Inference of Gene Regulatory Network (GRN) From Gene Expression Data Using K-means Clustering and Entropy Based Selection of Interactions

April, 2017

Undergraduate Thesis

Department of Computer Science and Engineering

BRAC University

Declaration

We, hereby declare that this thesis is based on the results found by ourselves. Materials of work found by other researchers are mentioned by reference. This thesis, neither in whole nor in part, has been previously submitted for any degree.

Signature of Students

Asadullah Al Galib

Mohammad Mohaimanur Rahman

Signature of Supervisor

Dr. Md. Haider Ali

Signature of Co-Supervisor

Eusra Mohammad

Abstract

Inferring regulatory network from gene expression data only is considered a challenging task in systems biology, and the introduction of various high-throughput DNA microarray technologies in the collection of expression data has significantly increased the amount of data to be analyzed by existing algorithms. All of these algorithms focus on different issues regarding the inference of gene regulatory network (GRN) and their methodologies work better only for certain types of datasets and/or regulatory networks. As a result, they have inherent limitations in dealing with different types of datasets. In this paper, we propose a novel method to infer gene regulatory network from expression data which utilizes *K-means Clustering* along with some properties of entropy from information theory. The proposed method has two main components, first grouping the genes of a dataset into given number of clusters and then finding statistically significant interactions among genes of each individual cluster and selected nearby clusters. To achieve this, an information theoretic approach based on *Entropy Reduction* is used to finally generate a regulatory interaction matrix consisting of all genes. The purpose of grouping genes in clusters based on the similarity of expression level is to minimize the search space of regulatory interactions among genes. The Entropy Reduction Technique (ERT) finds regulatory interactions with reduced number of genes. To assess the performance of our algorithm, we used datasets from *DREAM5 – Network Inference challenge* [6], *DREAM4 – In Silico Network challenge* [7] and one *in silico* dataset generated by *GeneNetWeaver* [8]. The performance of our algorithm was compared with the result of *ARACNE,* a popular information theoretic approach to reverse engineer gene regulatory network from expression dataset. We used precision and recall as performance measures. Our algorithm showed significant improvement in the precision and recall percentage over the network generated by ARACNE. We also compared our results among different threshold values and different numbers of clusters with three versions of our algorithm -*No Clustering*, *Unmerged Clustering* and *Selected Merged Clustering*.

Acknowledgements

We would like to thank our supervisor Professor Dr. Md. Haider Ali, for his invaluable support, encouragement in our work and the freedom that he allowed us to explore different concepts and techniques to finish our work. Whenever we had any difficulties with a concept or an implementation, we always found the solutions from his vast knowledge and experience in the field. Without his continuous guidelines, we would never be able to finish our work in time. We would also like to thank our co-supervisor Ms. Eusra Mohammad for her valuable advice and guidelines regarding the biological aspects of our thesis. We would like to express our gratitude also to Md. Khaledur Rahman from Department of CSE, United International University, for his tremendous help in starting our work in this topic.

Table of Contents

List of Figures:

Results for Dataset - 3 (GNW) 46

Chapter 1

Introduction

In our approach to infer gene regulatory network, we focused on merging a very powerful gene expression analysis technique, clustering with the Entropy Reduction Technique (ERT) from information theory in an attempt to achieve better performance than existing information theoretic approach such as *ARACNE* (Algorithm for the Reconstruction of Accurate Cellular Networks). The goal was to merge these two techniques to be able to deal with large datasets without being provided any information regarding the type of experimental conditions in which the expression levels of genes were measured. With the knowledge of regulatory network and the role that each gene plays in that network, fields such as drug discovery and personalized medicine can be revolutionized.

1.1 Gene Regulatory Network

Genes of a biological system do not act independently of each other. They work in a complex regulatory network with other genes and gene products such as RNA and protein to control the expression level of certain genes in the network determined by various cellular conditions. Gene regulatory networks (GRN) are part of the biological networks of a biological system. Various essential tasks which a biological system has to execute in order to survive and adapt to different types of environmental conditions are governed by proteins it creates which are gene products. The synthesis of proteins is controlled by other gene or a group of genes which are in the same biological system. That creates a complex biological network consisting of genes, RNAs and proteins in which the expression level of one is control by other, either through activation or inhibition regarding the expression level. Two genes can be considered connected by a regulatory interaction link if the expression level of one influences the expression level of the other. Here for our approach we use the microarray gene expression datasets provided by DREAM5 and DREAM4 challenges to infer transcriptional regulatory network solely from the datasets without using any domain knowledge or information about the type of experiments or the conditions of different types.

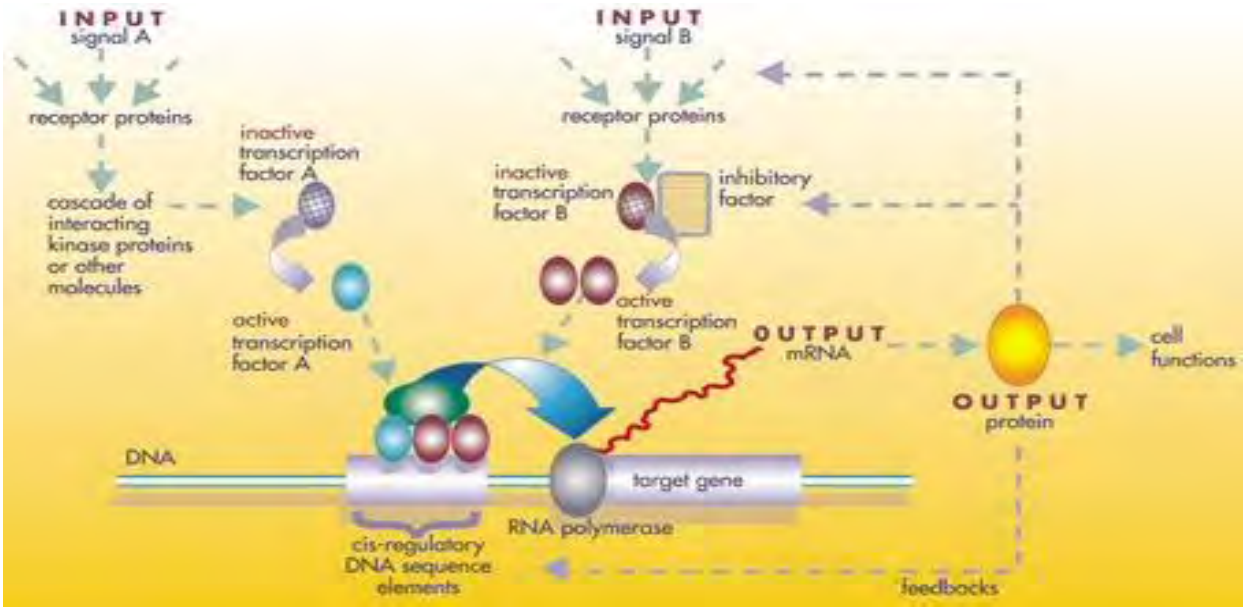

Figure 1.1: Structure of Gene Regulatory Network [16]

1.2 Clustering of Gene Expression Data

As a useful data mining technique, clustering has been used in the context of gene expression data to identify grouping that exists in the data and also to find hidden patterns among data points. Because of its effectiveness in finding unknown patterns in a dataset, various clustering techniques have been used for gene expression data analysis to handle large amount of biological data made possible by new DNA microarray technologies.

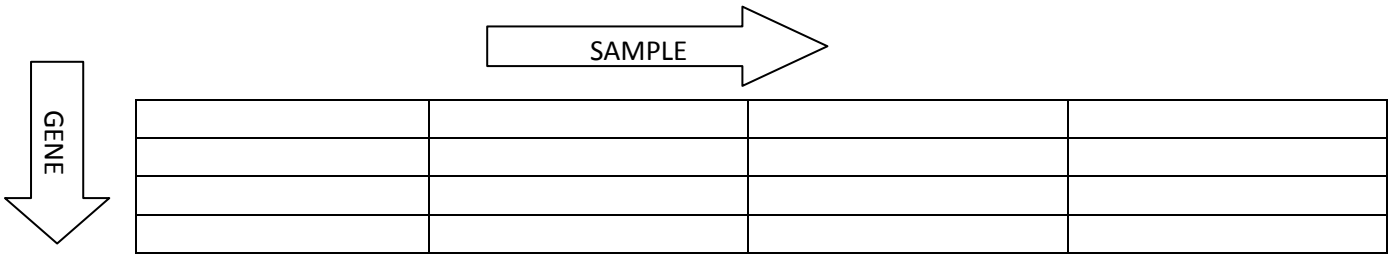

Figure 1.2: Gene expression data matrix table (row = genes & column = samples)

Common clustering algorithms used for gene expression data includes k-means clustering, hierarchical clustering, self-organizing map (SOM), graph-theoretical approaches [10]. In our algorithm, we have used k-means clustering because of its simplicity and ability to cluster large datasets containing 4000 to 5000 genes along with hundreds of samples in an efficient manner to be later used with our Entropy Reduction Technique (ERT). We used a function that uses the *Elbow Method* to find the optimal value for number of clusters k, as it is required for k-means clustering. The function clusters the data from k=2 to 100 and generates a plot by having number of clusters, k in the x-axis and the corresponding within cluster *sum of squared errors* (WSSE)in the y-axis to identify the value of k for which the reduction in WSSE is highest for lower cluster numbers.

We will discuss k-means clustering in details in chapter 2 and also define some other clustering algorithms mentioned above.

1.3 Inference of Gene Regulatory Network

The advent of high throughput microarray technologies enables us to analyze all the genes of the whole genome under various experimental conditions. One important assumption in systems biology is that if Gene A's expression level changes in proportion to Gene C's expression level, it can be assumed that a regulatory link exists between the two. But one limitation with this assumption is that, the genes which are not directly related but have a high co-expression level due to indirect relationship through another gene are also considered to have a regulatory link

between them. Different algorithms from a wide range of theoretical concepts have tried to separate correlation from causation regarding the expression levels of genes. Various models have been developed for the task of inferring regulatory networks, such as, models based on discrete variables which include methods like *Boolean Network Model, Probabilistic Boolean Network Model and Bayesian Network Model* and models which are based on continuous variables which include *Differential Equation Model.* Another type of model utilizes various information theoretic approaches such as *ARACNE* (Algorithm for the Reconstruction of Accurate Cellular Networks) and *CLR* (Context Likelihood of Relatedness) to determine true regulatory interactions among genes which are statistically significant and also to eliminate false positives.

The regulatory network can be represented as a graph G, where genes are represented as nodes and interactions among genes are represented as edges.

The fundamental process of inferring gene regulatory network can be described in the following way –

Input: Matrix M of dimension a x b, where a is the number of genes and b is the number of experiments or samples. Each cell defined by M (i,j) contains the expression level of i-th gene in j-th experiment.

Algorithm: A network inference algorithm which takes the matrix M as input and after finding regulatory relationships among genes, it returns a square matrix denoting regulatory relationship among genes either by a simple Boolean value or a weight value of the edge between two genes.

Output: Matrix O of dimension a x a, where O (i,j) indicate whether Gene i and Gene j has a regulatory relationship.

Because of the good performance of Information Theoretic approaches in DREAM5 – Network Inference challenge [6], we wanted to use a relatively new and less used entropy reduction technique with clustering to get more efficient result than *ARACNE* in inferring gene regulatory network. Our algorithm generates a connection matrix as output after using the predefined threshold value to eliminate edges between genes where the reduction of entropy is less than the threshold value. As we do not use any external information about experiment types or conditions and also to compare our algorithm with *ARACNE* easily, which is also a direction neutral algorithm due to its dependency on mutual information, we generate a network with only regulatory interactions among genes without any directionality in their edges. In chapter 3, we will discuss various network inference algorithms in details.

1.4 Entropy Reduction (ER)

We use entropy reduction approach in our algorithm for the purpose of determining statistically significant regulatory interactions among genes. In *Information Theory*, proposed by Shannon, Entropy is a fundamental concept. It can be defined as the measurement of uncertainty of a random variable [2].

$$\boldsymbol{H(x) = -\sum_{x \in \chi} p(x) \log p(x)}$$

Where H is the entropy, x is a discrete random vector with alphabet X, and p(x) is the probability mass function.

Entropy is very closely related to Mutual Information (MI), which is the measurement of amount of information about one random variable that is contained in another variable. So it reduces the uncertainty of one variable given that the information about another variable is provided [2]. In the biological context, if two genes have a regulatory interaction among them, then the mutual information between those two genes will be high. On the other hand, if two genes act independently in the biological process, they will have a mutual information measure close to zero.

The main component of entropy reduction technique is, if a variable A shares a regulatory link with another variable B, then

$$\mathbf{H(A|B) < H(A)}$$

Where, H (A|B) is the Conditional Entropy of A given B and H (A) is the Entropy of A [2].

Entropy Reduction Technique (ERT) is a relatively new approach to be applied to the task of inferring biological networks. The implementation of Entropy Reduction presented in this paper [2], focused mainly on small biological networks. To be able to apply ERT in the context of large datasets, we used clustering algorithm to minimize the search space so that ERT then can be applied on smaller group of genes in an efficient way.

1.5 Our Contribution

We are proposing a novel approach to infer gene regulatory network that combines clustering of genes with Entropy Reduction Technique to make this effective idea applicable on large datasets. We evaluated the performance of our algorithm using two datasets from DREAM5-Network Inference Challenge, one dataset from DREAM4-In-Silico Network Challenge and one In-Silico dataset of 1000 genes generated by GeneNetWeaver software. To assess the performance of our algorithm we used Precision, Recall and Area under the Precision-Recall (PR) Curve. We also used the DPI technique from ARACNE on the final connection matrix generated by our algorithm to reduce the false positives and also to verify whether integrating DPI technique with our algorithm brings significant changes in the result. We compared our result with the regulatory network generated by ARACNE and found much higher recall rate than ARACNE. We have also compared results generated from No Clustering, Unmerged Clustering and Selected Merged Clustering version of our algorithm to assess the effect of clustering in the regulatory networks. Even though No-Clustering version was the most effective one in determining regulatory interactions among genes, Selected Merged Clustering version also performed well across all datasets. We have also made a comparison among results obtained from different threshold values used after the ERT step to eliminate less significant interactions.

1.6 Organization of the Report

In this chapter, we have discussed what gene regulatory network is, how gene expression data can be clustered, basics of inferring gene regulatory network, basics of entropy reduction technique and our contribution in the inference of regulatory network. In chapter 2, we discuss various clustering algorithms in details including the K-means clustering algorithm that we are using in our approach. Chapter 3 deals with different supervised, semi supervised and unsupervised approaches used to infer gene regulatory networks. In chapter 4, we explain various information theoretic approaches to infer regulatory networks including the ERT algorithm. In chapter 5, we explain both the clustering part and ERT part of our algorithm in details. The results obtained from the algorithm and various comparisons with ARACNE are presented in chapter 6. In chapter 7 we discuss the results presented in chapter 5. And finally, in chapter 8 conclusions are drawn and plans of future improvements are mentioned.

Chapter 2

Clustering of Gene Expression Data

The main goal of clustering is to group data points which are similar into same cluster from a set of clusters and dissimilar data points into a different cluster. How the similarity between data points is defined is very important for the performance of different clustering techniques. In the context of genetics, the similarity measure can be the similar expression or co-expression level of genes [10]. If Gene A and Gene B are grouped in the same cluster based on expression level, then it can be deduced that they are part of the same biological process. So given the knowledge of functions of one gene, important functional information regarding another gene which was previously unknown can be obtained based on the clustering of genes. Moreover, strong co-expression level among genes also suggests co-regulation [10].

2.1 Clustering Algorithms

Clustering algorithms can be divided into two primary subgroups - hierarchical clustering and partitioning clustering. Hierarchical clustering and some other commonly used biological clustering are described below –

2.1.1 Hierarchical Clustering

It is a graphical representation of hierarchical series of Nested Cluster by a tree (*dendrogram)* which provides the formation and the specified numbers of cluster at any moment of clustering. Genes are represented as object on rows to find co-regulated and functionally related genes and samples are as features on columns to find sub types of related sample in gene base clustering. Sometimes two way clustering is used to identify the most important genes by combining sample based and gene based clustering. Hierarchical clustering algorithms has two approaches according to the formation of dendrogram, they are Agglomerative algorithms (bottom-up approach), Divisive algorithms (top-down approach). According to Agglomerative approach each objects form it's on cluster. Then merge their closest pair until one cluster (k cluster) using proximity Matrix. For updating proximity matrix different methods like single linkage, complete linkage, average linkage, centroid linkage are used which define the distance between clusters. In single linkage, minimum distance between two genes of two clusters and minimum spanning tree (MST) are used for finding closest pair till one cluster remains [12].

$$\boldsymbol{d(C_i, C_j) = min_{x \in C_i y \in C_j}\{d(x, y)\}}$$

Maximum distance of genes in different cluster is used for complete linkage

$$\boldsymbol{d(C_i, C_j) = max_{x \in C_i y \in C_j}\{d(x, y)\}}$$

And average of maximum and minimum for average linkage.

$$\boldsymbol{d(C_i, C_j) = avg_{x \in C_i y \in C_j}\{d(x, y)\}}$$

Its space complexity is O (N^2) where N is the number of points and time complexity is O (N^3) where N is steps and step size is N^2. Because of proximity matrix its complexity is high but it can be reduced to O (N^2 log (N)). Its advantage is it does not require the cluster number to before

because it computes. Without giving input parameters it computes a complete hierarchy of clusters [12].

