

Figure 6.4: The Beacon Inference Engine Pipeline used to predict edges between 33 genes in Seed Development Network1

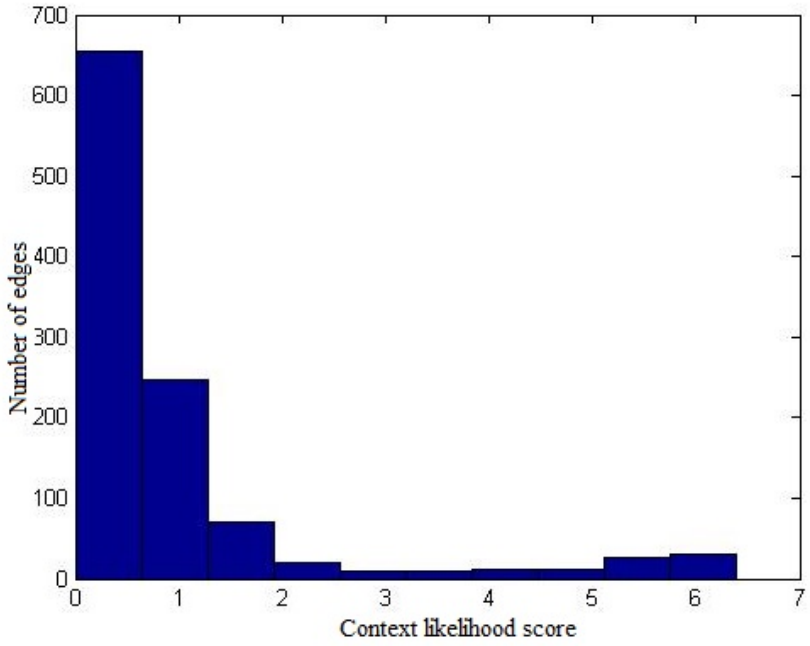

Figure 6.5: Histogram plot of CLR Score of all the possible edges between 33 genes in the time series data set.

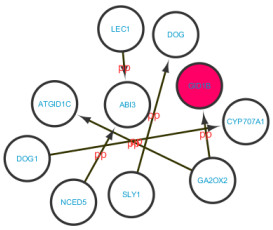

Figure 6.6: Sample of edges in Seed Development Network1 that are used as prior knowledge for the prediction of new edges in Seed Development Network1

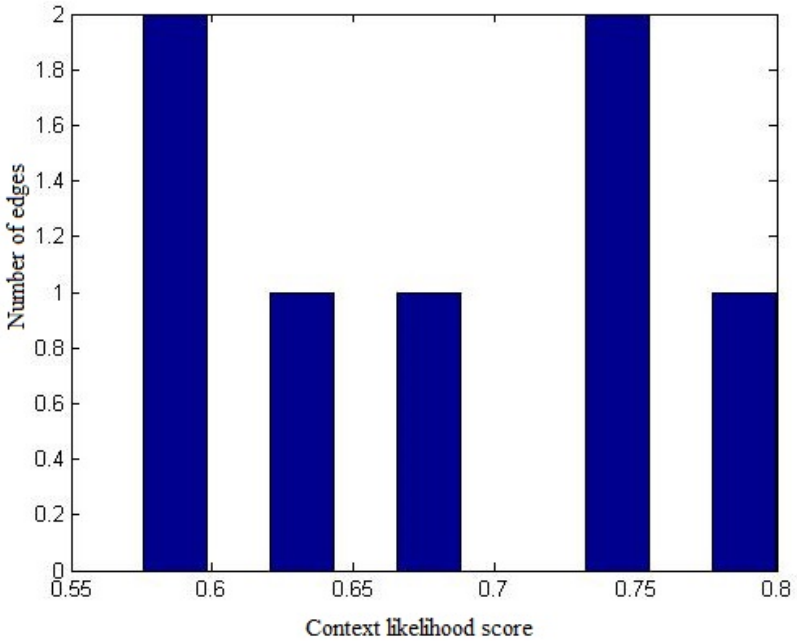

Figure 6.7: Histogram plot of CLR Score of prior knowledge in Seed Development Network1

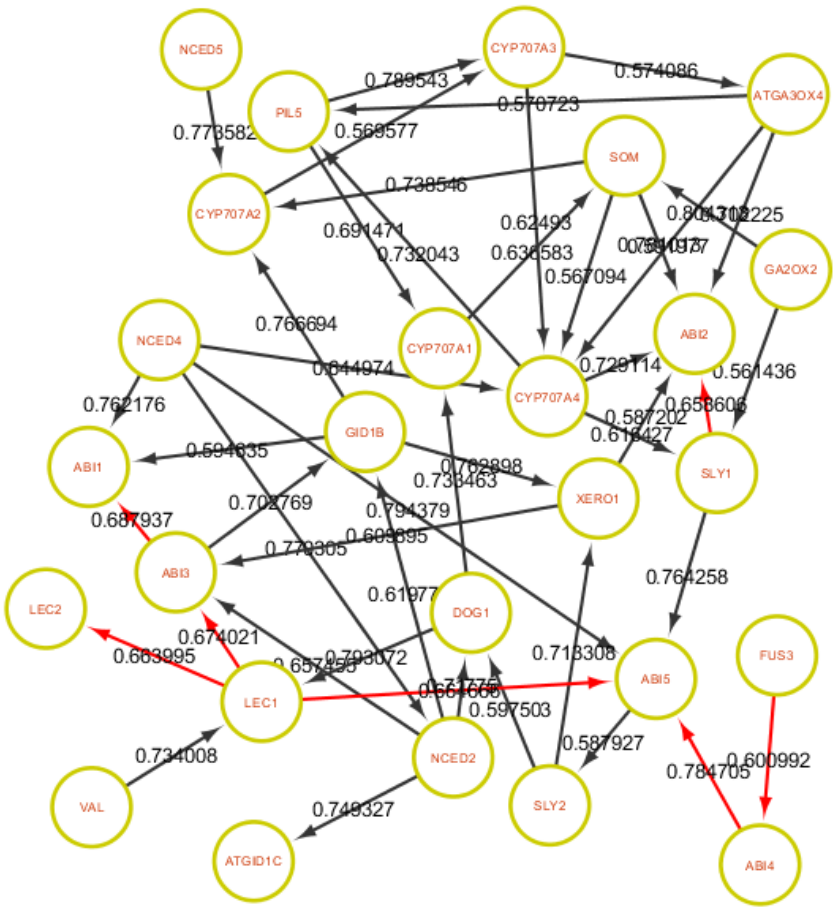

Figure 6.8: The Beacon inference predicted above edges in the Seed Development Network1 using the prior knowledge(Figure 6.1) and the 33 genes time series data set. The new predicted edges are highlighted in Red. The new predicted edges were absent in Seed Development Network1 and are inferred by the Beacon inference engine

