MACHINE LEARNING METHODS FOR MICROARRAY DATA ANALYSIS

by

Prasad Amaresh Gabbur

A Dissertation Submitted to the Faculty of the

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

In Partial Fulfillment of the Requirements
For the Degree of

DOCTOR OF PHILOSOPHY

In the Graduate College

THE UNIVERSITY OF ARIZONA

2010

THE UNIVERSITY OF ARIZONA
GRADUATE COLLEGE

As members of the Dissertation Committee, we certify that we have read the dissertation prepared by Prasad Amaresh Gabbur
entitled Machine Learning Methods for Microarray Data Analysis
and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

Kobus Barnard — Date: 28 April 2010

Jeffrey Rodriguez — Date: 28 April 2010

Hong Hua — Date: 28 April 2010

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copies of the dissertation to the Graduate College.
I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

Dissertation Director: Kobus Barnard — Date: 28 April 2010

STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at the University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

SIGNED: Prasad Amaresh Gabbur

ACKNOWLEDGEMENTS

Thanks are due to many people without whom this work would not have been possible. Firstly, my thanks go to the Arizona Biomedical Research Commission (ABRC) for providing me with the financial support to do this research and write the thesis. I am also grateful to the 3D Visualization and Imaging Systems Lab (3DVIS), BIO5 Institute (Rounsley Lab) and Arizona Research Labs (Hildebrand Lab) at the University of Arizona, who provided me with much needed support during the course of my doctoral study. Thanks to James Hoying and Kevin Greer for their microarray data and the CARMA tool.

I owe my deepest gratitude to my advisor, Kobus Barnard, for being a great mentor and keeping me inspired, motivated and challenged all these years. Working with him has taught me a great deal about research and life in general. Many thanks to my committee members Jeffrey Rodriguez and Hong Hua for taking time to review this work and provide useful feedback. I wish to thank Moshe Shaked for his course, which has been particularly helpful in carrying out this research.

Many thanks to the faculty of the Electrical and Computer Engineering department for providing me with a great learning experience through their excellent courses. I would like to thank the staff of both the Electrical and Computer Engineering and Computer Science departments for their help on a number of occasions. I cannot overstate my gratitude to my teachers Malur Sundareshan, Robin Strickland, Michael Marcellin, Subbanna Bhat, Sumam David, K.R. Ramakrishnan, S.R. Mathpathi, and A.Y. Naik, for their knowledge, support and inspiration to pursue a research-oriented path.

Special thanks to my colleagues and friends at the Computer Vision Lab, especially Joseph Schlecht, Quanfu Fan, and Ranjini Swaminathan for many thought provoking discussions. Writing this thesis would not have been possible without the patience and moral support from my parents, sisters and relatives. I owe a great deal to my grandmother Shantabai Patil for instilling in me some of her virtues during my formative years.

DEDICATION

This thesis is dedicated to my father who taught me that one of the most helpful virtues to have is patience, to my mother whose prayers and faith in God helped us overcome many obstacles during this path. To my sisters, Swetha and Deepa, who have been a great source of moral support all through. To my grandmother for her wisdom and her family for their support.

Also this thesis is dedicated to all my friends and colleagues who offered their unconditional help.

TABLE OF CONTENTS

TABLE OF CONTENTS – *Continued*

LIST OF FIGURES

LIST OF FIGURES – *Continued*

LIST OF FIGURES – *Continued*

LIST OF TABLES

ABSTRACT

Microarrays emerged in the 1990s as a consequence of the efforts to speed up the process of drug discovery. They revolutionized molecular biological research by enabling monitoring of thousands of genes together. Typical microarray experiments measure the expression levels of a large number of genes on very few tissue samples. The resulting sparsity of data presents major challenges to statistical methods used to perform any kind of analysis on this data. This research posits that phenotypic classification and prediction serve as good objective functions for both optimization and evaluation of microarray data analysis methods. This is because classification measures what is needed for diagnostics and provides quantitative performance measures such as leave-one-out (LOO) or held-out prediction accuracy and confidence. Under the classification framework, various microarray data normalization procedures are evaluated using a class label hypothesis testing framework and also employing Support Vector Machines (SVM) and linear discriminant based classifiers. A novel normalization technique based on minimizing the squared correlation coefficients between expression levels of gene pairs is proposed and evaluated along with the other methods. Our results suggest that most normalization methods helped classification on the datasets considered except the rank method, most likely due to its quantization effects.

Another contribution of this research is in developing machine learning methods for incorporating an independent source of information, in the form of gene annotations, to analyze microarray data. Recently, genes of many organisms have been annotated with terms from a limited vocabulary called Gene Ontologies (GO), describing the genes' roles in various biological processes, molecular functions and their locations within the cell. Novel probabilistic generative models are proposed for clustering genes using both their expression levels and GO tags. These models

are similar in essence to the ones used for multimodal data, such as images and words, with learning and inference done in a Bayesian framework. The multimodal generative models are used for phenotypic class prediction. More specifically, the problems of phenotype prediction for static gene expression data and state prediction for time-course data are emphasized. Using GO tags for organisms whose genes have been studied more comprehensively leads to an improvement in prediction. Our methods also have the potential to provide a way to assess the quality of available GO tags for the genes of various model organisms.

CHAPTER 1

INTRODUCTION

Microarrays were invented in the 1990s as a consequence of the efforts to speed up the process of drug discovery (Lenoir and Giannella, 2006) . Traditional drug discovery was based on developing a number of candidate drugs and trying them one-by-one against diseases of interest. The idea is to bring a good match between a ligand (molecule in the drug) and a disease target (e.g. protein) to which the ligand binds either stimulating or inhibiting the action of the target. It was later found that ligands could be synthesized as polypeptide sequences and the chemical diversity in these sequences amounted to diversity in the candidate ligands. The lengthy and expensive process of trial-and-error based drug discovery could be parallelized by synthesizing polypeptide sequences combinatorially from their basic building blocks–amino acids. A group of scientists at Affymax developed a photolithographic technique to accomplish this in a fashion similar to the synthesis of VLSI (Very Large Scale Integration) chips in the semiconductor industry (Fodor et al., 1991) . By selectively laying out particular amino acids at specific sites on a glass slide layer-by-layer, with the help of photo-protective masks, they synthesized spatially addressable polypeptide chains on the slide. This gave rise to the idea of synthesizing similarly spatially addressable nucleic acid chains from their building blocks–the four nucleic acid bases (A, T, G, and C). The first version developed by the sister company Affymetrix came to be known as the GeneChip. Simultaneous to these efforts, researchers at Pat Brown's lab of Stanford University (Eisen and Brown, 1999) developed a different type of microarray. They devised spotted cDNA microarrays by depositing small amounts of probe cDNA sequences at particular locations on a substrate. There are a few differences between the GeneChips or oligonucleotide arrays and spotted cDNA microarrays in how they are designed to measure the expression levels of target genes.

1.1 GeneChips versus Spotted cDNA arrays

GeneChips or oligonucleotide arrays measure a particular gene by making use of short (oligo) sequences of nucleotides spread across the chip. These are usually referred to as *probes* and are a few tens of bases in length, e.g. 25-mers. There are two sets of probes for a gene. Each probe in the first set (*Perfect Match (PM)*) is synthesized to be exactly complementary to a subset of the entire nucleotide sequence of the gene. The probes in the second set (*Mismatch(MM)*) have the same nucleotide sequences as their *PM* counterparts except that one nucleotide at the center is different. The probes in the *PM* set bind specifically to their target gene's mRNA or cDNA sequence due to their perfect complementary sequences. Probes in the *MM* set help determine non-specific binding because of the mismatch in the central nucleotide. The signal from the *MM* probes is subtracted from the corresponding *PM* probe signal to adjust for cross hybridization. GeneChips are manufactured on a large scale using a photolithographic process to grow *PM* and *MM* subsequences of target genes on a substrate.

Spotted cDNA microarrays have printed spots on a substrate where each spot could contain the entire complementary sequence of a target gene. Usually the spots are laid out using a robotic arrayer or an inkjet printing based technology. In this method it is not necessary to know the sequence information of a probe gene beforehand. And the entire sequence for a particular gene, regardless of its length, can be laid out at a single site on the microarray as opposed to using a number of short probe sequences spread across multiple sites on a GeneChip. Since the cost involved in preparing a spotted microarray is smaller compared to that of the oligonucleotide arrays, it is used by a number of researchers to build in-house chips for specific experiments.

To measure gene expression levels in a tissue, mRNA or the corresponding transcribed cDNA sample is prepared and labeled with a fluorescent dye which absorbs and emits a specific wavelength of light energy. The dyes in common use are referred to as Cy3 and Cy5, which emit red and green wavelengths respectively upon

excitation by light of suitable wavelengths. The sample is then allowed to hybridize with a microarray where the mRNA or cDNA in the sample bind to spots having their complementary sequences. The spots are then examined by shining a laser light of appropriate wavelength and recording the light emitted due to fluorescence of the material bound to the spots. The higher the expression level of a gene the greater the intensity of the corresponding spot(s). So the measured intensity level of a spot is an indicator of the abundance of the target gene in the sample.

As opposed to a single tissue hybridized to a GeneChip microarray, usually two samples are competitively hybridized to a spotted cDNA array. The goal is to measure relative abundance of genes between the two samples. Competitive hybridization leads to a spot being either green, red or a combination of the two colors in the scanned image. Hence spotted arrays are also called two-color or two-channel microarrays. The redness or greenness of a spot is an indicator of the relative abundance or dearth of a gene in a particular sample. Competitive hybridization is typical of experiments where a sample of interest (e.g. disease) is being examined for differentially expressed genes relative to another sample (e.g. normal). The spots corresponding to such genes ideally appear green or red but not yellow.

1.2 Data pre-processing

The data from microarrays is generally not directly usable for analysis purposes. This is because apart from the interesting biological variations there are a number of other sources of variation, which confound any kind of analysis carried out directly on the data. After a microarray slide is hybridized to a sample it is scanned by exciting the fluorescent material bound to the sample and recording an image of intensities for each fluorescent channel. A typical scanned image for a GeneChip array and a spotted cDNA array is shown in Fig. 1.1

The recorded images are put through a series of pre-processing steps to extract useful gene expression values from them. A typical microarray data preprocessing pipeline is illustrated with the help of a flowchart in Fig. 1.2. The first step is to

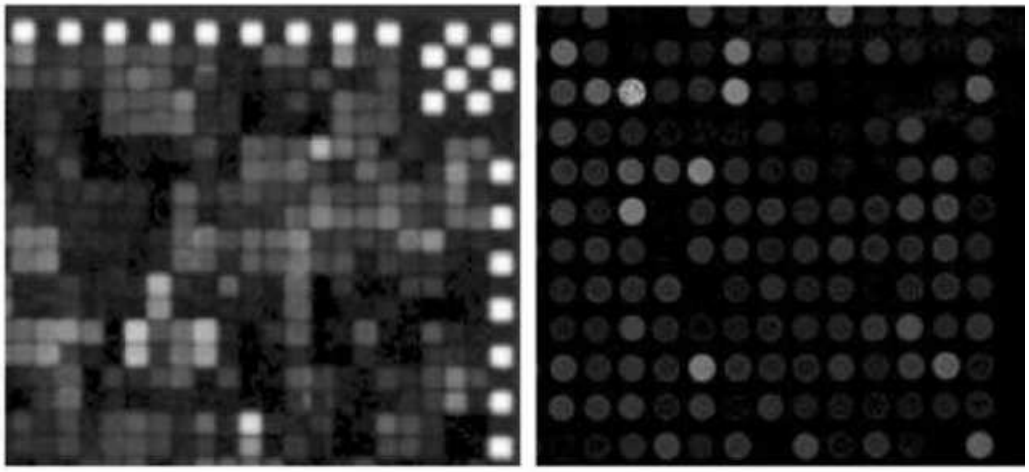


Figure 1.1: Sample scanned subimages of hybridized microarrays. The left image is an Affymetrix GeneChip array and the right one is an Agilent spotted cDNA array (Bolstad, 2007) . The spot intensities are proportional to the abundance of the corresponding genes but also have noise embedded in them.

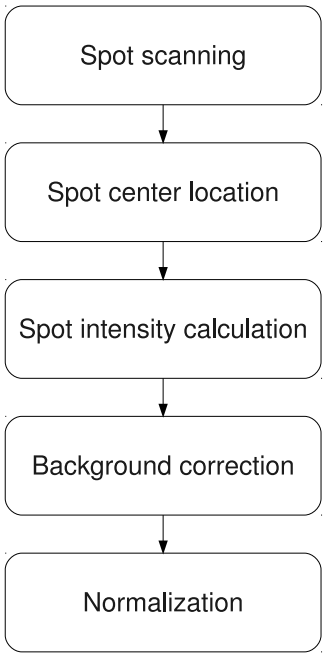


Figure 1.2: Microarray data pre-processing pipeline. Scanned image of microarray spot intensities is usually noisy and is subject to a number of preprocessing steps to locate the spots precisely, estimate their intensities robustly and correct for background activity and systematic technological noise.

locate the spots precisely on the chip. Usually a set of reference or control spots are laid out on the array to help align the spots on the array with a reference grid. For example, these are seen as bright square spots along the border of the Affymetrix scanned image in Fig. 1.1. The control spots are always bright or dark regardless of the hybridized sample. By aligning the control spots with the corresponding spots on the reference grid, the centers of the other spots on the array are located as the centers of the corresponding ones on the reference grid. Further measurements for each spot are taken within the localized region for that spot.

After the spots are precisely localized, a robust estimate of spot intensity is computed. In the case of GeneChip arrays, usually a higher quantile (e.g. 75th

percentile) of the histogram of pixel intensities within the spot region is used. For spotted cDNA arrays the spots are *segmented* out using appropriate intensity thresholds into foreground and background regions. A statistic (mean or median) of the foreground region serves as the spot intensity estimate. The spot intensities themselves are not pure estimates of gene expression because they generally include background activity due to effects such as noise and cross-hybridization. These effects are reduced in a background correction step.

There are a number of background correction methods for both GeneChip and spotted cDNA arrays. Usually the idea is to get an estimate of intensity due to background activity and subtract it from the spot intensity estimated in the previous step. For example, the Affymetrix MAS 5 Suite (Affymetrix, 2001) grids the entire array into a certain number of blocks and for each block computes the average spot intensity of the lowest 2% spots in the block. It then uses a weighted mean of these averages as estimate of background for a spot. The RMA and GCRMA software (Wu et al., 2004) fit a statistical model to the spot intensities and determine the background activity as a random variable for each spot. In the case of cDNA arrays the segmented background pixels can be used to estimate the background activity. A number of software suites perform this segmentation and use the result in different ways (Yang et al., 2002a) .

Apart from cross-hybridization there are other sources of noise that potentially confound gene expression data analysis. These are due to experimental effects, which manifest as systematic spurious variation in the data. The main causes are the dye bias, labeling efficiency, sample mRNA concentration, spot layout and scanner settings. A simple way to understand this variation is by considering an experiment where two mRNA samples prepared from the same source but labeled with different dyes are hybridized to two microarrays of the same type. The resulting spot intensities on one array can be approximated as scaled factors of the corresponding spots on the other array. The scaling factor takes into account the different binding affinities of the two dyes to the samples, the sample mRNA concentrations and different scanner settings when the two images were taken. Further this scaling factor may

be a function of the spot intensity due to various causes including an effect called *quenching*–a process where dye molecules reabsorb different amounts of fluorescent light from each other depending on their concentration. In the case of spotted cDNA arrays the scaling factor may be a function of the spot location because spots are usually laid out as separate grids on the array using separate print-heads. A suitably designed normalization method is used to adjust for the different systematic experimental effects introduced into the microarray data.

Normalization approaches to oligonucleotide arrays differ slightly from that of two-channel microarrays. The main difference is that the approaches for two-channel arrays do both within-array and between-array normalization while the approaches for oligonucleotide arrays do only the latter. In general the normalization methods for oligonucleotide arrays are directly applicable to two-channel arrays by treating each channel as a separate array. The widely used approaches to normalizing single channel arrays include geometric (Szabo et al., 2002) , rank (Tsodikov et al., 2002) and quantile normalization (Bolstad et al., 2003) . For normalizing between two channels of a single array, lowess normalization (Yang et al., 2002b) is often used. The problem of normalization is one of the main focus of this work and will be addressed in more detail in the subsequent chapters.

1.3 Microarray data analysis and evaluation

Regardless of the technology, microarrays make it possible to monitor the expressions or expression profiles of thousands of genes together. This enables identifying genes involved in the regulation of important biological processes. Observing the co-expression between genes allows determining interactions between them that work towards regulating a particular biological process. However, the extraction of useful information from microarray data requires modeling the noise in the data. A typical microarray dataset involves an assay of a set of genes under two or more different biological conditions. The expression measurements under each biological condition may have been taken using multiple tissue samples, possibly from different subjects,

to allow for biological averaging. Many experimental designs use combinations of different arrays, dyes and tissue samples. This is helpful in estimating the variability in data due to each of those experimental factors. However, a microarray experiment is expensive and as a result there are only a few samples compared to the number of genes assayed. This leads to the problem of working with very high dimensional data that is sparse–very small number of samples. The sparsity of data also makes evaluation of statistical models a challenging task.

Typical microarray experiments measure the expression levels of a large number of genes on very few tissue samples. The resulting sparsity of data presents major challenges to statistical methods used to perform any kind of analysis on this data. And more often than not there is no ground truth to compare the performances of different analysis methods. This research posits that phenotypic classification and prediction serve as good objective functions for both optimization and evaluation of microarray data analysis methods. This is because classification measures what is needed for diagnostics and provides quantitative performance measures such as leave-one-out (LOO) or held-out prediction accuracy and confidence. Since LOO performance is an indicator of generalization ability of a data analysis method, it helps predict the behavior of the particular analysis method with novel data. This is critical since the intention of designing any learning based analysis method is to apply it on new data.

As a first step, this research evaluates various normalization procedures using a classification framework based on class label permutation analysis and also using support vector machines (SVM) and linear discriminant based classifiers. A novel normalization technique is proposed based on minimizing the squared correlation coefficients between expression levels of gene pairs in a microarray experiment. The proposed normalization method is evaluated along with other methods within the classification framework. There has been no previous work using real microarray datasets to perform this kind of evaluation. The evaluations help quantify the extent to which normalization methods can be optimized towards achieving a high prediction accuracy. It is natural to ask if a normalization method can be designed

to maximize a classification objective. An attempt to answer this question in the maximum margin framework of linear SVMs is explored. This is because a SVM maximizes the margin between the class boundary and the nearest training sample, thereby providing a convenient objective function to represent classifier generalizability (Vapnik, 1998; Bishop, 2006) .

1.3.1 Use of Gene Ontologies

A second contribution of this research is in developing machine learning methods for incorporating an independent source of information, in the form of gene annotations, to analyze microarray data. Recently, genes of many organisms have been annotated with terms from a limited vocabulary called Gene Ontologies (GO) (Ashburner, 2000) . These terms describe the genes' roles in various biological processes, molecular functions and their locations within the cell. Clustering is frequently used with microarray data to group together genes exhibiting similar expression profiles, e.g. over samples or time. This is a form of data compression with the assumption that the genes involved in the same biological processes end up in the same cluster. Noise in the expression data often confounds these clustering methods resulting in a grouping of genes that are not necessarily involved in the same biological function(s). Novel generative models for clustering using both gene expression levels and GO tags are explored to reduce this confound. These models are similar in essence to the ones used for multimodal data, such as images and words (Barnard et al., 2001; Barnard and Forsyth, 2001) .