2.1.2 Self-Organizing Map (SOM)

A self-organizing map (SOM) is a type of artificial neural network (ANN) which produce lower dimensional grid of neighborhood stricture which represent mapped data point of input output neurons with the help of closest reference vector [10]. For making the automatic map, inputted data are trained through a competitive process. As a result we found the trained reference vectors having the direction towards denser areas. This training process of neurons makes SOM more effective approach than K-means clustering in highly noisy data. This process needs two parameter like cluster number and grid structure map of the neuron map. SOM cannot do the effective identification of irrelevant data point merged into few clusters like one or two because vast majority cluster populate the data.

2.1.3 Graph Theoretic Approach:

It is a technique where graph theory like proximity graphics used to solve the problem of clustering a dataset by retrieving minimum cut or maximal cliques [10].Normally, the connections with an edge of pair of objects are measured by proximity. For some method connection with an edge is measured only when the proximity is 1 and else not because it is mapped as 0 or 1on the basis of some threshold.

2.1.4 K-means Clustering

K-means clustering falls into the subgroup of clustering called Partitioning Clustering, which is a clustering technique where each data point belongs to only one of the non-overlapping groups or clusters. It is a very simple and fast unsupervised clustering technique. The most essential element of k means is the value of k or number of clusters.

Basic K-means algorithm or Lloyd's Algorithm [11]:

Input: Dataset A where rows contain data points and columns contain features or variables, the number of clusters K.

Algorithm:

1. Initially choose K random points as Centroids for clusters.

2. Using a proximity measure between data points and Centroids, assign each data point to its nearest Centroid and thus group data points into K number of clusters.

3. Find the new Centroid (e.g. mean) for each formed cluster from its data points.

4. Repeat steps 2 to 3 on updated Centroids and assignments of data points until Centroids do not change or number of maximum iterations is exceeded.

Another important element for K-means algorithm is the proximity measure which is used to calculate the nearest Centroid for each data points. The proximity measure varies for different dataset types and also the goal of clustering on that dataset. For data points in Euclidean space, Euclidean distance can be used as the proximity measure [11].To measure the quality of the clustering, Sum of Squared Error (SSE) is computed for a clustering of data points [11]. Given a clustering, the Squared Error or Euclidean distance is calculated for each data points and a sum of all the squared errors is calculated. Because the initial Centroids are chosen randomly, given the algorithm, K-means can converge for local minimums. So for a given dataset, multiple run of K-means for same number of clusters, the SSE is calculated for each clustering and the version of clustering has the smallest value of SSE is chosen to have a better representation of the clustering of data points [11].

$$\boldsymbol{SSE} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{K}} \sum_{\boldsymbol{X \in C_i}} \boldsymbol{dist}(\boldsymbol{C_i}, \boldsymbol{X})^{\boldsymbol{2}}$$

Where dist is the Euclidean distance between two points in Euclidean space and C_i is the centroid of i-th cluster which is defined by,

$$\boldsymbol{c_i = \frac{1}{m_i} \sum_{x \in c_i} x}$$

Depending on the value of k, the similarity or distance function used in the algorithm and the initial random points chosen as cluster centers, the performance of K-means varies significantly.

For our algorithm, we used Lloyd's version of K-means with twenty runs for each value of K to find the minimum SSE where different initial Centroids are chosen for each run. To find the optimal value of K for a given dataset, we use a function to run K-means algorithm on that dataset for K=2 to K=100, and plot a within cluster SSE against the number of clusters to identify the maximum reduction of SSE for a smaller value of K which is known as the Elbow Method.

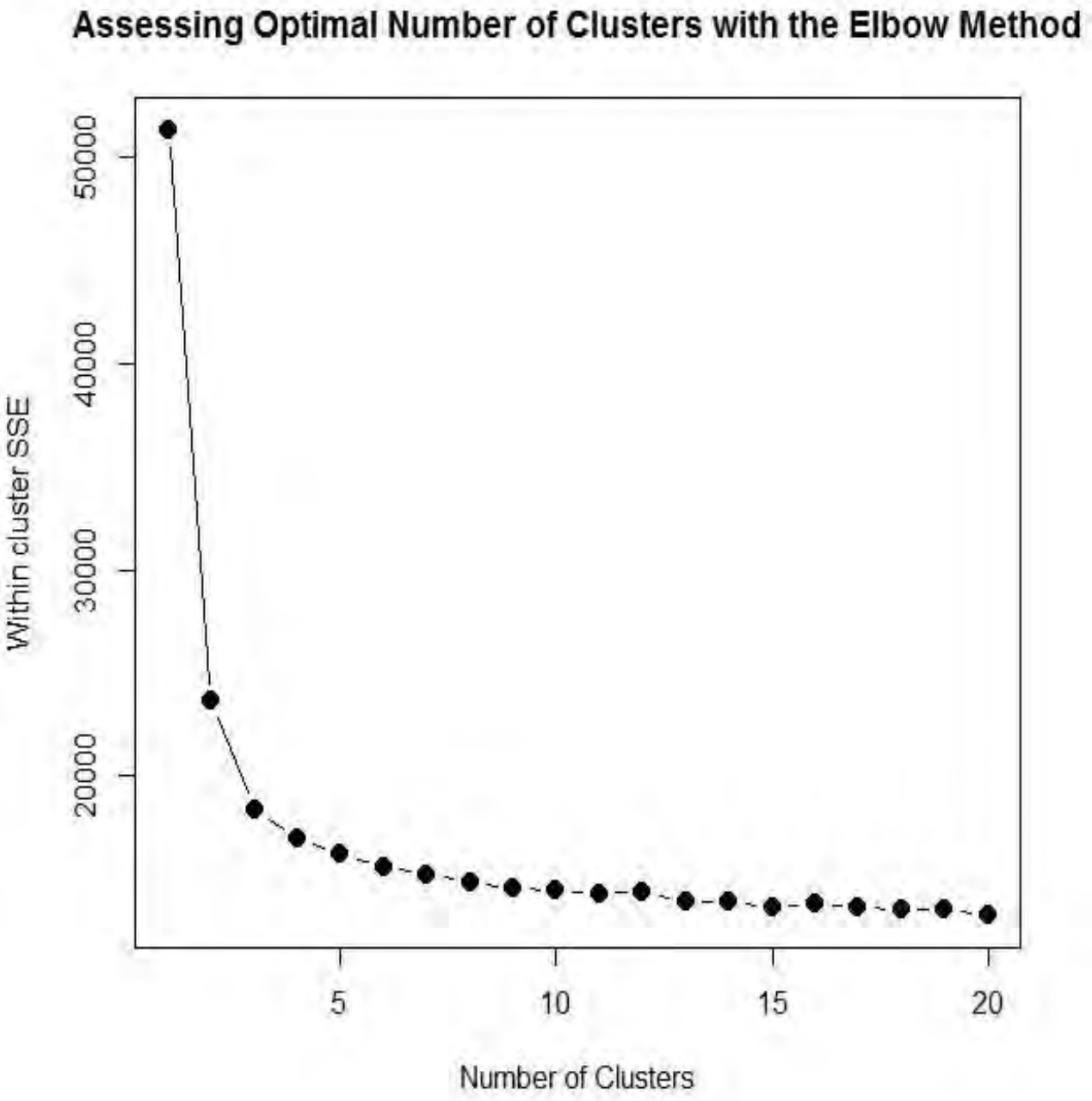

Figure 2.1: Elbow Method to find the optimal value of K

Gene expression data can be clustered in two ways – i) by row (gene), to cluster genes in different groups and treating samples as features and ii) by column (samples), to cluster samples and treating genes as features. Both of them have their practical purposes. For our method we used row-based or gene-based clustering to group genes by their similar co-expression levels.

We used three different versions of clustering together with the ERT step, to be compared for their effects on the efficiency of the ERT process to use the grouped data to find correct regulatory interactions. The versions are described below:

- No Cluster Version: In this version, we did not cluster the datasets; instead we ran the ERT step on the entire dataset to find the true regulatory interactions among genes. Given the complexity of the ERT step, it took more time to finish generating the connection matrix than the other two versions.
- Unmerged Version: For this version, we clustered the datasets into given number of clusters and ran ERT on genes of each cluster in a separate way. Then we merged the connection matrix returned from running ERT on each cluster and merged them to generate the n x n connection matrix where n is the number of genes. In this version, the genes that had true regulatory interactions with other genes which were in a different cluster were not identified.
- Selected Merged Version: With this version, after running the unmerged version of the algorithm, an additional merging among "close" clusters was carried out. We calculated which clusters were close to a given cluster by first finding the Euclidean distance of the nearest cluster and then multiplying the distance by two and finally identifying which other clusters' distance is less than the multiplied distance. We did the merging of all clusters with close clusters in the described way and finally merged all the connection matrix into a n x n connection matrix where n is the number of genes. We used this version for the clustering part of the algorithm.

One limitation of K-means is that it clusters data points better when the natural grouping of the data points is globular. From the theoretical knowledge of the structure of gene regulatory network, we decided to use K-means for the clustering purpose.

Chapter 3

Inference of Gene Regulatory Network

The main goal of Inference of Gene Regulatory Network is to predict which gene is responsible for the regulation of other genes. It also provides deep knowledge of biological process of regulatory network. This inference can be done by learning from dataset. Different types of methods are used for inferring regulatory networks. Some of them are good for specific type of genes but for another type of data their accuracy is low. Day by day different type of modification is implemented to improve the algorithms of inference. There are three types of learning methods - they are supervised, semi supervised and unsupervised learning methods.

3.1 Supervised Algorithms

A supervised learning is a machine learning process where algorithm infers function by analyzing from training data to use in new example mapping. It can give good accuracy for the same type of training among the various types of algorithm **Support Vector Machines** (**SVM)** is used successfully for making the inference of gene regulatory network [14].

3.1.1 Support Vector Machines (SVM)

SVM needs to calculate optimization problem over Lagrange multipliers α where ones feature vector is determined by another [5]. After finding the multipliers a feature vector can be added by measuring the signed distance to decision boundary where distance can provide confidence Value. SVM are trained by maximizing a constrained

$$\max_{\alpha} L(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j \gamma_i \gamma_j x^T{}_i x_j$$

For making feature vector outer product is used because it is communicative as a result predicted interactions are therefore symmetric and undirected. +1 and -1 is used to determine whether it is interacted or not. Prediction will be unnecessary if we know all the interactions of network that finds from feature vector. After all the process accuracy is measured by providing sample training set of data where all networks are connected [5].

3.2 Semi Supervised Algorithms

Semi supervised leanings is extended part of supervised leanings. Almost all case, detecting interaction is highly preferred so there so negative data do not get importance. In the training data there might be many negative data which are not concerned in supervised data as a result practical gene regulation network cannot be inferred [5]. So unsupervised learning is used. Here unleveled samples of SVM are relabeled with negatives. As a result it can handle the unleveled samples.

3.3 Unsupervised Algorithms

In this section different types of unsupervised methods like Relevance Networks (RN), Correlation, and SPEARMAN-C will be discussed [5]. Where increase of accuracy is maintained, particularly each method tries to improve the accuracy. Unsupervised methods are important in a sense that it can be a general algorithm to inference Gene Regulatory Network where in supervised or semi supervised methods are for specific training data set. And they will try to inference network according to the same process but different data set of gene may have different type relation.

3.3.1 Relevance Network

In this method the profiles of genes are used to infer the interaction of genes. To infer this Butte and Kohane [13] used Mutual Information of gene profiles. The equation of Mutual Information is

$$\boldsymbol{I(X_i, X_j)} = \sum_{x_i \in X_i} \sum_{x_j \in X_j} \boldsymbol{p(x_i, x_j)} \log \left(\frac{\boldsymbol{p(x_i, x_j)}}{\boldsymbol{p(x_i)p(x_j)}} \right)$$

Here for X_i, X_j are discrete variables and $p(x_i) and p(x_j)$ are the marginal probabilities and $p(x_i, x_j)$is joint probability distribution of the discrete variables [5].

3.3.2 Correlation

According to this method, the pair of genes of correlated expression levels is indicative of a regulatory interaction. Correlation coefficients of correlated expression levels has the range of +1 to -1, where '+1' means the pair has activating interaction and '-1' means inhibitory interaction

$$\boldsymbol{corr(X_i, X_j)} = \frac{\boldsymbol{cov(X_i, X_j)}}{\boldsymbol{\sigma(X_i).\sigma(X_j)}}$$

Where i and j are the genes of X_i and X_j expression levels and $cov(X_i, X_j)$denotes the covariance, and $\sigma(X_i).\sigma(X_j)$are the standard deviation [5]. And the correlation Coefficients is measured by

$$\boldsymbol{\omega_{ij}} = |\boldsymbol{corr(X_i, X_j)}|$$

3.3.3 SPEARMAN-C

It is a modification of spearman method. Spearman is also modified from correlation method for ranked expression values, for Here correlation Coefficient is multiplying with the mean correlation of one gene with all other gene [5]

$$\omega_{ij}=|corr(X_i, X_j) . \frac{1}{n} \sum_k^n corr(X_i, X_k) |$$

Information theoretic approaches will be discussed in details in next chapter

Chapter 4

Information Theoretic Approaches in Gene Regulatory Network Inference

Several types of information theoretic approaches have been used to reverse engineer gene regulatory network from expression data. Some of these algorithms including the Entropy Reduction Technique we used in our algorithm are described below:

4.1 ARACNE

ARACNE (Algorithm for the Reconstruction of Accurate Cellular Networks) is an information theoretic algorithm that uses microarray gene expression data to generate transcriptional regulatory network [3]. It identifies links between genes as true regulatory interactions if the statistical dependency is irreducible between those genes. ARACNE defines potential regulatory interactions between two genes based on their Mutual Information (MI). After generating pair-wise mutual information, it uses a threshold value to eliminate links between gene pairs as not

significant if they are below the threshold value. But the problem with this MI-based approach is that it also labels indirect interactions between genes which are highly co-regulated due to their relationship with a third gene, as true regulatory interactions, which results in large amount of false positives. ARACNE solves this problem by using a technique called the Data Processing Inequality (DPI) [3].

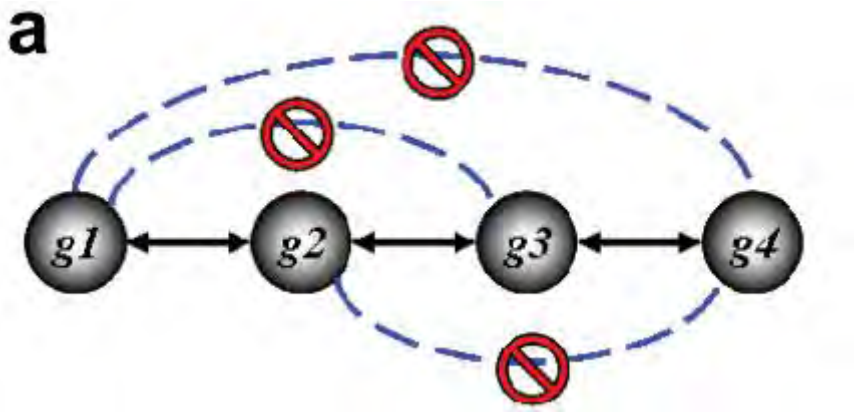

Figure 4.1: DPI Technique [3]

The idea of DPI is, if Gene g1 and Gene g3 [above figure] have indirect interactions through a third Gene g2 then the link between g1 and g3 will be removed by DPI because, where I is the MI between the gene pairs.

$$I(g1, g3) <= \min [I(g1, g2), I(g2, g3)]$$

4.2 CLR

CLR (Context Likelihood of Relatedness) is another information theory based algorithm to infer transcriptional regulatory network. Like ARACNE, CLR also uses mutual information as potential regulatory interactions. First, CLR calculates MI between gene pairs. Then it modifies the value using the information about the context of the network [2]. It gives priority to the MI

values of gene pairs for probable regulatory relationship where the MI value is greater than the background distribution of the interactions which are related to the gene pairs.

4.3 Entropy Reduction Technique

The reason of using concepts from information theory such as Entropy, Mutual Information is to generate biological network without any theoretical knowledge regarding the problem. The most essential concept is Entropy which can be defined as follows [2] -

$$\boldsymbol{H(x) = -\sum_{x \in \chi} p(x) \log p(x)}$$

Where H is the entropy, x is a discrete random vector with alphabet X, and p(x) is the probability mass function.