LEC1	ABI3	0.864666
LEC1	LEC2	0.663995
LEC1	ABI5	0.664666
ABI3	ABI1	0.687937
FUS3	ABI4	0.600992
ABI4	ABI5	0.784705
SLY1	ABI2	0.794008

Table 6.1: The tabular representation of the new edges inferred between the genes in Seed Development Network1. The new predicted edges were absent in Seed Development Network1 and are inferred using the Beacon inference engine

Beacon Inference Engine Validation			
Number of transcripts	Total Predicted Present edges	New Edges	Known Edges
33	47	7	40
70	102	40	52

Table 6.2: The predicted edges for Seed Development Network3 and Seed Development Network1 using Beacon Inference Engine

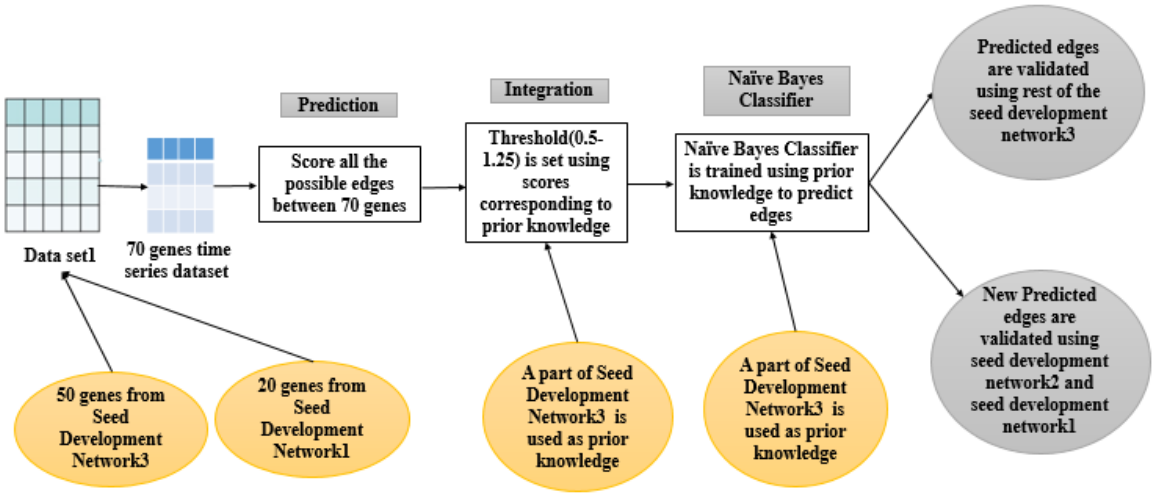

Figure 6.9: The Beacon Inference Engine Pipeline used to expand Seed Development Network3 with respect to the 20 genes that are present exclusively in Seed Development Network1

6.2 Beacon inference engine validation to infer new edges in signal transduction pathway with respect to additional biological entities

In addition to explore new edges Beacon inference engine provides the provision of expanding the signal transduction pathways with respect to additional biological entities. The Beacon inference engine tests this approach using Seed Development Network3 in the plant *Arabidopsis thaliana* [52]; see Figure 6.9. The Beacon inference engine expands Seed Development Network3 with respect to the 20 genes that are present exclusively in Seed Development Network1 [20]; see in Table 6.3.

The time course data of expression of all the genes that are present in Seed Development Network1 and Seed Development Network3 is taken from the Data Set1;ref Chapter 4.2. It consists of 70 genes out of which 20 genes are present exclusively in Seed Development Network1. The Beacon inference engine uses CLR scoring scheme along with mutual information to predict all the possible 1600 edges between the 70 genes. The distribution of the CLR score of all the predicted edges is shown in Figure 6.10.

The predicted edges consists of false positive edges that are eliminated using the prior knowledge

Gene Name
CYP707A2
NCED5
ATGID1C
DOG1
XERO1
SLY1
NCED4
NCED2
CYP707A4
SOM
ABI2
ABI1
SLY2
PIL5
VAL 1
ABI4
GA2OX2
G1D1B
ATGA3OX4
SLY2

Table 6.3: 20 genes present exclusively in the Seed Development Network1 which are used for the expansion of Seed Development Network3 by inferring new edges

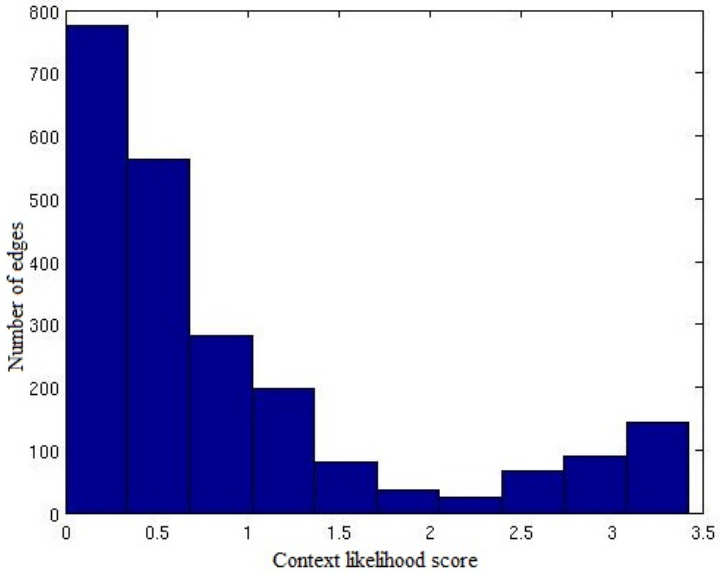

Figure 6.10: Histogram plot of the CLR score of all possible edges between 70 genes in the time series data set

