Title
Analysis of Large-Scale Genetic Perturbation with Linear Regression of Microarray and Bayesian Networks

Author
Jiang, Ruifu

Publication Date
2018

UNIVERSITY OF CALIFORNIA

Los Angeles

Analysis of Large-Scale Genetic Perturbation with Linear Regression of Microarray and Bayesian Networks

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Applied Statistics

by

Ruifu Jiang

2018

ABSTRACT OF THE THESIS

Analysis of Large-Scale Genetic Perturbation
with Linear Regression of Microarray and Bayesian Networks

by

Ruifu Jiang
Master of Applied Statistics
University of California, Los Angeles, 2018
Professor Qing Zhou, Chair

This paper aims to examine how large-scale genetic perturbations reveal regulatory network and an abundance of gene-specific repressors by analyzing data from a published paper (Kemmeren et al., 2014) [35]. The main goal is to uniformly determine the effect of different components on the expression of all other genes. The idea of their experiment is doing gene deletion of one-quarter of yeast genes individually and then observing the mRNA expression genomewide. Then genetic perturbation would be resulted, which also shows some properties including the architecture of protein complexes and pathways, identification of expression changes compatible with viability, and the varying responsiveness to genetic perturbation. And all data collected from this experiment is constructed as a genetic perturbation network which present a varying connectivities among regulators. Finally it provides a regulation network with analysis result from R package limma and sparsebn.

The thesis of Ruifu Jiang is approved.

Nicolas Christou

Ying Nian Wu

Qing Zhou, Committee Chair

University of California, Los Angeles

2018

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

Introduction

As it is well-known, cell function is performed by plenty of molecular interactions. Studying those interactions on how to affect the cell function in molecular-wide is still one of the toughest part nowadays (Ideker et al., 2001) [1]. Effort on this study is quite urgent as people realize that combination by few simple interaction would result in complicated properties. All interaction mainly divided into two categories, one is physical interaction which is easily observed, e.g. protein-protein interactions (Walhout and Vidal, 2001) [2], another is abstract interaction which is harder to study, e.g. genetic interactions (Costanzo et al., 2010) [3]. Both type interactions provide important information to studying the individual function or system properties. Hence in the past few years, a bunch of dataset were created rapidly. Particularly among these dataset, one related to mRNA expression is generally used to study how regulatory network rely on mRNA expression based on analyzing how mRNA expression influenced by cellular components genome-wide.

It is generally agreed that analysis of individual perturbation on genome-wide expression would go further reveal the deeper function of the entire system (DeRisi et al., 1997) [4]. In order to study the function among different component, regulatory pathways (Roberts et al., 2000) [5] and protein complexes (van de Peppel et al., 2005) [6] would be used due to less difficulties to solve and more unexpected results revealed. For example, one study of 276 mutants in the yeast Saccharomyces cerevisiae provided a result revealing a quite larger combination of genetic perturbation expression signatures (Hughes et al., 2000) [7]. This is a very complicated study as it analyzed individual factors, entire dlasses of regulators, (Hu et al., 2007) [8] and incorporation with other types of perturbation (Chua et al., 2006) [9].

According to the first compendium (Hughes et al., 2000) [7], although a large number of

such studies has been done in recent years, the result of genetic perturbations analysis did not expand as much as we expected. Within those genetic perturbations analysis, analysis of entire system is still a tough part as it is difficult to compare gene expression data generated across the different conditions, genetic backgrounds, technology platforms, types of controls, and degrees of replication in different studies. This paper will analyze the genetic perturbation to determine the properties of regulatory system.

CHAPTER 2

Experiment Design

Due to the complexity of processing nucleic acid microarrays, it is crucial to design an appropriate experiment in order to generate the most accurate measurements of the genes. This experimental design used microarrays, which are widely used in simultaneously analyzing the expression of thousands of genes. These microarrays can hold a large quantity of nucleic acid fragments, and are capable of providing information on gene expression levels or measuring specific genetic variations. The technology of microarray utilizes the action of hybridization, or the formation of double-stranded RNA by base pairing between the complementary sequences of nucleic acid molecules. Two-channel (two-color) microarrays containing pairs of separately labeled nucleic acid samples on the same array were used in this design. These two-channel microarrays can measure the difference in gene expression between pairs of competitively cohybridized nucleic acid samples.

A common reference experiment design was chosen for the microarrays. To prepare samples for the microarrays, a large quantity of RNA was extracted from the wild-type yeast strain cultures as the reference RNA. The reference RNA sample was placed in a channel on the arrays for each hybridization. The average expression levels for each mutated gene relative to the wild-type expression were statistically obtained from the reference RNA sample. Pairs of independent samples were hybridized separately on different arrays. As shown in Figure 2.1, the hybridization of the two mutants (mt 1, mt 2) and the wild-type control sample (wt) was carried out by base paring a Cy5 (red fluorescent dye) labeled mutated RNA sequence to a Cy3 (green fluorescent dye) labeled complementary sequence from the wild-type reference sample (unmutated). These pairs of separately labeled sequences can hybridize to form new double-stranded RNA, measurements were taken from the products

of the hybridization on the microarrays. In this experiment design, each sequence pair had a replication with switching the dyes of the sequences. Therefore, every mutated gene was hybridized twice, leading to a total of 4 measurements for each mutated gene on the microarrays. As the control, the microarrays also contained hybridized RNA samples of the wild-type cultures from the same day to eliminate day-specific factors that influence the results.

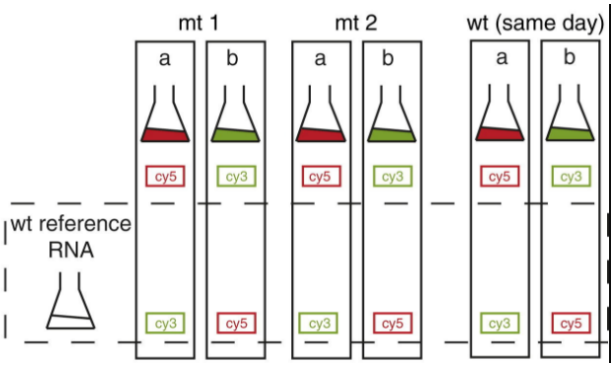

Figure 2.1: Experiment design (Kemmeren et al., 2014) [35].