Entropy is closely related to Mutual Information which is the measurement of how much information one variable contains about another variable. Mutual Information can be described in terms of both Joint Entropy H(X, Y) and Conditional Entropy, H(Y|X) Entropy in the following way [2]

$$\boldsymbol{H(X,Y) = -\sum_{x} \sum_{y} p(x,y) \log p(x,y)}$$

$$\boldsymbol{H(Y|X) = -\sum_{x} p(x)(Y|X = x)}$$

$$\boldsymbol{= -\sum_{x} p(x) \sum_{y} p(y|x) \log p(y|x)}$$

$$\boldsymbol{= -\sum_{x} \sum_{y} p(x,y) \log p(y|x)}$$

The basic idea of Entropy Reduction technique is, if a variable A does not depend on a set of variables B, and then the Entropy of A given the set of variable B is equal to the Entropy of A. But if variable A has dependency on set of variables B, then the Entropy of A given B is less than the Entropy of variable A [2].

$$\boldsymbol{H(A|B) = H(A)}\textbf{, A and B are independent of each other}$$

$$\boldsymbol{H(A|B) < H(A)}\textbf{, A and B have dependency relationship}$$

This technique was used to infer small biological networks in [2] by starting the ERT algorithm for a variable A with an empty set of variables B of regulatory interactions with A, H (A|B). Then in an iterative process new variables were added to the list of regulatory relationship in H (A|B) if the conditional entropy of adding new variable with previous set of variable, H (A|B,C) is less than the previous conditional entropy H (A|B). The adding of new variables continued until all the other variables except A were added to that list or the conditional entropy of adding new variable was equal to the previous conditional entropy.

The focus for this paper [2] was small biological networks which made it possible to carry out the Entropy Reduction Technique up to three to four variables by calculating the conditional entropy of one variable given three to four variables. But for large networks containing thousands of genes, this is an extremely time consuming and unfeasible to apply in real applications.

So for our algorithm, we only carried out one step Entropy Reduction by considering for each variable A to have potential regulatory interactions with all the other variables if the conditional entropy of a variable with A, H (A|B) is less than the entropy of A, H (A). After that the amount of entropy reduced for each variable with other variables identified from the Entropy Reduction step was calculated. This produced a matrix of size n x n (n is the number of genes), of potential regulatory interactions among genes. To reduce the false positives by eliminating the less significant interactions with very low entropy reduction, we used the mean of all entropy reduction values as the threshold value. To evaluate our algorithm for other threshold values, we ran the algorithm for a range of threshold values including the mean value.

Chapter 5

Clustering and ERT based Algorithm

In this chapter we will describe the complete algorithm in details. The algorithm can be divided in two main parts, the Clustering part and the Entropy Reduction part.

5.1 The Clustering Part

Input: Data matrix A of dimension n x m, where n is the number of genes and m is the number of samples or experiments, the value of number of clusters K, algorithm for K-means clustering and maximum number of iterations for K-means.

Algorithm:

1. Cluster the dataset into K different clusters using the given algorithm.

2. Generate a list of K elements L, where each element contains all the data points assigned to a cluster. This list is used in the Cluster Merging and ERT steps.

3. Calculate a distance matrix D of dimension K x K where distance between each pair of cluster centers is stored.

Output: L, list of K elements and distance matrix D

5.2 Entropy Reduction Part for One Cluster [2]

Input: Cluster ID

Algorithm:

1. Collect data points n of the given cluster ID from the list L generated in Clustering Part

2. Generate data matrix TD of the genes from the main data matrix A where columns contain genes and rows contain samples.

3. Discretize TD.

4. Calculate Mutual Information (MI) matrix M of dimension n x n and normalize the mutual information in an n x n dimensional matrix NMI using Linfoot definition of normalization to have the mutual information values in the range of 0 to 1 [2].

M [i,j] = MutualInformation(TD[,i], TD[,j])

5. Calculate the single entropy matrix E of dimension n.

E[i] = Entropy (TD [,i])

6. Calculate an n x n dimensional conditional entropy matrix CE between all pair of variables.

CE [i,j] <- ConditionalEntropy(TD[,i],TD[,j])

7. Calculate an n x n dimensional Reduced Entropy matrix RE between each pair of variables i,j using mutual information matrix M and single entropy matrix E using the equation,

RE [i,j] = (M[i,j]) / E[i]

8. Generate an n x n dimensional ERT matrix ERTM using the following condition,

If CE [i,j] = E[i] , then ERTM [i,j] = 1

Else ERTM [i,j] = 0

9. Generate a connection matrix C of n x n dimension using the following condition,

If ERTM [i,j] == 1, then C [i,j] = RE [i,j]

Else C [i,j] = 0

After this step, each cell of the connection matrix contains the reduced entropy between two genes.

Output: Connection matrix C.

To apply the ERT algorithm on two clusters for the Selected Merged version of the algorithm, first identify the closest clusters of a given cluster by the process described in chapter 2, Selected Merged Version, and then run the ERT algorithm described above with the exception of all n x n matrix are replaced with n x m matrix where n is the number of genes of first cluster and m is the number of genes in the second cluster. The returned connection matrix C is also of dimension n x m.

After running ERT on all individual clusters and all pairs of closest clusters, all the returned matrices are combined together in a connection matrix of dimension n x n. Different threshold values are applied in the connection matrix to evaluate the performance of the algorithm.

The overall algorithm can be described as follows:

1. Cluster the dataset into given number of clusters

2. Apply ERT algorithm on each cluster.

3. Merge a cluster with its closest clusters

4. Apply ERT algorithm on all the merged clusters and combine all the results of connection matrices from ERT to generate an n x n final connection matrix where n is the number of genes.

We also used the DPI technique from ARACNE algorithm after the final connection matrix of genes was generated by our algorithm for different number of clusters and threshold values in an attempt to reduce the false positives of our algorithm and also to verify whether combining the DPI technique with our algorithm improves the accuracy of the generated regulatory network.

Chapter 6

Results

To evaluate our algorithm, we used three datasets of different sizes. Two of our datasets were from DREAM Challenges, which is an organization of researchers that holds annual community challenges on gene regulatory network inference. Dataset-1 is the network 1 from DREAM5 - Network Inference Challenge, Dataset-2 is the 100-Genes *in silico* network from DREAM4 - *In Silico* Network Challenge, and Dataset-3 is from GeneNetWeaver (GNW), software to generate artificial gene expression data.

For each dataset, we generated multiple types of graphs and bar charts to evaluate the results obtained from the algorithm by visual means. First the different types of graphs are explained below and then for each dataset we present the results in terms of these graphs.

1. To identify the relationship between cluster numbers and threshold values in the context of True Positives, we generated “True Positive VS Cluster Number” and “True Positive VS Threshold Values” graphs both for “Before DPI” and “After DPI” results.

2. Same graphs were also generated for False Positives.

3. To compare the performance of the results in terms of True Positives among ARACNE, No Clustering Version and Selected Merged Clustering Version and Unmerged Clustering Version of our algorithm, we generated bar charts for different threshold values both for “Before DPI” and “After DPI” results.

4. Same bar charts were generated for False Positives as well.

5. To compare the performance in terms of Precision and Recall among different inference method including different versions of our algorithm, we generated bar charts for different threshold values both for “Before DPI” and “After DPI” results.

We use precision and recall as performance measures for our algorithm. Precision is defined by

$$\mathbf{Precision} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FP}}$$

Where TP is the True Positive prediction of a regulatory interaction between a pair of genes and FP is the False Positive prediction.

Recall is defined by,

$$\mathbf{Recall} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FN}}$$

Where FN is a False Negative Prediction of regulatory interaction.

Results for Dataset - 1 (DREAM5)

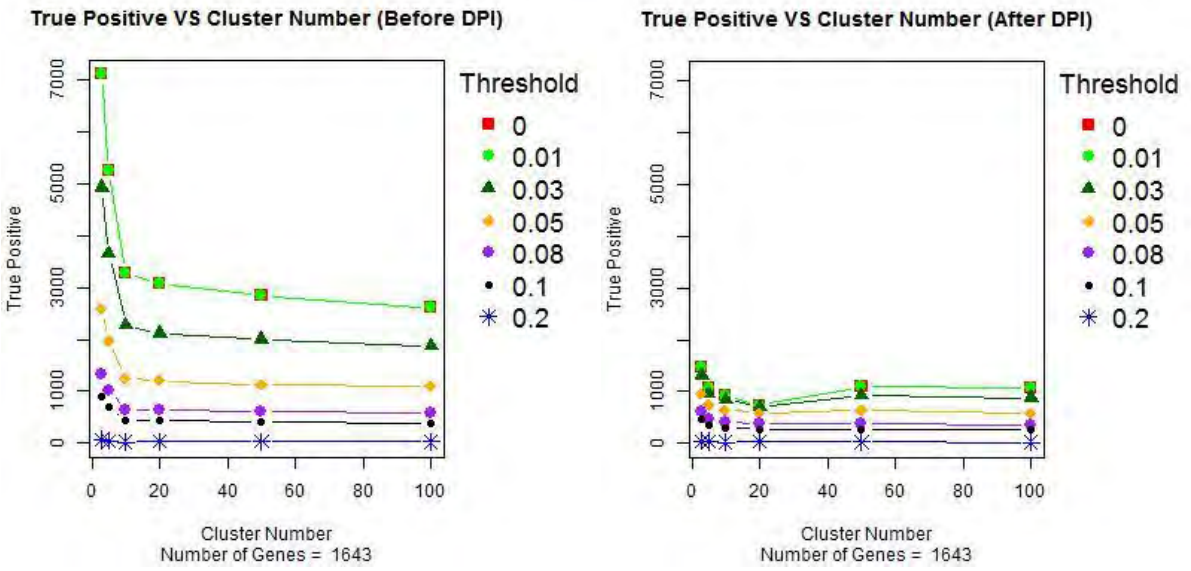

Figure 6.1: True Positive VS Clusters for Different Threshold values. Before DPI (Left), After DPI (Right)

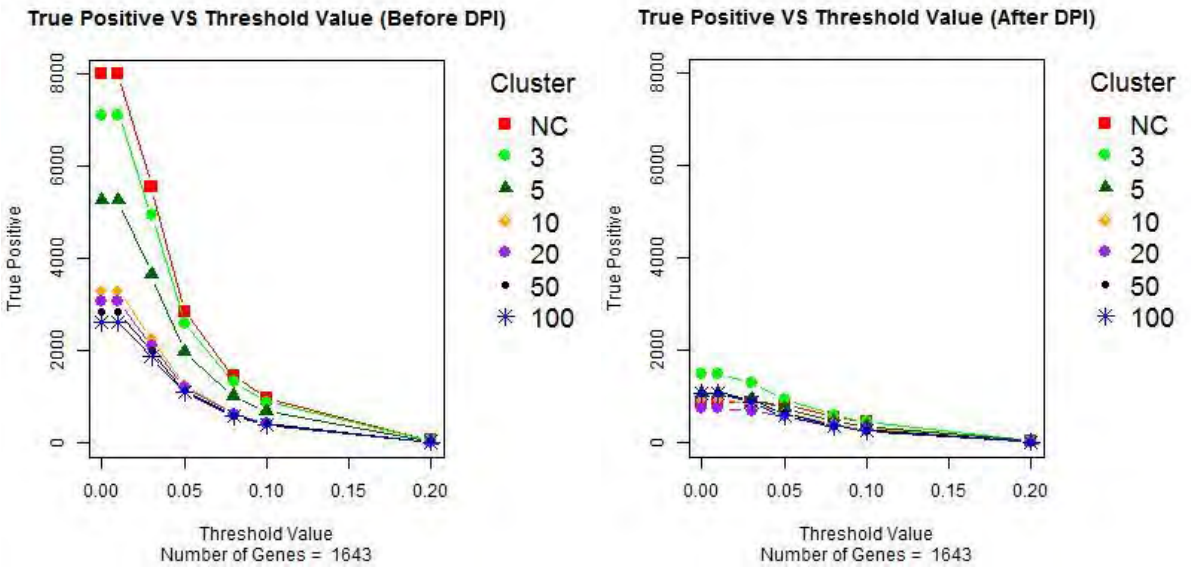

Figure 6.2: True Positive VS Threshold for Different Cluster Numbers. Before DPI (Left), After DPI (Right)

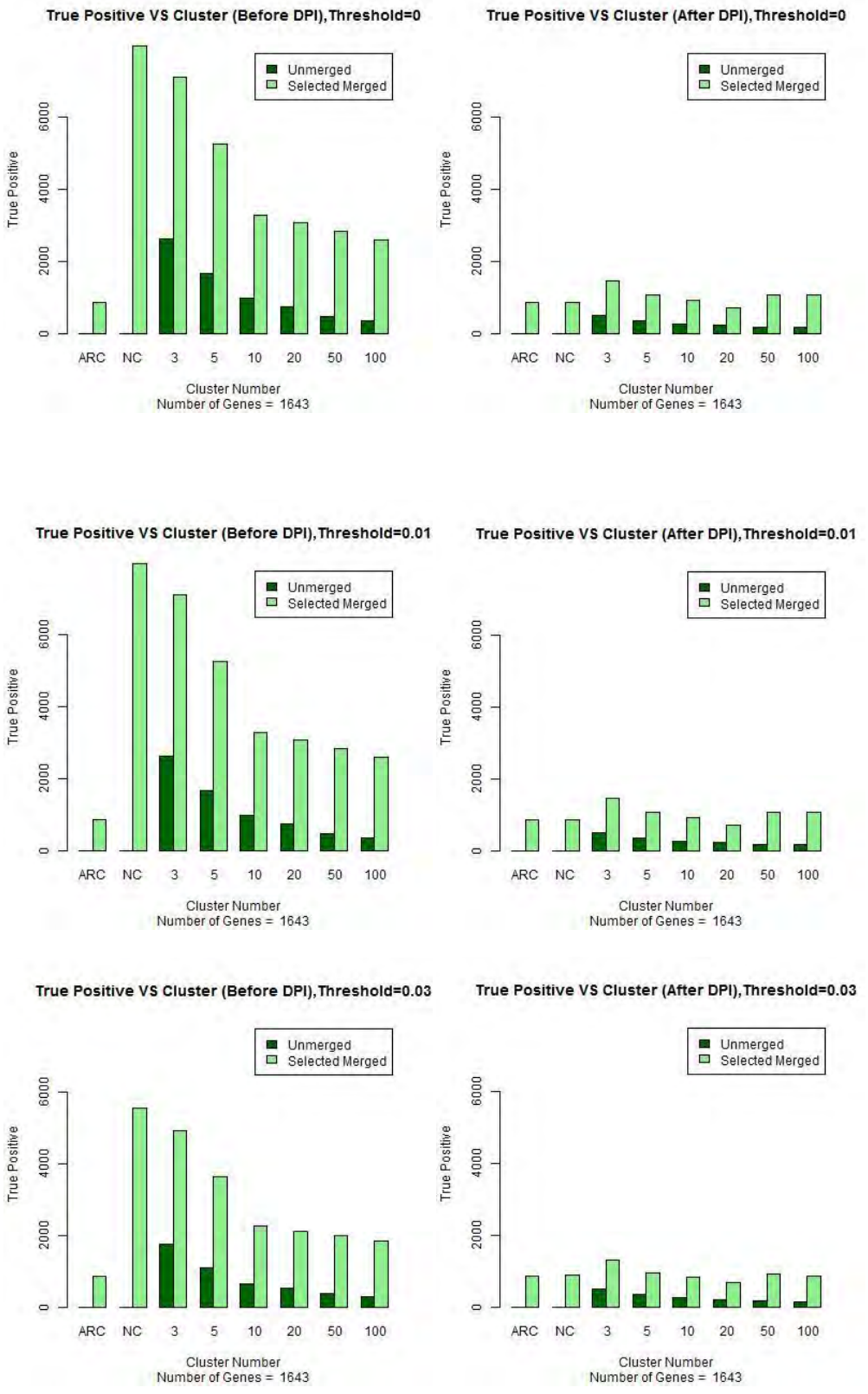

Figure 6.3: True Positive VS Cluster for ARACNE, No-Clustering, Selected Merged, Unmerged. Before DPI (Left), After DPI (Right) (Continued)