The multimodal generative models are used for phenotypic class prediction to demonstrate that using GO tags in conjunction with quantitative gene expression data leads to an improvement in prediction. More specifically, the problem of state prediction for time course gene expression data and phenotype prediction (e.g. disease) for static data is emphasized. To account for different behaviors and correlations of gene expressions across samples, models assuming independence between samples as well as those pooling them together are considered. The models assuming independence between time points include multimodal variants of a mixture of

Gaussians (GMM). Constrained Hidden Markov Models (HMM) (Rabiner, 1989) and their mixtures with multimodal likelihood functions are used for joint modeling of expression data and GO tags. Prediction on novel data is done in a Bayesian maximum likelihood framework using the optimized parameters learnt during training.

Experiments are performed using a number of microarray datasets including the colon cancer data of Alon et al. (Alon et al., 1999) and the angiogenesis data of Hoying et al. (Greer et al., 2006) . Time-course gene expression data and data with multiple phenotypic classes (states) are also analyzed within the proposed framework. More specifically, the time-course datasets are the yeast cell cycle data of Cho et al. (Cho et al., 1998) and human cell cycle data of Whitfield et al. (Whitfield et al., 2002) in addition to the Hoying angiogenesis data.

The following chapter introduces a few widely used normalization methods for microarray data. A new normalization method based on minimizing the correlation between expression profiles of pairs of genes is also proposed and its ability to do so is demonstrated on a synthetically generated microarray dataset. Chapter 3 evaluates the various normalization methods on the phenotype class label prediction task. A hypothesis testing framework based on class label permutation analysis is used for the evaluations along with two classifiers: Golub linear discriminant classifier and a Support Vector Machine with Recursive Feature Elimination (SVM-RFE). Gene Ontologies (GO) are introduced in Chapter 4 along with the formulation of a number of probabilistic generative models for clustering genes using their expression profiles and associated GO tags. A Bayesian maximum likelihood prediction framework using the proposed models is also developed in the same chapter. Experimental results on the task of phenotype class prediction and state prediction in the case of static and time-course microarray data respectively are reported in Chapter 5.

CHAPTER 2

NORMALIZATION

Regardless of the technology, microarrays make it possible to monitor the expressions or expression profiles of thousands of genes together. This enables identifying genes involved in the regulation of important biological processes. Observing the co-expression between genes points to interactions regulating particular biological processes. Another use of microarray data is to provide diagnostic indicators. In this case, the main goal is to use the data to predict the presence or stage of a biological process. For example, the classification problem might be associating a tissue sample with a particular type of cancer or one of its possible subtypes. These two tasks are related — genes that can be used to classify are either involved in the underlying process or strongly correlated with it.

Success at either of these tasks has many applications. However, exploiting microarray technology to address them is hampered by the sparseness of the data. In particular, the number of predictors (genes) is typically much larger than the number of replicates over cases. This means that standard tests of statistical significance need substantive adjustments that require strong assumptions on the underlying statistics which are generally not well known. Often the only realistic goal is to find candidates of relevant genes for further study or a diagnostic suite that is bound to include both useful and spurious genes. Hence we argue that classification performance, which measures what is wanted for diagnostics, is the preferred way to evaluate methods for preprocessing microarray data and subsequent gene selection.

Microarray data preprocessing typically targets removing experimental effects such as effects due to dyes or arrays. Doing so is referred to as normalization, as it often is achieved by scaling subsets of the data by factors associated with various experimental effects. Researchers generally assume that removing such effects are important, if not critical, for making use of microarray data. Hence a number of

normalization methods have been proposed, each with its own assumptions behind the nature of the sources for experimental variation (e.g. (Yang et al., 2002b; Qiu et al., 2005; Szabo et al., 2002)). Different methods transform the data in different ways to account for the experimental effects.

It is natural to ask how effective applying any of these methods to data are for the end goal. While ground truth for genes that are involved in a particular process is rare on a large scale, classification performance can be quantified using well adopted principles. Hence we study various normalization methods by using them as a preprocessing stage to prepare the data for subsequent gene selection method coupled with classification. In particular, we experiment with approaches proposed by Golub et al. (Golub et al., 1999) and Guyon et al. (Guyon et al., 2002) , and report classification accuracy and confidence on data held out from the gene selection processes and classifier training (cross-validation). Performance in this paradigm provides some indication as to how effective a proposed normalization method might be for an end goal. We further study normalization as a preprocessing stage for the class label hypothesis testing framework proposed by Golub et al. (Golub et al., 1999) as an alternative to strict cross validation.

2.1 Normalization methods

We consider five normalization strategies: geometric normalization, quantile normalization, rank normalization, δ-sequences and a new normalization technique proposed here based on minimizing the sum of squared correlation coefficients between pairs of genes (MIN-SS-CORR). Although the δ-sequences is not truly a normalization method, their computation entails the nullification of experimental effects to some extent (Klebanov and Yakovlev, 2007) . There is an important distinction between the normalization methods considered. Some normalization methods transform values on an array using information from only that array. Geometric, rank normalization and δ-sequences fall under this category. And others use information across all the arrays to do normalization. Ours and the quantile normalization

method fall under this category. In a previous work it was pointed out that methods of the latter category are favorable over the ones of the former category (Bolstad et al., 2003) .

2.1.1 Geometric normalization

Geometric normalization is a global linear normalization method with the assumption of an array-specific multiplicative effect. Data normalization is achieved by dividing the expression value of a gene on an array by the geometric mean of the expression values of all the genes on that array (Szabo et al., 2002) . Equivalently it amounts to normalizing each array to have zero mean in the log space by doing a global subtraction of an array-specific constant from the log transformed gene expression values. An interesting alternate view of geometric normalization is given in Appendix A.

2.1.2 Quantile normalization

Quantile normalization (Bolstad et al., 2003) is based on the assumption that if two random variables are identically distributed then their quantiles should be equal. This amounts to all the quantiles residing along the line from origin to $(1, 1)$ in the two-dimensional Cartesian space with the orthogonal axes defined by the two random variables' quantiles divided by the corresponding maximum value (100 percentile). The idea is extended to the case of more than two random variables. The gene expressions on each array are assumed to be realizations of a random variable corresponding to the condition being measured on the array. These are used to compute sample quantiles for each array, which correspond to points in a multidimensional space with the dimensionality equal to the number of arrays. The resulting points are linearly projected on to the diagonal $(1, 1, \ldots, 1)$ line and the gene expression values are replaced by the projected values. This causes the quantiles of the distributions corresponding to the arrays to be equal and their empirical distributions to be identical, thus achieving normalization. In a comparison with a

few other normalization methods, the quantile normalization was reported to reduce bias and variance of the normalized expression values relative to the raw expressions (Bolstad et al., 2003) .

2.1.3 Rank normalization

Rank normalization (Tsodikov et al., 2002; Szabo et al., 2002) replaces the expression value of a gene on an array by its normalized rank relative to all the genes on the array. For example, on a hypothetical array measuring 10 genes the least gene expression value corresponding to rank 1 gets replaced by 1/10. The ranking is evaluated based on the raw expression values. The assumption is that any systematic experimental effect produces a monotone transformation, not necessarily linear, on all the true gene expression values of a particular array. Therefore the ranking should be invariant to such a transformation. This assumption is similar to the geometric normalization method but allows for more general transformations than linear. But it comes at a cost of altering the actual gene expression measurements due to replacing them with their discrete ranks.

2.1.4 δ-sequences

The δ-sequence is a sequence of random variables computed adaptively from a microarray gene expression dataset (Klebanov and Yakovlev, 2007) . The log transformed gene measurements on the arrays are first re-ordered according to their sample variances computed across the arrays. The measurements of the gene with the highest sample variance are at the top followed by the measurements of the gene with the next highest variance and so on. With the resulting ordering of the genes, the differences between successive non-overlapping pairs of genes constitutes the δ-sequence of random variables. For each array there are half the number of variables in the δ-sequence as the number of genes. Although the purpose of the transformation is to obtain a significantly less correlated sequence of random variables, the δ-sequence computation achieves normalization indirectly. For example,

in the case of a multiplicative array-specific effect, the differencing operation of two log transformed measurements on the same array results in canceling out the effect term (Klebanov and Yakovlev, 2007) . So the resulting values are devoid of the effect term.

2.1.5 A new normalization method: Minimizing the correlation between gene expression levels (MIN-SS-CORR)

Klebanov et al.'s (Klebanov and Yakovlev, 2007) work on δ-sequences suggests a possible normalization method for microarray data sets. They note that most of the positive correlations observed in the raw data or the log-transformed data are due to systematic experimental effects. A simple and generally accepted way to model some of these effects is to assume an array-specific multiplicative effect. Then a constant additive term forms a part of the measured gene expression levels in the log space. If x_{ij} is the true expression value of the ith gene on the jth array and x^*_{ij} is its observed counterpart, then the array-specific systematic effect ϵ_j manifests as

$$log(x^*_{ij}) = log(x_{ij}) + \epsilon_j \tag{2.1}$$

The systematic effect contributes to experimental variation of the observed measurements across experiments and also results in a positive correlation between gene pairs even if they are biologically uncorrelated. This can be easily verified, for example by assuming that the ϵ_j arise from a certain population sampled from a distribution having zero mean and finite variance. Consider a pair of genes whose true expression values are statistically independent of each other and also independent of the effects. Computing the correlation coefficient between their expression levels in the presence of systematic effects leads to a positive correlation coefficient between them, which would otherwise be zero.

This implies that we can exploit cross-correlation between gene-pairs to estimate the systematic array-specific experimental effects. More specifically, the idea is to estimate the offsets ϵ_j that minimize the cross-correlations between all pairs of genes

measured in an experiment. We propose to minimize the sum of squared correlation coefficients between all gene pairs over the space of offsets. This tends to minimize the spurious cross-correlations introduced by the array-specific experimental effects. More general models of experimental effects assume offsets that are different for different subgroups of genes. For example, subgroups laid out differently on the array can be associated with different offsets. It is possible to estimate these subgroup-specific offsets by considering all the gene pairs within the subgroups separately and minimizing the sum of their squared correlation coefficients. The method developed below can be easily adapted to the general case. However, here we address the simple case of array-specific effects.

Consider a microarray experiment where N genes are measured using a set of M arrays. Let g_{ij} denote the measurement (log-transformed) of the ith gene on the jth array. This measurement incorporates the effect ϵ_j as described in the equation 2.1. We are trying to estimate the ϵ_j's that minimize the sum of squared correlation coefficients between all pairs of genes p and q used for the experiment. Let o_j's denote our estimates of these array-specific offsets and $\tilde{g_{pi}}$ and $\tilde{g_{qi}}$ denote the offset adjusted counterparts of the corresponding observed measurements: $\tilde{g_{pi}} = (g_{pi} - o_i)$ and $\tilde{g_{qi}} = (g_{qi} - o_i)$. Similarly let the quantities $\tilde{\mu_p}$ and $\tilde{\mu_q}$ denote the sample means of genes p and q after offset adjustment. The correlation coefficient between the expression levels of genes p and q after accounting for the offsets is given by:

$$r(p,q) = \frac{C^2(p,q)}{\sqrt{C^2(p,p)C^2(q,q)}} \tag{2.2}$$

where, $C^2(p,q)$ is the covariance between the genes p and q and $C^2(p,p)$ and $C^2(q,q)$ are their variances respectively:

$$C^2(p,q) = \frac{1}{M}\sum_{i=1}^{M}(\tilde{g_{pi}} - \tilde{\mu_p})(\tilde{g_{qi}} - \tilde{\mu_q}) \tag{2.3}$$

$$C^2(p,p) = \frac{1}{M}\sum_{i=1}^{M}(\tilde{g_{pi}} - \tilde{\mu_p})^2 \tag{2.4}$$

$$C^2(q,q) = \frac{1}{M}\sum_{i=1}^{M}(\tilde{g_{qi}} - \tilde{\mu_q})^2 \tag{2.5}$$

The offsets O are estimated by minimizing the objective function

$$S(O) = \sum_{p=1}^{N}\sum_{q=1}^{N} r^2(p,q) \tag{2.6}$$

$$= \sum_{p=1}^{N}\sum_{q=1}^{N} \frac{(C^2(p,q))^2}{C^2(p,p)C^2(q,q)} \tag{2.7}$$

It is convenient to write the objective function S as a matrix vector product as

$$S = \boldsymbol{v}^T Q \boldsymbol{v} \tag{2.8}$$

For simplicity we have omitted the dependence of S on O. The vector $\boldsymbol{v}$ is the vector of inverse variances of all the genes:

$$\boldsymbol{v} = \begin{pmatrix} \frac{1}{C^2(1,1)} \\ \frac{1}{C^2(2,2)} \\ \vdots \\ \frac{1}{C^2(N,N)} \end{pmatrix} \tag{2.9}$$

and the matrix Q is obtained from the covariance matrix of offset adjusted gene expression levels by squaring the individual elements:

$$Q = \begin{pmatrix} (C^2(1,1))^2(C^2(1,2))^2 \ldots (C^2(1,N))^2 \\ (C^2(2,1))^2(C^2(2,2))^2 \ldots (C^2(2,N))^2 \\ \vdots\vdots\cdots\vdots \\ (C^2(N,1))^2(C^2(N,2))^2 \ldots (C^2(N,N))^2 \end{pmatrix} \tag{2.10}$$

There is no closed form solution for the offsets o_i that minimizes the objective function in equation 2.7. Therefore we estimate the offsets by gradient descent on the objective function over the space of offsets, starting from a suitable initial point. The initial point could be the vector of all zero offsets, which is used for all the

experiments reported here, or a random vector. The choice of a zero initial offset vector is with the optimistic assumption of no significant experimental effects.

The gradient of the objective function can be computed using the product rule of derivatives. The partial derivative of S with respect to a particular offset o_l can be expressed as

$$\frac{\partial S}{\partial o_l} = \left(\frac{\partial \boldsymbol{v}}{\partial o_l}\right)^T Q\boldsymbol{v} + \boldsymbol{v}^T \left(\frac{\partial Q}{\partial o_l}\right) \boldsymbol{v} + \boldsymbol{v}^T Q \left(\frac{\partial \boldsymbol{v}}{\partial o_l}\right) \tag{2.11}$$

The vector and matrix partial derivatives in the above formulation are easy to compute as we need to know only the partial derivative of the covariance term $C^2(p,q)$ for any gene pair (p,q) and the variance term $C^2(p,p)$ for any gene p. These are given by:

$$\frac{\partial C^2(p,q)}{\partial o_l} = \frac{1}{M}(2o_l - g_{pl} - g_{ql}) + \frac{1}{M^2}\sum_{i=1}^{M}(g_{pi} + g_{qi} - 2o_i) \tag{2.12}$$

$$\frac{\partial C^2(p,p)}{\partial o_l} = \frac{2}{M}(o_l - g_{pl}) + \frac{2}{M^2}\sum_{i=1}^{M}(g_{pi} - o_i) \tag{2.13}$$

Making use of the following expressions

$$\frac{\partial (C^2(p,q))^2}{\partial o_l} = 2C^2(p,q)\frac{\partial C^2(p,q)}{\partial o_l} \tag{2.14}$$

$$\frac{\partial \left(\frac{1}{C^2(p,p)}\right)}{\partial o_l} = -\frac{1}{(C^2(p,p))^2}\frac{\partial C^2(p,p)}{\partial o_l} \tag{2.15}$$

$$\tag{2.16}$$

it is possible to compute the individual derivative terms in equation 2.11 to obtain the partial derivative of the objective function with respect to a particular offset o_l and similarly for other offsets. At every iteration t of gradient descent, the offset vector O^t is updated by subtracting from it a vector of certain magnitude along the current gradient vector direction

$$O^{(t+1)} = O^t - \eta_t \nabla S(O^t) \tag{2.17}$$

In the above $\nabla S(O^t)$ denotes the gradient vector direction at iteration t. The rate of gradient descent η_t determines the magnitude of update at iteration t. We start with an initial value of 1.0 for this parameter and reduce it by half every time the descent procedure crosses over to the other side of the valley formed by a local minimum. This event can be determined by the gradients of two successive iterations being at an obtuse angle with each other.

If the offset vector O^* is a minimizer of the sum of square correlation coefficients then any vector O^*_λ such that $O^*_\lambda = O^* + \lambda \mathbf{1}$, where $\mathbf{1}$ is a vector of all 1's and λ is any arbitrary scalar, is also a minimizer. The λ can be chosen such that a particular offset goes to zero, which amounts to treating the particular array as the reference with respect to which the other arrays would be normalized.

Experiments on synthetic data

We tested the proposed normalization method on a synthetic dataset that simulates the assumptions of array-specific systematic effects in a microarray experiment. We consider a small scale experiment simulating the measurement of 500 genes on 10 arrays under two biological conditions. The first condition is represented by the first five arrays and the second condition by the next five. We start with a 500x10 matrix of all zeros. We then add independent zero mean unit variance Gaussian noise to all the elements to simulate measurement noise. In the first five columns, values of +2.0 and -2.0 are added to the first and second set of 125 genes respectively (i.e., the top-left 250x5 submatrix is changed). This simulates a situation where 25% of the genes are differentially over-expressed and another 25% are differentially under-expressed between the two conditions. This dataset has biological correlations among genes that are either over-expressed or under-expressed between the two conditions. Otherwise there are no correlations. A plot of the histogram of correlation coefficients between all pairs of genes is shown in

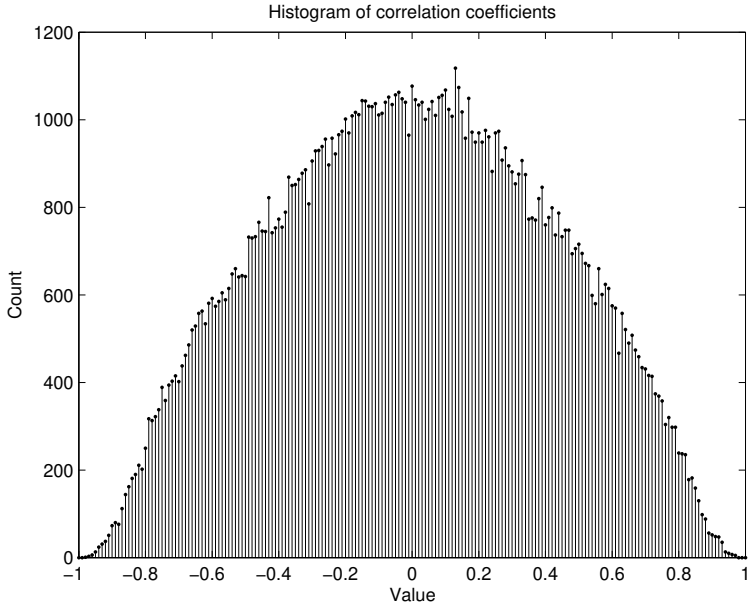


Figure 2.1: Histogram of correlation coefficients between all pairs of genes in the synthetic dataset. By construction, there are almost equal number of positive and negative correlations due to the equal number of over and under expressed genes in the two conditions. The majority of the gene pairs are uncorrelated because half of the total number of genes are not differentially expressed.