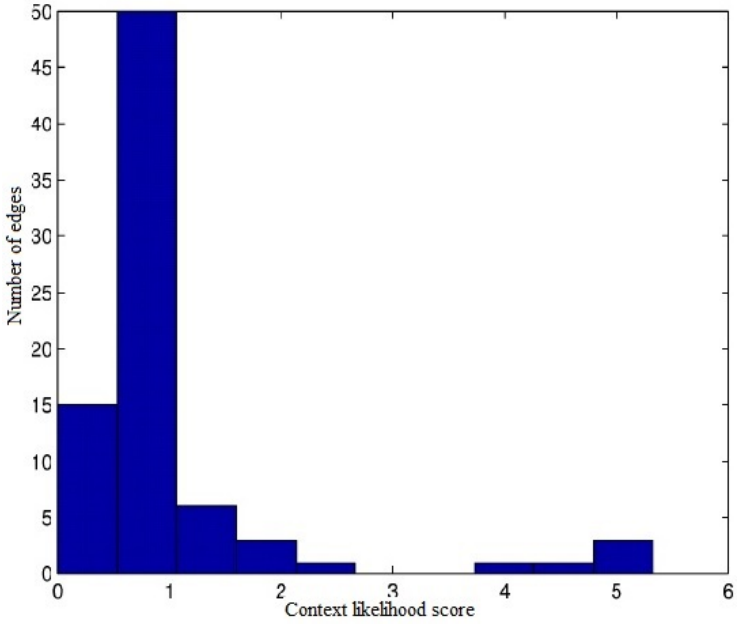

Figure 6.11: Histogram plot of CLR Score of prior knowledge in seed development network

integration step of Beacon inference engine. The prior knowledge consists of 67 edges present in signal transduction pathways as shown in Figure 6.12. The distribution of the CLR score corresponding to the prior knowledge is shown in Figure 6.11 defines the score range 0.5 and 1.25 to eliminate the false positive predicted edges from the previous step and reduces the total predicted edges to 460. The naive Bayes classifier step of the Beacon inference engine uses naive Bayes for the reduction of false positive predicted edges. The positive samples consists of the 20 edges from prior knowledge in score range 0.5-1.25 and the negative sample consist of edges provided by user that are not present in the prior knowledge. The final output from Bayes classifier predicts 40 new connections as shown in Figure 6.13,Table 6.4 which is 24% of the total edges predicted(present and absent), i.e., 365. The new connections represent the edges not present in the Seed Development Network1 and are verified through Seed Development Network2. The precision of the classifier is .814. Precision is the number of true edges out of the total edges. To calculate precision we have not considered the edges we do not have any information about. The Beacon inference engine predicts 102 directed edges as shown in Figure 6.14.

Recall rate is number of true edges out of the total known edges. The Seed Development Network3 has 67 connections. The Beacon inference engine recalls 52 connections giving us the recall rate of .776. The above results validates the Beacon inference engine as a tool that can integrate different signal transduction pathways. In this the integration of Seed Development Network1 is done with Seed Development Network3 by expanding the Seed Development Network3 with respect to genes present exclusively in Seed Development Network1.

6.3 Infer new edges for Transcripts TCONS_00020995 and TCONS_00020996

The above data set validates the Beacon inference engine across different functions proposed in the thesis. After the validation of the Beacon inference tool, it is used to study gene HSI2; see Figure 6.15. The Beacon inference tool explores new edges for the pathway component HSI2. The predicted edges are classified as the forward and the backward edges on the basis of edges direction. The forward edges indicates that HSI2 is the source and regulates gene while backward edge suggests that HSI2 is the target gene, suggesting possible negative regulation of the gene. The Beacon inference engine uses Data Set2 to predict new edges;refer chapter4.3. The Data Set2 has transcripts $TCONS\_00020995$

NCED5	ABI5	0.747901
NCED5	CYP707A2	0.773582
NCED4	NCED2	0.770305
NCED2	ABI3	0.657455
NCED2	DOG1	0.71775
NCED2	GID1B	0.619778
NCED2	ATGID1C	0.749327
NCED4	ABI5	0.794379
NCED4	ABI1	0.762176
NCED4	CYP707A4	0.644974
LEC1	LEC2	0.663995
SLY1	ABI5	0.764258
ABI5	SLY2	0.587927
LEC1	ABI5	0.664666
ABI3	ABI1	0.687937
XERO1	ABI3	0.609895
ABI3	GID1B	0.702769
GID1B	ABI1	0.594635
XERO1	ABI2	0.587202
CYP707A4	ABI2	0.729114
SOM	ABI2	0.781013
ATGA3OX4	ABI2	0.702225
SLY1	ABI2	0.658606
DOG1	CYP707A1	0.733463
SLY2	DOG1	0.597503
DOG1	LEC1	0.793072
GID1B	XERO1	0.762898
SLY2	XERO1	0.713308
PIL5	CYP707A1	0.691471
CYP707A1	SOM	0.636583
CYP707A2	CYP707A3	0.569577
CYP707A3	CYP707A4	0.62493
PIL5	CYP707A3	0.789543
CYP707A3	ATGA3OX4	0.574086
SOM	CYP707A2	0.738546
GID1B	CYP707A2	0.766694
VAL	LEC1	0.734008
LEC1	ABI3	0.674021
CYP707A4	PIL5	0.732043
SOM	CYP707A4	0.567094
ATGA3OX4	CYP707A4	0.591977
CYP707A4	SLY1	0.616427
ATGA3OX4	PIL5	0.570723
GA2OX2	SOM	0.804313
GA2OX2	SLY1	0.561436
FUS3	ABI4	0.600992
ABI4	ABI5	0.784705

Table 6.4: The tabular representation of the new edges inferred between the genes in Seed Development Network3 and the 20 genes used for expansion of Seed Development Network3. The new predicted edges were absent in Seed Development Network3 and are inferred using the Beacon inference engine

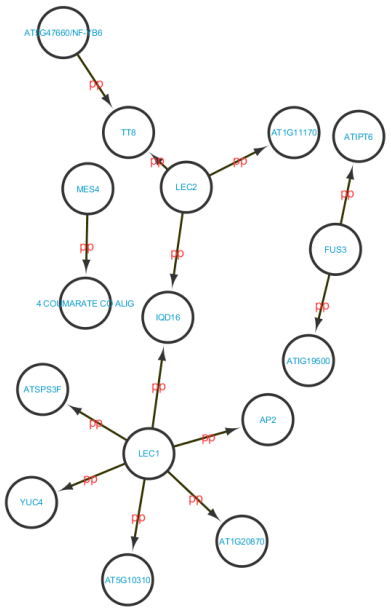

Figure 6.12: Sample of edges in Seed Development Network1 that are used as prior known edges to infer new edges in Seed Development Network3

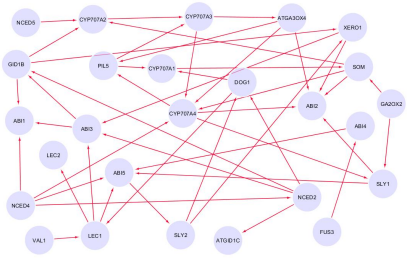

Figure 6.13: The graphical representation of the new edges inferred between the genes in Seed Development Network3 and the 20 genes used for expansion of Seed Development Network3. The new predicted edges were absent in Seed Development Network3 and are inferred using the Beacon inference engine