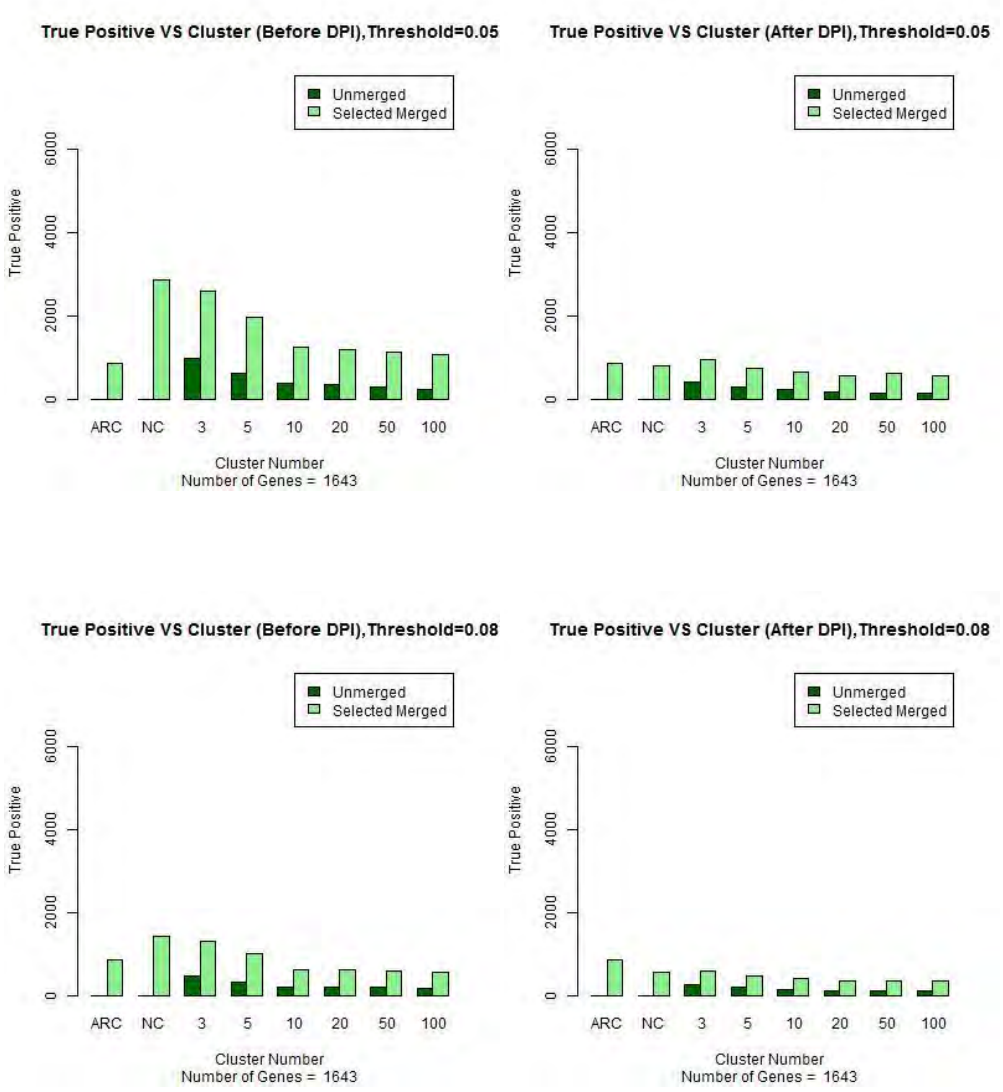

Figure 6.3: True Positive VS Cluster for ARACNE, No-Clustering, Selected Merged, Unmerged. Before DPI (Left), After DPI (Right)

Same graphs were also generated for False Positives:

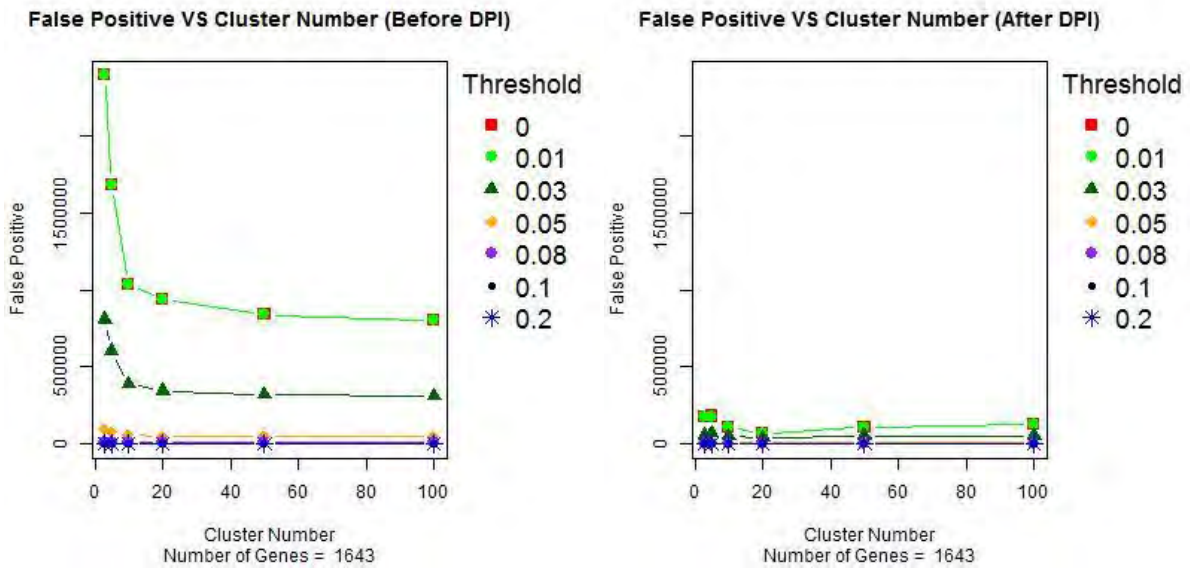

Figure 6.4: False Positive VS Clusters for Different Threshold values. Before DPI (Left), After DPI (Right)

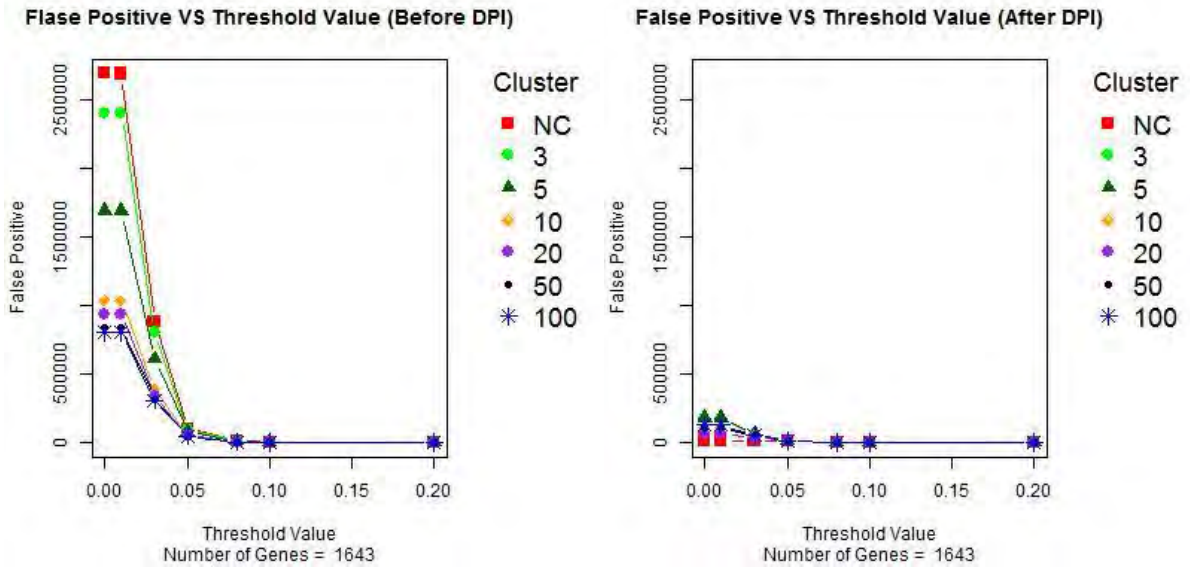

Figure 6.5: False Positive VS Threshold for Different Cluster Numbers. Before DPI (Left), After DPI (Right)

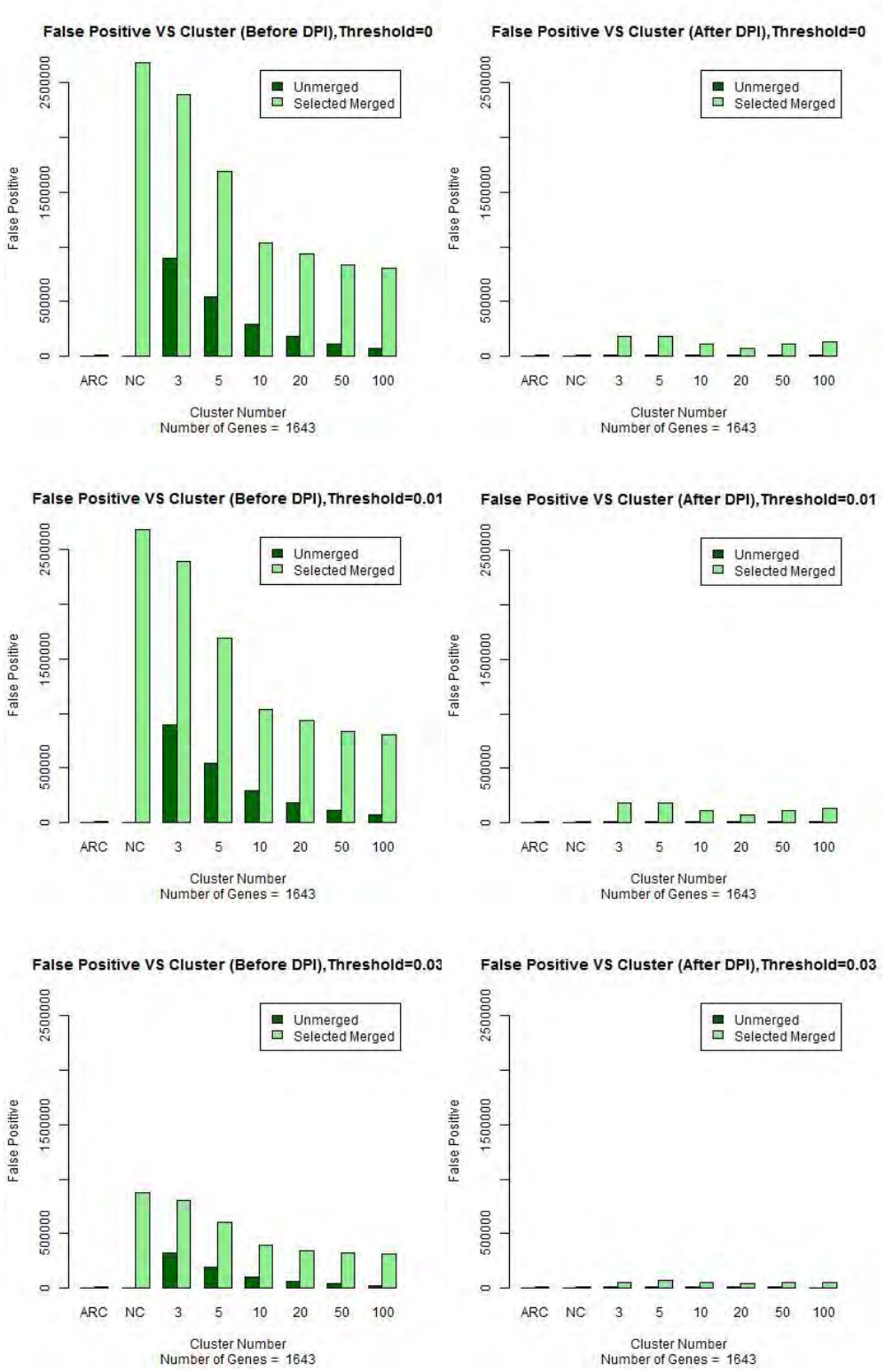

Figure 6.6: False Positive VS Cluster for ARACNE, No-Clustering, Selected Merged, Unmerged. Before DPI (Left), After DPI (Right) (Continued)

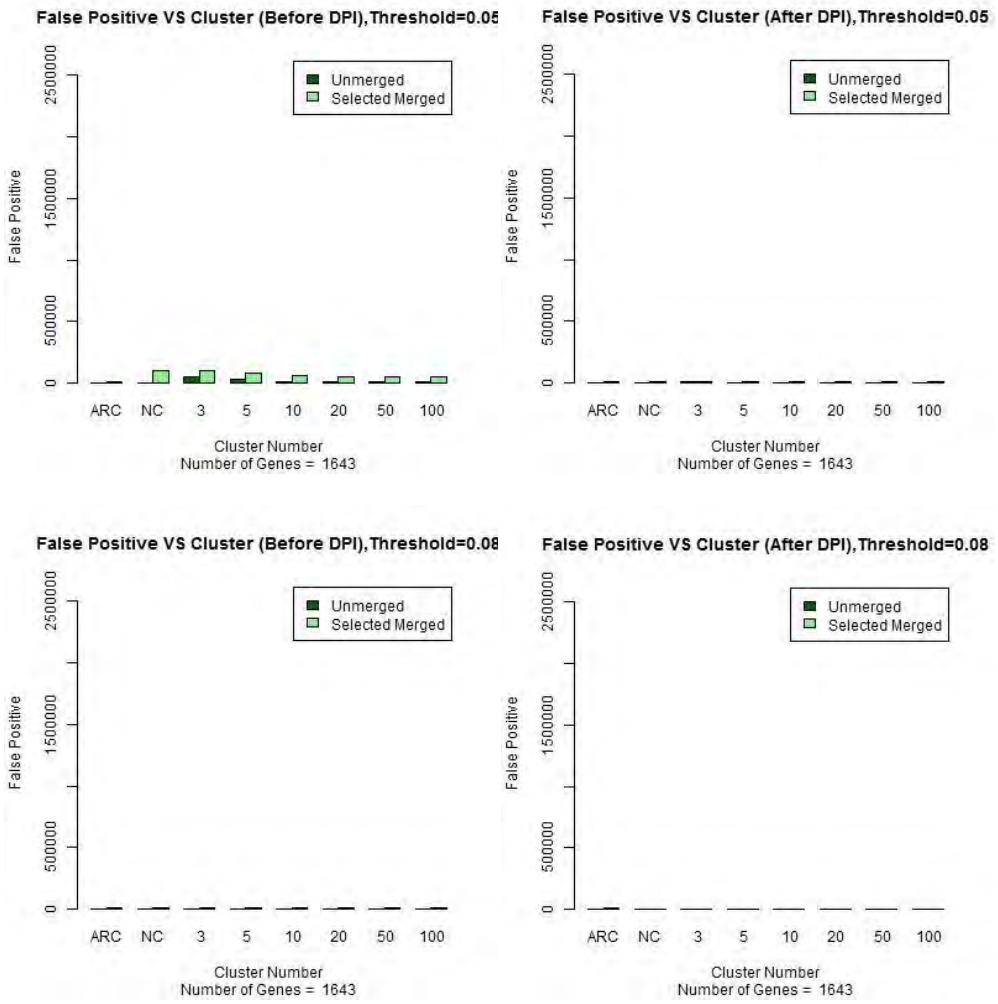

Figure 6.6: False Positive VS Cluster for ARACNE, No-Clustering, Selected Merged, Unmerged. Before DPI (Left), After DPI (Right)

Precision and Recall Graphs for Different Threshold values:

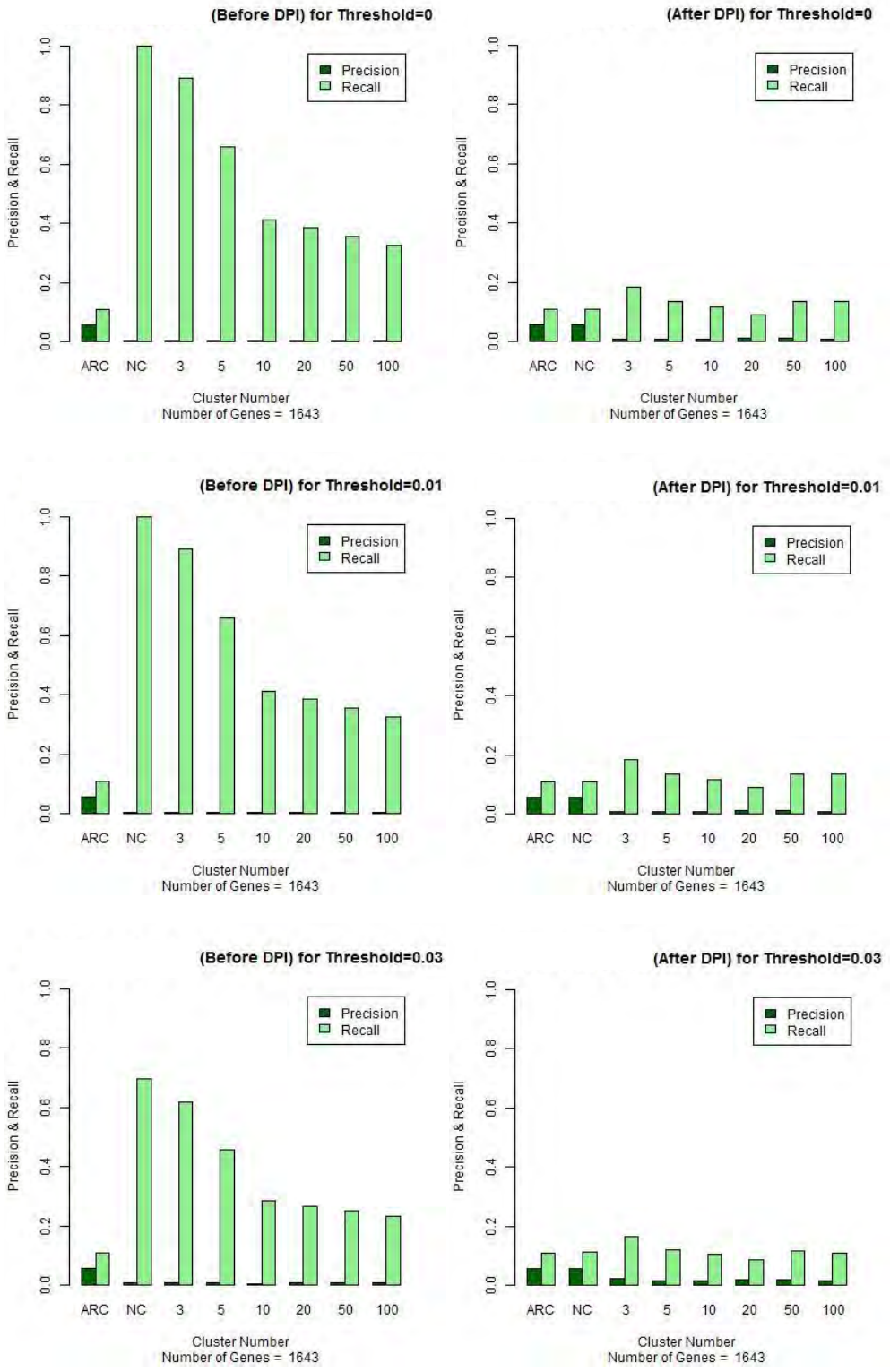

Figure 6.7: Precision and Recall for ARACNE, No-Clustering and Selected Merged Clustering. Before DPI (Left) and After DPI (Right) (Continued)