CHAPTER 3

Methodologies

3.0.1 Limma

The first methodology used is "limma" which is linear model for microarray or RNA-Seq Data. This R package is a response to the rapid increasing on data related to microarray or RNA-Seq. It's popular also for its stability while analyzing small number of arrays. It shows a good capacity as well on dealing with complicated experiment with larger number of predictors and response variables. As it is widely used nowadays, two main trends have been divided for different purpose. Comparing to the past tools of analyzing only microarray, limma would also provide efficient analysis on RNA sequencing data as well. For users, it is a great choice to use the similar method and procedure to analysis the an essential dataset that previous downstream analysis tools unable to do. Another features is that, limma can provide much more ways of analyzing gene expression than traditional analysis. For example, it can abstract the gene expression into a higher-order expression signature, which make it possible to interpret the gene expression differences in biological perspective (Matthew et al., 2015) [10].

Limma's primary capabilities fit this experiment quite well. In general, tools for gene expression analysis are used regularly to determine how gene expression changes under some certain treatments or conditions. And these changes usually bring some perturbations as well. Such studies go with many observations, and also include a big number of different effect result and covariates. The harder part is that the number of replicates is much smaller than those influence factors. Dataset with big number of variables and small of observation is always tough to analysis, which inspire people to use some more special statistical method to encounter such situation. After developed limma with many generations, limma package

has become a very powerful tools to process more kinds of microarray data using various flexible statistical methods.

The basic idea of limma is doing linear regression on a gene expression matrix. Each row of the matrix is a gene we interested in the study, and each column is one observation, and fill up the matrix with an expression value respectively. Then each row could generate a linear model, and combine those feature model with weights in some ways. It use the result from gene-wise models and abstract their result to determine the nature of genomic data, which make the conclusion convincing even though small number of samples are observed.

Distinct to single linear model, limma use aone linear model to analyze the integrated matrix as a whole rather than treat each gene row separately. The different part is that, rows are not compared individually, they are more like contacting each other and share all information together. This would give us an advantage to find whether correlations exist between genes.

This approach make it flexible for further adjustment. For example, researchers can extract any pair of two gene and test whether they are independent or dependent. This implies that assumptions for this test is not as strict as comparison test. It is loose enough to allow more flexible test like interaction effects or multiple comparisons.

One key method limma uses is global parameter. It is estimated maybe from the entire dataset which include all genes in the experiment and available for single linear model build on each gene. This is possible due to the big linear model among all genes, and each gene in the big model share information together. The global parameter or global hyper-parameter has the same effect as correlation, it can be used as correlation or variation between genes.

A special statistical method is parametric empirical Bayes which allow information between genes to be used in an efficient way (Efron and Morris, 1973) [24]. Each linear model of single gene has a residual variance and the model is improved by adjusting the residual variance. However, the residual variance is not adjusted individually, it has to consider the residual got from the global model involving all genes. By modifying in this way, the degrees of freedom is more effective. And further help the model have a higher reliability to process

data with small number of samples (Smyth, 2004) [25].

As long as the empirical Bayes applied in limma, more powerful results are exposed which make empirical Bayes procedure mare reliable. As explained above, each model gives a residual variance and combining them together will get a variance trend which called mean-variance trend. Since the global variance are affected by those single variances, they all are not isolated but cooperate with each other. This make limma more special and reliable on processing data with small number of replicates while other gene expression tools fail to do. What's more, the relative weighting of single gene and the global are not necessary to be same any more. This permits the feasibility of a more robust procedure that helps researchers to find hyper-variables genes which has to be treated separately (Phipson et al., 2013) [26]. This is how one of the special statistical methods mentioned earlier works to help limma build a good reputation on generating reliable reference in flexible ways.

Another special statistical method limma uses is quantitative weight. It is very flexible that can be used in any part of the analysis. For example, it can be used in normalization in order to control probes with more emphasis. There isn't a strict rule for setting weights, it could be set from any quality information even beyond the experiment or estimated from the dataset (Ritchie et al., 2006)[27].

As limma getting more and more widely used, it has been explored to analysis high-level expressin signatures involving genes regulating each other. Instead of using expression matrix, the data includes interesting genes and and their log-fold changes. Then use rotation test to test the significance based on linear regression model of those genes (Wu et al., 2010)[28]. The rotation test allow it to combine the information like direction of the expression signature with the contribution of each gene to the expression signature, which means limma link the new expression data with the previous experimental data.

The expression data are collected as intensities, which means each microarray has to do related correction and normalization before doing any analysis. ANd limma includes various type of method to do this procedure, for example maximum likelihood and quantitative weights.

The steps of using limma is first importing data and preprocessing including correction and normalization, then analyzing with linear modeling or differential expression, and finally testing model. Figure 3.1 is a clear limma workflow.

Most microarrays are generated by image analysis program including Agilent Feature Extraction, ArrayVision, BlueFuse, GenePix, and so on. Limma use function read.maimages to read output images with format from prior programs or Stanford Microarray Database.

Usually, when a microarray image is readm the foreground and background intensities are also read automatically. The background correction is removing those non-specific background intensities from the foreground intensities. Background correction is not directly removing all background from foreground intensities, but uses a wise method based on normal distribution convolutionand normal-exponential convolution.

In order to set all samples in the same measurement scale as possible as it could be, normalization is implemented after background correction. The ideal result of doing normalization is remaining the biological differences only by removing other systematic differences. Some methods for normalization can be called by using functions like normalizeWithinArrays, normalizeBetweenArrays, and so on.