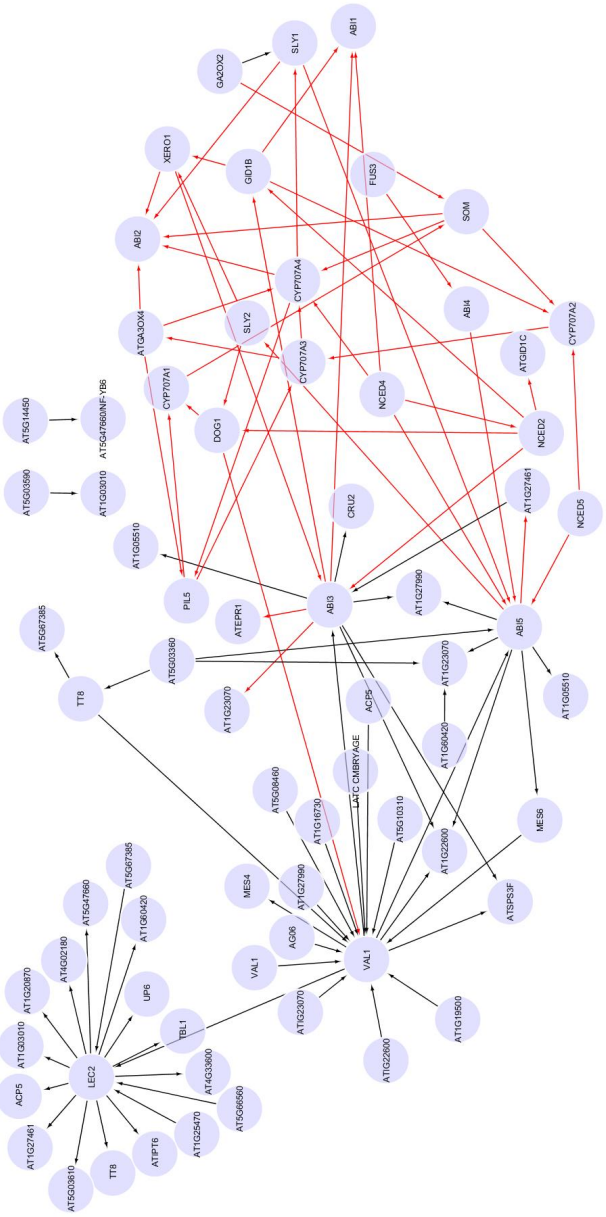

Figure 6.14: The Beacon inference engine predicted above edges in the Seed Development Network3 using the prior knowledge and the 70 genes time series data set. New predicted edges are highlighted in RED

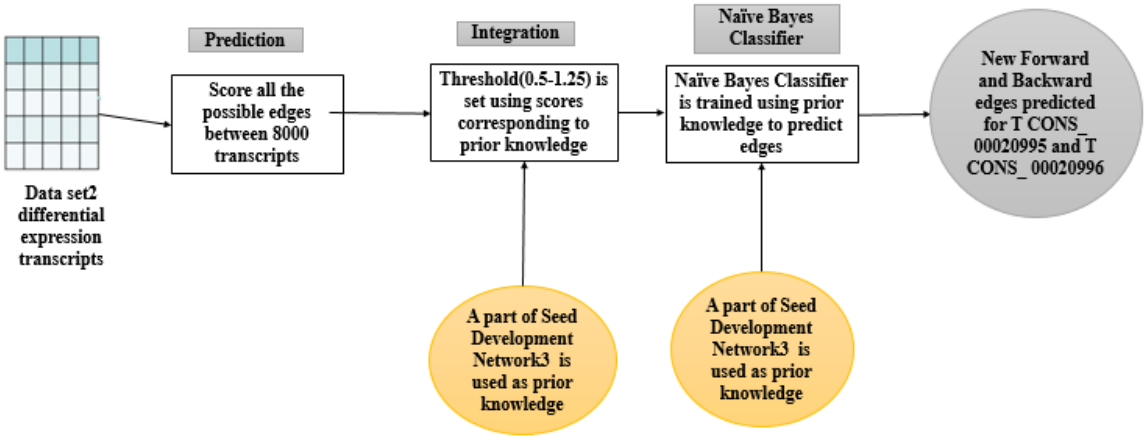

Figure 6.15: The Beacon Inference Engine Pipeline used to infer new edges for Transcripts TCONS_00020995 and TCONS_00020996

and $TCONS\_00020996$, corresponding to gene HSI2. The input to prior knowledge integration step of the Beacon inference engine are the prior known seed development networks [52].

The Beacon inference engine predicts all the possible edges between transcripts present in the time series data set using mutual information and CLR algorithm. The CLR probability distribution for all the predicted edges is shown in Figure 6.11. The prior knowledge integration step eliminates the false positives by defining the bounds for the CLR score. The CLR score distribution corresponding to prior knowledge is 0.5-1.85. The edges having CLR score outside 0.5-1.85 are ignored and the remaining edges are used as an input for naive Bayes classification to predict new edges. The naive Bayes classifier is trained using the prior knowledge as the positive sample. The Beacon inference tool predicted 251 forward edges and 279 backward edges; see Table 6.5.

The interesting observation is the relatively low redundancy of edges between the two splice variants of the HSI2 and the other transcripts present in time course expression transcripts data. The transcript $TCONS\_00020995$ has 121 forward edges as shown in Figure 6.21, while transcript $TCONS\_00020996$ has 128 forward edges as shown in Figure 6.20. We observe that transcripts $TCONS\_00020995$ and $TCONS\_00020996$ has 25 common edges as shown in Figure 6.19. The common edges strengths the edge prediction between the HSI2 and the other transcripts present in time course expression transcripts data. The transcript $TCONS\_00020995$ has 139 backward edges as shown in Figure 6.17, while transcript $TCONS\_00020996$ has 139 backward edges as shown in Fig-

Predicted edges using Beacon Inference Engine for gene HSI2			
Transcript Name	Total Predicted Present edges	Forward edges	Backward edges
TCONS_00020995	260	121	139
TCONS_00020996	267	128	139

Table 6.5: Predicted edges for transcripts TCONS_00020995 and TCONS_00020996 using the Beacon Inference Engine

ure 6.16. There are 100 common edges between the predicted edges for transcript $TCONS\_00020995$ and $TCONS\_00020996$ shown in Figure 6.18.