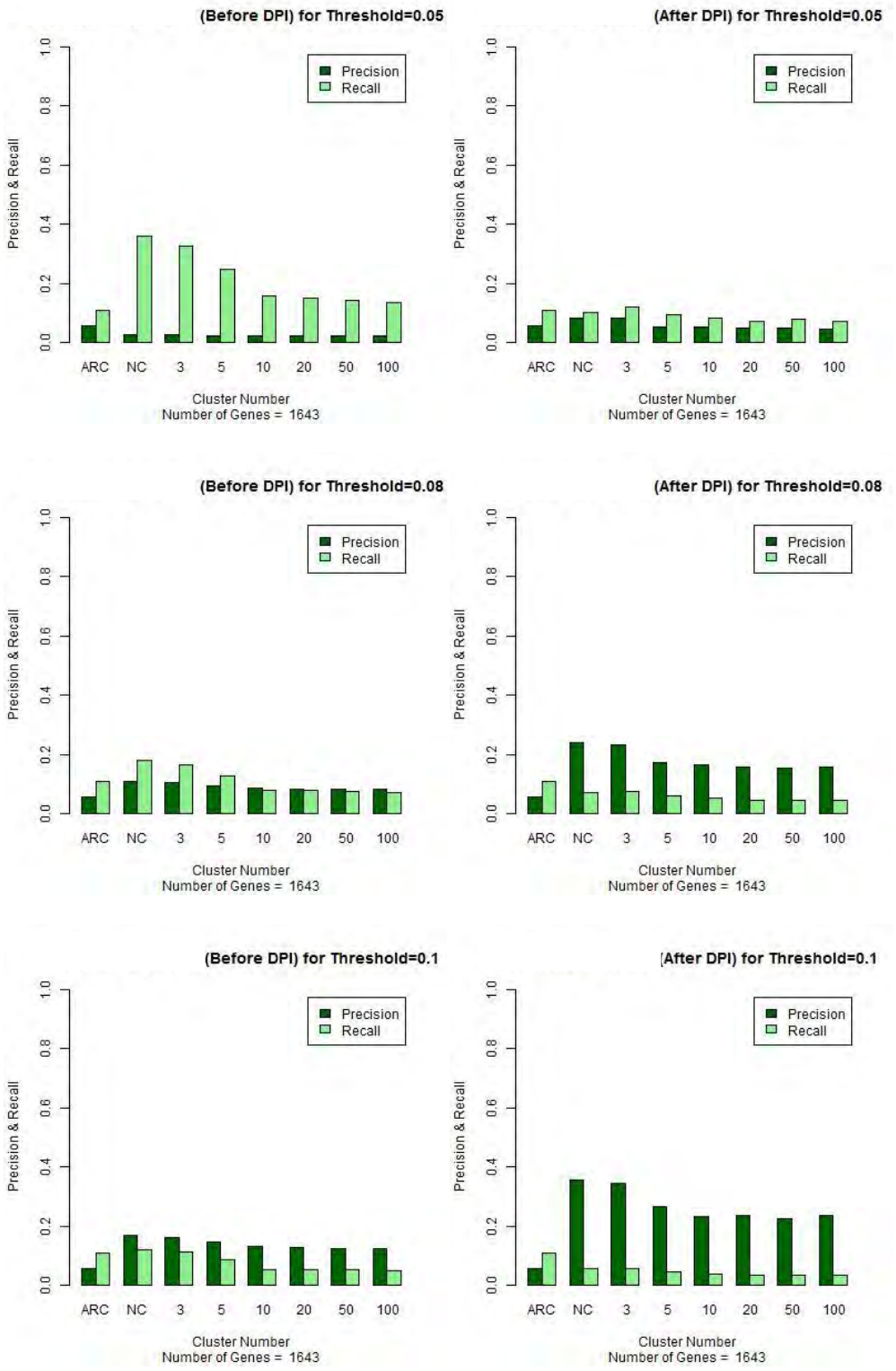

Figure 6.7: Precision and Recall for ARACNE, No-Clustering and Selected Merged Clustering. Before DPI (Left) and After DPI (Right) (Continued)

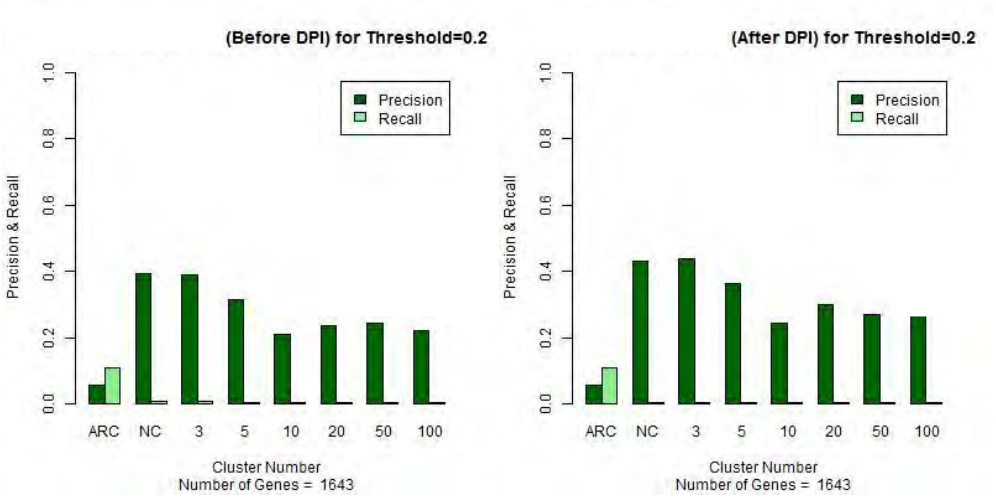

Figure 6.7: Precision and Recall for ARACNE, No-Clustering and Selected Merged Clustering. Before DPI (Left) and After DPI (Right)

Results for Dataset - 2 (DREAM4)

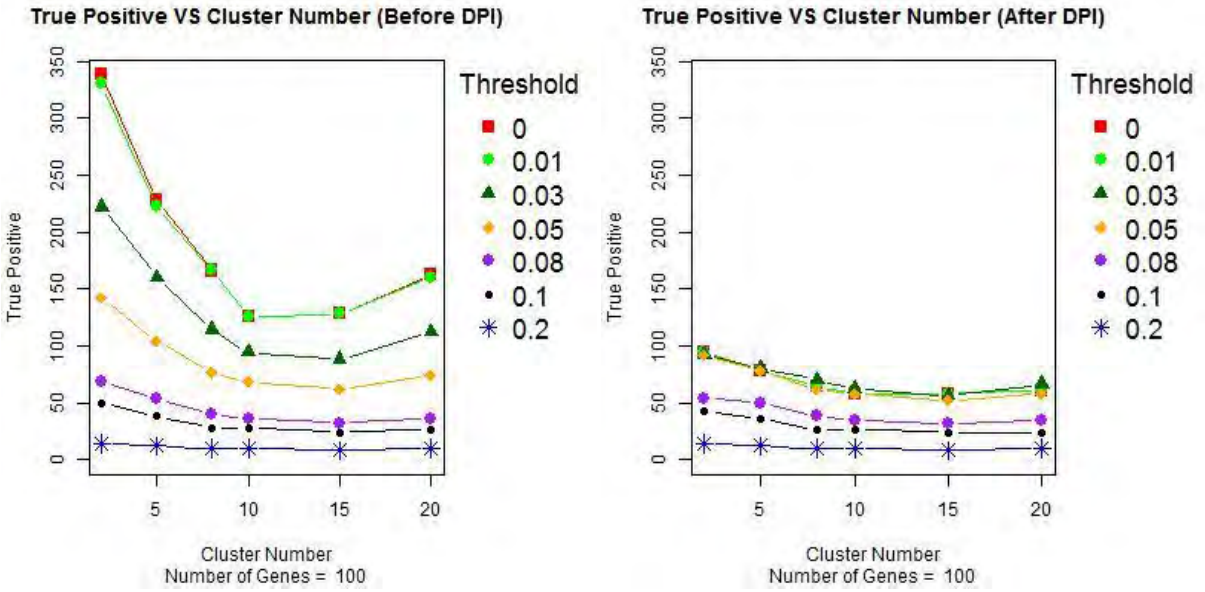

Figure 6.8: True Positive VS Clusters for Different Threshold values. Before DPI (Left), After DPI (Right)

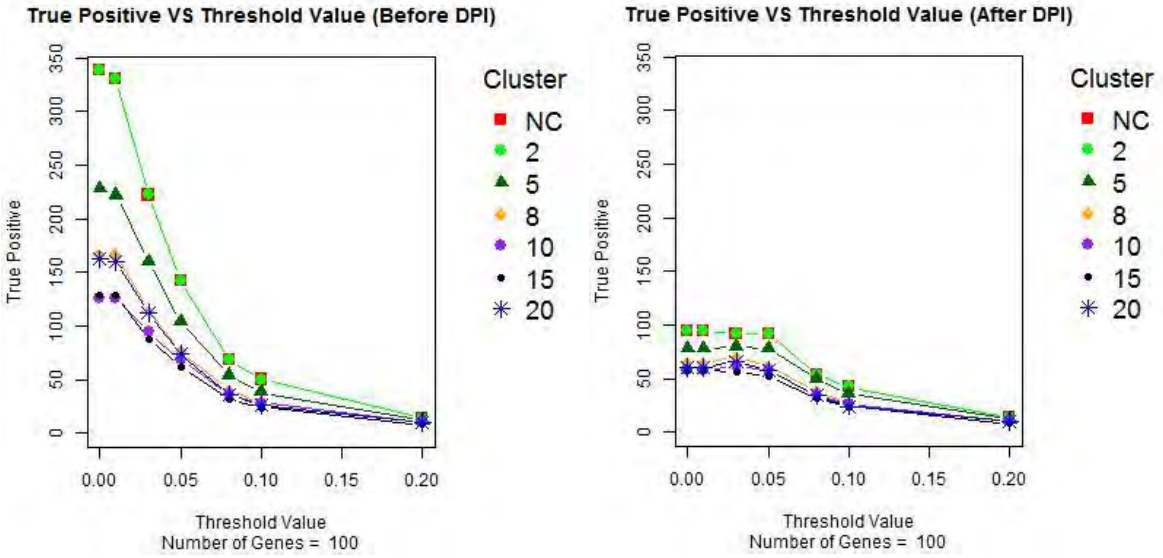

Figure 6.9: True Positive VS Threshold for Different Cluster Numbers. Before DPI (Left), After DPI (Right)

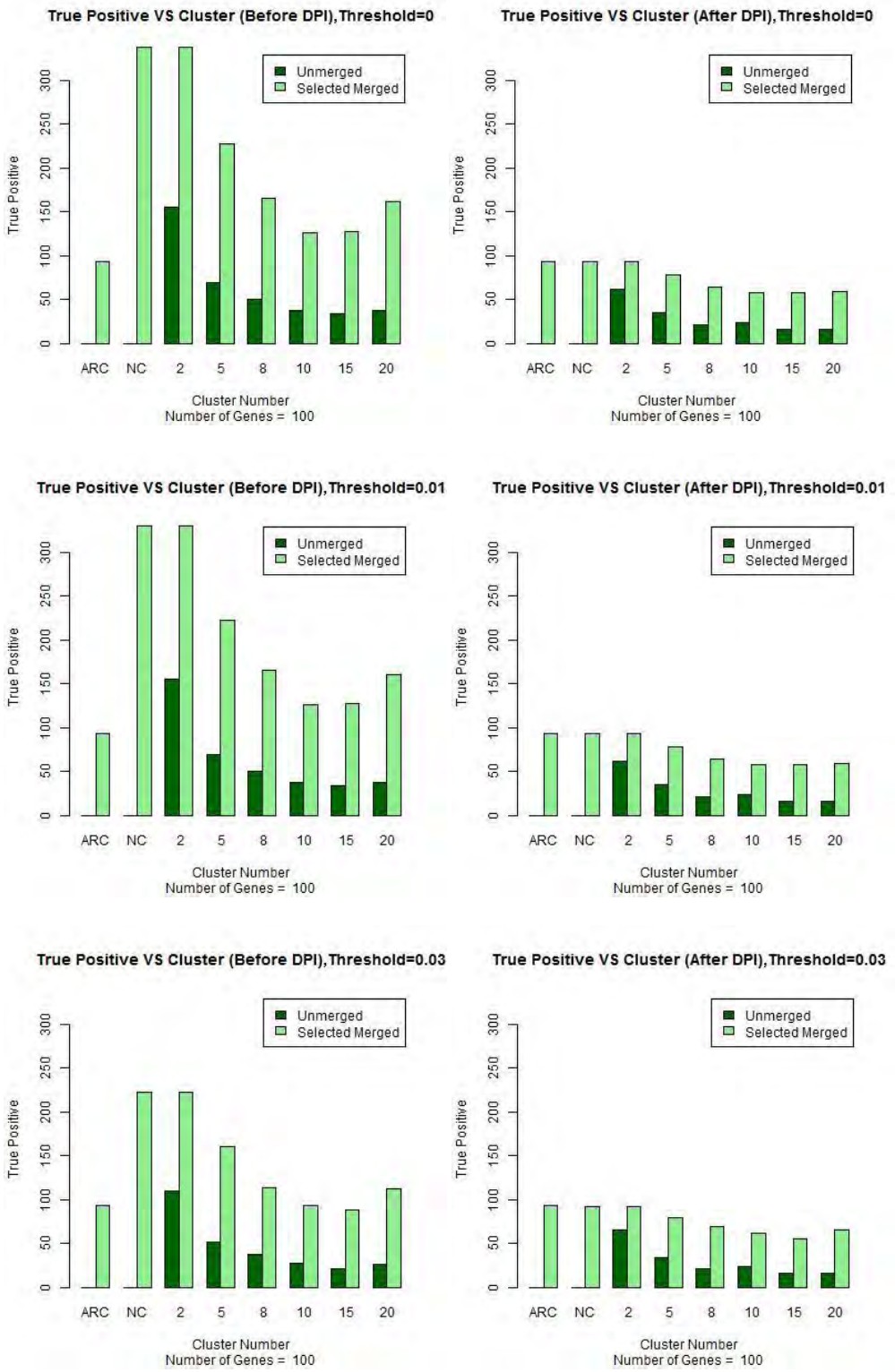

Figure 6.10: True Positive VS Cluster for ARACNE, No-Clustering, Selected Merged, Unmerged. Before DPI (Left), After DPI (Right) (Continued)