Fig. 2.1.

The array-specific multiplicative effects are simulated by adding a fixed offset to all the log transformed measurements in a particular column and repeating for all the columns. The fixed offsets for different columns are sampled from a zero mean Gaussian distribution with variance 5. This introduces spurious correlations between gene pairs and most of the correlations are closer to 1.0 than to 0.0 due to the large magnitude of the array-specific effects, as seen in the resulting histogram of correlation coefficients between gene pairs (Fig. 2.2). Although the magnitude of the effects seems a bit too extreme, this is not too far from reality based on what is observed in the correlation analyses of a number of real microarray datasets (Klebanov and Yakovlev, 2007) .

We use this dataset to test the ability of the proposed normalization method in recovering the array-specific systematic effects by minimizing the sum of square

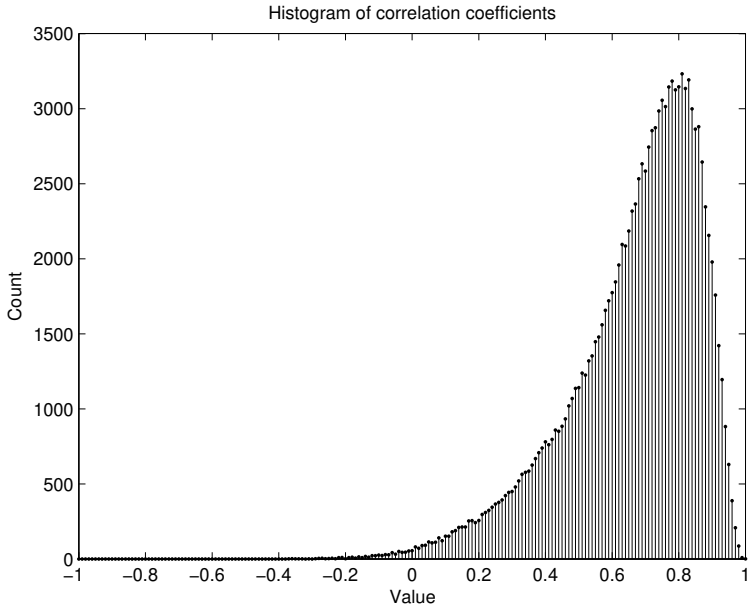


Figure 2.2: Histogram of correlation coefficients between all pairs of genes in the synthetic dataset after introducing array-specific systematic effects. Most of the correlation coefficients between genes are closer to 1.0 than to 0.0 due to the large relative variance of the systematic effects compared to the measurement noise variance.

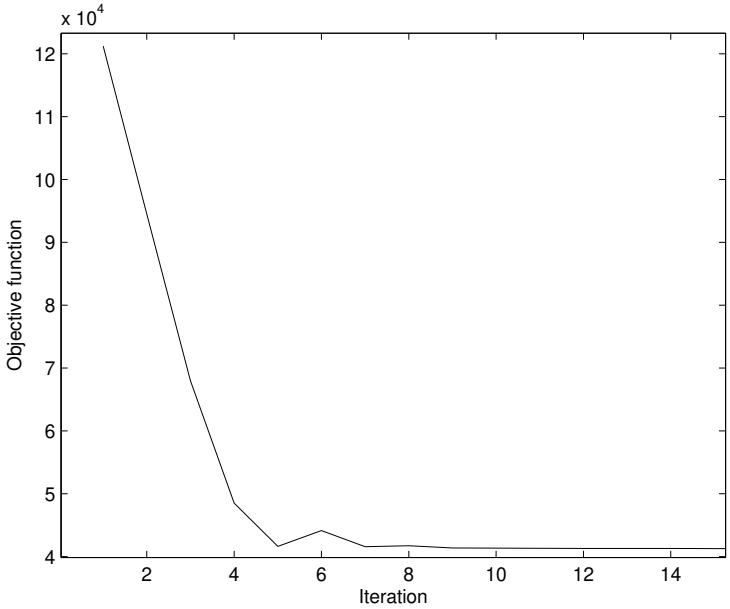


Figure 2.3: Trail of the objective function over the first 15 iterations of gradient descent starting from an all zero initial offset vector. The optimization procedure almost converges within this number of iterations indicating a well-behaved objective function. Starting from many different random initial points led to the same value of the objective function minimum.

correlation coefficients between all pairs of genes. Starting from a initial offset vector O^0 of all zeros, corresponding to no systematic effects, we perform 100 iterations of gradient descent. A local minimum is reached within the first 15 iterations and the trail of the objective function over those iterations is shown in Fig. 2.3.

The initial, true and recovered offset values are plotted in Fig. 2.4. The recovered offsets are close to the true offsets suggesting that the sum of square correlation coefficients is a reasonable objective function to minimize under the assumptions of array-specific effects. More so because the number of gene pairs in a microarray dataset is typically much larger than the number of systematic effects introduced in the experiment. The objective function is most likely well behaved with local minima corresponding to locations of the true systematic effects.

Subtracting the estimated offsets from the data should remove the spurious non-biological correlations between genes. Plotting the histogram of the correlation

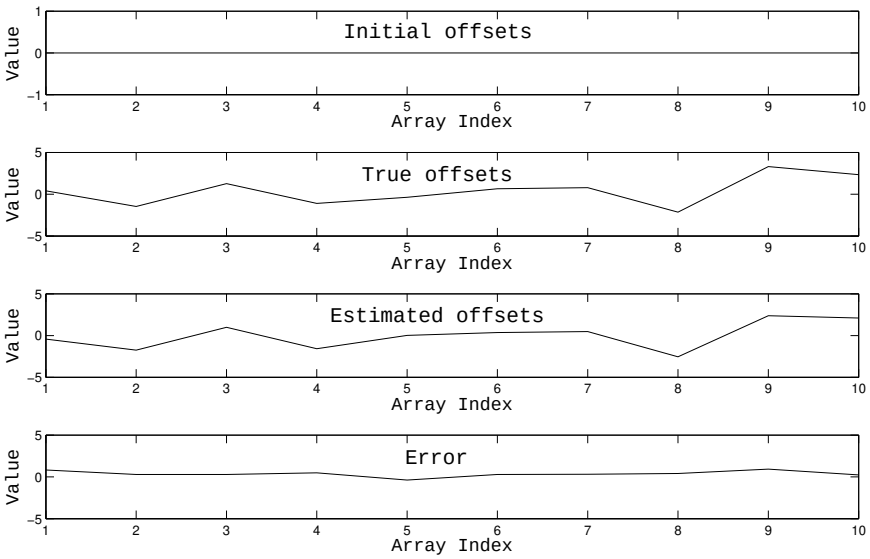


Figure 2.4: Profiles of the initial (top), true and estimated (middle) offsets representing systematic array-specific effects in the synthetic data. Starting from an all-zero offset vector the gradient descent procedure estimates the true offsets to a good accuracy. The root mean square error (bottom) in recovering the offsets is 0.5. Note that any other offset vector that is obtained by adding an arbitrary constant to the estimated offset vector elements above also corresponds to a local minimum as explained in section 2.1.5. However our optimization procedure has recovered an offset vector close to the true one given the initial starting point of an all-zero offset vector.

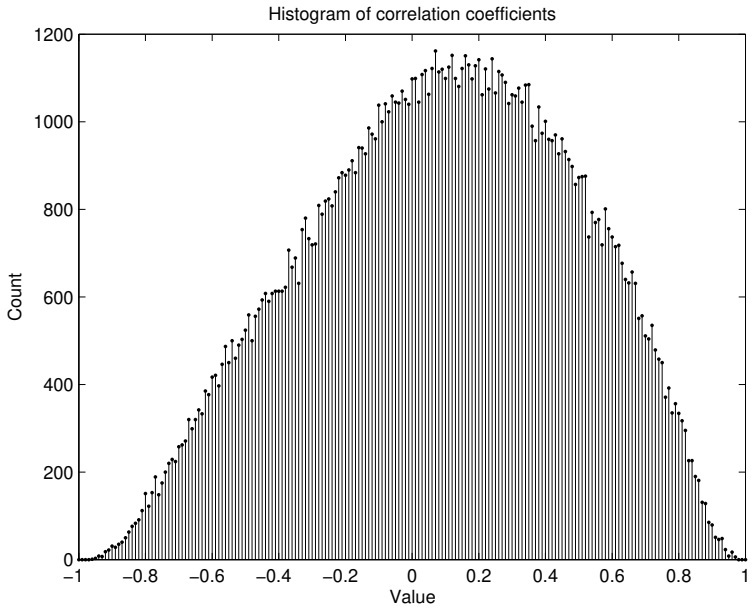


Figure 2.5: Histogram of correlation coefficients between gene pairs in the synthetic dataset with array-specific systematic effects after offset adjustment with offsets estimated by the gradient descent procedure. Most of the spurious correlations introduced by the array-specific effects are removed. The histogram has a structure similar to that obtained for the data prior to introducing the effects (Fig. 2.1), with a small bias towards positive correlation. This is perhaps due to the approximation of the offsets that are estimated. The error in estimating the offsets, as is evident from the bottom plot of Fig. 2.4 not being a perfectly horizontal line, still leaves a small residual spurious positive correlation in the data.

coefficients as before for the offset adjusted data in Fig. 2.5 shows that this is in fact the case.

CHAPTER 3

NORMALIZATION AND CLASSIFICATION

Our belief is that a normalization method should support the inherent classifiability of the data into its phenotypic classes. So we seek to evaluate the effects of normalization on the classifiability of the resulting data. To our knowledge there has been no previous work on the evaluation of normalization methods using real microarray data from this perspective. Hua et al. (Hua et al., 2006) use synthesized data to compare a few normalization methods and observe that those methods are generally beneficial for classification. Observing the effects on real data is perhaps more revealing due to the presence of effects that can not be effectively modeled. Other authors have tried to evaluate normalization techniques based on bias-variance (Bolstad et al., 2003) , correlation between gene-pairs and their consequent effects on selecting differentially expressed genes (Qiu et al., 2005) . We evaluate the above described normalization methods using two schemes. One is the hypothesis testing framework and the other by doing actual classification employing two classifiers. Further details about the evaluation schemes are described below.

3.0.6 The hypothesis testing method

This evaluation scheme is inspired by Golub et al.'s work (Golub et al., 1999) of gene expression based classification of leukemia into its two subtypes (AML and ALL). From each gene's expression values it is possible to compute distance measures between the two classes or conditions. One such distance measure is the difference between the mean expression values of the gene under the two conditions normalized by the within-class variances. The mean and the variance under a particular condition are measured using all the available tissue samples under that condition. The higher the distance for a gene, the better is its classification potential. At a set

distance value a count of the number of genes that imply a distance greater than or equal to the set value is kept. If for the phenotypic class labeling the above gene count is better than that of any other possible labeling of the tissue samples, then the phenotypic labeling makes sense. In other words, there is significant evidence for the phenotypic classes from the measured data. Repeating this for different set distance levels, Golub et al. (Golub et al., 1999) confirm that the particular labeling of AML and ALL samples is in fact supported by the measured gene expression data. Hence a classification based solely on gene expression profiles is possible.

In general, it is useful to confirm that there is enough evidence in the data to support the phenotypic classification or labeling of the tissue samples. In the case of a cancer dataset, each of the tissues can be assumed to carry a label of either +1 or -1 depending on whether the tissue is normal or cancerous. This implies a particular pattern of +1 and -1's on the tissue samples. If the data is to be classifiable based on only the gene expression levels, then there should exist a good number of genes whose expressions have a high correlation with the above pattern of +1 and -1's. Given that the data is noisy, it is hard to specify the minimum amount of correlation for a gene to support the hypothesis. This makes it difficult to estimate how many genes actually support the hypothesis. Following Golub et al. (Golub et al., 1999) we can test whether the correlations are statistically significant with respect to a null hypothesis. In this case, the null distribution is the distribution over the number of genes that support an arbitrary pattern of +1 and -1's at a given correlation level. To quantify this we compute the Pearson correlation coefficient between the gene expression levels and a pattern of +1 and -1's and count the number of genes that have their absolute correlation coefficient above a certain fixed value. The null distribution is modeled by considering 1000 random class labeling patterns of the tissue samples. If the number of genes having their absolute correlation coefficient value above the fixed value is greater than the 1% or 5% level of the null distribution at that fixed threshold value, then the null the hypothesis can be rejected. In other words, the data supports the phenotypic labeling significantly better than an alternative possible random labeling. We repeat this for different set

threshold values sampled uniformly within the range [0, 1] resulting in a comparison with the null distribution at different absolute correlation coefficient cutoff levels.

3.0.7 Classification

Different normalization methods influence the inherent classifiability of the gene expression data differently. Although the hypothesis testing framework helps quantify these influences in one way, a more useful test would be to perform classification using the data. To this end we built two previously studied classifiers for cancer gene expression data: one is a variant of the classifier proposed by Golub et al. (Golub et al., 1999) and the other is a Support Vector Machine (SVM) based classifier by Guyon et al. (Guyon et al., 2002) . Both are instances of linear discriminant classifiers (Guyon et al., 2002) but with different assumptions about correlations between features and the classifier objective function. In addition to classifying data into two classes they have an inherent mechanism to either select or eliminate features that are the most or least useful for classification respectively. Thus the two classifiers jointly address the problems of gene selection and classification.

A good indicator of classifier generalization performance is the leave-one-out (LOO) error. Leave-one-out testing is a special case of k-fold cross-validation where the available data is randomly split into k equal parts. One part is held out for testing and the remaining $(k-1)$ parts are used for training. The average classification accuracy over all the k different parts held out for testing in different experiments is a measure of generalization performance. The special case of leave-one-out cross-validation results from setting k to the total number of points in the available data. This is particularly useful here due to the relatively small number of samples available.

Each classifier's performance is a function of the genes used for classification. Further the best set of genes of a given size might vary for the different classifiers and is still a debated question. So we evaluate the classifier performances by varying the number of genes from 1 to the maximum number available. The question of which genes to use for a given number is addressed by letting each of the above

classifiers use their inherent feature selection mechanism.

With gene selection being done as part of classifier training, it is possible to use all the available data or only the training samples of each leave-one-out set. Both of these approaches have been followed in earlier works (Guyon et al., 2002; Furey et al., 2000) . In one experiment we use all the available data for gene selection in an initial training step. But after the selection is performed, the data is split into training and test sets. The classifiers are retrained and tested using the split data, using only the selected genes, to measure LOO performance. In a different experiment, gene selection is done as part of the training procedure using only training portion of the split data. The results of these experiments are reported separately.

Using the two classifiers and the leave-one-out cross-validation framework we measure the generalization ability of the classifiers using both the raw data and the data resulting from the different normalizing transformations. The generalization performance is measured using two metrics. One is the prediction accuracy and the other is the average prediction confidence. The prediction confidence quantifies the confidence of the classifier in predicting the classes for the leave-one-out samples. This metric is specific to the classifier and will be described in detail later. Since each of the normalizing transformations affect the raw data differently based on their assumptions of the systematic experimental effects, it will be possible to characterize the helpfulness or detrimental effects of the various normalizing methods for classification. The main motivation behind this evaluation approach is that one of the main goals of analyzing microarray gene expression data is classification. So there is a need to assess normalization methods on the basis of their support for classification.

Golub classifier

This classifier was originally proposed to discriminate between the two subtypes of Leukemia (ALL and AML) based on microarray gene expression measurements (Golub et al., 1999) . It can be called a *feature-inclusion* type of classifier that

begins by incrementally selecting a subset of features that are the most useful for classification. The most useful features are the ones whose measurements have the highest correlations with the class labels of the training samples. Computing each feature's correlation coefficient with the class label vector of the training data leads to a rank-ordering of the features with the one having the highest correlation coefficient at the top. Further the correlation coefficients are used as weights in a voting scheme based classification where each feature contributes to the decision according to its weight. Classification is performed based on the measurements of the first few features in the rank ordered list.

Using the measurements of the selected features, class-specific means are computed for each of those features from the training data. In a binary classification setting with the class labels -1 and $+1$, each feature i's measurements in the training set yield the two class-specific means $\mu_{i,-1}$ and $\mu_{i,+1}$. Let μ_i denote the mean of the two class-specific means:

$$\mu_i = \frac{\mu_{i,-1} + \mu_{i,+1}}{2} \tag{3.1}$$

The classifier parameters include the means $\mu_{i,-1}$, $\mu_{i,+1}$, μ_i and weight w_i for each included feature i. Given a new test sample, each feature i of the included set votes for the class -1 or $+1$ depending on whether its measurement x_i is closer to the class-specific mean $\mu_{i,-1}$ or $\mu_{i,+1}$ respectively. The magnitude of the vote is $V(i) = w_i|x_i - \mu_i|$. By aggregating the votes of all included features for both the classes, the one with the highest votes is assigned to the test sample. To quantify the uncertainty associated with the decision, a class prediction confidence is tied to it. The prediction confidence (η) is given by the magnitude of the surplus votes for the winning class relative to the total number of cast votes for both the classes

$$\eta = \frac{V_w - V_l}{V_w + V_l} \tag{3.2}$$

where V_w and V_l are the total votes of the winning and losing classes respectively. Note that the prediction confidence is always a number in the range [0,1] with a

higher value indicating higher confidence. The weights w_i, which are a measure of correlation between the gene expression measurements and the class labels can be computed in a number of ways. Golub et al. (Golub et al., 1999) use the difference between the class-specific means weighted by the sum of class-specific standard deviations as the measure of correlation. This leads to a special case of Fisher's linear discriminant (Duda et al., 2000) with the assumption of uncorrelated features or diagonal covariance matrix (Guyon et al., 2002) under normality. However we use the absolute sample correlation coefficient between the gene expression measurements and the class labels vector for the training data. This is in keeping with the experiments in section 2.1 and most likely not very different from the earlier work. Other authors have proposed different ways to compute this correlation metric (Pavlidis et al., 2001; Furey et al., 2000) .

Support Vector Machine classifier with Recursive Feature Elimination (SVM-RFE)

Support vector machines (SVM) (Vapnik, 1998; Bishop, 2006) are being widely used for classification of biological datasets including microarray data due to their ability to effectively handle very high dimensional feature spaces. Being maximum margin classifiers they have an inherent regularization mechanism enabling them to have a good generalization performance. This is especially useful for microarray gene expression datasets due to the relatively large number of genes involved compared to the number of samples. At the same time they naturally yield a feature selection mechanism to identify genes that are the most useful for classification. This was illustrated by Guyon et al. (Guyon et al., 2002) with the help of a recursive feature elimination (RFE) technique in conjunction with a linear kernel SVM classifier. In contrast to the Golub classifier described in section 3.0.7, SVM-RFE is a *feature-exclusion* type of classifier that incrementally eliminates features that are the least useful for classification until the desired number of features remain.