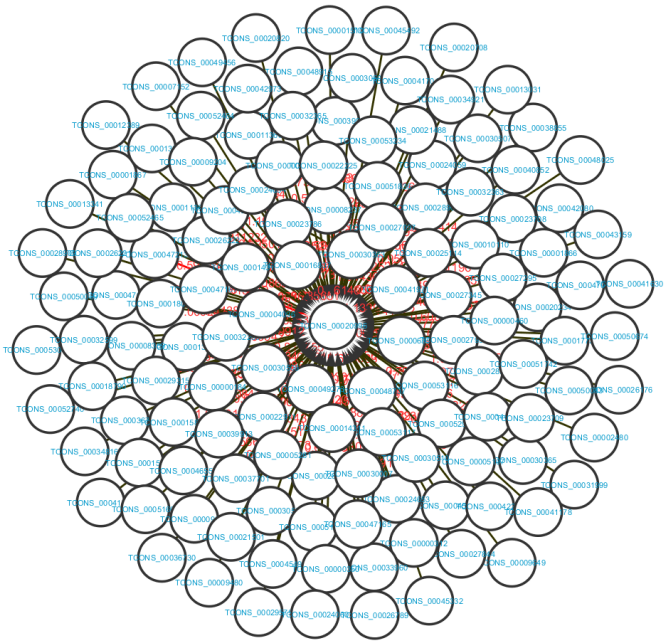

Figure 6.16: The backward edges predicted for $TCONS\_00020996$ using the Beacon inference engine.

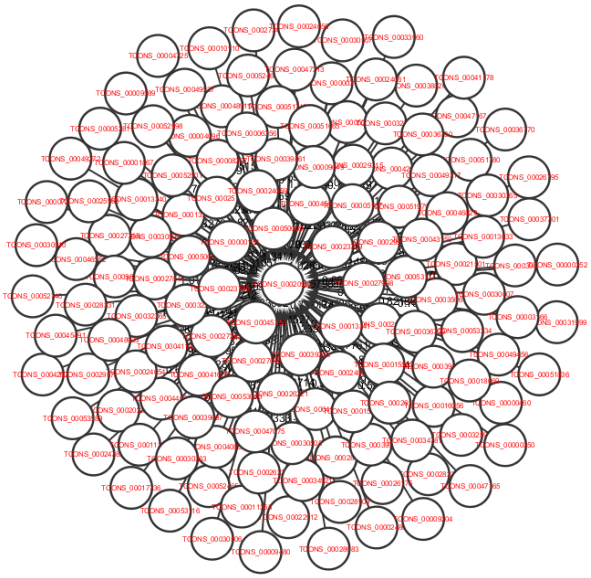

Figure 6.17: The backward edges predicted for $TCONS\_00020995$ using the Beacon inference engine.

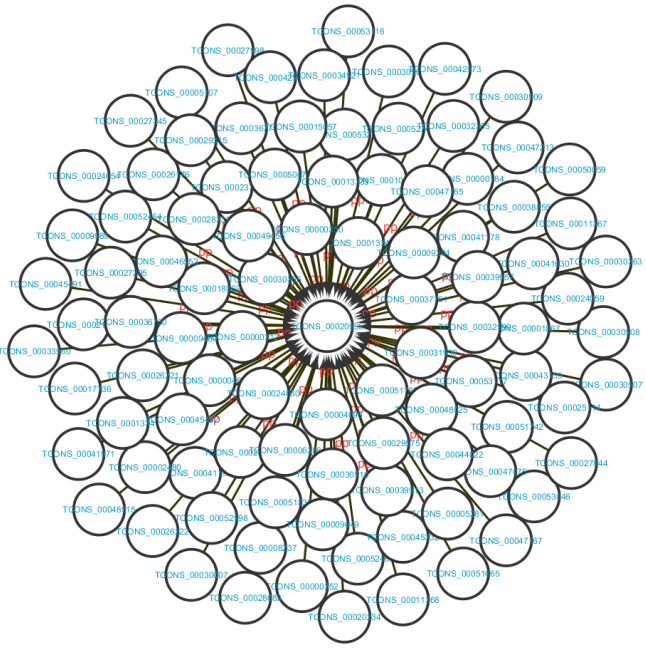

Figure 6.18: The common backward edges between $TCONS\_00020996$ and $TCONS\_00020995$.

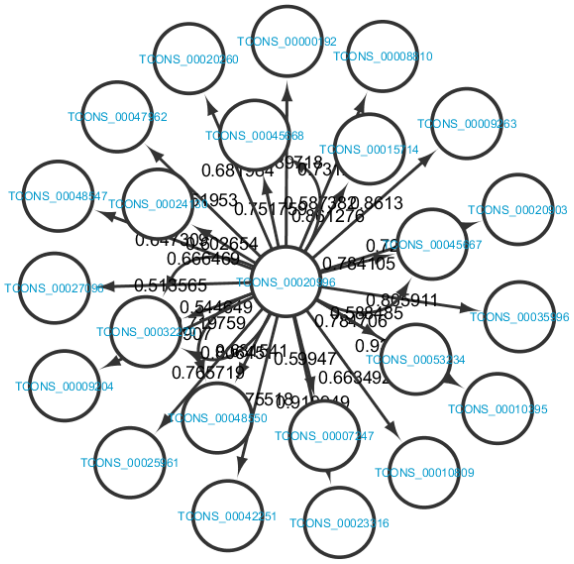

Figure 6.19: The common forward edges between $TCONS\_00020996$ and $TCONS\_00020995$.

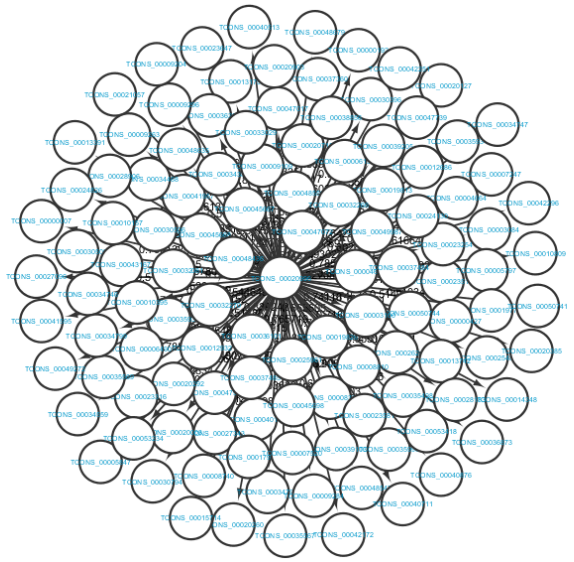

Figure 6.20: The forward edges predicted for $TCONS\_00020996$ using the Beacon inference engine.