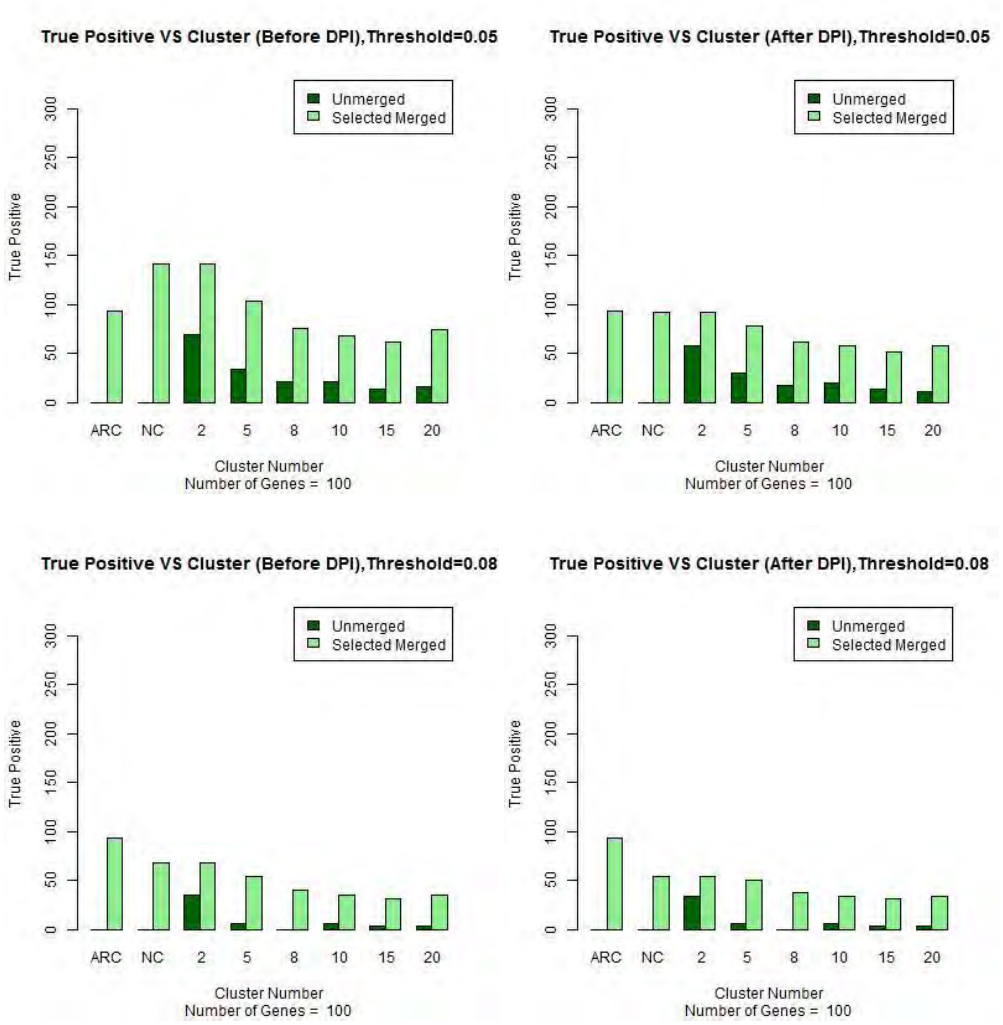

Figure 6.10: True Positive VS Cluster for ARACNE, No-Clustering, Selected Merged, Unmerged. Before DPI (Left), After DPI (Right)

Same graphs were also generated for False Positives:

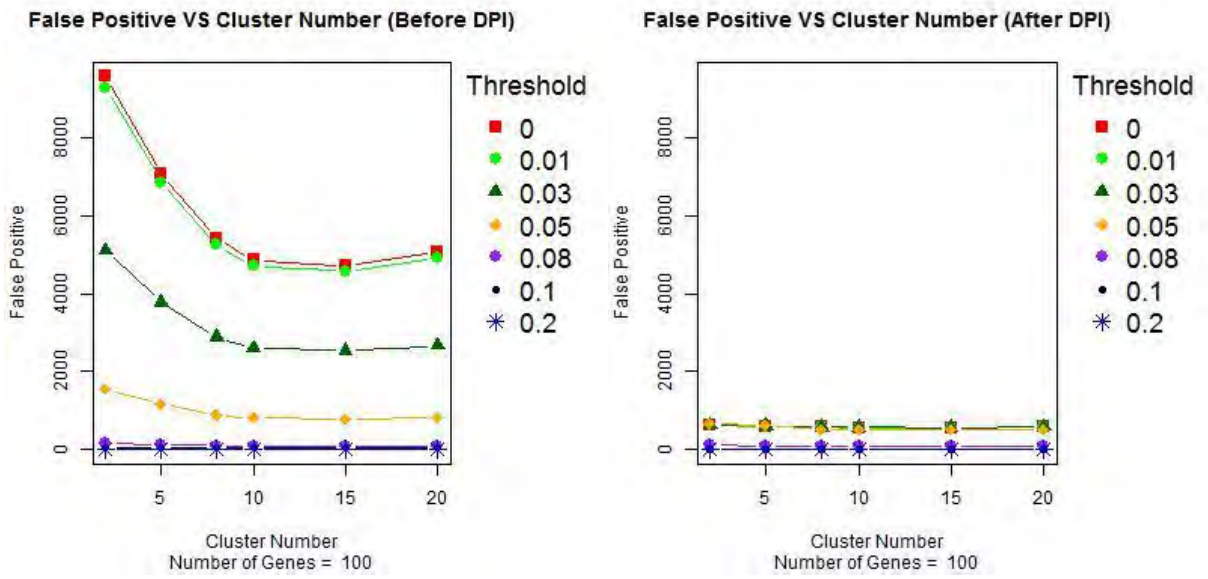

Figure 6.11: False Positive VS Clusters for Different Threshold values. Before DPI (Left), After DPI (Right)

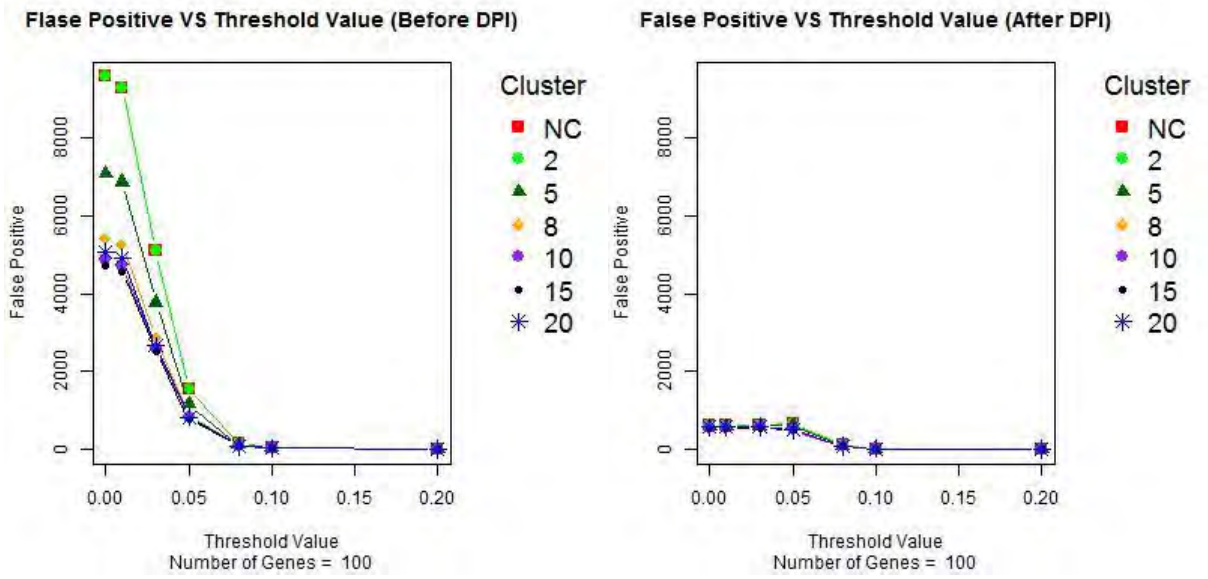

Figure 6.12: False Positive VS Threshold for Different Cluster Numbers. Before DPI (Left), After DPI (Right)

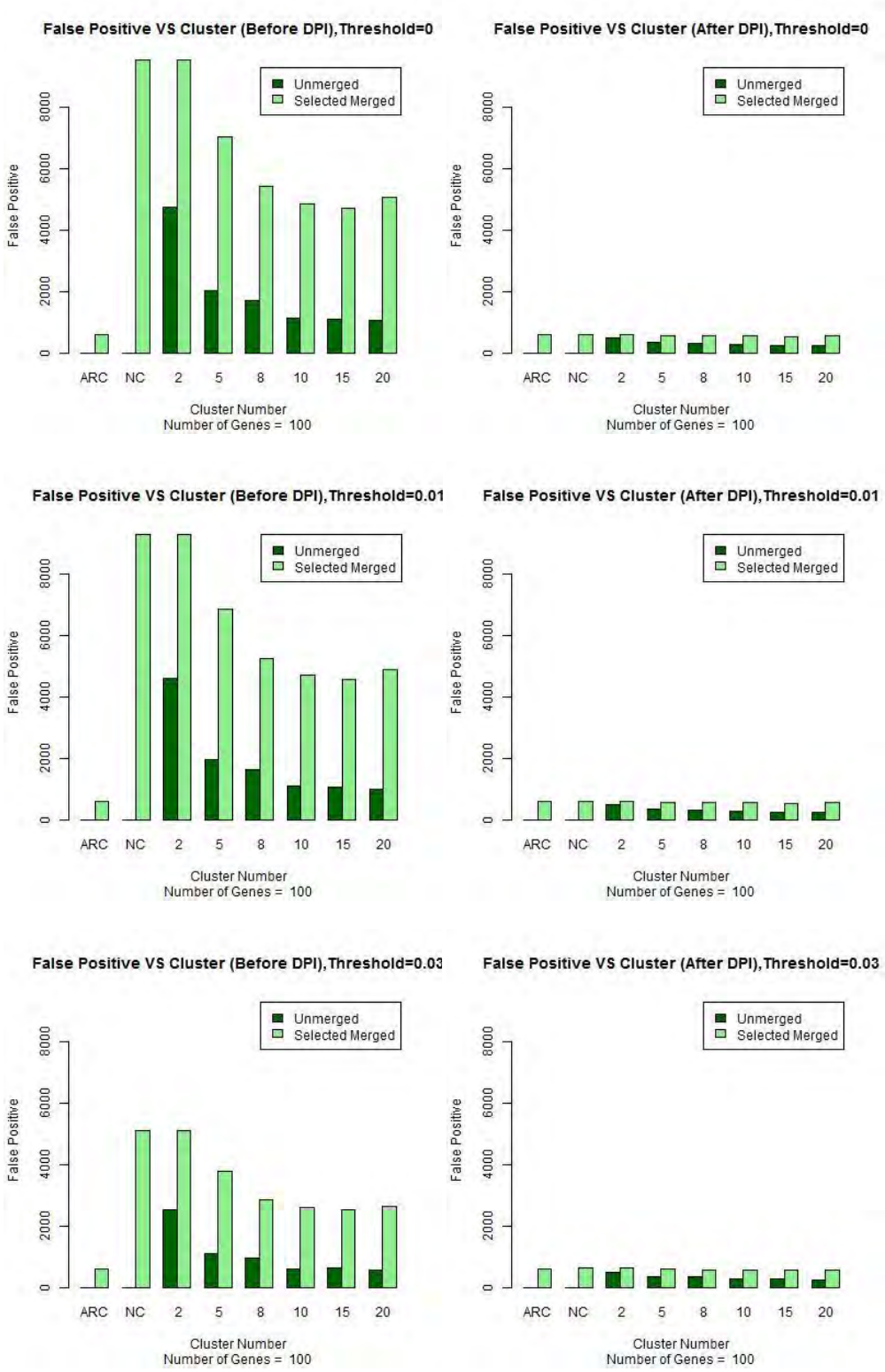

Figure 6.13: False Positive VS Cluster for ARACNE, No-Clustering, Selected Merged, Unmerged. Before DPI (Left), After DPI (Right) (Continued)

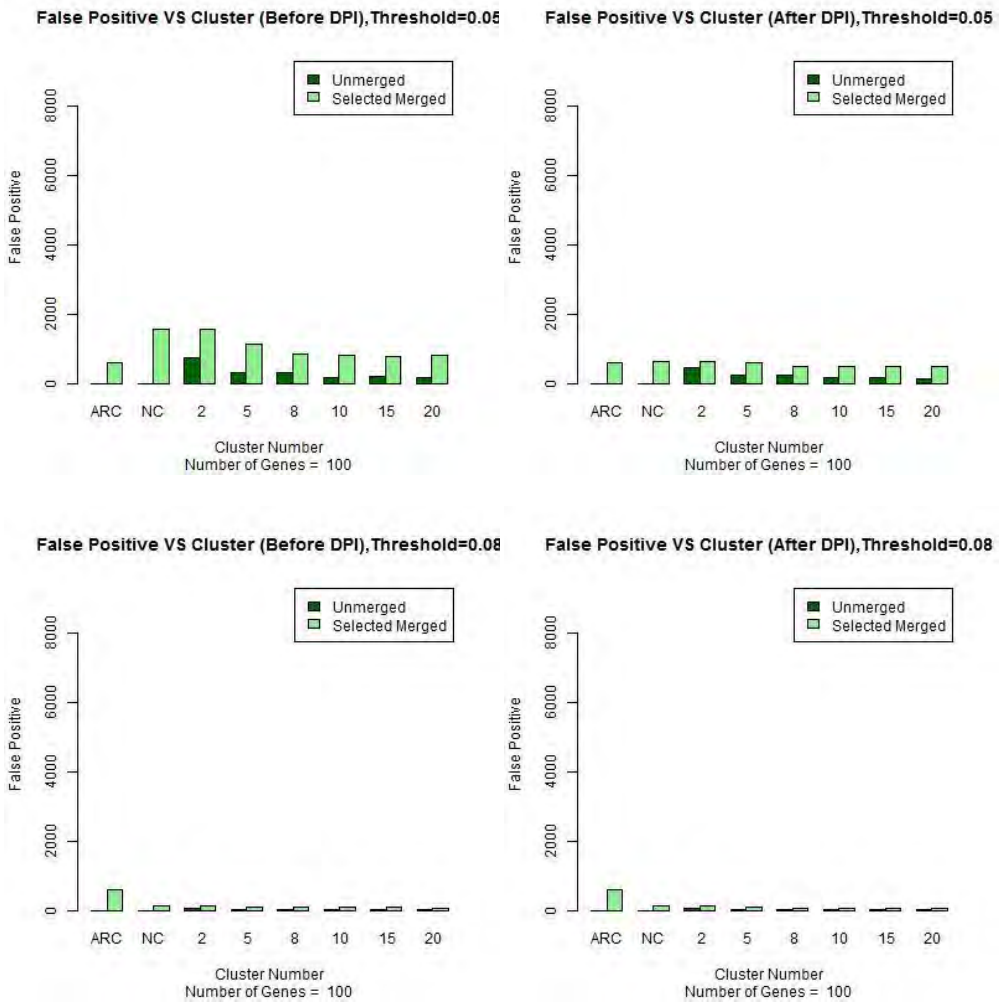

Figure 6.13: False Positive VS Cluster for ARACNE, No-Clustering, Selected Merged, Unmerged. Before DPI (Left), After DPI (Right)

Precision and Recall Graphs for Different Threshold values:

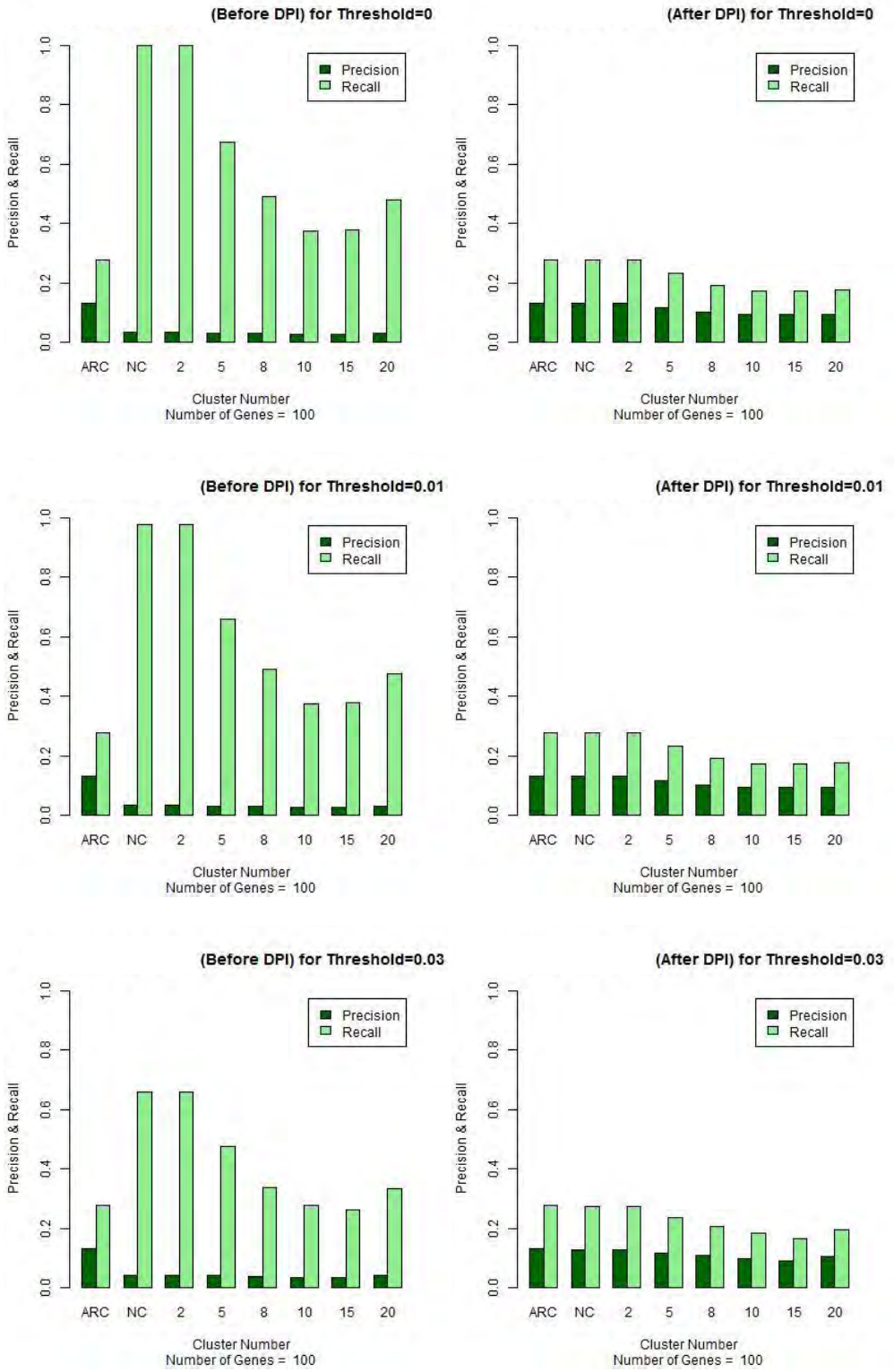

Figure 6.14: Precision and Recall for ARACNE, No-Clustering and Selected Merged Clustering. Before DPI (Left) and After DPI (Right) (Continued)