The motivation for the feature-exclusion principle comes from analyzing the importance of weights associated with features in a linear classifier for multivariate

data (Guyon et al., 2002) . Due to the inherent correlations among features it is not always guaranteed that the importance of a feature for classification is implied by the magnitude of its weight. Of course this is the case when the features are uncorrelated, which is the assumption behind the Golub classifier of section 3.0.7 but uncorrelatedness is not always guaranteed. Instead the weight magnitude is a definite indicator of how much loss is incurred by eliminating the particular feature assuming a quadratic objective function in weights as is the case with linear SVMs (Cun et al., 1990) . So at each step, the SVM-RFE trains a linear kernel SVM with all the retained features and eliminates the one among them that has the smallest magnitude weight associated with it. This is repeated until a desired number of features are left.

Mathematically, a linear two-class SVM is characterized by a weight vector $\boldsymbol{w}$ and a bias b that determine a hyper-plane in the feature space. In the case of separable data, all the points $\boldsymbol{x}_n$ satisfy $y_n(\boldsymbol{x}_n^T\boldsymbol{w} + b) \geq 1$ where y_n is the class label (-1 or +1) associated with $\boldsymbol{x}_n$. The hyperplane $(\boldsymbol{w}, b)$ maximizes the margin $\frac{1}{\|\boldsymbol{w}\|^2}$ between itself and the closest points on either side of it. The closest points, the ones that satisfy the equality in the constraint above are called the support vectors. In the case of non-separable data the points are allowed to be misclassified but misclassifications are penalized by introducing a slack variable ζ_n for each point $\boldsymbol{x}_n$. The points now satisfy the constraints $y_n(\boldsymbol{x}_n^T\boldsymbol{w} + b) \geq (1 - \zeta_n)$, where $0 \leq \zeta_n < 1$ for the correctly classified points and $\zeta_n > 1$ for the misclassified points. The optimum hyperplane is the one that minimizes $\|\boldsymbol{w}\|^2 + C\sum_n \zeta_n$ where C is a parameter to adjust the relative importance of the misclassification penalty. The maximum-margin hyperplane is solved by setting up an optimization problem in the dual space (Bishop, 2006) . Each data point $\boldsymbol{x}_n$ is associated with a Lagrange multiplier α_n and the objective function to maximize, L, is a quadratic:

$$L = -\frac{1}{2}\sum_{l,m}\alpha_l\alpha_m(\boldsymbol{x}_l^T\boldsymbol{x}_m) + \sum_n \alpha_n \tag{3.3}$$

with the linear constraints:

$$0 \leq \alpha_n \leq C$$
$$\sum_n \alpha_n y_n = 0$$

In the resulting solution the points for which $\alpha_n = 0$ are correctly classified and are outside the margin. Other points for which $0 < \alpha \leq C$ are the support vectors which can be either correctly classified (lie inside or on the margin) or misclassified. The weight vector $\boldsymbol{w}$ is given by:

$$\boldsymbol{w} = \sum_n \alpha_n y_n \boldsymbol{x}_n \tag{3.4}$$

and the bias is calculated from the support vectors that are correctly classified and lie on the margin ($0 < \alpha_n < C$) (Bishop, 2006) . The class of a new test point $\boldsymbol{x}_{test}$ is assigned based on which side of the hyperplane it falls:

$$Class(\boldsymbol{x}_{test}) = sign(\boldsymbol{x}_{test}^T \boldsymbol{w} + b) \tag{3.5}$$
$$= sign(\sum_n \alpha_n y_n (\boldsymbol{x}_{test}^T \boldsymbol{x}_n) + b) \tag{3.6}$$

The absolute distance of the test point from the separating hyperplane can be thought of as characterizing the confidence of the classifier. We use this quantity as the prediction confidence η of the SVM-RFE classifier:

$$\eta = \frac{|\boldsymbol{x}_{test}^T \boldsymbol{w} + b|}{\|\boldsymbol{w}\|} \tag{3.7}$$

Note that unlike the Golub classifier, this quantity is not guaranteed to be in the range [0, 1]. But it is useful in a comparative analysis of how far away the leave-one-out points lie from the separating hyperplanes on average.

In addition to being a useful classifier for high-dimensional datasets, SVMs provide a convenient way to formulate the normalization problem in a classification setting. Is there a normalization technique that is optimal in terms of its ability

to transform the data to be highly predictive of phenotype labels? An attempt to answer this question is made here using the maximum margin framework of SVMs. Since one of the goals behind formulating a maximum margin classifier is to achieve better generalization, it makes sense to use this as an objective function to design a normalization method. With the assumption of array-specific offsets it turns out that the optimum offsets are the same as the ones given by geometric normalization. See Appendix A for mathematical details about the optimization and some experimental results.

3.1 Experiments

We compare the performances of the five normalization methods using the two comparison schemes described above. Raw gene expression data or the suitably log-transformed versions from two microarray datasets are subjected to normalization transformations of the different methods. One dataset is the colon cancer dataset of Alon et al. (Alon et al., 1999) and the other is the in-house angiogenesis data of Hoying et al. (Greer et al., 2006) .The details of each of the datasets follow.

3.1.1 Colon cancer dataset

A widely used dataset in the cancer classification literature using microarray gene expression measurements is the colon cancer dataset of Alon et al. (Alon et al., 1999) . It consists of 62 tissue samples of which 40 are tumor samples and 22 are normal. The samples were obtained from different patients and from some of them both normal and cancerous tissues were collected. Affymetrix arrays containing more than 6500 oligonucleotide sequences complementary to human genes were used to measure the gene expression profiles across the samples. After a filtering process, 2000 genes with the highest minimal intensity across all the samples were retained. The goal is to classify the tissues into cancerous or normal based solely on gene expression measurements.

3.1.2 Angiogenesis dataset

This microarray dataset was collected by the research group of Hoying et al. (Greer et al., 2006) to identify genes that help to characterize different stages of vascularization. The data was obtained by hybridizing tissue samples extracted from an experimental model (in vivo) of tissue vascularization in SCID mice with the implants obtained from either tie2:GFP mouse or rat adipose. The vascularization proceeded in the relative absence of non-vascular cells. So the hypothesized diagnostic genes were specific to vascular cells. Tissue samples were extracted from the implanted constructs at discrete time points – days 3, 7, 14, 21 and 28. In a unique experimental design these samples and a day 0 sample (implant source) were hybridized using two channel cDNA microarrays to obtain measurements of gene expression for 15600 genes. Care was taken so that biological variations were averaged out in the measurements. Each gene was measured twice on each of the 4 microarrays for a given time point. This resulted in 8 measurements per gene per time point. The intensity measurements were background subtracted and linlog transformed (Greer et al., 2006) for variance stabilization and only those measurements that were consistently well above the background level were retained.

For the experiments reported here, we averaged the two background subtracted and linlog transformed measurements of a gene on the same microarray. This reduced the number of measurements to 4 per gene per time point. A working concept of vascularization is hypothesized to involve two main stages–Angiogenesis (day 3 through 7) and Maturation (day 14 through 28). We considered the tissue sample collected on day 3 as representative of Angiogenesis and that of day 21 as representative of Maturation. Although it is possible to use the information from other samples, it is necessary to account for the possible time variations in gene expression levels. We chose to avoid this complexity for now but will be addressed in subsequent work. The choice of the samples as representatives of the stages of vascularization was based on the evidence in the data for the two discrete phenotypic classes (Angiogenesis and Maturation) based on hypothesis testing. This yielded a

set of 16 samples (8 in each class) for the experiments.

3.1.3 Experiments on colon cancer dataset.

The raw colon cancer (Alon et al., 1999) data is suitably log transformed and input to the different normalization methods described in section 2.1. The results of evaluation on the raw and normalized data using the two evaluation methods are described below.

Hypothesis testing results

Both the raw data and the normalized data were subjected to hypothesis testing to seek evidence in the resulting gene expression levels for the phenotypic class labels. The resulting plots are shown in Fig. 3.1. For the raw data, the TEST curve falls well above the RANDOM-1-percentile curve for absolute correlation coefficient levels up to around 0.2. This confirms that the data supports the phenotypic class labeling significantly better than alternate possible labellings. This provides a great hope for classification methods to work with this data because this behavior is optimistic in the absence of any normalization. The three normalization methods: geometric, quantile and MIN-SS-CORR improve the support for the classes slightly compared to raw data.

However, using the rank normalized data, the support for the phenotypic labeling of the tissue samples is not statistically significant. As seen in Fig. 3.1e, the curve for the hypothesis of interest falls well within the median curve of the null hypothesis for larger threshold values of the absolute correlation coefficient. This is perhaps due to the replacement of gene expression values with their discrete ranks. It is possible to test the classifiability of the data based on δ-sequence of random variables with the help of the class label hypothesis testing as before. Instead of using the gene expression levels to compute the correlation coefficients with the class label patterns on the tissue samples, the δ-sequence values are used. The resulting plot is shown in Fig. 3.1f. It is clear that at all cut-off levels of the absolute correlation coefficient

the number of variables that support the phenotypic labeling are well below the median number for the null hypothesis.

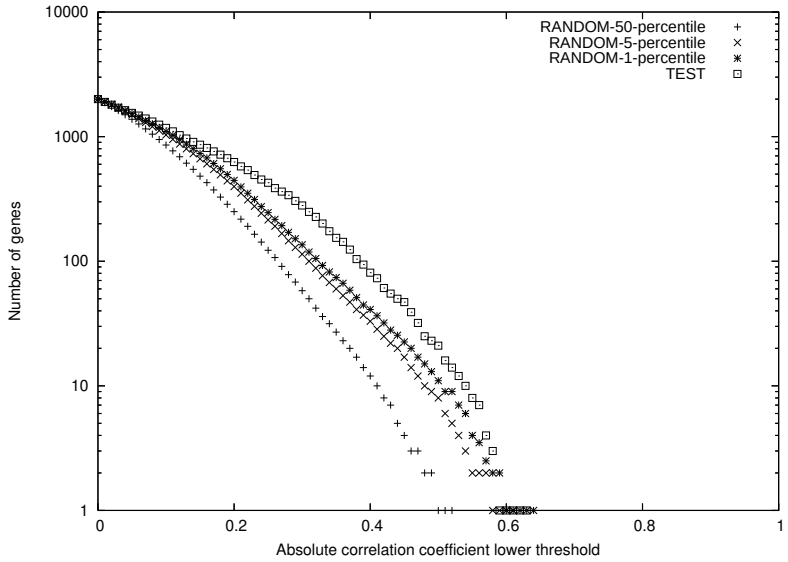
Classification results using Golub classifier

We evaluate the prediction accuracy and the average confidence of the Golub classifier as a function of the number of genes included for classification. Gene selection for a given number of genes is done as part of the classifier training (see Section 3.0.7). Both raw data and the data resulting from the different normalization methods, described in section 2.1 are used as input to the classifier in different experiments. The results are plotted in Fig. 3.2.

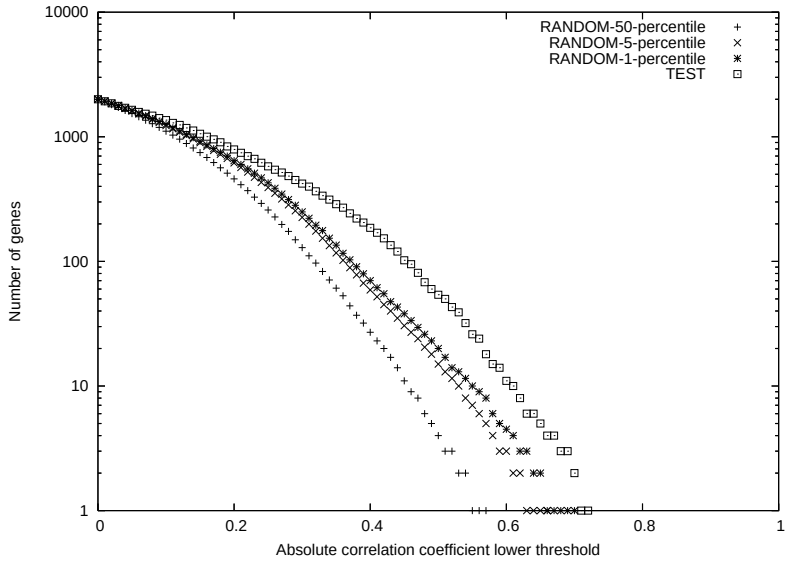
It is clear that the prediction accuracy for all the normalization methods, except δ-sequences and rank normalization for number of genes less than about 8, is at least as good as or better than the raw data. For number of genes greater than about 16 rank normalized data seems to help the classification the most. It is not very surprising that rank normalized data can be well-classified because the phenotypic class labeling is supported as well as any others (see Fig. 3.1e). However the average prediction confidence is an indicator of how well distinguished the classes are in the data. The rank normalization and δ-sequences seem to perform poorly in this respect. The other three normalization methods (geometric, quantile and MIN-SS-CORR), whose behavior in the permutation analyses were similar, seem to perform better than the raw data in average prediction confidence regardless of the number of genes.

Further insight into the behavior of different normalization methods can be obtained by fixing the number of genes and looking at the number of correct predictions over the LOO test samples as the prediction confidence decreases. Plotting this for 50, 100 and 500 genes results in Fig. 3.3.

The data resulting from the three normalization methods: geometric, quantile and MIN-SS-CORR is better than the raw data both in terms of prediction accuracy and confidence and improves with the increase in the number of genes. The three methods seem to perform equally good. Although rank normalized data helps

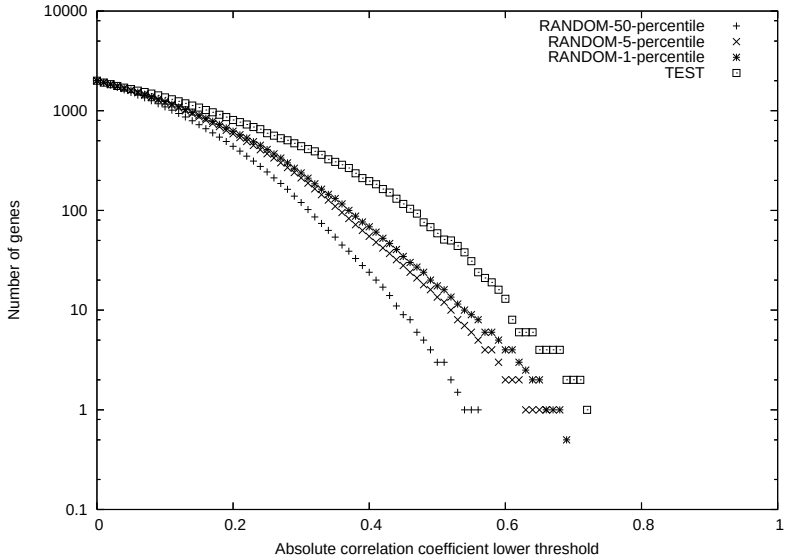


(a) Raw data

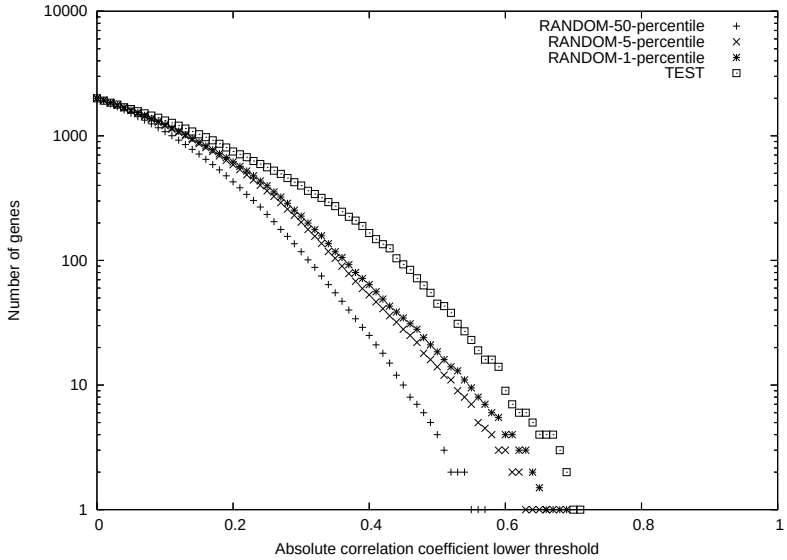


(b) Geometric normalization

Figure 3.1: Class label hypothesis testing of the Alon colon cancer dataset (Alon et al., 1999) before and after normalization. TEST refers to the number of genes that have their absolute correlation coefficient value above a certain level with the correlation coefficient being computed with the phenotypic pattern of class labeling. See Section 3.0.6. RANDOM refers to statistics of the distribution obtained by considering 1000 different random binary labeling patterns on the tissue samples (null distribution). Plotting the median (50 percentile), 5-percentile and 1-percentile points of the null distribution at different minimum levels of the absolute correlation coefficient results in the three curves RANDOM-50-percentile, RANDOM-5-percentile and RANDOM-1-percentile respectively. The higher the TEST curve above the RANDOM curves the better is the statistical significance suggested by the data for the phenotypic classes.

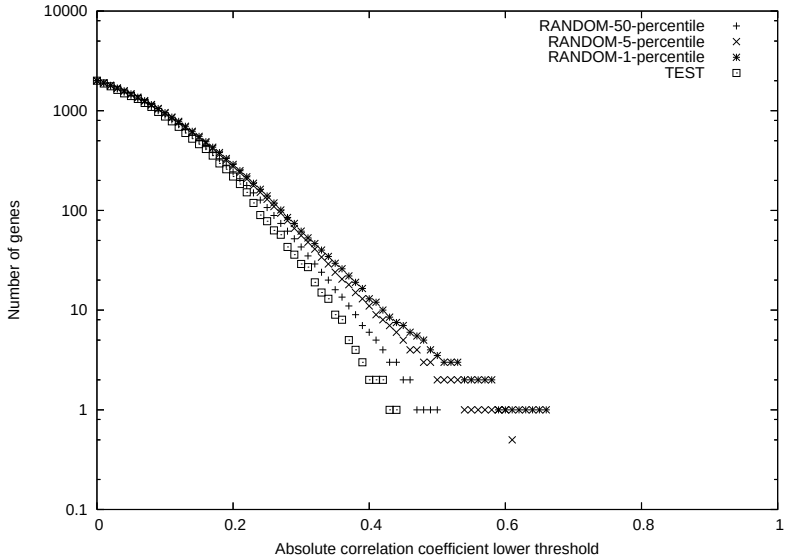


(c) Quantile normalization

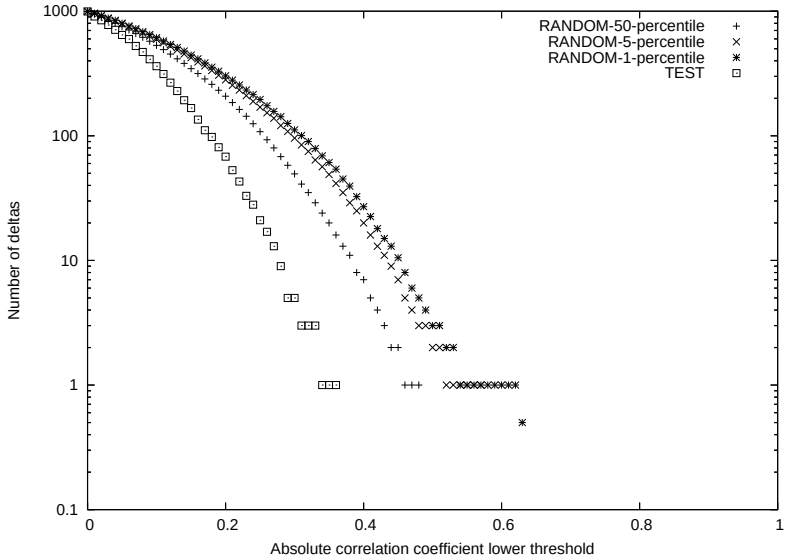


(d) MIN-SS-CORR normalization

Figure 3.1: continued.