After data pre-processing, plotting some diagnostic graphs usually help researchers have a brief inspect on the data. For example, using function plotMA with some sample genes by comparing the log-expression for a mean-difference plot tells us the trend of intensity-dependent .

Major analysis starts from determining the differential expression (DE) of genes. Plotting the realtive differences between samples with plotMDS function usually show a obvious result.

The next step and the most important step is building a gene-wise linear model. Similar to general linear model, the gene-wise linear model is to estimate the log-ratio if data is two channel or log-intensities if data is single channel among samples simultaneously.

The formula is

$Y_{ij} = \mu_j + \epsilon_{ij}$

Step in Analysis	Functioin
Data Import	read.maimages / read.ilmn / read.idat
	readTargets / read.ilmn.targets
	readGAL / readSpotTypes
	controlStatus
Preprocessing & Quality Assessment	backgroundCorrect / nec
	normalizeWithinArrays
	normalizeBetweenArrays / neqc
	voom / vooma / voomaByGroup
	plotMA
	plotDensities
	plotFB
	imageplot
	plotMDS
	arrayWeights / voomWithQualityWeights
	removeBatchEffect
Model Analysis	modelMatrix
	lmFit
	lmscFit
	avereps
	duplicateCorrelation
	makeContrasts
	eBayes
	topTable
	treat
	topTreat
	decideTests
	write.fit
	propTrueNull
	genas
	volcanoplot
	heatdiagram / heatDiagram
	plotSA
	vennDiagram
Gene Set Testing	id2indices
	goana
	geneSetTest / wilcoxGST
	camera
	roast / mroast
	romer
	barcodeplot

Figure 3.1: The limma workflow.

Y_{ij} is the expression value, μ is the mean, ϵ is the error term, i is the ith sample, j is the jth treatment.

Each linear regression start within an expression matrix with rows for genes and columns for samples, and stored in a row wise fashion with related regression coefficients and errors. By comparison among interested samples, statistics are obtained for further gene ranking.

Correlation between samples are inevitable, limma uses the random effect model with all genes constrained to hold the same intrablock correlation.

Based on linear model of gene expression, more and more higher-level analysis is exploited like interaction or independence with multiple genes and molecular pathways decomposed from gene signatures.

3.0.2 Bayesian

The second methodology is Bayesian network. Bayesian network is a type of probabilistic graphical model that represents a set of variables and their conditional dependencies via a directed acyclic graph (DAG). And the package related to Bayesian network algorithm is called "sparsebn" (Aragam et al., 2017)[36]. The basic model is a p-dimensional random vector X with joint distribution P. The distribution X follows a multivariate Gaussian distribution if the data is continuous, or it would be assumed that each X is a factor with r levels. For more details about the methods in this package, see Fu and Zhou (2013) [37], Aragam and Zhou (2015) [38] and Gu et al (2018)[39] .

The traditional methods of Bayesian networks is developing algorithms from a graph-theoretic perspective based on its definition (Spirtes and Glymour 1991)[29]. In order to follow that approach, we usually have to hold its restrictive assumptions such as strong faithfulness (Uhler et al., 2013[30]; Zhang and Spirtes, 2002[31]), which make it uneasy to practice. Refer to the main idea of package sparsebn, a more general approach is adopted via structural equation models. In this approach, each conditional probability distribution (CPD) is directly modeled via a generalized linear model.

For continuous data

$X = B^T X + \epsilon$

This is called a structural equation model for X. B is the weighted adjacency matrix of a directed graph by writing $B = |\beta_1| \cdots |\beta_p| \in R^{p \times p}$ and $\epsilon \sim N(0, w_j^2)$. In this approach, we have to make sure the adjacency matrix B to be acyclic.

For discrete data

$P(X_j = l|z) = \frac{exp(z^T \beta_{lj})}{\sum_{m=1}^{r_j} exp(z^T \beta_{mj})}, l = 1, ..., r_j$

where $\beta \in R^d$ is the coefficient vector for X.

This is the conditional distribution under the parametrization. Instead of traditional product multinomial modelfor discrete data, sparsebn takes a multi-logit model. In the multi-logit model, each X is encoded by $d_j = r_j - 1$ dummy variables, and a vector of dummy variables $z_j = (z_{jk}, k = 1, ..., r_j - 1) \in \{0, 1\}^{d_j}$

Algorithms for building Bayesian network are generally divided into three type: constraint-based methods, score-based methods and hybrid methods.

The basic idea of constraint-based methods is learning the structure of a network via repeated conditional independence test to determine edges that would not exist in a DAG. This procedure would always satisfied as long as faithfulness assumption holds. The building steps are first sketch the skeleton of the network with independence tests and then enrich the edges with v-structures (Koller and Friedman, 2009)[32].

Comparing to constraint-based methods, the score-based methods use kinds of scoring functions, e.g. log-likelihood. By optimizing a certain scoring function, a DAG would be found. This algorithm is usually faster, but also likely to predict too many edges in high-dimensional setting (Cooper and Herskovits 1992) [33].

The last hybrid method is a combination of constraint-based and score-based methods. The procedure of hybrid method is that first create a search space with constraint-based method and then get the optimal DAG structure with score-based method (G amez, Mateo, and Puerta 2011)[16].

Learning Bayesian network usually start from learning using a score-based method by

regularizing the maximum likelihood estimation. Assuming an observed matrix without any missing value is $X \in R^{n \times p}$ and l represent the negative log-likelihood and ρ_λ is regularizer.