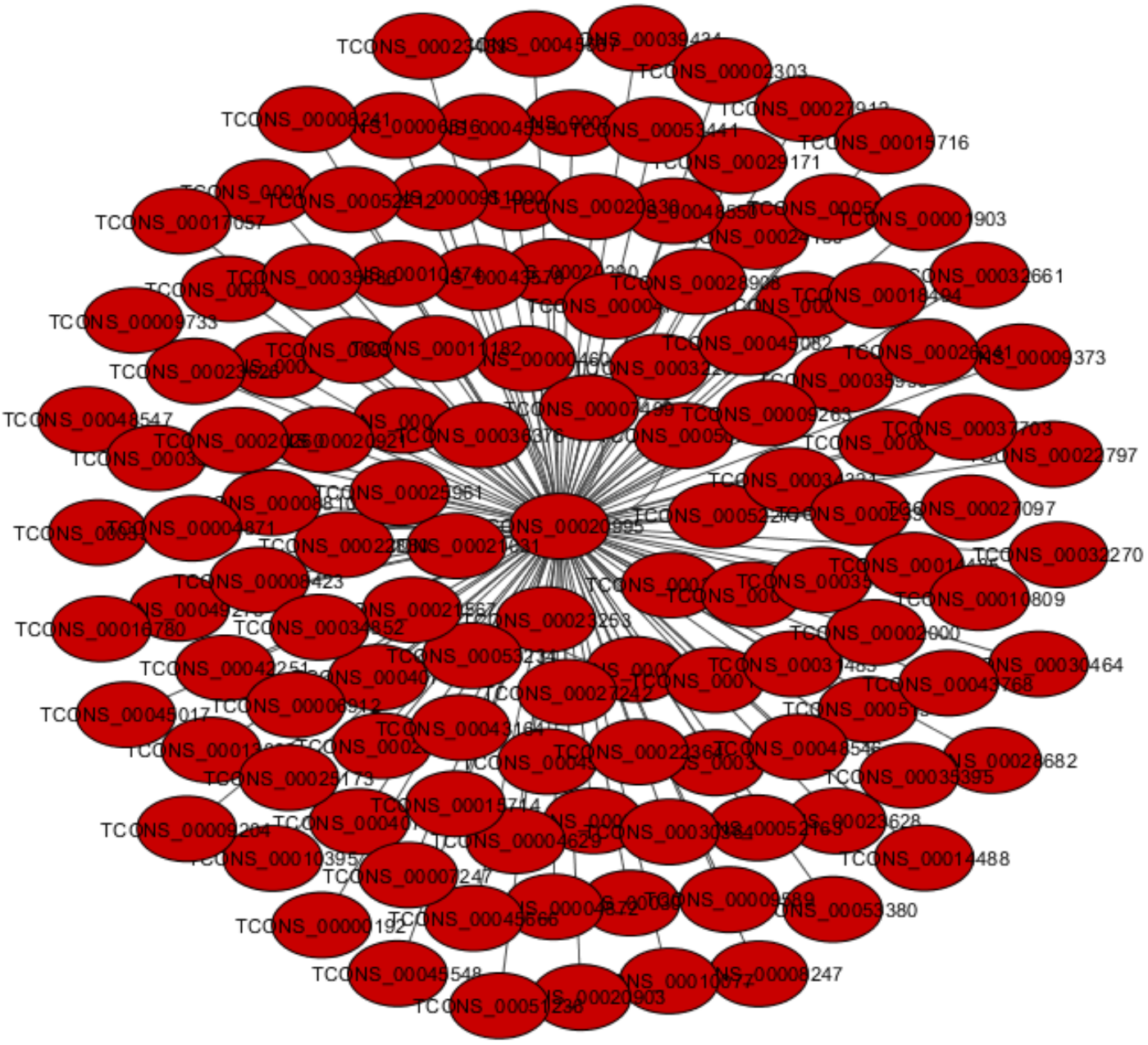

Figure 6.21: The forward edges predicted for *TCONS_00020995* using the Beacon inference engine.

Chapter 7

CONCLUSIONS

Signal transduction pathways are key to understand regulatory networks in plants in response to stimuli. However, the amount of information lags behind the available time series data sets. Thus we need computational methods to expand signal transduction pathways. In this work, the Beacon inference tool has been used to explore new information in seed development networks. The Beacon inference tool uses networks identified by experts as prior knowledge to predict new edges. The new edges are predicted between the biological entities present in seed development networks along with the entities added as per the time series data set. The problem in predicting edges with the available methods is that the egdes are undirectional . Also, present computational method predicts a large number of false positives egdes. The novelty in the Beacon inference engine is the use of directed inference to predict directed edges between the biological entities along with the provision to expand the well established signal transduction pathways with respect to time series data set. The Beacon inference engine uses the directed mutual information measure along with the context likelihood scoring scheme to predict edges among biological entities.

In this thesis, the Beacon inference engine has been developed and validated by using seed development networks pathways in the model plant *Arabidopsis thaliana*. The Beacon inference engine's ability to predict new edges between the biological entities present in signal transduction pathways is validated by using Seed Development Network1 whereas the function to predict new edges with respect to additional components as per the time series data set is verified by using the Seed Development Network3 along with the Seed Development Network1. After validation, the

Beacon inference engine has been used to explore new connections for gene HSI2 using Data Set2.

The Beacon inference engine, when validated, predicted 47 existing edges between the 33 genes present in the Seed Development Network1. The 40 predicted edges were verified using the Seed Development Network1; however the remaining 7 were considered as new edges and are validated using the Seed Development Network2. The Beacon inference engine predicted 40 new edges, when implemented for Seed Development Network3 with respect to additional pathway components in the time series data set. The new edges are the edges between the genes that were not present in the prior established seed development network, but were present in the time series data set. The 40 new predicted edges represent 24% of total predicted edges. The recall rate and the precision of the Beacon inference engine is determined by using the seed development network as the ground truth. The recall rate represents the number of true edges predicted from the total known edges, whereas precision represents the number of true edge predicted from the total predicted edges. The recall rate of the Beacon inference engine is .776 and the precision is .814.

After validation, the Beacon inference tool is used to predict new edges for gene HSI2 using the Data Set2. The splice variant data set has two transcripts corresponding to gene HSI2 with different nearest neighbors that is $TCONS\_00020995$ and $TCONS\_00020996$.The Beacon inference engine predicted 251 forward and 279 backward edges for transcripts $TCONS\_00020995$ and $TCONS\_00020996$. There were 125 edges that were common between the transcripts $TCONS\_00020995$ and $TCONS\_00020996$.

One of the drawbacks of the Beacon inference engine is that it cannot distinguish between the inhibitory relationships and the ones in which activation is involved. The Beacon inference engine can predict the direction of information flow between the genes as it uses directed mutual information for directed inference which may help in inferring inhibitory relationships. The definition of upper bound and lower bound for scoring scheme using prior knowledge penalizes the prediction of new edges outside that range. The present prior knowledge is taken from the established signal pathways for *Arabidopsis thaliana*. However, much information still needs to be discovered for *Arabidopsis thaliana* signal pathways.