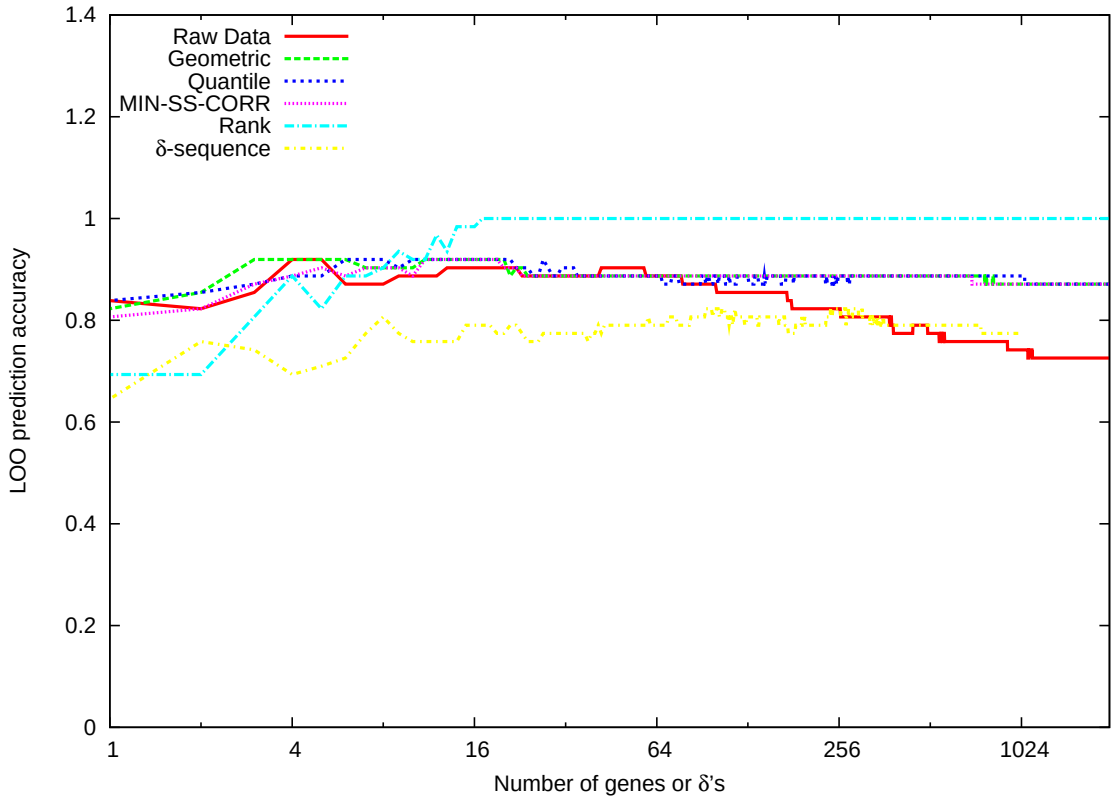


(e) Rank normalization



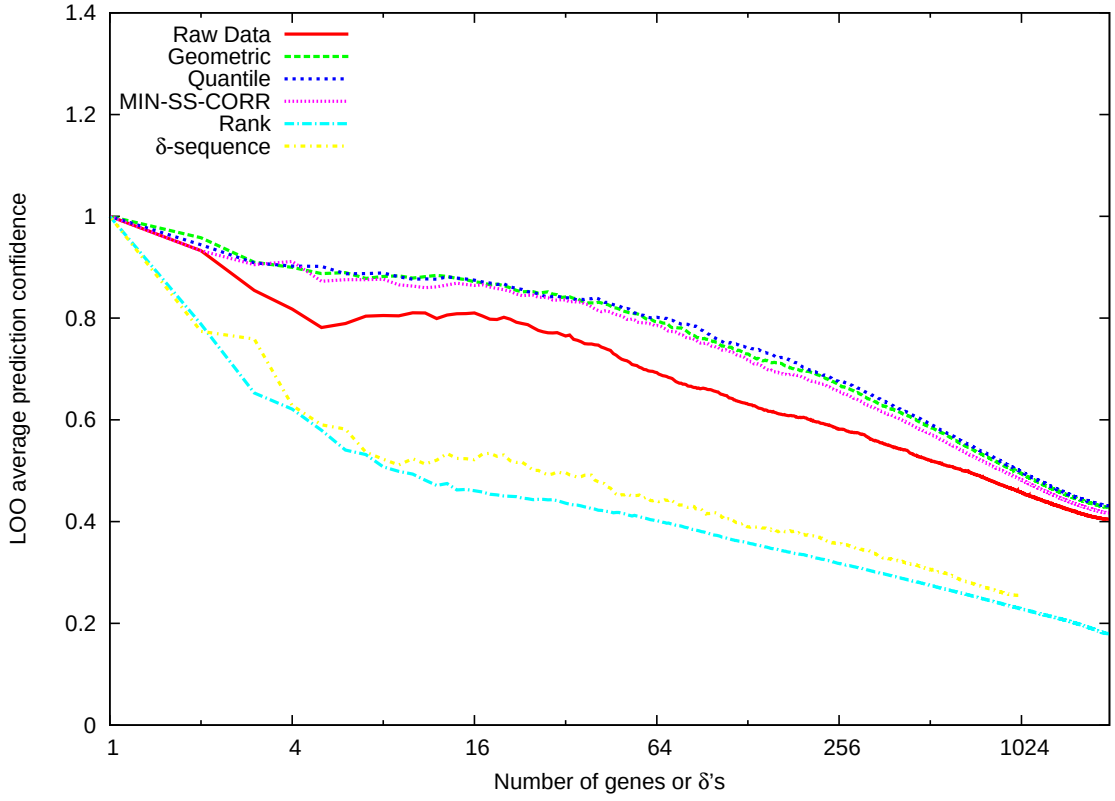
(f) δ-sequences

Figure 3.1: continued.



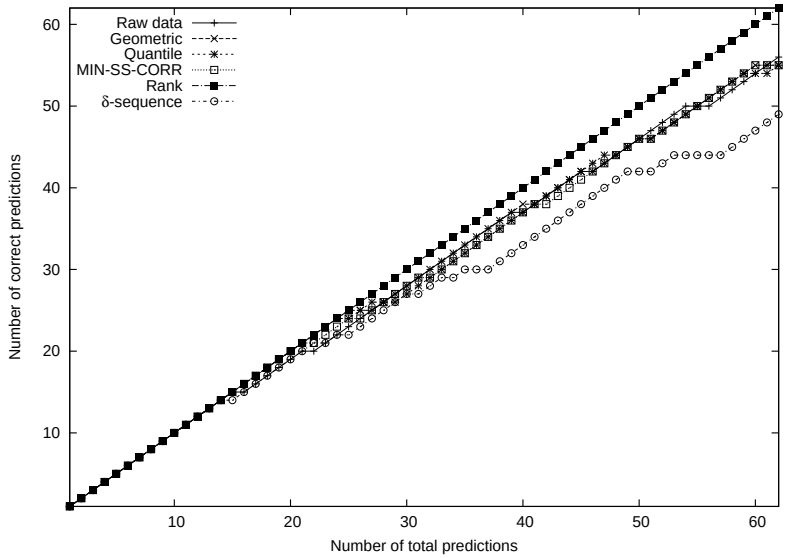
(a)

Figure 3.2: (GOLUB CLASSIFIER, ALON DATA) Leave-one-out prediction accuracy as a function of the number of genes or δ's used for classification. Note that the maximum number of δ's available for this dataset is 1000. The fact that rank normalization helps classification better than other normalization methods for number of genes greater than about 16 is not very surprising given that this data supports any random class labeling hypothesis almost equally well beyond this gene count as seen in the plot of Fig. 3.1e. But these predictions are not made with as high a confidence as the others (Fig. 3.2b).

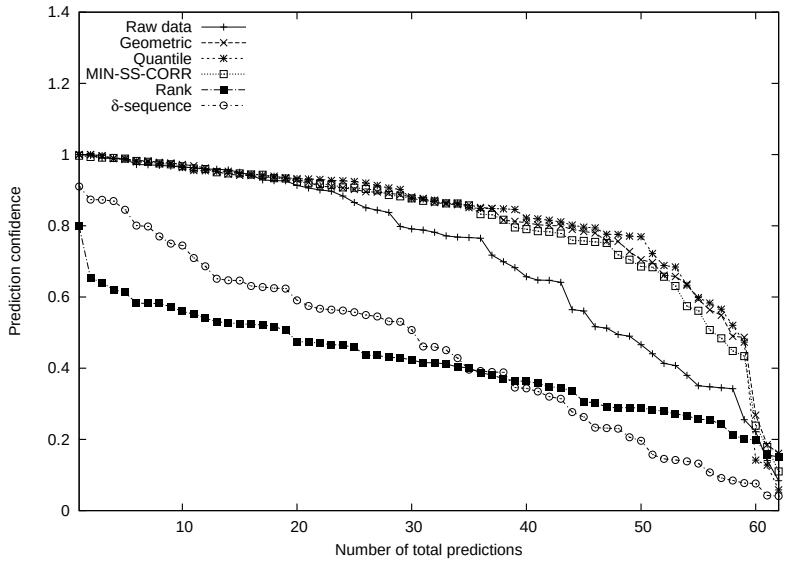


(b)

Figure 3.2: continued. Leave-one-out average prediction confidence as a function of the number of genes or δ's used for classification. Note that the maximum number of δ's available for this dataset is 1000. The behavior of the classifiers using normalized data is better overall compared to the raw data except for the δ-sequences and rank normalization. Although the prediction accuracy is high, the average prediction confidence is low for rank normalized data compared to raw data and other normalization methods except the δ-sequences. The three normalization methods (geometric, quantile and MIN-SS-CORR) that support the phenotypic class labeling significantly better than others in permutation analyses perform almost the same.

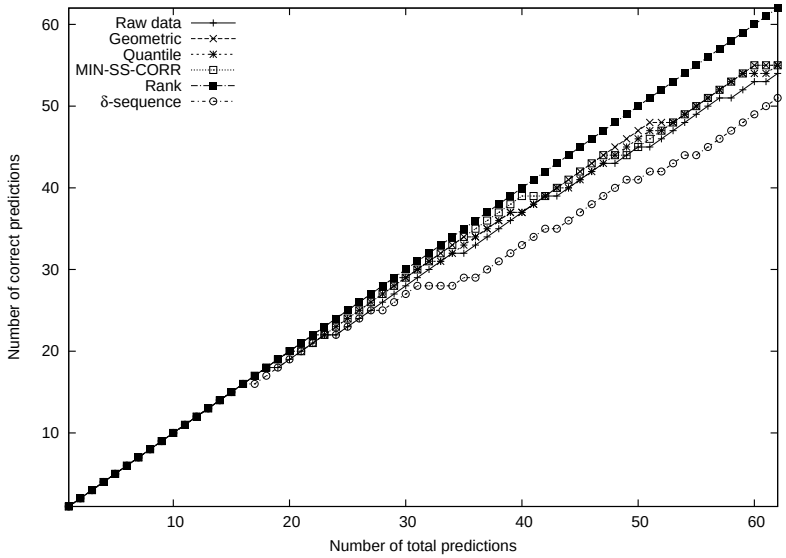


(a) 50 genes or δ's

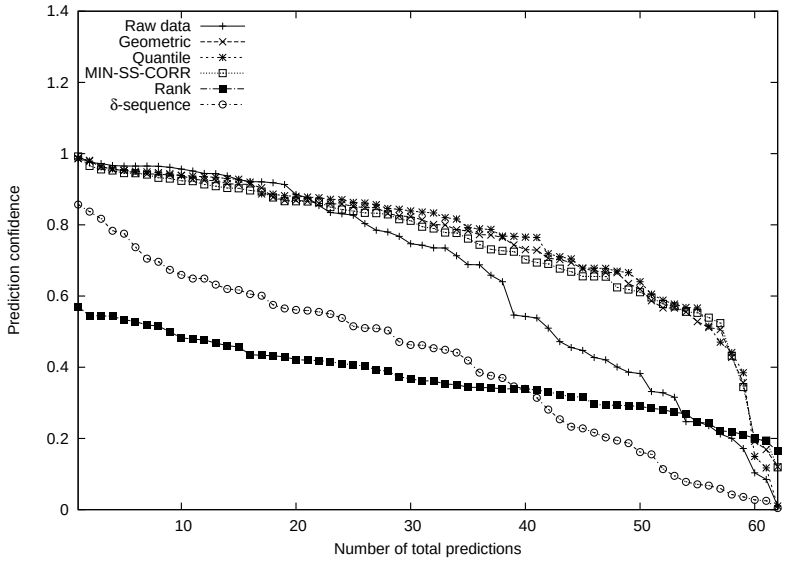


(b) 50 genes or δ's

Figure 3.3: (GOLUB CLASSIFIER, ALON DATA) Leave-one-out prediction behavior for a fixed number of genes or δ's : 50, 100 and 500. The cumulative number of correct predictions as a function of the ordered LOO sample indices is plotted on the top panel and the respective prediction confidences are plotted on the bottom panel. The test samples are ordered in the decreasing order of their prediction confidences given by the corresponding classifiers. Using data normalized by applying the three normalization methods: geometric, quantile and MIN-SS-CORR is at least as good or better than the raw data both in terms of prediction accuracy and confidence. Although the rank normalized data is accurately predictable, the prediction confidences associated with the leave-one-out samples are much smaller. The δ-sequence data is hard to predict both accurately and confidently.

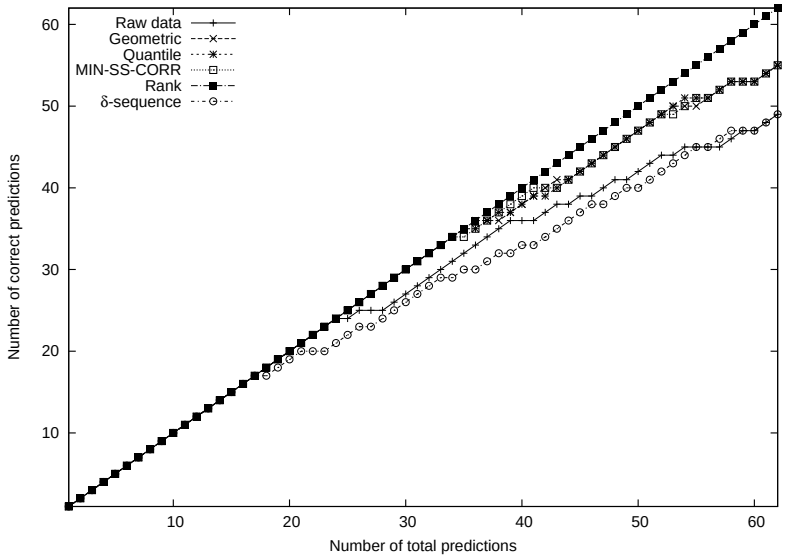


(c) 100 genes or δ's

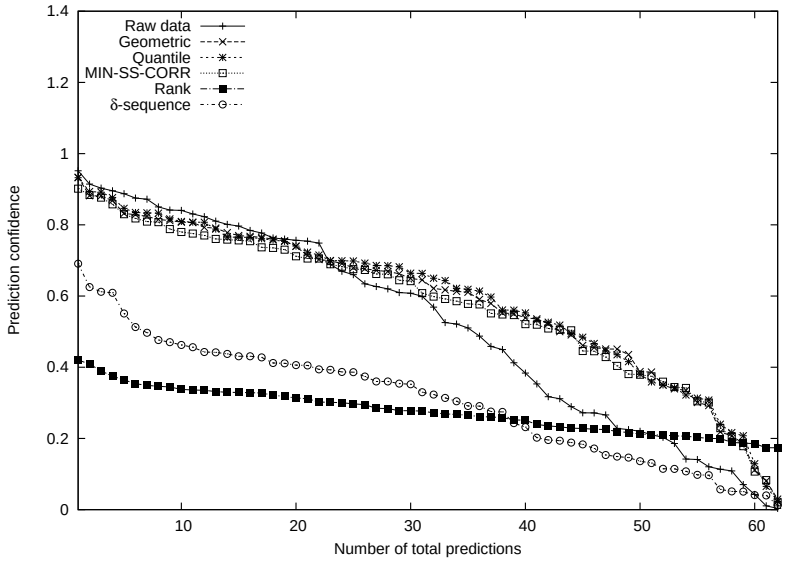


(d) 100 genes or δ's

Figure 3.3: continued.



(e) 500 genes or δ's



(f) 500 genes or δ's

Figure 3.3: continued.

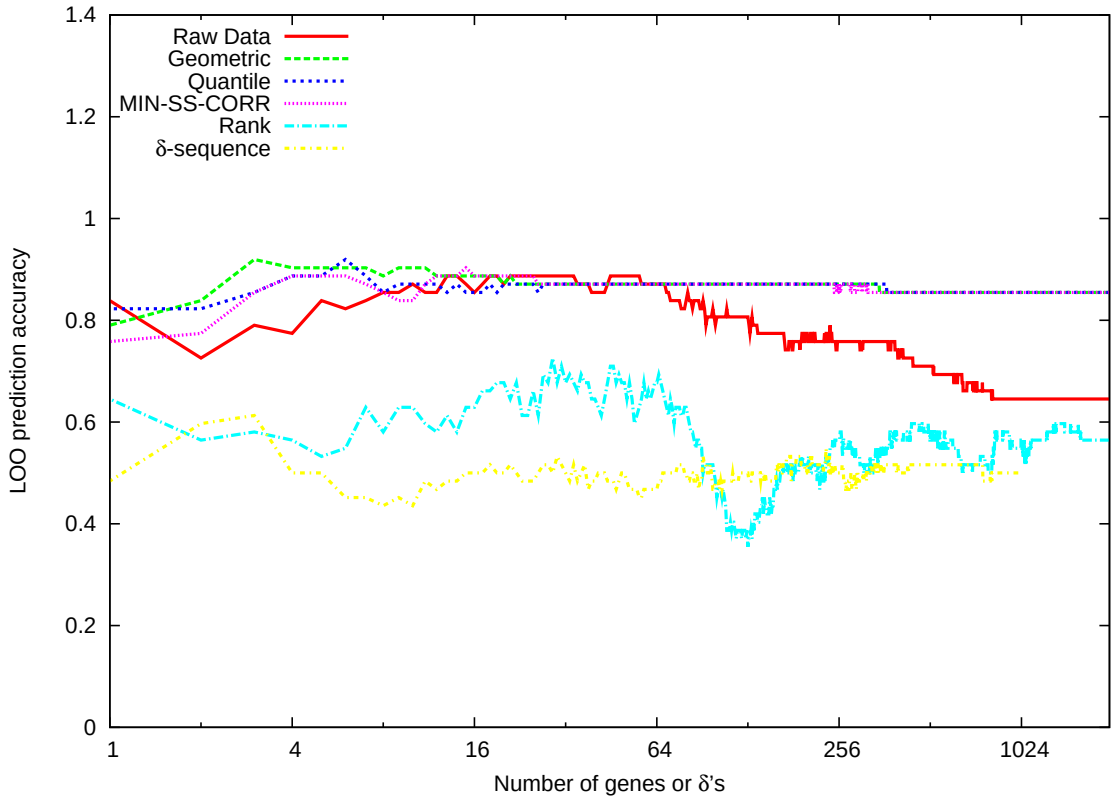
accurate prediction, the associated confidences are small. In a scenario where a tissue sample would be declared cancerous or not based on prediction confidence, this would amount to refraining from making a decision in most of the cases. The δ-sequence data is hard to predict both accurately and confidently.