$\min_{B \in D} l(B; X) + \rho_\lambda(B)$

assuming $D \in R^{n \times p}$ are some weighted adjacent matrices. This formula contains the nonconvexity resulted from the constraint D, loss function l, and regularizer ρ_λ

If the data is continuous, the loss function l should be combined with a Gaussian likelihood got from the structural equation model mentioned earlier. If the data is discrete, the lasso penalty group should be combined with a multi-logit model got from the conditional distribution under z parametrization. Suppose $M \subset (1, ..., p)$ is a subset of variables under the intervention, which follow the distribution that:

$P(X_1, ..., X_p) \propto \prod_{i \notin M} P(X_i | pa(X_i))$

By setting L_i as the row index of matrix X, and $O_j = 1, ..., n - L_j$ as the subset of observation that not under the intervention, the negative log-likelihood factorization is:

$l(B; X) = -\sum_{j=1}^{p} \sum_{h \in O_j} log f_{\beta_i}(X_{hj} | pa(X_{hj}))$

Where B is a set of β, f_{β_i} is the conditional density in jth node, and the X_{hj} is the hth value of X_j. By using this factorization, the orientation of the edges in Bayesian network is completed.

The algorithm of above procedure is:

1. Build up a outer loop in which:

a. Tuning (β_{kj}, β_{jk}) where $j \neq k$ to minimizing $\min_{B \in D} l(B; X) + \rho_\lambda(B)$ while other parameters are fixed.

b. If there is cycle in the edge between j and k, update β as 0.

c. Run inner loop

2. Build up an inner loop that minimizing $\min_{B \in D} l(B; X) + \rho_\lambda(B)$ with edge weights β_{kj} for $(k, j) \in E$ where E is the fixed edge set get from outer loop.

3. Run outer loop until meeting a certain stopping criterion set prior.

As we can see from the main formula used in above algorithm, the value of λ has to be provided before any action. A common way of solving this is using an algorithm named solution path instead of single DAG estimate, which is also known as regularization path(Friedman et al., 2010)[34].

Hence when the algorithm is implemented, some steps with methods of coordinate descent are added to accelerate the running speed.

1. Once the $\hat{\beta}(\lambda_{l-1})$ got from previous solution path, it would be used as an initial guess for next iteration on calculating $\hat{\beta}(\lambda_l)$. This is why λ_0 is chose for $\hat{\beta}(\lambda_0) = 0$ as default in package sparsebn.

2. Rewrite the inner loop for updating β_{jk}, β_{kj} at the same time rather than updating only one of them each time. This needs a conditional that neither β_{jk} nor β_{kj} is 0.

3. A special structure named sparse data structure is used to store the result, which saves more memory and shortens the time of calculation at each iteration.

The next mission after learning Bayesian network structure is estimating parameters of conditional distribution. Those parameters decide how larger the effect size is between parents and children.

Method of least squares regression is used to regress between node and its parent if the data is continuous. This method need a precondition that the number of parents is no more than n. To improve this, let $\hat{B} = (\hat{\beta}_j)$ as a weighted adjacent matrix like before, and use it to estimate the conditional variance by given formula:

$(\hat{w_j})^2 = var(x_j - X\hat{\beta}_j)$

Apply $\hat{\Omega} = diag(\hat{w}_1^2, ..., \hat{w}_p^2)$ as a variance matrix and combining $(\hat{B}, \hat{\Omega})$ to calculate the variance covariance matrix Σ.

Multi-logit regression is used to regress between node and its parent if the data is discrete. The different part is a four way array $\hat{B}$ is obtained rather than a coefficient matrix for continuous dataset.

For this experiment, sparsebn package is used to build a Bayesian network for its various

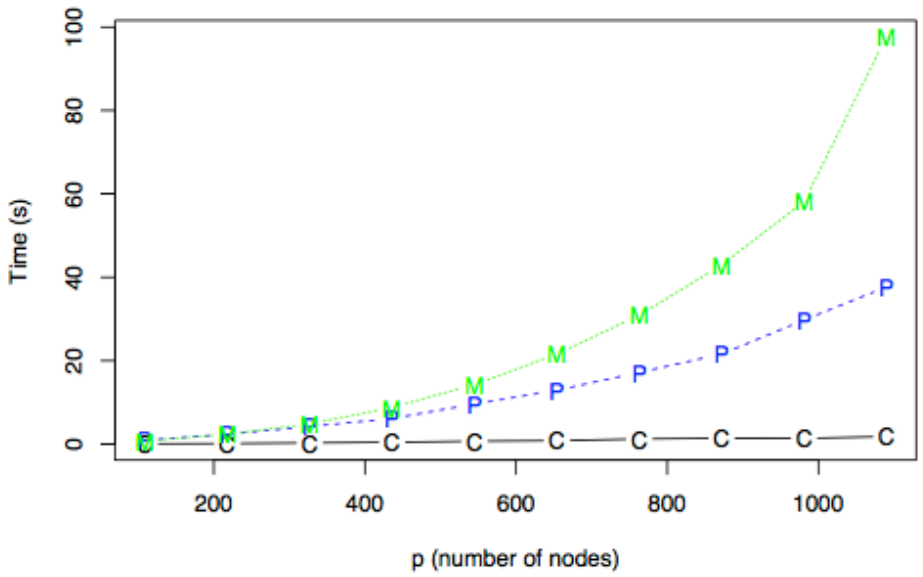

Figure 3.2: Timing comparison (in second). C (solid black line) is for sparsebn. P(dashed blue line) is for pcalg, and M (dotted green line) is for bnlearn (Aragam et al., 2017)[36].

features. Sparsebn shows a great advantage on speed while processing large-scale dataset with even thousands nodes (Aragam et al., 2017)[36]. Comparing to other two popular package pcalg and bnlearn, it run much faster even when the number of nodes p increased greatly. Figure 3.2 shows the timing comparision of three package while processing continuous data.

Another feature sparsebn presents is its capability for dealing with data mixed with the observational and the experimental. Adding experimental interventions enhance the significance of observational DAGs and further make the true causal DAG more obvious. That means sparsebn has the power to improve the estimation of true causal DAG with experimental interventions involved.

CHAPTER 4

Data Treatment

The dataset consists of approximately 40 million expression measurements. The data is a bunch set of two color microarray. It looks like a black test board filling with color points(Figure 4.1). Each point represent a value of expression.