There are many computational methods to predict edges between the biological entities in signaling pathways. However there is still a lot to be done to achieve precise and biological meaningful predictions. The ideal computational method to explore signal transduction pathways will be the one that can predict new directed edges inferring inhibitory relationships . Also, the Beacon inference

engine uses a context likelihood relatedness scoring scheme to score the edges insignal transduction pathways. The future work could consist in providing a facility to biologist to use different scoring schemes to score the information present in signal transduction pathways in order to have more meaningful predictions.

Chapter 8

REFERENCES

[1] T. Ando, S. Imoto, and S. Miyano, *Functional data analysis of the dynamics of gene regulatory networks*, in Knowledge Exploration in Life Science Informatics, Springer, 2004, pp. 69–83.

[2] A. H. Ang and W. H. Tang, *Probability concepts in engineering*, Planning, 1 (2004), pp. 1–3.

[3] B. Ari and H. A. Güvenir, *Clustered linear regression*, Knowledge-Based Systems, 15 (2002), pp. 169–175.

[4] M. Bansal, V. Belcastro, A. Ambesi-Impiombato, and D. Di Bernardo, *How to infer gene networks from expression profiles*, Molecular Systems Biology, 3 (2007).

[5] Z. Bar-Joseph, G. K. Gerber, D. K. Gifford, T. S. Jaakkola, and I. Simon, *Continuous representations of time-series gene expression data*, Journal of Computational Biology, 10 (2003), pp. 341–356.

[6] J. M. Berg, J. L. Tymoczko, and L. Stryer, *Signal- transduction pathways: An introduction to information metabolism*, (Biochemistry. 5th edition.WH Freeman. New York:2002.).

[7] S. K. Botting, J. P. Trzeciakowski, M. F. Benoit, S. A. Salama, and C. R. Diaz-Arrastia, *Sample entropy analysis of cervical neoplasia gene-expression signatures*, BMC Bioinformatics, 10 (2009), p. 66.

[8] R. J. Brooker, *Genetics: Analysis and Principles*, Addison Wesley Longman, Inc, 1999.

[9] A. J. Butte and I. S. Kohane, *Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements*, in Pacific Symposium on Biocomputing (PSB), vol. 5, 2000, pp. 418–429.

[10] A. J. Butte, P. Tamayo, D. Slonim, T. R. Golub, and I. S. Kohane, *Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks*, Proceedings of the National Academy of Sciences, 97 (2000), pp. 12182–12186.

[11] C. Caragea, D. Caragea, A. Silvescu, and V. Honavar, *Semi-supervised prediction of protein subcellular localization using abstraction augmented markov models*, BMC Bioinformatics, 11 (2010), p. S6.

[12] T.-J. Cham and R. Cipolla, *Automated B-spline curve representation incorporating mdl and error-minimizing control point insertion strategies*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 21 (1999), pp. 49–53.

[13] C. Cheadle, M. P. Vawter, W. J. Freed, and K. G. Becker, *Analysis of microarray data using z score transformation*, The Journal of molecular diagnostics, 5 (2003), pp. 73–81.

[14] M. S. Cheong and D.-J. Yun, *Salt-stress signaling*, Journal of Plant Biology, 50 (2007), pp. 148–155.

[15] W. B. Copeland, B. A. Bartley, D. Chandran, M. Galdzicki, K. H. Kim, S. C. Sleight, C. D. Maranas, and H. M. Sauro, *Computational tools for metabolic engineering*, Metabolic Engineering, 14 (2012), pp. 270–280.

[16] C. O. Daub, R. Steuer, J. Selbig, and S. Kloska, *Estimating mutual information using b-spline functions–an improved similarity measure for analysing gene expression data*, BMC Bioinformatics, 5 (2004), p. 118.

[17] R. Elkon, R. Vesterman, N. Amit, I. Ulitsky, I. Zohar, M. Weisz, G. Mass, N. Orlev, G. Sternberg, R. Blekhman, et al., *Spike–a database, visualization and analysis tool of cellular signaling pathways*, BMC Bioinformatics, 9 (2008), p. 110.

[18] J. Ernst and Z. Bar-Joseph, *Stem: A tool for the analysis of short time series gene expression data*, BMC Bioinformatics, 7 (2006), p. 191.

[19] J. J. Faith, B. Hayete, J. T. Thaden, I. Mogno, J. Wierzbowski, G. Cottarel, S. Kasif, J. J. Collins, and T. S. Gardner, *Large-scale mapping and validation of Escherichia coli transcriptional regulation from a compendium of expression profiles*, PLoS Biology, 5 (2007), p. e8.

[20] R. Finkelstein, *Abscisic acid synthesis and response*, The Arabidopsis Book, (2013), p. e0166.

[21] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, *Using Bayesian networks to analyze expression data*, Journal of Computational Biology, 7 (2000), pp. 601–620.

[22] A. Greenfield, C. Hafemeister, and R. Bonneau, *Robust data-driven incorporation of prior knowledge into the inference of dynamic regulatory networks*, Bioinformatics, 29 (2013), pp. 1060–1067.

[23] A.-C. Haury, F. Mordelet, P. Vera-Licona, and J.-P. Vert, *Tigress: Trustful inference of gene regulation using stability selection*, BMC Systems Biology, 6 (2012), p. 145.

[24] G. H. John and P. Langley, *Estimating continuous distributions in bayesian classifiers*, in Proceedings of the Eleventh conference on Uncertainty in artificial intelligence, Morgan Kaufmann Publishers Inc., 1995, pp. 338–345.

[25] C. Kaleta, A. Göhler, S. Schuster, K. Jahreis, R. Guthke, and S. Nikolajewa, *Integrative inference of gene-regulatory networks in Escherichia coli using information theoretic concepts and sequence analysis*, BMC Systems Biology, 4 (2010), p. 116.

[26] S. Koussevitzky, N. Suzuki, S. Huntington, L. Armijo, W. Sha, D. Cortes, V. Shulaev, and R. Mittler, *Ascorbate peroxidase 1 plays a key role in the response of Arabidopsis thaliana to stress combination*, Journal of Biological Chemistry, (2008).

[27] A. Krishnan and A. Pereira, *Integrative approaches for mining transcriptional regulatory programs in arabidopsis*, Briefings in Functional Genomics & Proteomics, 7 (2008), pp. 264–274.

[28] J. C. Liao, R. Boscolo, Y.-L. Yang, L. M. Tran, C. Sabatti, and V. P. Roychowdhury, *Network component analysis: Reconstruction of regulatory signals in biological systems*, Proceedings of the National Academy of Sciences, 100 (2003), pp. 15522–15527.