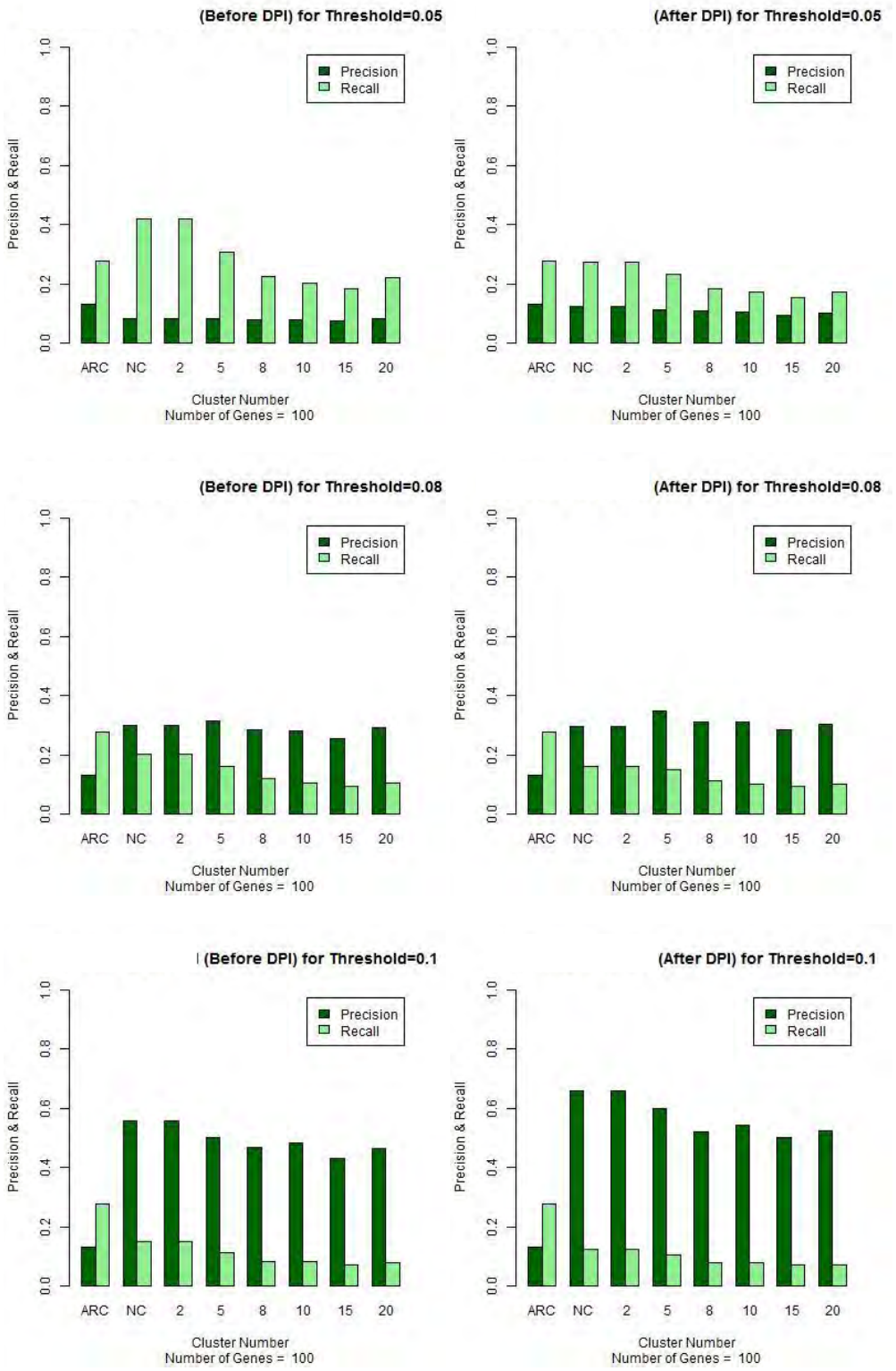

Figure 6.14: Precision and Recall for ARACNE, No-Clustering and Selected Merged Clustering. Before DPI (Left) and After DPI (Right) (Continued)

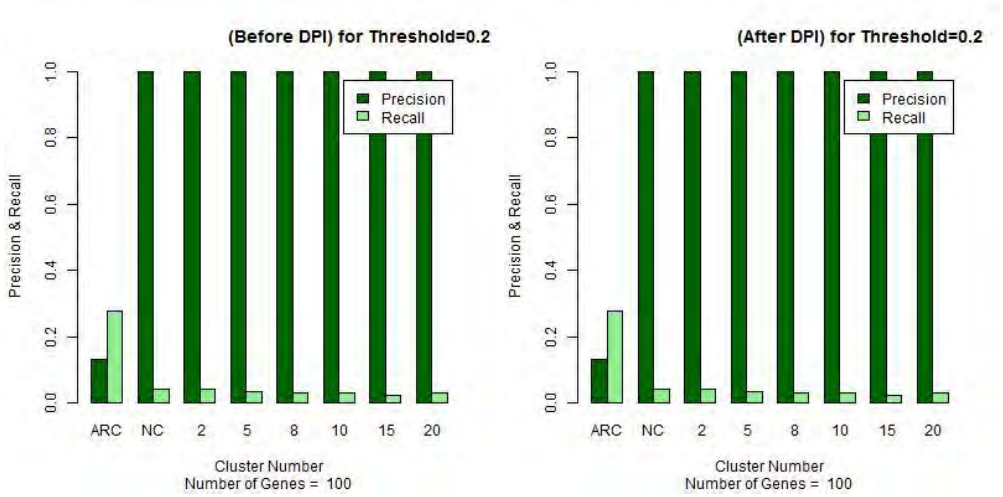

Figure 6.14: Precision and Recall for ARACNE, No-Clustering and Selected Merged Clustering. Before DPI (Left) and After DPI (Right)

Results for Dataset - 3 (GNW)

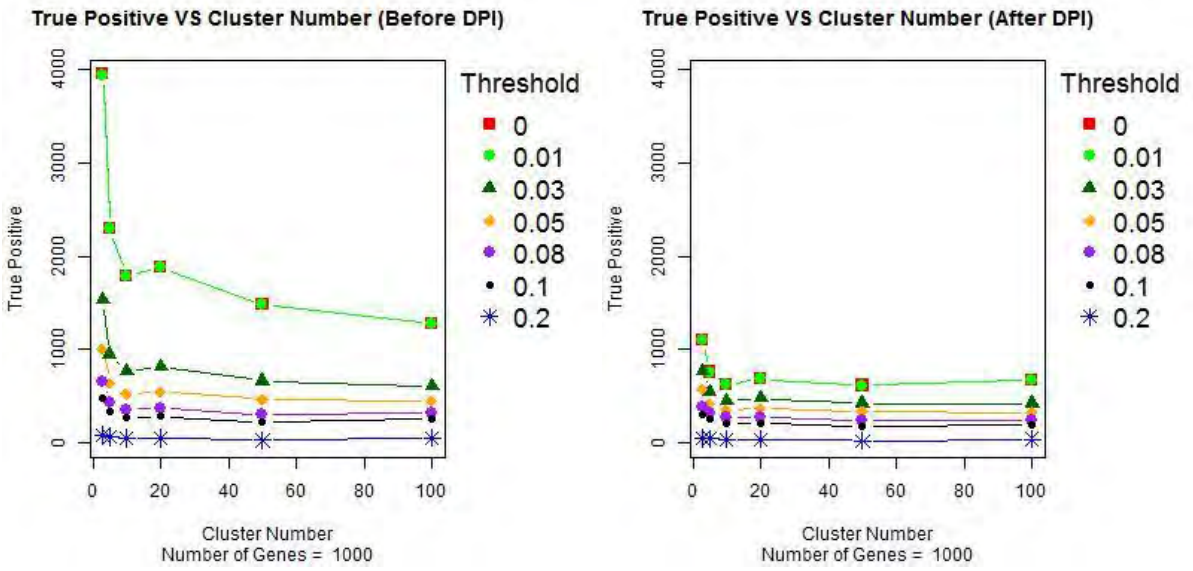

Figure 6.15: True Positive VS Clusters for Different Threshold values. Before DPI (Left), After DPI (Right)

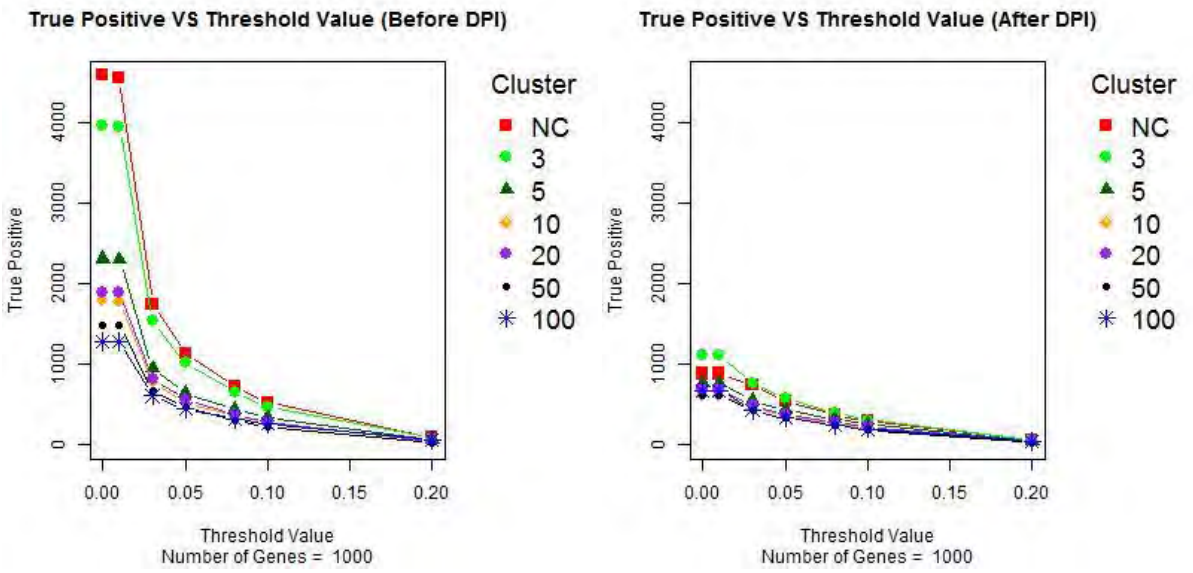

Figure 6.16: True Positive VS Threshold for Different Cluster Numbers. Before DPI (Left), After DPI (Right)

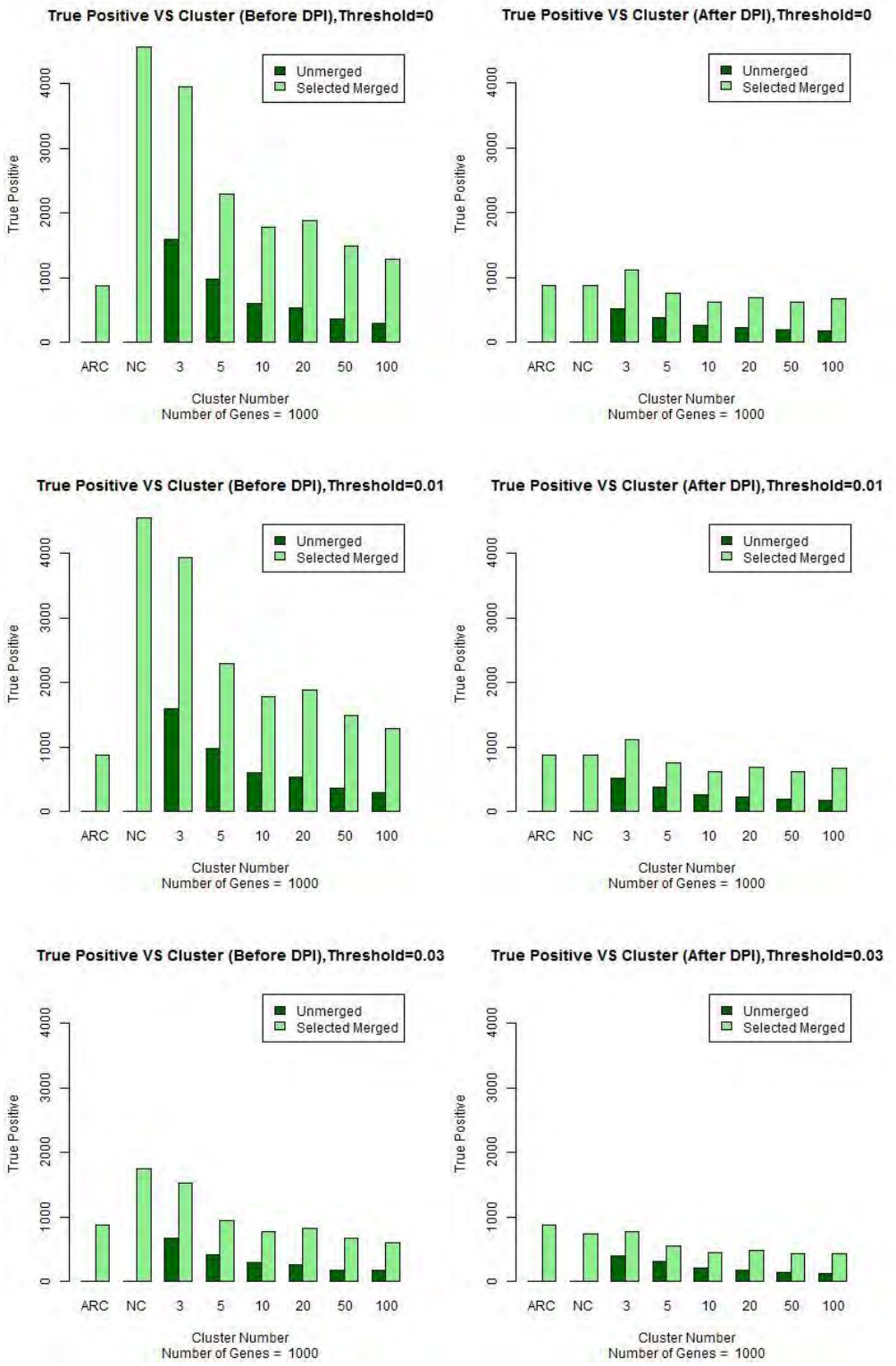

Figure 6.17: True Positive VS Cluster for ARACNE, No-Clustering, Selected Merged, Unmerged. Before DPI (Left), After DPI (Right) (Continued)

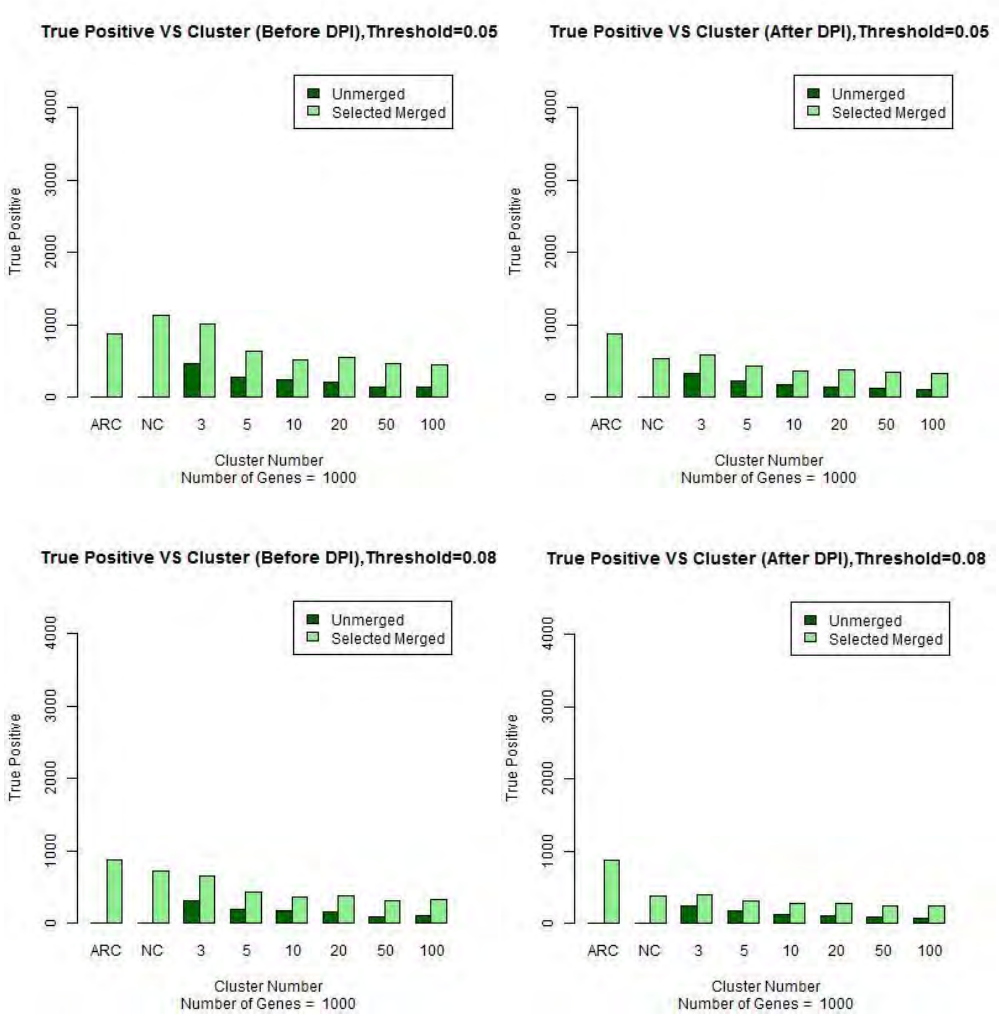

Figure 6.17: True Positive VS Cluster for ARACNE, No-Clustering, Selected Merged, Unmerged. Before DPI (Left), After DPI (Right)