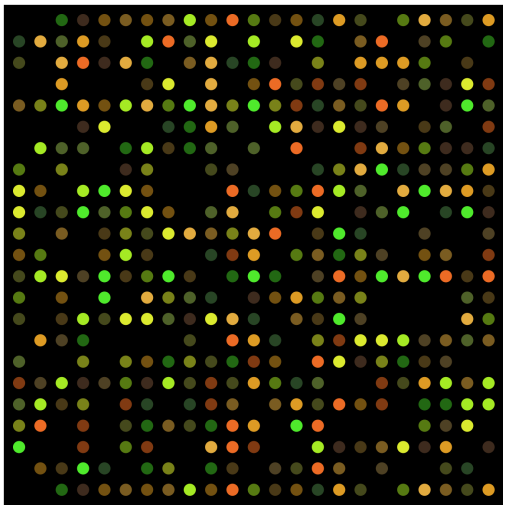

Figure 4.1: Two-color microarray

First read all those image format data in to R with limma package and transform them into a numerical form. As discussed in methodology part, dataset has to be done with background correction and normalization before any analysis. Use print-tip LOESS to perform the microarray data normalization on mean intensity and run in R with package named marray (Yang et al., 2002)[17]. And then use R package dyebias to correct the Gene-specific dye bias(GSDB) after the LOESS normalization. The correction consists of two parts: the intrinsic dye bias of a specific probe and the degree of dye bias observed in a specific slide.

The correction formula is : $M^*_{ij} = M_{ij} + GSDB_{ij} = M_{ij} + iGSDB_i \times F_j$

M^*_{ij} is the log_2 fold-change of gene i in hybridization j, which is biased and M_{ij} is the unbiased one. $GSDB_{ij}$ is the gene-specific dye bias component of M^*_{ij}. F_j is the slide bias of hybridization j. Then calculate the corrected log_2 fold-change M_{ij} with the formula $M_{ij} = M^*_{ij} - iGSDB_i \times F_j$.

After doing linear regression of micoarray, extract all statistical significant gene and create a new expression matrix for further analysis with Bayesian network. The expression matrix has 6170 columns each represents for interested deletion gene, and 1320 rows each for affect gene, and the filled up with expression values corresponding to each pair of genes.

CHAPTER 5

Analysis of Result

5.0.1 Limma Analysis

Each time a mutant was applied to replicate hybridizations, it was treated with two independent cultures and compared to WT culture, and they were all tested at the same day in order to control for day-specific effects. As the MATalpha mutants grown in the plate shaker with Tecan, the WT pool involved 200 MATalpha replicates. As the MATa mutants grown in the plate shaker of Tecan, the WT pool involved 20 WT-MATa replicates. As mutants grown within Erlenmeyers, the MATalpha WT pool involved 200 WT-MATalpha replicates. As MATa mutants grown within Erlenmeyers, the WT pool involved 8 WT MATa replicates.

Use R package limma to get the p-values after the Benjamini-Hochberg FDR correction. If the fold-change is greater than 1.7 and the p-value is less than 0.05, then we believe genes have a significant change after a certain mutant added. Though some changes are statistical significant, they are not considered as the change respective of the targeted mutant by applying the 200WT versus WT comparisons of the two large WT MATalpha replicate pools (ArrayExpress accession E-TABM-773 and E-TABM-984). Then omit the top 5 percent of expression outliers, the WT variable gene list is : AI1, AI2, AI4, AI5-ALPHA, AI5-BETA, ATP8, BIO3, BIO4, BIO5, BSC1, DDR2, FIT2, GLK1, GSY1, HSP12, HSP30, HSP42, HXK1, NCE103, OLI1, PHO84, PRM7, SOL4, SPL2, SRO9, STP4, TPS2, TSL1, VAR1, VTC3, YDL038C, YDR170W-A, YDR210C-C, YIG1, YJR154W, YKR075C, YNL284C-A, YOR343W-B, yrO2, ZEO1, AIM33, CTR1, GPD2, GPH1, PHO12, PKH2, RIF2, RTC3, RTC4, VTC1, YDL177C, YDR210W-B, YER053C-A, YFL002W-B, YMR046C, YPR158W-A, ZRT1. All those genes in the list would be removed for further analysis.

In order to determine a threshold to test whether mutants affected mRNA expression significantly, all deletion mutants and the WT cultures grown under the same condition were ranked according to the number of genes changed significantly. The percentage of WTs that had greater than 3 changing genes is less than 6 percent. With the help of this percentage, threshold was set as 4 which means if the deletion mutants had more than 4 changing genes, they were marked as responding, or they were marked as nonresponding. Based on those settings, Limma was applied for further analysis and got the analysis results.

After classification of responding mutants, all responding mutants were extracted and knitted as a gene expression matrix. Then used the gene expression matrix to plot a network graph which consisted of edges and nodes. Nodes means each kind of deletion mutant and edges are are the significant changing between each pair deletion mutants.

As the publisher did not provide the data of tow-color microarray, we can not redo the limma analysis. And getting some new result with Bayesian method is much meaningful than repeating past analysis. But according to the original paper, we still could have a glance of the result from original paper (Kemmeren et al., 2014)[35]. A brief result of original paper is attached below.

Other complicated motifs can be found in the regulatory system in addition to the nested effects. One category of these motifs is feed-forward loops (FFLs) which have eight subtypes. Previous studies do not provide enough information for large-scale analyses of these FFLs. Thus, the FFLs were identified from the genetic perturbation network. For example, the incoherent 2 FFL is recognized from the overlapping in genes (X and Y) with upregulation due to the deletion of two genes. In this case, one of the deleted genes can also be upregulated due to the deletion of the other gene. To assure that the chosen FFLs are the most consistent ones, only X-Y gene pairs with a considerable overlap in the downstream effective genes (Z) were included. For the following network sketching, distinct X-Y FFL pairs were only considered once instead of more than one times for every mutual downstream gene. By combining 1,120 X-Y FFL pairs, an easily comprehended network graph was generated with different colors indicating different subtypes of FFLs.