[29] B. Liu, A. de La Fuente, and I. Hoeschele, *Gene network inference via structural equation modeling in genetical genomics experiments*, Genetics, 178 (2008), pp. 1763–1776.

[30] S. Ma and H. J. Bohnert, *Integration of Arabidopsis thaliana stress-related transcript profiles, promoter structures, and cell-specific expression*, Genome Biology, 8 (2007), p. R49.

[31] A. Madar, A. Greenfield, E. Vanden-Eijnden, and R. Bonneau, *DREAM3: Network inference using dynamic context likelihood of relatedness and the inferelator*, PloS ONE, 5 (2010), p. e9803.

[32] S. R. Maetschke, P. B. Madhamshettiwar, M. J. Davis, and M. A. Ragan, *Supervised, semi-supervised and unsupervised inference of gene regulatory networks*, Briefings in bioinformatics, (2013), p. bbt034.

[33] D. Marbach, J. C. Costello, R. Küffner, N. M. Vega, R. J. Prill, D. M. Camacho, K. R. Allison, M. Kellis, J. J. Collins, and G. Stolovitzky, *Wisdom of crowds for robust gene network inference*, Nature Methods, 9 (2012), pp. 796–804.

[34] F. Mordelet and J.-P. Vert, *Sirene: supervised inference of regulatory networks*, Bioinformatics, 24 (2008), pp. i76–i82.

[35] R. Munns and M. Tester, *Mechanisms of salinity tolerance*, Annual Revision Plant Biology, 59 (2008), pp. 651–681.

[36] E. Parzen, *On estimation of a probability density function and mode*, The Annals of Mathematical Statistics, (1962), pp. 1065–1076.

[37] J. Pearl et al., *Causal inference in statistics: An overview*, Statistics Surveys, 3 (2009), pp. 96–146.

[38] S. Pemmaraju and S. Skiena, *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica*, Cambridge University Press, 2003.

[39] B. Pham, *Offset approximation of uniform b-splines*, Computer-Aided Design, 20 (1988), pp. 471–474.

[40] V. Pieterse and P. E. Black, *Dictionary of Algorithms and Data Structures [online]*, http://www.nist.gov/dads/HTML/directedGraph.html, 20 November 2008.

[41] C. R. Quinn, R. Iriyama, and D. D. Fernando, *Computational predictions and expression patterns of conserved micrornas in loblolly pine (pinus taeda)*, Tree Genetics & Genomes, 11 (2015), pp. 1–11.

[42] A. S. Raghavendra, V. K. Gonugunta, A. Christmann, and E. Grill, *ABA perception and signalling*, Trends in Plant Science, 15 (2010), pp. 395–401.

[43] M. Rodbell, *Proteins in membrane transduction*, Nature, 284 (1980), p. 17.

[44] D. F. Rogers and J. A. Adams, *Mathematical Elements for Computer Graphics*, McGraw-Hill Higher Education, 1989.

[45] D. Sabbadin and S. Moro, *Supervised molecular dynamics (SUMD) as a helpful tool to depict gpcr–ligand recognition pathway in a nanosecond time scale*, Journal of Chemical Information and Modeling, 54 (2014), pp. 372–376.

[46] M. Seki, A. Matsui, J.-M. Kim, J. Ishida, M. Nakajima, T. Morosawa, M. Kawashima, M. Satou, T. K. To, Y. Kurihara, et al., *Arabidopsis whole-genome transcriptome analysis under drought, cold, high-salinity, and aba treatment conditions using tiling array and 454 sequencing technology*, in Plant and Cell Physiology, vol. 48, Oxford University Press, 2007, pp. S8–S8.

[47] H. Shi, B. Schmidt, W. Liu, and W. Müller-Wittig, *Parallel mutual information estimation for inferring gene regulatory networks on GPUs*, BMC Research Notes, 4 (2011), p. 189.

[48] G. S. Shieh, C.-M. Chen, C.-Y. Yu, J. Huang, W.-F. Wang, and Y.-C. Lo, *Inferring transcriptional compensation interactions in yeast via stepwise structure equation modeling*, BMC Bioinformatics, 9 (2008), p. 134.

[49] K. Shinozaki and E. S. Dennis, *Cell signalling and gene regulation: global analyses of signal transduction and gene expression profiles*, Current Opinion in Plant Biology, 6 (2003), pp. 405–409.

[50] K. Shinozaki and K. Yamaguchi-Shinozaki, *Gene expression and signal transduction in water-stress response.*, Plant Physiology, 115 (1997), p. 327.

[51] R. R. Sinden, *DNA Structure and Function*, Elsevier, 2012.

[52] N. Sreenivasulu and U. Wobus, *Seed-development programs: A systems biology-based comparison between dicots and monocots*, Annual Review of Plant Biology, 64 (2013), pp. 189–217.

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

[54] M. Stitt, *Progress in understanding and engineering primary plant metabolism*, Current Opinion in Biotechnology, 24 (2013), pp. 229–238.

[55] J. S. Thaler, R. Karban, D. E. Ullman, K. Boege, and R. M. Bostock, *Cross-talk between jasmonate and salicylate plant defense pathways: Effects on several plant parasites*, Oecologia, 131 (2002), pp. 227–235.

[56] A. Venelli and V. Dupaquis, *Efficient entropy estimation for mutual information analysis*, TSI-Technique et Science Informatiques, 30 (2011), p. 1217.

[57] R. F. Weaver and P. W. Hedrick, *Genetics*, Wm. C. Brown Publishers, 1997.

[58] D. B. West, *Introduction to Graph Theory*, Prentice Hall, Upper Saddle River, 2001.

[59] Wikipedia, *Signal transduction — wikipedia, the free encyclopedia*, 2015. [Online; accessed 16-July-2015].

[60] L. Wu, Z. Zhang, H. Zhang, X.-C. Wang, and R. Huang, *Transcriptional modulation of ethylene response factor protein jerf3 in the oxidative stress response enhances tolerance of tobacco seedlings to salt, drought, and freezing*, Plant Pysiology, 148 (2008), pp. 1953–1963.

[61] C.-W. Yao, B.-D. Hsu, and B.-S. Chen, *Constructing gene regulatory networks for long term photosynthetic light acclimation in Arabidopsis thaliana*, BMC Bioinformatics, 12 (2011), p. 335.

[62] C. Yoo and E. M. Brilz, *The five-gene-network data analysis with local causal discovery algorithm using causal bayesian networks*, Annals of the New York Academy of Sciences, 1158 (2009), pp. 93–101.

[63] A. K. Zoran Nikoloski, Sabrina Hempel and J. Kurths, *Unraveling gene regulatory networks from time-resolved gene expression data - A measures comparison study: 2011 update*, Network Analysis, 12 (2011).