Precision and Recall Graphs for Different Threshold values:

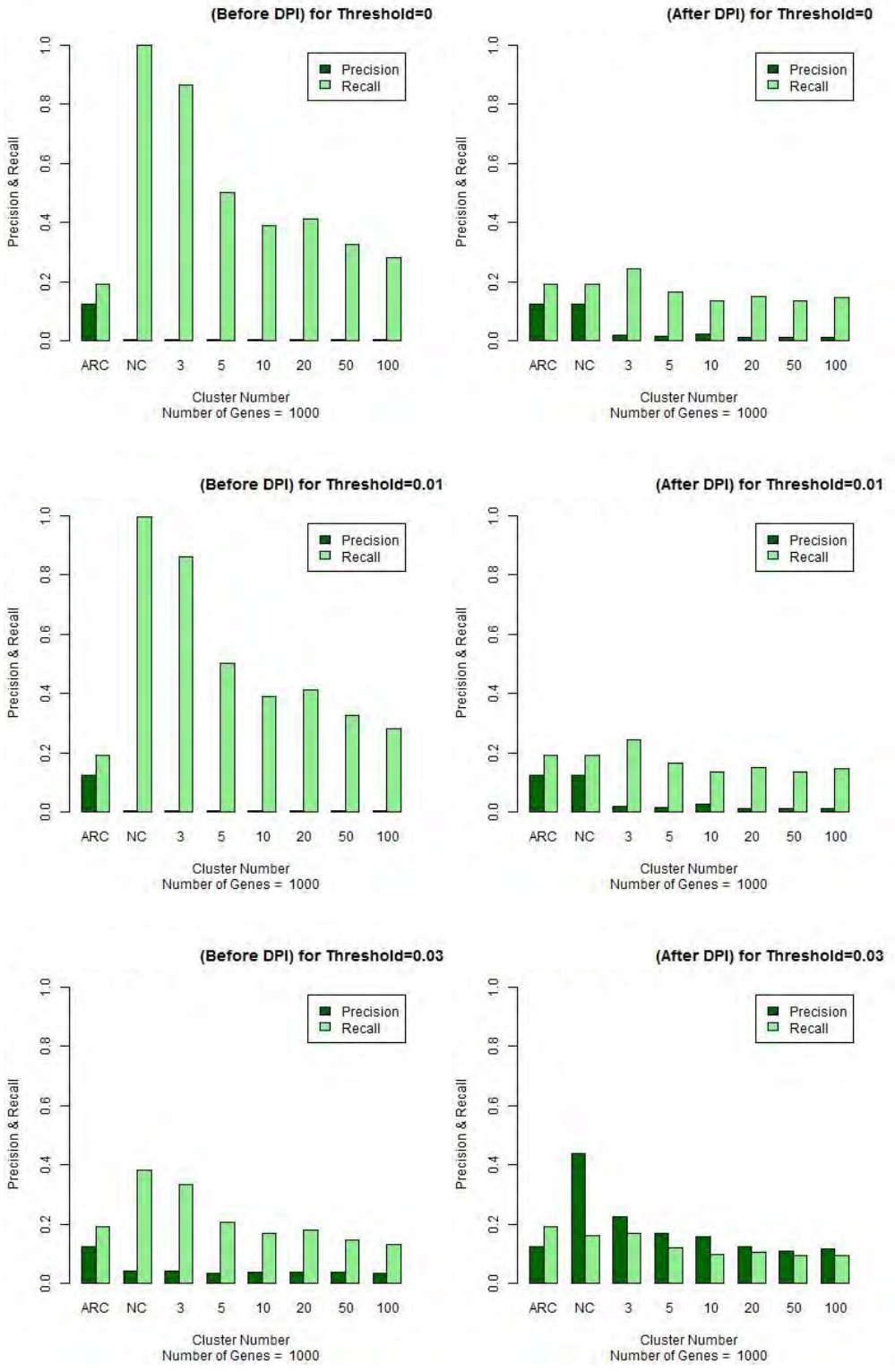

Figure 6.18: Precision and Recall for ARACNE, No-Clustering and Selected Merged Clustering. Before DPI (Left) and After DPI (Right) (Continued)

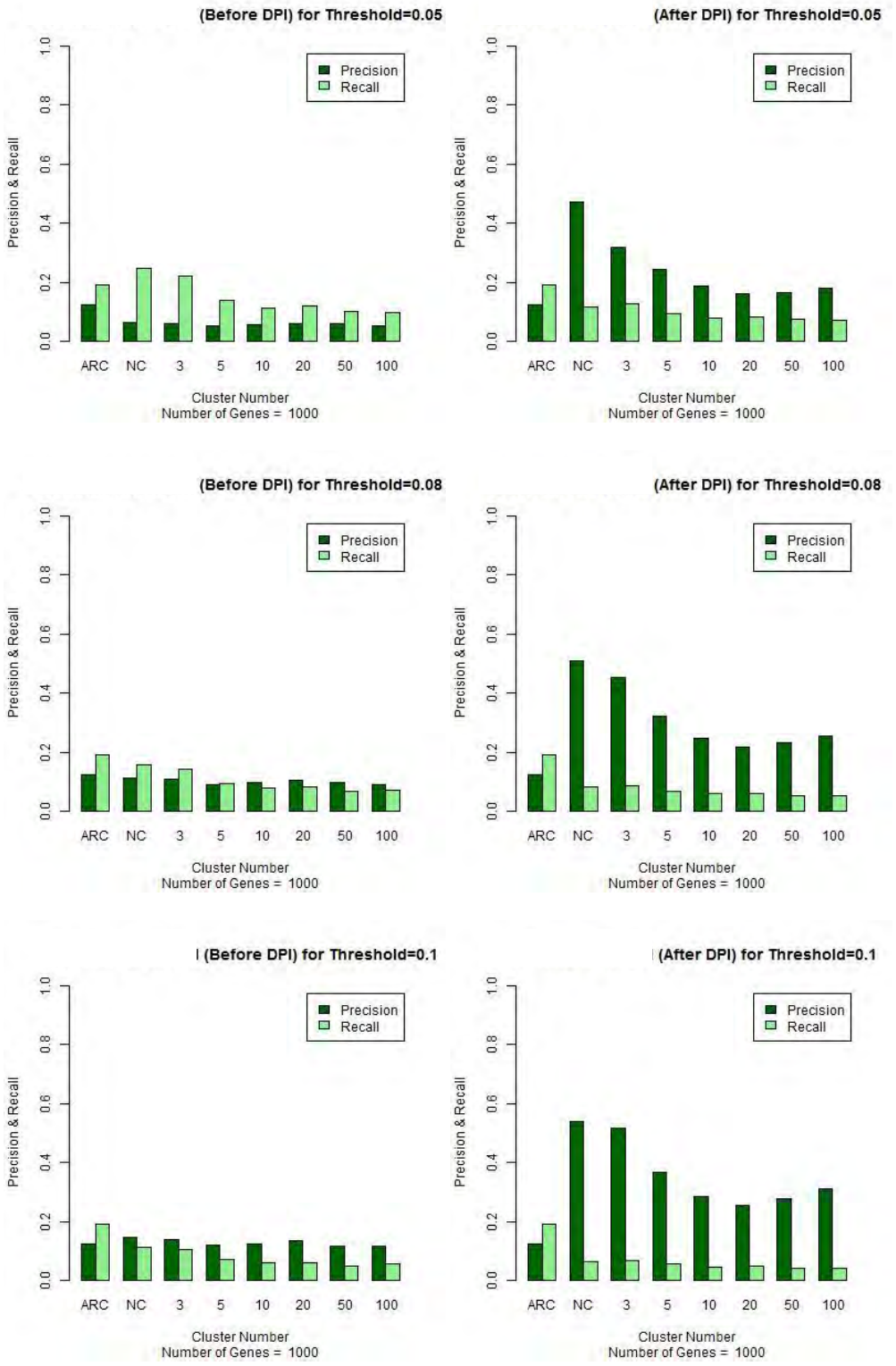

Figure 6.18: Precision and Recall for ARACNE, No-Clustering and Selected Merged Clustering. Before DPI (Left) and After DPI (Right) (Continued)

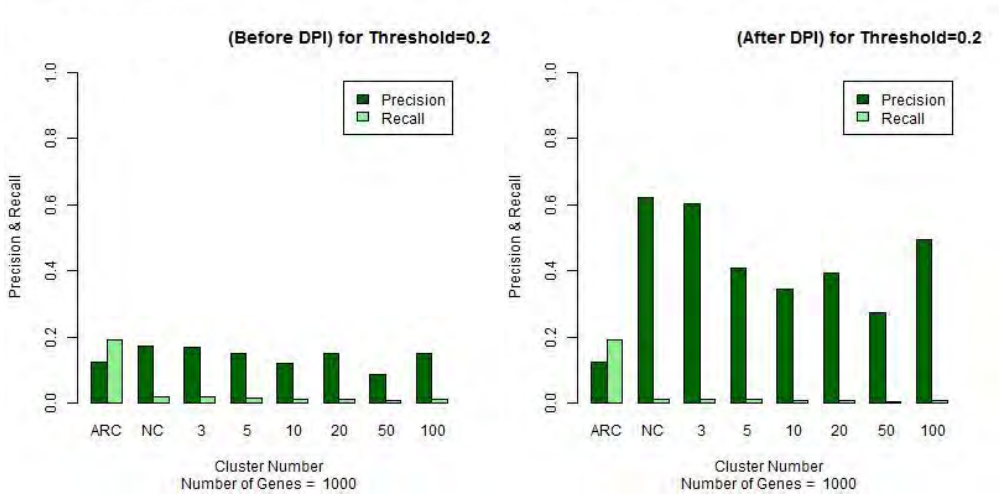

Figure 6.18: Precision and Recall for ARACNE, No-Clustering and Selected Merged Clustering. Before DPI (Left) and After DPI (Right)

Chapter 7

Discussions

The performance of our algorithm in inferring true regulatory interactions before using DPI technique showed great improvements for lower threshold values over the ARACNE approach. We also found that No-Clustering version of our algorithm was ranked highest among ARACNE, Unmerged Clustering and Selected Merged Clustering version of our algorithm in identifying true interactions before using DPI technique. But lower values for the number of clusters in selected merged version also produced similar results to No-Clustering version for finding true interactions. For higher values of cluster numbers, both the true positives and false positives were reduced comparing to lower values of cluster numbers. Using threshold values lower than the mean of all non-zero values of final connection matrix generated by the algorithm which contained the reduced entropy for each pair of genes, resulted in high number of true and

false positives. For values greater than the mean, the reduction of false positives was much greater than the reduction of true positives. As it is evident from the graph of precision and recall for different cluster numbers, our algorithm produced very high recall rate before using DPI technique than ARACNE. But after using the DPI technique, the results were similar with that of ARACNE which suggests the applicability of using the entropy reduction technique to identify statistically significant interactions. Another important observation from the precision and recall graph is that, using smaller threshold values our algorithm produced high recall rate. On the other hand, using higher threshold values produced high precision rate. So for our algorithm to be used in real applications, reasonable threshold values depending on the goal of the task have to be chosen for it to perform well.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

Using clustering algorithm with entropy reduction is a novel approach in inferring gene regulatory network. With Selected Merged Clustering, our algorithm produced large number of false positives comparing to true positives which contributed to low precision rate. Using a high number of clusters reduced the number of false positives but as many true positives were also eliminated, the recall rate became low. The entire process of clustering genes and then using ERT on each individual cluster and selected merged clusters is a time consuming process. For few numbers of clusters, the algorithm took considerable amount of time to generate the regulatory network.

8.2 Future Work

For the clustering part of our algorithm we have only used K-means clustering for its simplicity. But in future we want to use different types of clustering algorithm such as spectral clustering and affinity propagation clustering with the Entropy Reduction Technique to compare the performance with our current approach. For the Selected Merged Clustering part of the algorithm we hope to use better measurement to identify close clusters. For the current implementation of our algorithm, we consider each gene pair to have true interactions if the conditional entropy is less than the single entropy. We hope to find some threshold values from the data to consider interactions to be true only if they are above these threshold values. With the current approach, as we do not use any information regarding the experiments, we are unable to give direction to the edges. In future we hope to use time series data with the entropy reduction technique to infer directions. We also want to use Parallel Processing for the ERT part of our algorithm to reduce the amount of time it takes to infer regulatory interactions.

Bibliography

1. Marbach, Daniel, James C Costello, Robert Küffner, Nicole M Vega, Robert J Prill, Diogo M Camacho, Kyle R Allison, et al. "Wisdom of Crowds for Robust Gene Network Inference." NatureMethods 9, no. 8 (July 15, 2012): 796–804.
2. Villaverde, A. F., Ross, J., Morán, F., & Banga, J. R. (2014). MIDER: Network Inference with Mutual Information Distance and Entropy Reduction. *PLoS ONE,9*(5). doi:10.1371/journal.pone.0096732
3. Margolin, A. A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Favera, R., & Califano, A. (2006). ARACNE: An Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context. *BMC Bioinformatics,7*(Suppl 1). doi:10.1186/1471-2105-7-s1-s7
4. Dimitrakopoulos, G. N., Maraziotis, I. A., Sgarbas, K., & Bezerianos, A. (2014). A Clustering based Method Accelerating Gene Regulatory Network Reconstruction. *Procedia Computer Science,29*, 1993-2002. doi:10.1016/j.procs.2014.05.183
5. Maetschke, S. R., Madhamshettiwar, P. B., Davis, M. J., & Ragan, M. A. (2013). Supervised, semi-supervised and unsupervised inference of gene regulatory networks. *Briefings in Bioinformatics,15*(2), 195-211. doi:10.1093/bib/bbt034
6. Sage Synapse : Contribute to the Cure. (n.d.). Retrieved April 08, 2017, from https://www.synapse.org/#!Synapse:syn2787209/wiki/70349
7. Sage Synapse : Contribute to the Cure. (n.d.). Retrieved April 08, 2017, from https://www.synapse.org/#!Synapse:syn3049712/wiki/74628
8. Schaffter, T., Marbach, D., & Floreano, D. (2011). GeneNetWeaver: in silico benchmark generation and performance profiling of network inference methods. *Bioinformatics, 27*(16), 2263-2270. doi:10.1093/bioinformatics/btr373
9. Lee, W., & Tzou, W. (2009). Computational methods for discovering gene networks from expression data. *Briefings in Bioinformatics*. doi:10.1093/bib/bbp028
10. Jiang, D., Tang, C., & Zhang, A. (2004). Cluster analysis for gene expression data: a survey. *IEEE Transactions on Knowledge and Data Engineering, 16*(11), 1370-1386. doi:10.1109/tkde.2004.68
11. Tan, P., Steinbach, M., & Kumar, V. (2015). *Introduction to data mining*. Dorling Kindersley: Pearson.
12. Valentini, G. (n.d.). *Hierarchical clustering for gene expression data analysis*. Lecture.
13. Butte, A. J., & Kohane, I. S. (1999). Mutual Information Relevance Networks: Functional Genomic Clustering Using Pairwise Entropy Measurements. *Biocomputing 2000*. doi:10.1142/9789814447331_0040
14. Mordelet, F., & Vert, J. (2008). SIRENE: supervised inference of regulatory networks. *Bioinformatics,24*(16), I76-I82. doi:10.1093/bioinformatics/btn273
15. Cerulo, L., Elkan, C., & Ceccarelli, M. (2010). Learning gene regulatory networks from only positive and unlabeled data. *BMC Bioinformatics,11*(1), 228. doi:10.1186/1471-2105-11-228

16. Gene regulatory network. (2017, April 13). Retrieved April 17, 2017, from https://en.wikipedia.org/wiki/Gene_regulatory_network#/media/File:Gene_Regulatory_Network.jpg