After the removal of the nested effects, the relative frequencies of 4 FFL types are calculated. The incoherent type 2 has the highest frequency compared to the other types. It also shows the quantity of the different families of genes that participate in the 4 FFLs types, and identifies them as the upstream (X) or the downstream (Y) gene. Regulators of chromatin are usually found as upstream participants in coherent FFL types. This implies that the cooperation with downstream participants could enhance the expression of certain genes. Additionally, a noteworthy finding is that the incoherent 2 FFLs showed higher activities in metabolic pathways. On one hand, a large portion of genes that participate in small-molecule metabolic pathways were identified as downstream Y nodes. On the other hand, downstream Y genes were also found to be highly represented in incoherent 2 FFLs.

5.0.2 Bayesian Networks Learning

For further analysis, we analyzed the gene expression matrix the publisher provided (Kemmeren et al., 2014) with sparsebn R package. The main procedure is represented below.

In order to use sparsebn package, we have to import the original data into a special frame in which related support information nested, which aims to provide the discernibility on different types of data. We can easily use rcode sparsebnData() on the raw data, but some components in this code have to be specified. The first one is setting the data type as "continuous" or "descrete", or it will be set as "discrete" as default. Another one is the interventions are extracted for each row (observation). After transformation, there are 1320 total rows with 1310 rows omitted.

```
R> ivn27 <- as.list(rownames(dat27))
R> dat <- sparsebnData(dat27, type = "continuous", ivn = ivn27)
R> dat
```

When data is loaded in a proper way, the next step is structure learning. In R, we usually use estimate.dag() based on the algorithm explained in methodology part. The rcode below is a simple example of Bayesian structure leaning with default parameters.

```
R> gene.learn <- estimate.dag(data = dat)
```

```
R> gene.learn
```

In terms of tuning, two main parameters are frequently adjusted to improve the model. One is the regularization parameter λ which can be set by length with lambdas.length() or by grid. There are two ways of setting grid, one is use linear scale, another is using log scale. A better grid of lambdas could improve the algorithm obviously as discussed in methodology is also explained in methodology part.

```
R> scalelambdas <- generate.lambdas(lambda.max = 10,
        lambdas.ratio = 0.001, lambdas.length = 10, scale = "linear")
R> scalelambdas

 [1] 10.00  8.89  7.78  6.67  5.56  4.45  3.34  2.23  1.12  0.01

R> scalelambdas <- generate.lambdas(lambda.max = 10,
        lambdas.ratio = 0.001, lambdas.length = 10, scale = "log")
R> scalelambdas

 [1] 10.00000000  4.64158883  2.15443469  1.00000000  0.46415888
 [6]  0.21544347  0.10000000  0.04641589  0.02154435  0.01000000
```

Another parameter is the threshold of edges, which give the algorithm a threshold which once touched by the number of edges, the algorithm will stop immediately and provide the result. Considering these two parameter, we got an improved result by setting these two parameters with appropriate numbers.

```
R> BN<- estimate.dag(data = dat,
        lambdas = scalelambdas, edge.threshold = 2*ncol(dat27))
```

Sometimes, researchers may know the relationship between a certain pair of genes prior the analysis. Then they can use whitelists() to link these tow genes while doing the structure.

On contrast, they can also set a certain pair of genes unlinked with blacklists() if they was known to have no relationship.

The output of the structure is a collection of solution paths. We can easily use index to view a path graph. For example, the third index graph shows that there is 6170 nodes and 6167 edges. We even can use show.parents() to show parents of certain child genes.

```
R> BN[[3]]

CCDr estimate
1320 observations
lambda = 2.15443469003188
DAG:
Directed graph with 6170 nodes and 6167 edges.
```

However, all result we got above are only structure paths involving path graphs. The next step is to find the parameters with estimate.parameters(). The output is a list combining weighted adjacent matrices of coefficients and diagonal matrices of conditional variances. Similar to the solution path, we still can use index to view detail results. Then we can use select.parameter()t to get the optimal parameter with the output of an index number. For this data, the best case is the 4th. By using the index number 4 we will get the optimal matrix of parameters which is omit to show here as it's too big like 6170 x 6170.

```
R> param<- estimate.parameters(BN, data = dat)
R> optimallambda<- select.parameter(BN, dat)
R> param[[optimallambda]]$coefs
```

Finally plot the graph of solution paths with the optimal parameters. Used the rcode below and got the plot shown in Figure 5.1.

```
R> plot(BN[[optimallambda]], vertex.size = 2, edge.lwd = 0.1,
vertex.label = NA, layout=layout_components)
```