Gene selection with LOO

The results above are based on doing gene selection using all the samples in the data. As gene selection is integral to classifier training, all data samples are used in an initial training step for the purpose of selecting the most useful genes for a given number of them. Then separate classifiers are trained (with only the selected genes) using the training data part of the leave-one-out sets to compute prediction performance metrics. It can be argued that true leave-one-out tests should make the gene selection process also blind to the held-out samples. Fig 3.4 through Fig 3.6 show the corresponding results from experiments where only the training data samples of the leave-one-out sets were used for gene selection. In general, the prediction performance of a classifier is superior when all the samples are used for gene selection and this was followed by one of the previous works (Guyon et al., 2002) . The contention is that there is probably an information leak when all the samples are used for selection (Furey et al., 2000) A fair test would be to do gene selection using only the training samples.

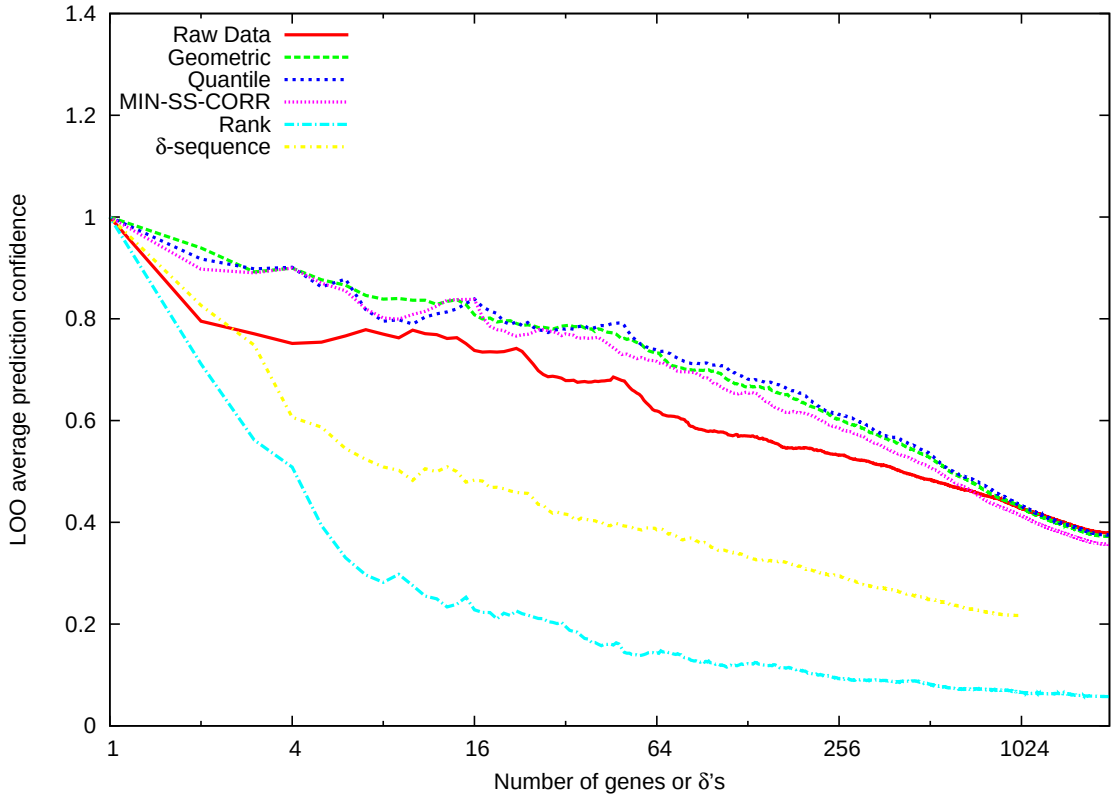
Classification results using SVM-RFE classifier

Similar to the analysis using Golub classifier we used the raw data and the normalized data obtained from different methods for simultaneous gene selection and classification of the tissues into two classes (cancerous or not) using SVM-RFE. The parameter C was set to 1000 based on the observation that most of the training data was linearly separable for number of genes greater than or equal to about 4. This was true for the raw and normalized data and the different training and test subsets of the LOO analysis. The training data was not completely linearly separable below



(a)

Figure 3.4: (GOLUB CLASSIFIER, ALON DATA: GENE SELECTION USES LOO) Leave-one-out prediction accuracy as a function of the number of genes or δ's used for classification. Note that the maximum number of δ's available for this dataset is 1000. The behavior of the classifiers using normalized data is better overall compared to the raw data except for the δ-sequences and rank normalization. The three normalization methods (geometric, quantile and MIN-SS-CORR) that support the phenotypic class labeling significantly better than others in permutation analyses perform almost the same.



(b)

Figure 3.4: continued. Leave-one-out average prediction confidence as a function of the number of genes or δ's used for classification. Note that the maximum number of δ's available for this dataset is 1000. The behavior of the classifiers using normalized data is better overall compared to the raw data except for the δ-sequences and rank normalization. The relative performance of the three normalization methods: geometric, quantile and MIN-SS-CORR, is better than the raw data and the other two methods: δ-sequences and rank normalization.

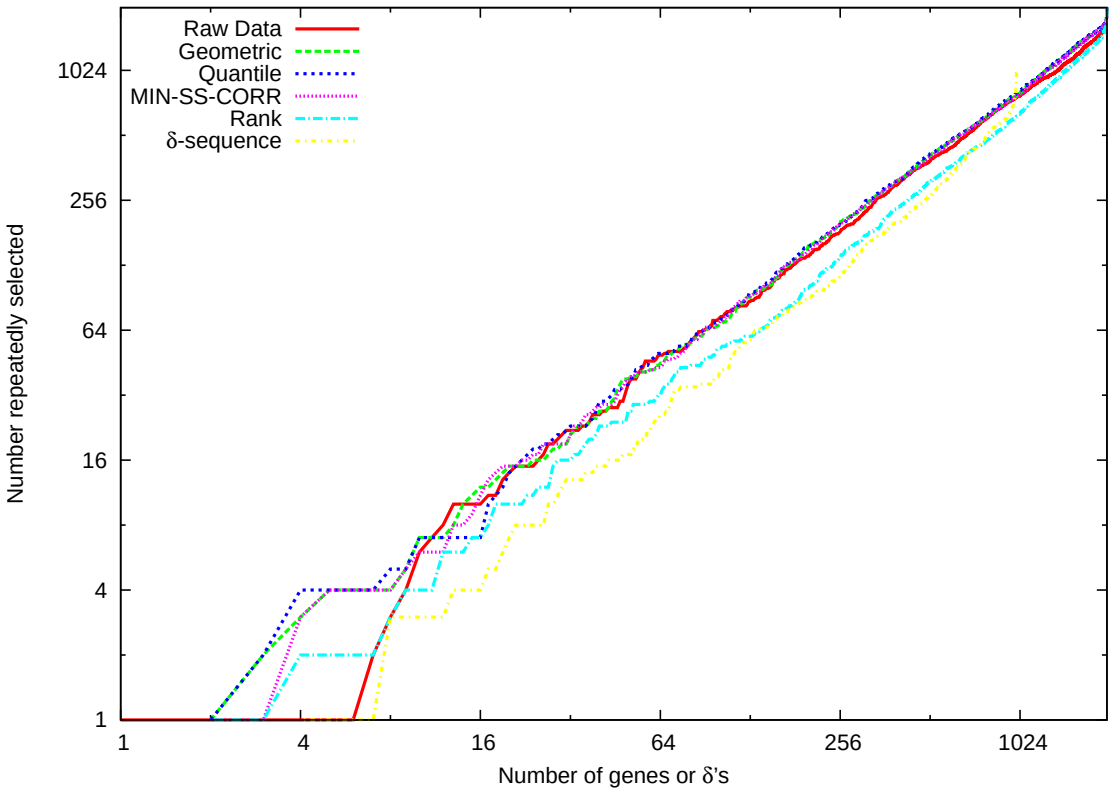
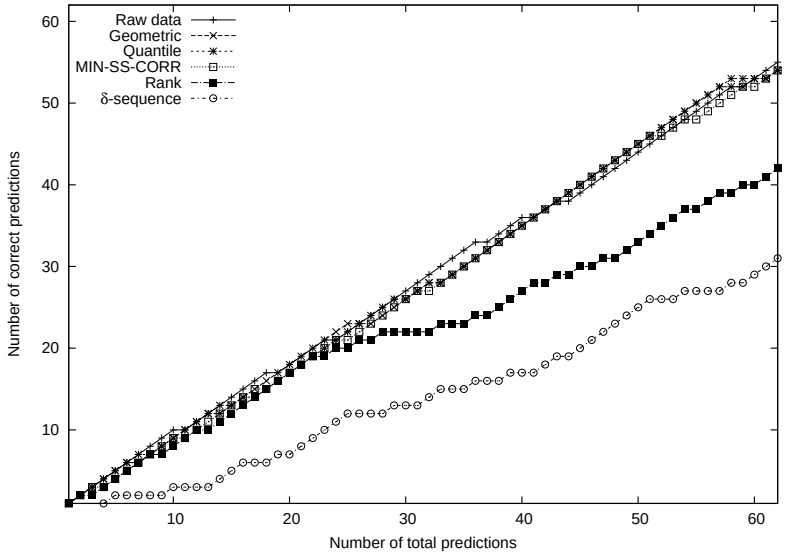
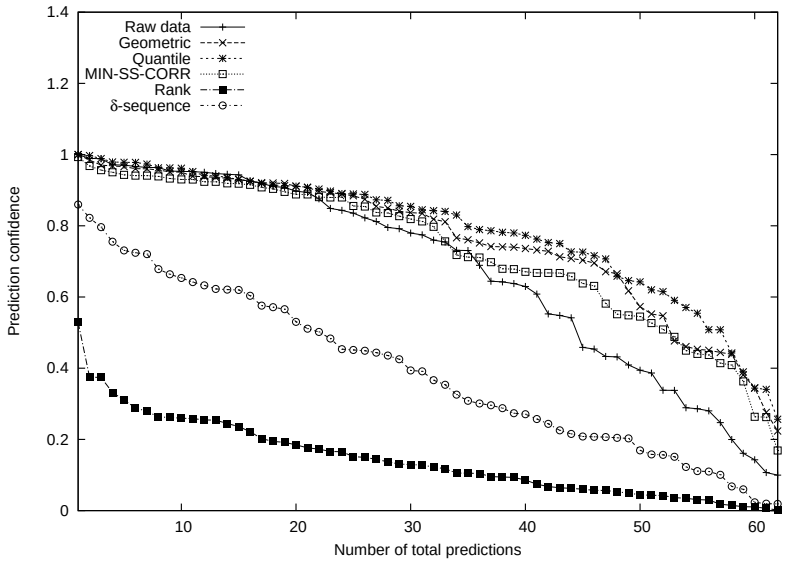


Figure 3.5: (GOLUB CLASSIFIER, ALON DATA: GENE SELECTION USES LOO) Number of genes or δ's that are repeatedly selected across all the divisions of the available data into training and test sets of leave-one-out analysis. Note that the maximum number of δ's available for this dataset is 1000. The independent axis is the total number of genes used by the classifier. The higher the number on the dependent axis the lower the variation in the genes that are selected as the most useful for classification across different LOO training sets. Data normalized using the three methods: geometric, quantile and MIN-SS-CORR results in the corresponding classifier having lower variability in the selected genes, generally better than the raw data, rank normalized data and δ-sequences. Note that as the number of genes or δ's reach their maximum then all of them are repeatedly selected due to which the curve for the δ-sequence data crosses over the other curves at 1000. Similarly for the other datasets.

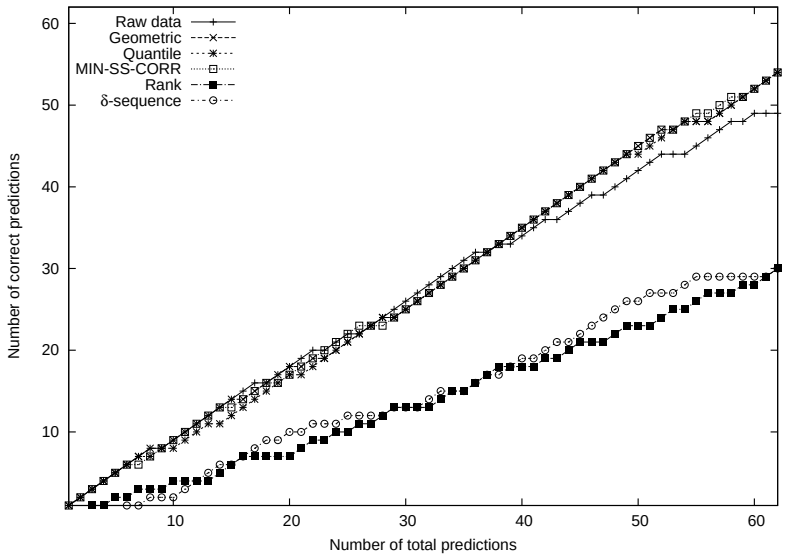


(a) 50 genes or δ's

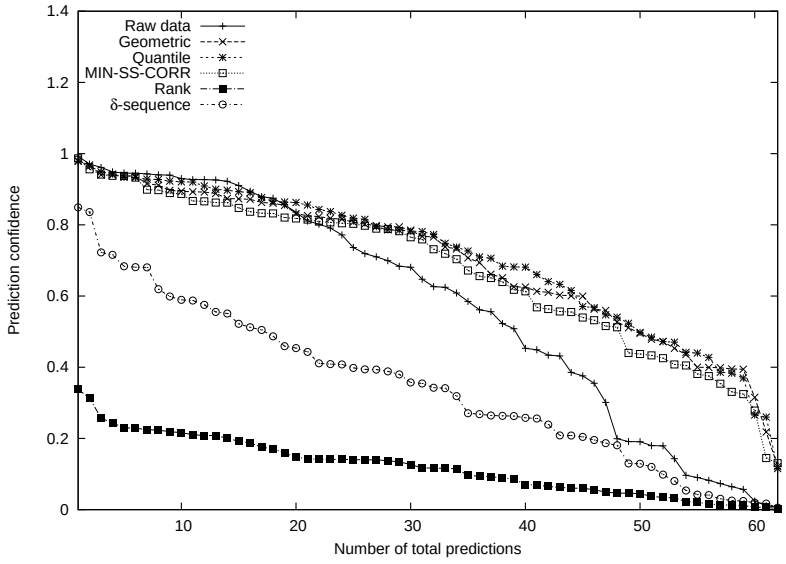


(b) 50 genes or δ's

Figure 3.6: (GOLUB CLASSIFIER, ALON DATA: GENE SELECTION USES LOO) Leave-one-out prediction behavior for a fixed number of genes or δ's : 50, 100 and 500. The cumulative number of correct predictions as a function of the ordered LOO sample indices is plotted on the top panel and the respective prediction confidences are plotted on the bottom panel. The test samples are ordered in the decreasing order of their prediction confidences given by the corresponding classifiers. Using data normalized by applying the three normalization methods: geometric, quantile and MIN-SS-CORR is at least as good or better than the raw data both in terms of prediction accuracy and confidence. The δ-sequence and rank normalized data is hard to predict both accurately and confidently.

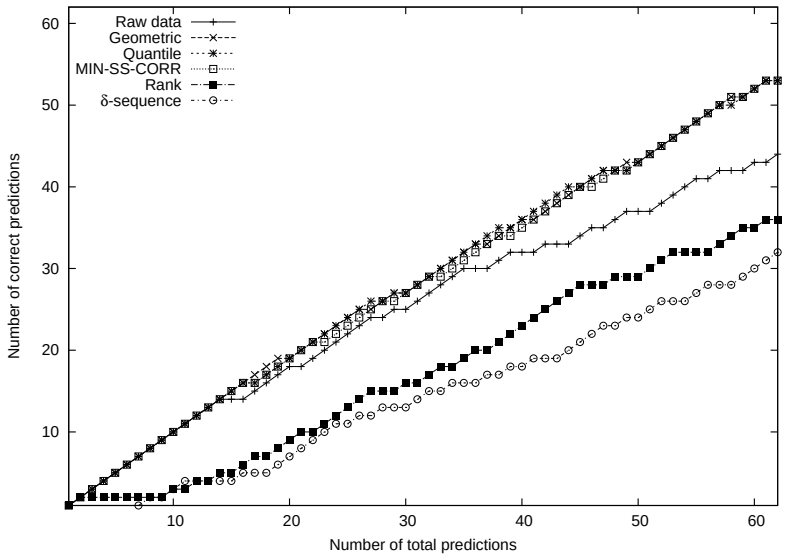


(c) 100 genes or δ's

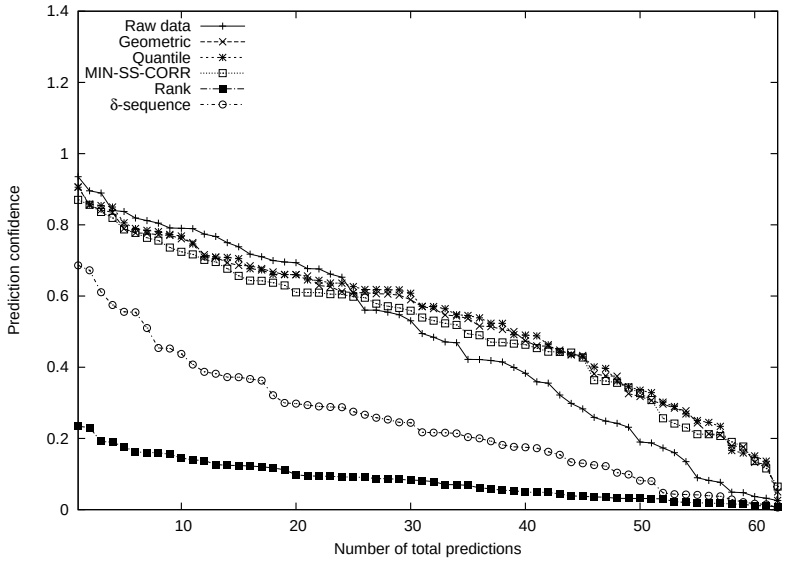


(d) 100 genes or δ's

Figure 3.6: continued.