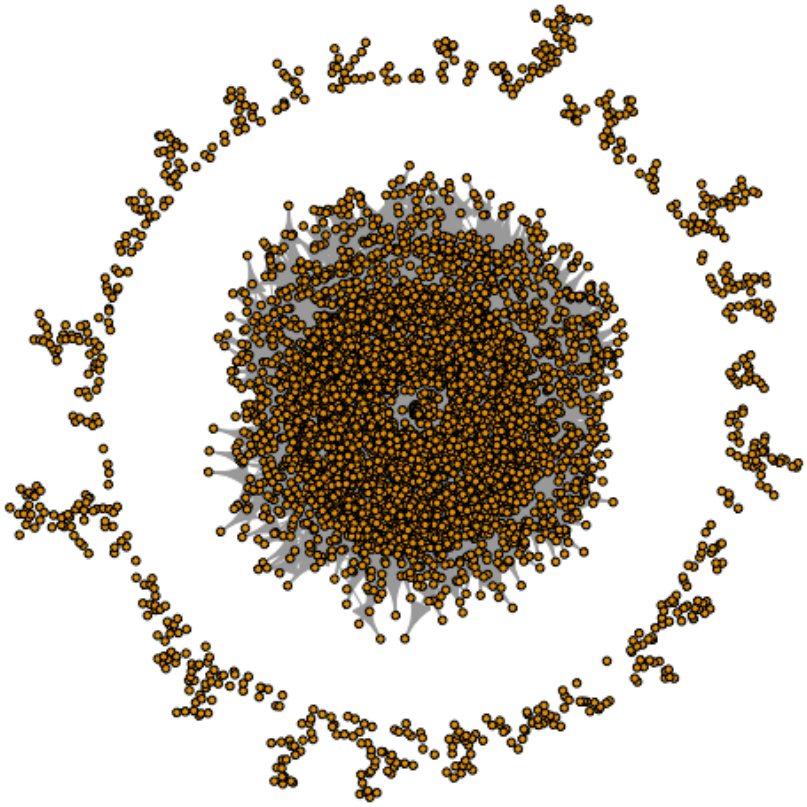

Figure 5.1: Bayesian Network

CHAPTER 6

Discussion

The data after treatment would be a good tool to analyze various properties of mRNA expression, the regulatory, and the genetic perturbation network. In addition to the properties of the network, the regulatory networks are highly interrelated, suggested by the number of straight and cyclic pathways, nested effects, and FFLs. These results are consistent with a number of previous studies on the patterns of protein and genetic interactions (Breitkreutz et al., 2010)[18]. The gene types having effects on transcription related to viability, the pattern of various types of FFLs in the network, and the properties of nonresponsive deletions were revealed in this study. The perturbation signatures showed the complexity of the structures of protein complexes and pathways. This result provides an insight into complex gene expression studies in the future.

In this study, gene-specific transcription factors (GSTFs) were thoroughly analyzed. The results showed a surprisingly high abundance of gene-specific repressors. This was unexpected since some studies suggested that the majority of eukaryotic GSTFs were activators (Fuda et al., 2009[19]). The results showed that these repressors have a high prevalent activity. This is consistent with the distributive transcription of the eukaryotic genome (David et al., 2006[20]), suggesting that the chromatin is not solely restricting to transcription. These results agree with previous studies which indicated that transcription often requires factors to avoid unwanted gene expression (Spitz and Furlong, 2012)[21]. The repression of transcription by the actions of GSTFs may also lead to upregulation of certain genes due to the inactivation of their repressors. It is unclear whether the abundance of gene-specific repressors suggests that gene-specific activators are unnecessary for transcription in eukaryotes. It is possible that the activators are not always needed. It would be useful to analyze GSTFs

together with other classes of regulators (e.g., chromatin factors) in the future. A general classification of GSTFs has not been attempted. Previous large-scale data also need to be revisited and revised. The modified genetic location data and studies revealing the correlation between activators and repressors will likely provide more information into building a classification system.

Overall, the data indicated that unexpected results can be observed from genetic perturbation. Future studies on characterizing uniformly acquired perturbation datasets can be conducted. It would be critical to measure, analyze and classify the effects of each individual gene deletion, and integrate these data into the overall picture of a large-scale genetic perturbation.

REFERENCES

[1] T. Ideker, T. Galitski, L. Hood *A new approach to decoding life: systems biology* Annu. Rev. Genomics Hum. Genet., 2 (2001), pp. 343-372.

[2] A.J.M. Walhout, M. Vidal *Protein interaction maps for model organisms* Nat. Rev. Mol. Cell Biol., 2 (2001), pp. 55-62.

[3] M. Costanzo, A. Baryshnikova, J. Bellay, Y. Kim, E.D. Spear, C.S. Sevier, H. Ding, J.L.Y. Koh, K. Toufighi, S. Mostafavi, et al. *The genetic landscape of a cell* Science, 327 (2010), pp. 425-431.

[4] J.L. DeRisi, V.R. Iyer, P.O. Brown *Exploring the metabolic and genetic control of gene expression on a genomic scale* Science, 278 (1997), pp. 680-686.

[5] C.J. Roberts, B. Nelson, M.J. Marton, R. Stoughton, M.R. Meyer, H.A. Bennett, Y.D. He, H. Dai, W.L. Walker, T.R. Hughes, et al. *Signaling and circuitry of multiple MAPK pathways revealed by a matrix of global gene expression profiles* Science, 287 (2000), pp. 873-880.

[6] J. van de Peppel, N. Kettelarij, H. van Bakel, T.T.J.P. Kockelkorn, D. van Leenen, F.C.P. Holstege *Mediator expression profiling epistasis reveals a signal transduction pathway with antagonistic submodules and highly specific downstream targets* Mol. Cell, 19 (2005), pp. 511-522.

[7] T.R. Hughes, M.J. Marton, A.R. Jones, C.J. Roberts, R. Stoughton, C.D. Armour, H.A. Bennett, E. Coffey, H. Dai, Y.D. He, et al. *Functional discovery via a compendium of expression profiles* Cell, 102 (2000), pp. 109-126.

[8] Z. Hu, P.J. Killion, V.R. Iyer *Genetic reconstruction of a functional transcriptional regulatory network* Nat. Genet., 39 (2007), pp. 683-687.

[9] G. Chua, Q.D. Morris, R. Sopko, M.D. Robinson, O. Ryan, E.T. Chan, B.J. Frey, B.J. Andrews, C. Boone, T.R. Hughes *Identifying transcrip- tion factor functions and targets by phenotypic activation* Proc. Natl. Acad. Sci. USA 103, 12045-12050

[10] Matthew E. Ritchie, Belinda Phipson, Di Wu, Yifang Hu, Charity W. Law, Wei Shi and Gordon K. Smyth *limma powers differential expression analyses for RNA-sequencing and microarray studies* Nucleic Acids Research (2015), pp. e47.

[11] Spirtes P, Glymour C *An Algorithm for Fast Recovery of Sparse Causal Graphs* Social Science Computer Review (1991), pp. 62-72.

[12] Uhler C, Raskutti G, Bu hlmann P, Yu B *Geometry of the Faithfulness Assumption in Causal Inference* The Annals of Statistics (2013), pp. 436-463

[13] Zhang J, Spirtes P *Strong Faithfulness and Uniform Consistency in Causal Inference* In Proceedings of the nineteenth conference on uncertainty in artificial intelligence (2002), pp. 632639.

[14] Koller D, Friedman N ,Probabilistic Graphical Models: Principles and Techniques. MIT press (2009).

[15] Cooper GF, Herskovits E *A Bayesian Method for the Induction of Probabilistic Networks from Data* Machine Learning (1992), pp. 309-347.

[16] G'amez JA, Mateo JL, Puerta JM *Learning Bayesian Networks by Hill Climbing: E cient Methods Based on Progressive Restriction of the Neighborhood* Data Mining and Knowledge Discovery (2011), pp. 106-148.

[17] Y.H. Yang, S. Dudoit, P. Luu, D.M. Lin, V. Peng, J. Ngai, T.P. Speed *Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation* Nucleic Acids Res., 30 (2002), p. e15.

[18] A. Breitkreutz, H. Choi, J.R. Sharom, L. Boucher, V. Neduva, B. Larsen, Z.-Y. Lin, B.-J. Breitkreutz, C. Stark, G. Liu, et al. *A global protein kinase and phosphatase interaction network in yeast* Science, 328 (2010), pp. 1043-1046.

[19] N.J. Fuda, M.B. Ardehali, J.T. Lis *Defining mechanisms that regulate RNA polymerase II transcription in vivo* Nature, 461 (2009), pp. 186-192.

[20] L. David, W. Huber, M. Granovskaia, J. Toedling, C.J. Palm, L. Bofkin, T. Jones, R.W. Davis, L.M. Steinmetz *A high-resolution map of transcription in the yeast genome* Proc. Natl. Acad. Sci. USA, 103 (2006), pp. 5320-5325.

[21] F. Spitz, E.E.M. Furlong *Transcription factors: from enhancer binding to developmental control* Nat. Rev. Genet., 13 (2012), pp. 613-626.

[22] T.R. Hughes, C.G. de Boer *Mapping yeast transcriptional networks* Genetics, 195 (2013), pp. 9-36.

[23] M. Bonke, M. Turunen, M. Sokolova, A. Vhrautio, T. Kivioja, M. Taipale, M. Bjrklund, J. Taipale *Transcriptional networks controlling the cell cycle* G3 (Bethesda), 3 (2013), pp. 75-90.

[24] Efron,B. and Morris,C *Steins estimation rule and its competitorsan empirical Bayes approach* J. Am. Stat. Assoc., 68 (1973), pp. 117-130.

[25] Smyth,G *Linear models and empirical Bayes methods for assessing differential expression in microarray experiments* Stat. Appl. Genet. Mol. Biol., 3(2004), Article 3.

[26] Phipson,B., Lee,S., Majewski,I.J., Alexander,W.S. and Smyth,G.K. *Empirical Bayes in the presence of exceptional cases, with application to microarray data.* Technical Report. Bioinformatics Division, Walter and Eliza Hall Institute of Medical Research, Melbourne, Australia (2013).

[27] Ritchie,M., Diyagama,D., Neilson,J., Van Laar,R., Dobrovic,A., Holloway,A. and Smyth,G. *Empirical array quality weights in the analysis of microarray data* BMC Bioinformatics, 7 (2006), pp. 261.

[28] Wu,D., Lim,E., Vaillant,F., Asselin-Labat,M., Visvader,J. and Smyth,G *ROAST: rotation gene set tests for complex microarray experiments* Bioinformatics, 26(2010), pp. 2176-2182.

[29] Spirtes P, Glymour C *An Algorithm for Fast Recovery of Sparse Causal Graphs* Social Science Computer Review, 9(1991), pp. 62-72.

[30] Uhler C, Raskutti G, Bu hlmann P, Yu B *Geometry of the Faithfulness Assumption in Causal Inference* The Annals of Statistics, 41(2013), 436-463.

[31] Zhang J, Spirtes P *Strong Faithfulness and Uniform Consistency in Causal Inference* In Proceedings of the nineteenth conference on uncertainty in artificial intelligence (2002), pp. 632-639.

[32] Koller D, Friedman N *Probabilistic Graphical Models: Principles and Techniques* MIT press (2009).

[33] Cooper GF, Herskovits E *A Bayesian Method for the Induction of Probabilistic Networks from Data* Machine Learning, 9 (1992), 309-347.

[34] Friedman J, Hastie T, Tibshirani R *Regularization Paths for Generalized Linear Models via Coordinate Descent* Journal of statistical software, 33(2010), pp. 1.

[35] Kemmeren P, Sameith K, van de Pasch LA, Benschop JJ, Lenstra TL, Margaritis T, ODuibhir E, Apweiler E, van Wageningen S, Ko CW, van Heesch S *Large-scale genetic perturbations reveal regulatory networks and an abundance of gene-specific repressors* Cell, 157(2014), pp.740-752.

[36] Aragam B, Gu J, Zhou Q. /textitLearning Large-Scale Bayesian Networks with the sparsebn Package arXiv preprint arXiv:1703.04025 (2017).

[37] Fu, F. and Zhou, Q. /textit Learning sparse causal Gaussian networks with experimental intervention: Regularization and coordinate descent Journal of the American Statistical Association, 108: 288-300(2013).

[38] Aragam, B. and Zhou, Q. *Concave penalized estimation of sparse Gaussian Bayesian networks* Journal of Machine Learning Research, 16: 2273-2328 (2015).

[39] Gu, J., Fu, F., and Zhou, Q. *Penalized estimation of directed acyclic graphs from discrete data* Statistics and Computing, DOI: 10.1007/s11222-018-9801-y (2018).