(e) 500 genes or δ's



(f) 500 genes or δ's

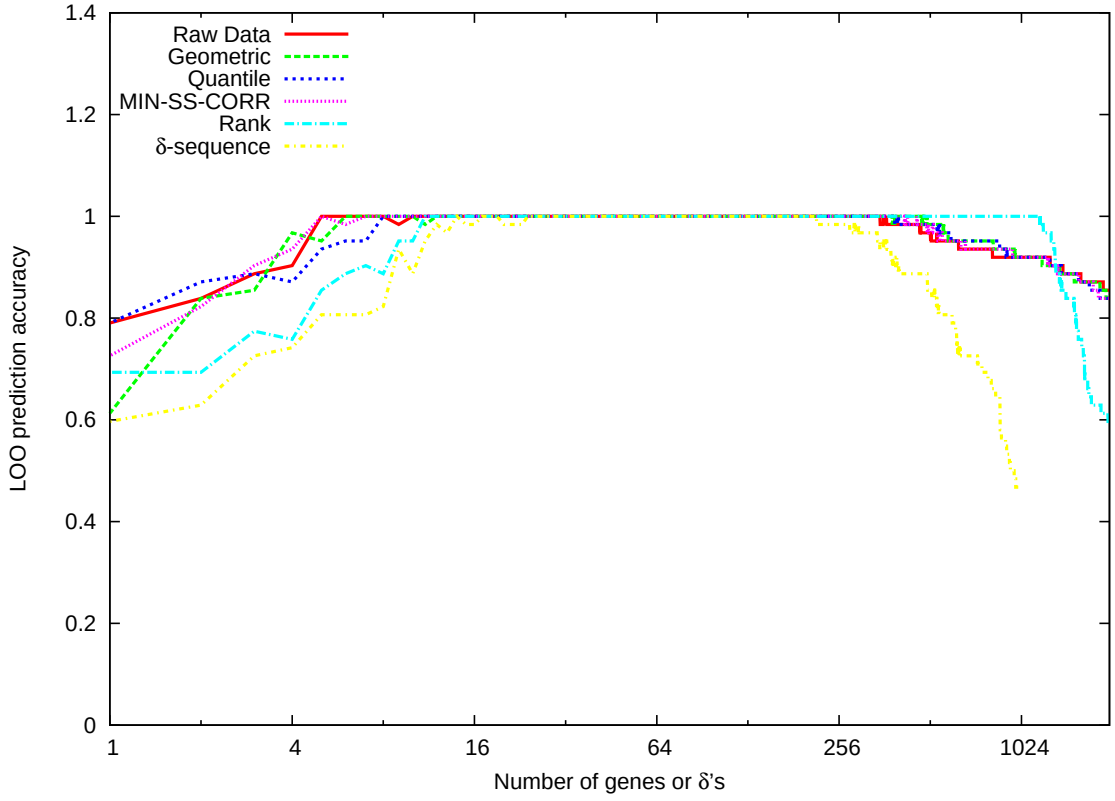
Figure 3.6: continued.

this number of genes most of the time and the resulting SVM coefficients α_n did not satisfy the box constraints using the Matlab solver. The plots of LOO prediction accuracy and average prediction confidence are shown in Fig. 3.7.

On average the performance of a SVM-RFE classifier is superior to the Golub classifier, as was also seen earlier (Guyon et al., 2002) . Using raw or any of the normalized data it is possible to achieve 100% LOO prediction accuracy over a range of the number of included genes. Similar to the Golub classifier, the three normalization methods (geometric, quantile and MIN-SS-CORR) perform equally well and slightly better than the raw data both in terms of prediction accuracy and average confidence. The behavior of the classifier using rank normalized data is more variable with the prediction accuracy deteriorating quickly beyond about 1200 genes. The average prediction confidence is lower than that for the raw data and the above three normalization methods for number of genes greater than about 300 and decreases beyond this point. The classifier using δ-sequences data although achieves 100% LOO prediction accuracy for a certain range of the number of δ's, its average confidence is relatively low in this range.

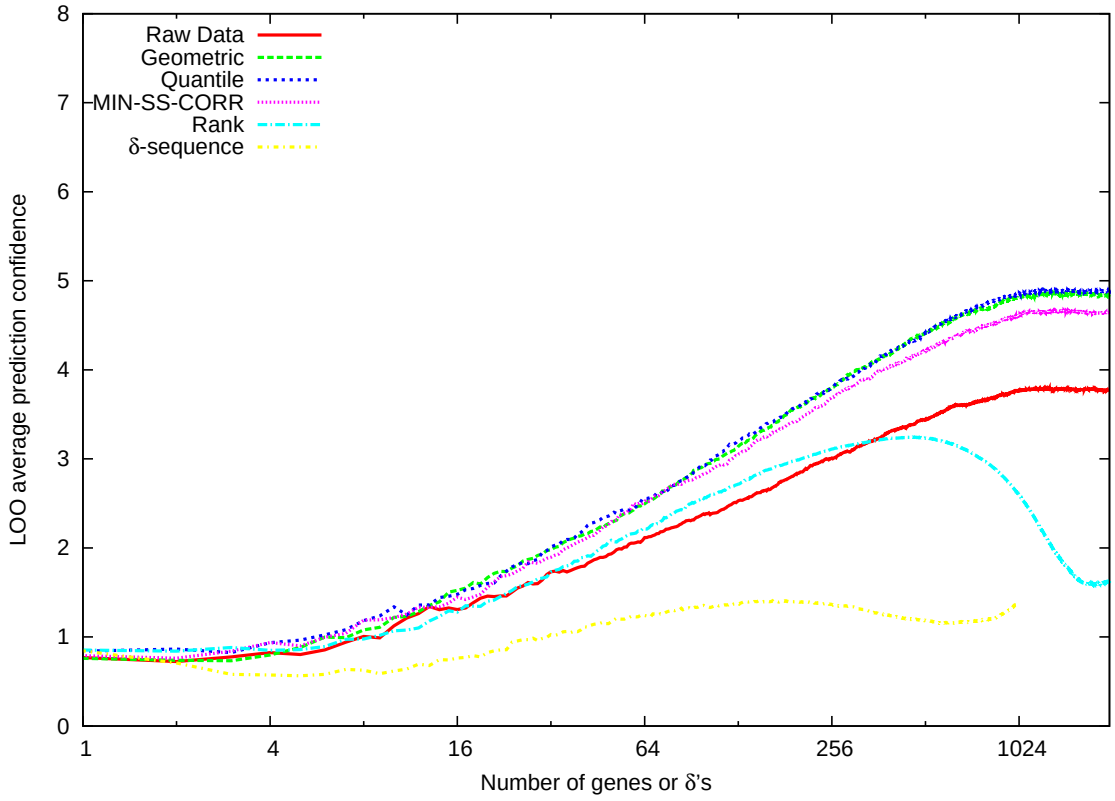
Fixing the number of genes and plotting the number of correct predictions and the prediction confidence as a function of the number of LOO test samples (arranged in the decreasing order of prediction confidence) results in the plots of Fig. 3.8.

The results based on doing gene selection using only the training samples of the LOO sets are shown in Fig 3.9 through Fig 3.11. The trend is similar to the results using Golub classifier.



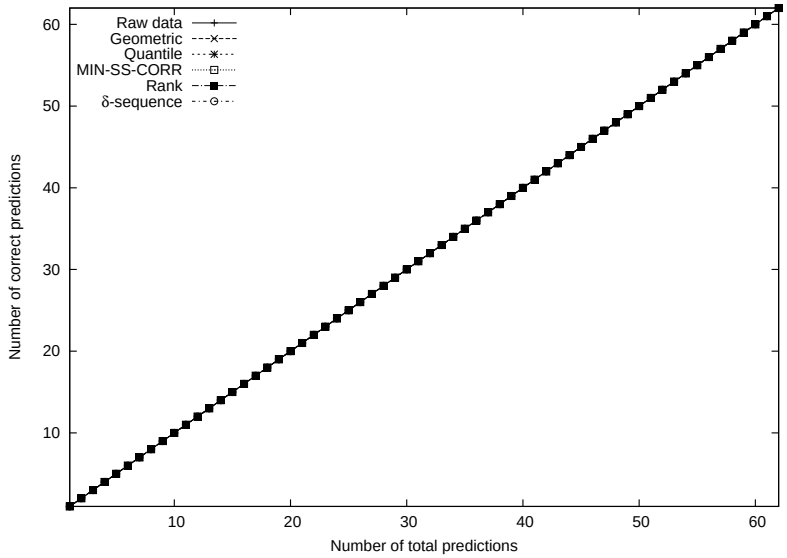
(a)

Figure 3.7: (SVM-RFE CLASSIFIER, ALON DATA) Leave-one-out prediction accuracy as a function of the number of genes or δ's used for classification. Note that the maximum number of δ's available for this dataset is 1000. The classifiers using different input data are all able to achieve 100% LOO accuracy over a range of the number of included genes or δ's. The behaviors of the classifiers using data obtained from the three normalization methods: geometric, quantile and MIN-SS-CORR are almost the same and better than that using just the raw data (See also Fig. 3.7b). The behavior of rank normalization is more variable with prediction accuracy dropping sharply beyond about 1200 genes.

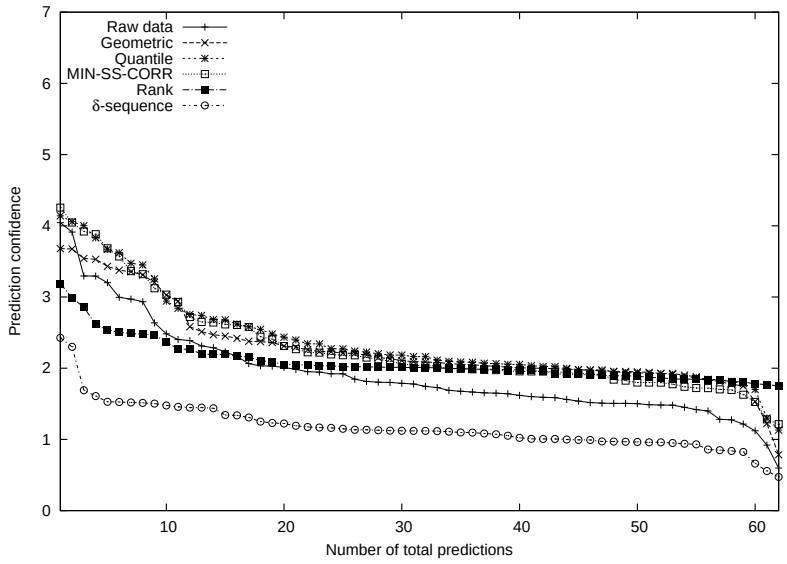


(b)

Figure 3.7: continued. Leave-one-out average prediction confidence as a function of the number of genes or δ's used for classification. Note that the maximum number of δ's available for this dataset is 1000. The average prediction confidence for rank normalized data starts decreasing for number of genes greater than about 300. Similarly the prediction confidence achieved by the classifier using δ-sequences is relatively low over the range of its 100% LOO prediction accuracy.

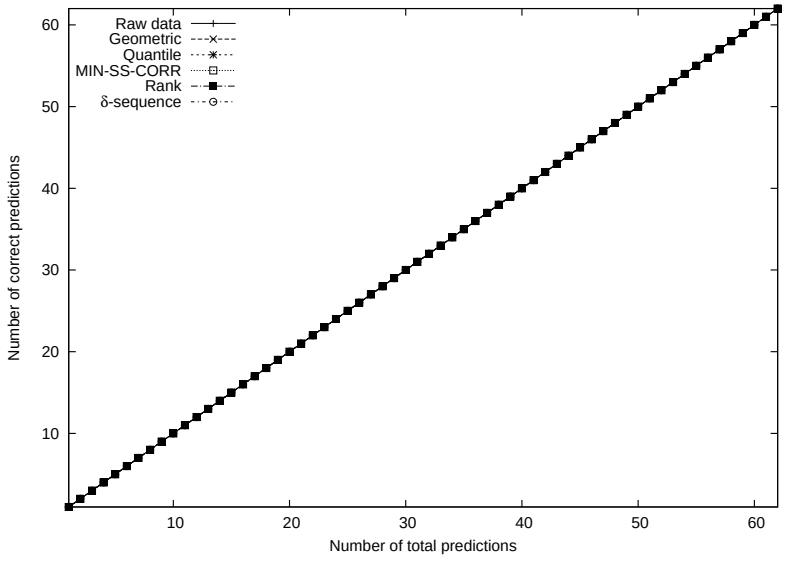


(a) 50 genes or δ's

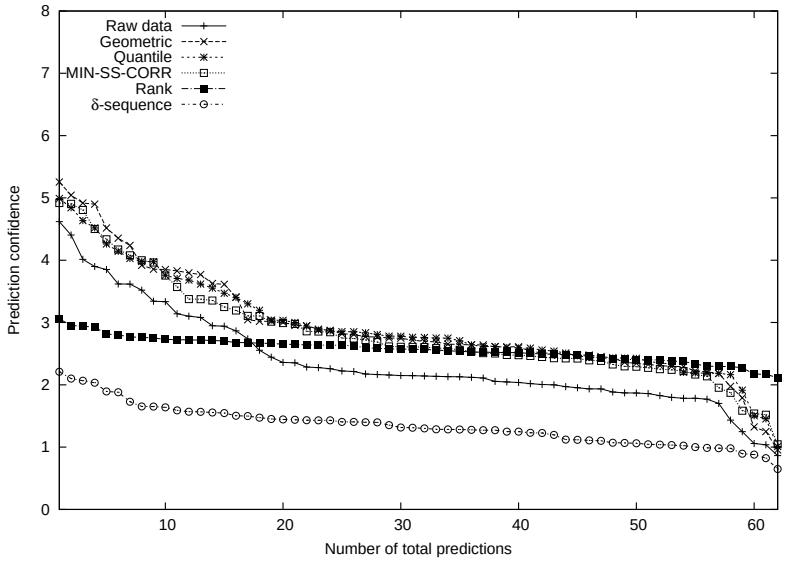


(b) 50 genes or δ's

Figure 3.8: (SVM-RFE CLASSIFIER, ALON DATA) Leave-one-out prediction behavior for a fixed number of genes or δ's: 50, 100 and 500. The cumulative number of correct predictions as a function of the ordered LOO sample indices is plotted on the top panel and the respective prediction confidences are plotted on the bottom panel. The test samples are ordered in the decreasing order of their prediction confidences given by the corresponding classifiers. Most of the classifiers achieve almost 100% LOO prediction accuracy except for the one that uses δ-sequences (500 δ's). The prediction confidences of the classifiers using the data normalized by the three normalization methods:geometric, quantile and MIN-SS-CORR are usually higher than those using the raw, rank normalized data or δ-sequences. Note that since all the methods have 100% prediction accuracy above, their plots overlap.

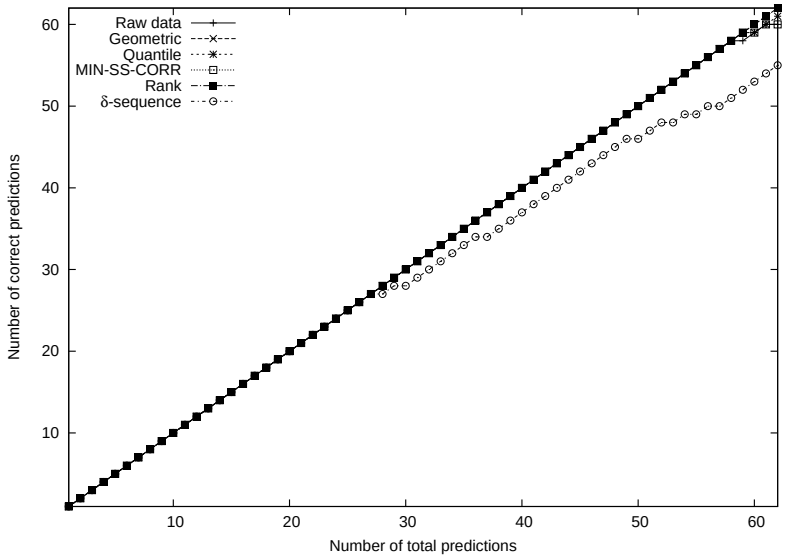


(c) 100 genes or δ's

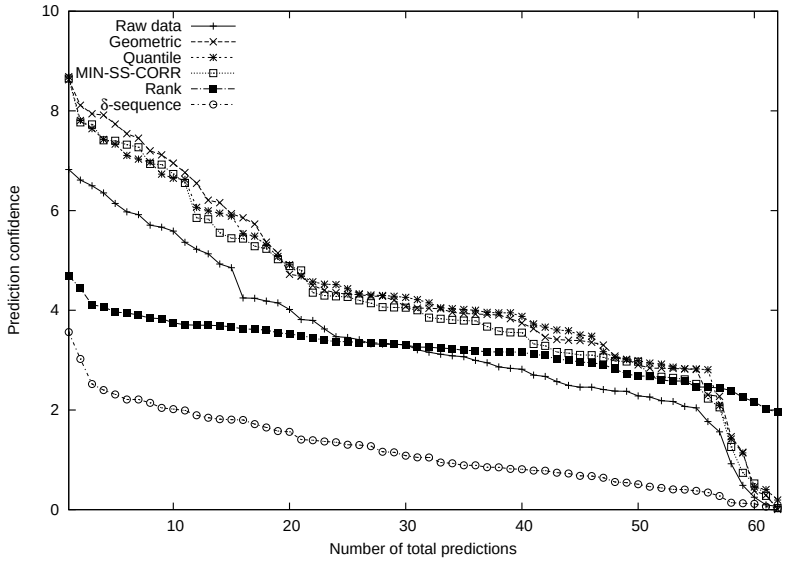


(d) 100 genes or δ's

Figure 3.8: continued. Note that since all the methods have 100% prediction accuracy above, their plots overlap.

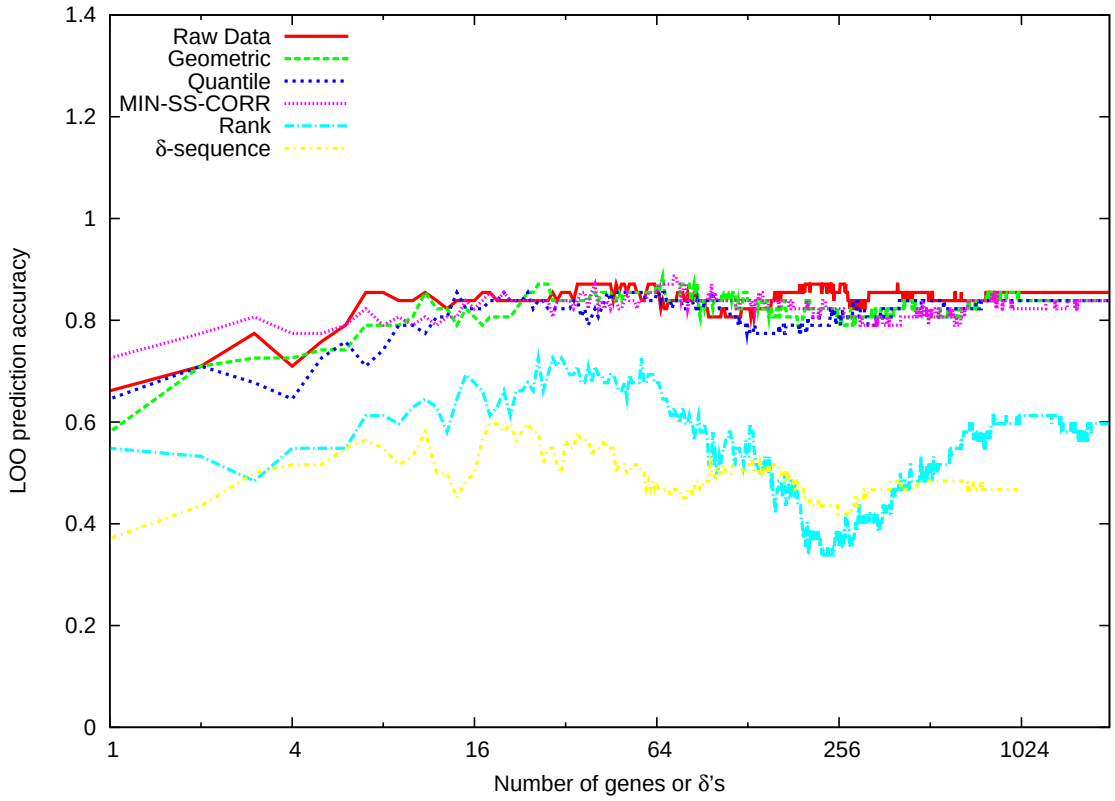


(e) 500 genes or δ's



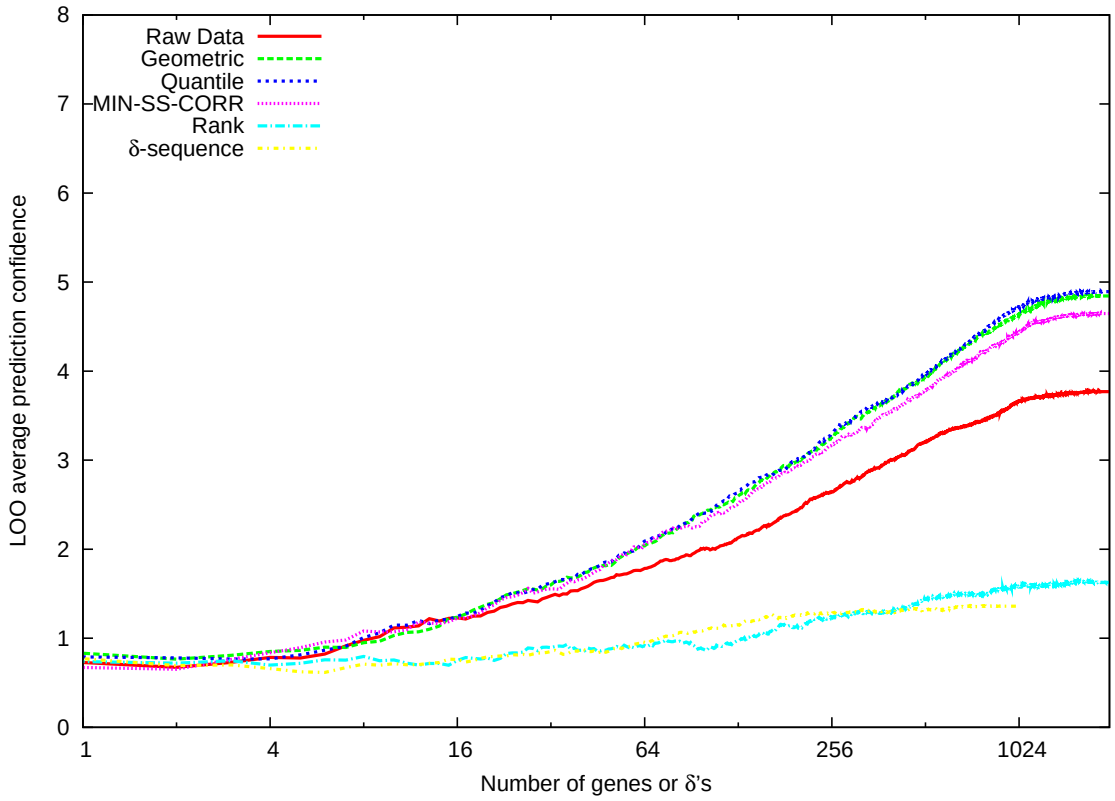
(f) 500 genes or δ's

Figure 3.8: continued.



(a)

Figure 3.9: (SVM-RFE CLASSIFIER, ALON DATA: GENE SELECTION USES LOO) Leave-one-out prediction accuracy as a function of the number of genes or δ's used for classification. Note that the maximum number of δ's available for this dataset is 1000. The behaviors of the classifiers using data obtained from the three normalization methods: geometric, quantile and MIN-SS-CORR are almost the same and better than that using just the raw data (See also Fig. 3.9b). The classifiers using rank normalized and δ-sequence data exhibit lower prediction accuracy overall. and lower average prediction confidence, especially as the number of genes increases.



(b)

Figure 3.9: continued. Leave-one-out average prediction confidence as a function of the number of genes or δ's used for classification. Note that the maximum number of δ's available for this dataset is 1000. The behaviors of the classifiers using data obtained from the three normalization methods: geometric, quantile and MIN-SS-CORR are almost the same and better than that using just the raw data. The classifiers using rank normalized and δ-sequence data show lower average prediction confidence values relative to the those using raw data or normalized using other methods, especially as the number of genes increases.

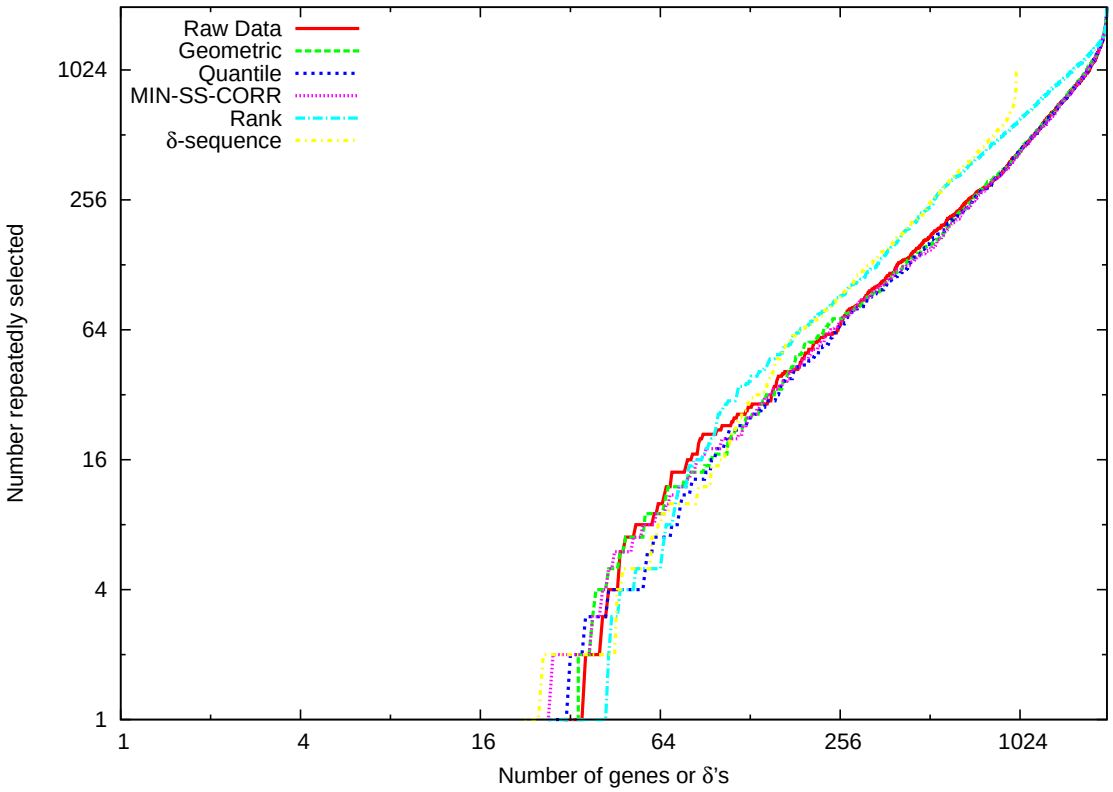
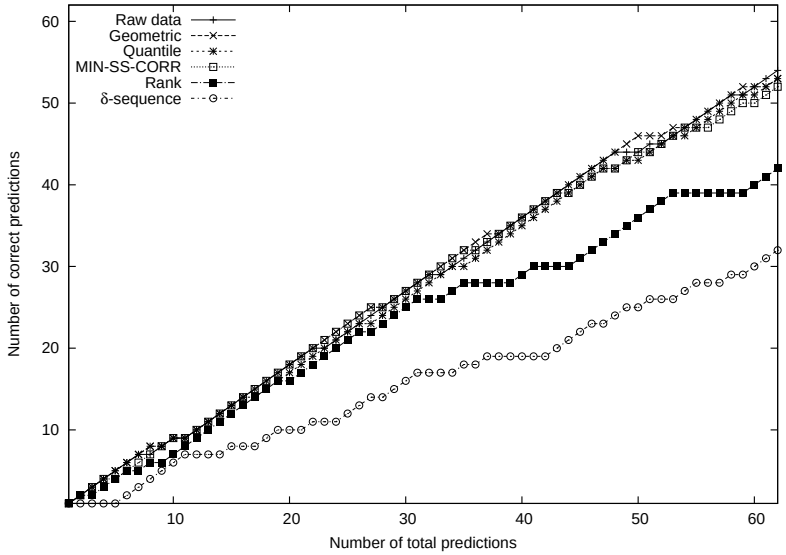
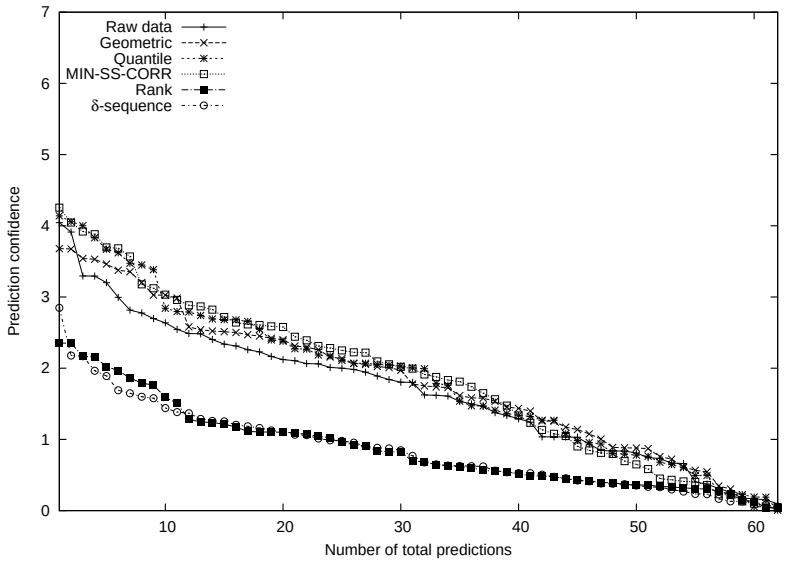


Figure 3.10: (SVM-RFE CLASSIFIER, ALON DATA: GENE SELECTION USES LOO) Number of genes or δ's that are repeatedly selected across all the divisions of the available data into training and test sets of leave-one-out analysis. Note that the maximum number of δ's available for this dataset is 1000. The independent axis is the total number of genes used by the classifier. The higher the number on the dependent axis the lower the variation in the genes that are selected as the most useful for classification across different LOO training sets. None of the classifiers show genes that are repeated until about 20 of them are used for classification. Data normalized using the three methods: geometric, quantile and MIN-SS-CORR results in the corresponding classifier show similar variability in the selected genes, generally better than or similar to that using raw data. Rank normalized data and δ-sequences seem to be slightly better as the number of genes or δ's increases beyond about 64. Note that as the number of genes or δ's reach their maximum then all of them are repeatedly selected due to which the curve for the δ-sequence data intersects the ideal curve at 1000. Similarly for the other datasets.

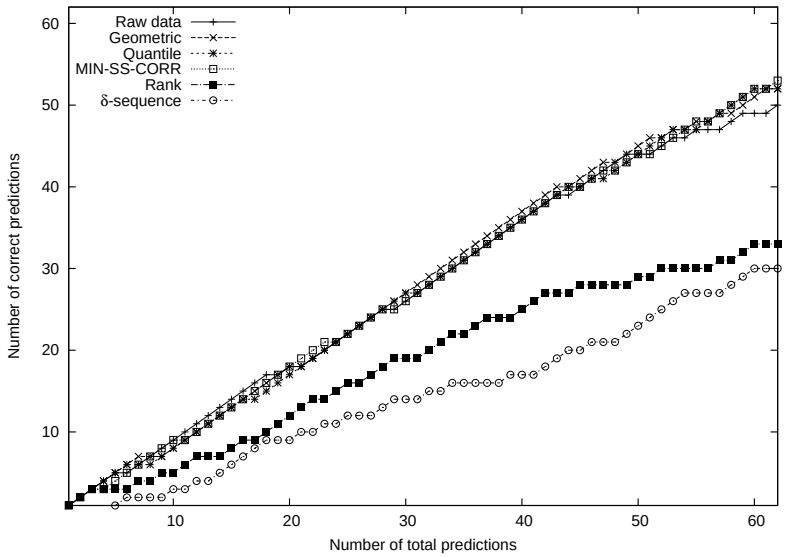


(a) 50 genes or δ's

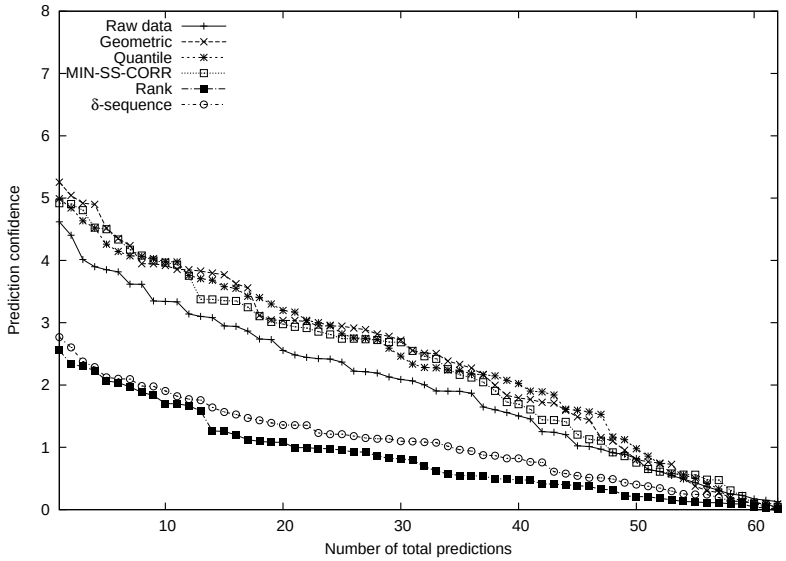


(b) 50 genes or δ's

Figure 3.11: (SVM-RFE CLASSIFIER, ALON DATA: GENE SELECTION USES LOO) Leave-one-out prediction behavior for a fixed number of genes or δ's: 50, 100 and 500. The cumulative number of correct predictions as a function of the ordered LOO sample indices is plotted on the top panel and the respective prediction confidences are plotted on the bottom panel. The test samples are ordered in the decreasing order of their prediction confidences given by the corresponding classifiers. The prediction confidences of the classifiers using data normalized by the three normalization methods:geometric, quantile and MIN-SS-CORR are usually higher than those using the raw data, rank normalized data or δ-sequences.

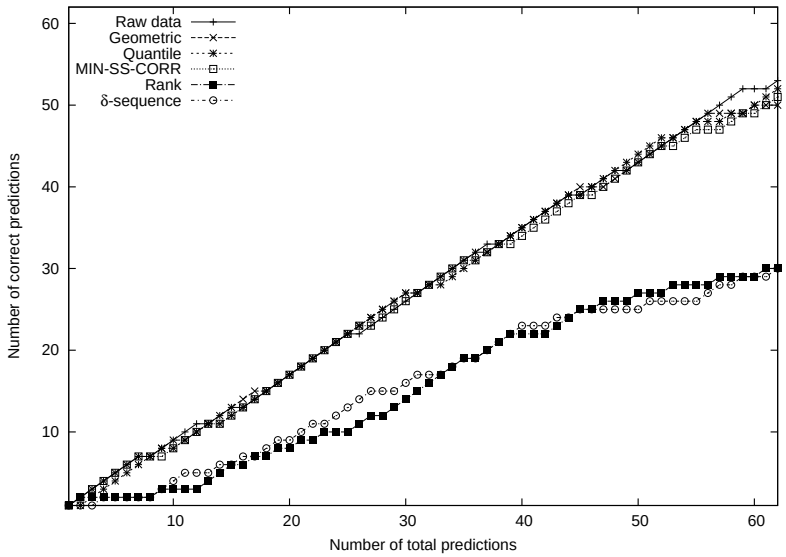


(c) 100 genes or δ's

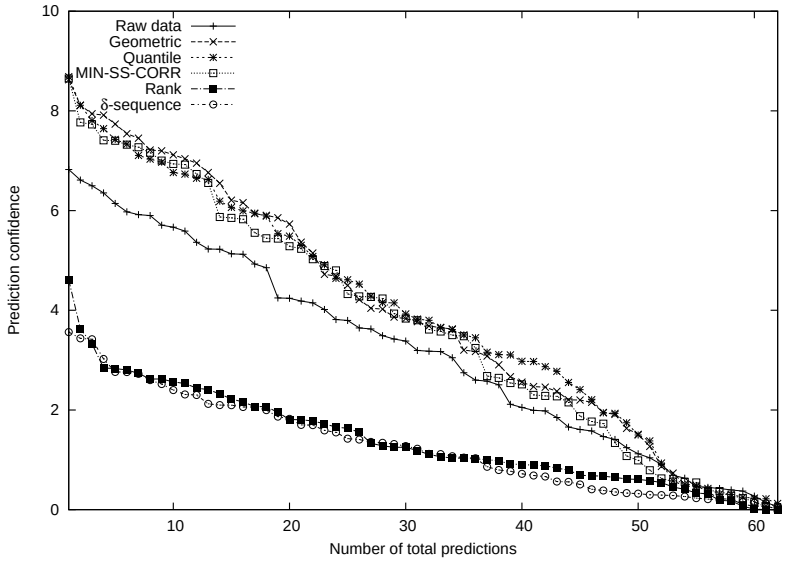


(d) 100 genes or δ's

Figure 3.11: continued.



(e) 500 genes or δ's



(f) 500 genes or δ's

Figure 3.11: continued.

3.1.4 Experiments on angiogenesis dataset

The raw data and the normalized versions of Hoying dataset are evaluated similar to the colon cancer dataset using the two evaluation schemes. The Hoying dataset has a relatively large number of genes (15600) compared to the colon cancer dataset (2000). The MIN-SS-CORR normalization method aims to minimize the sum of squared correlation coefficients between all pairs of genes. This would increase the number of pairs exponentially with the number of genes in the dataset. It is perhaps not necessary to consider all pairs given that the number of free parameters during the optimization process is much small relative to the number of all gene pairs. So we perform the optimization using only the first 500 genes in the dataset and use the resulting offsets to normalize the entire data.

The increase in the number of genes also causes an increase in the computation time for the different classifiers if a single gene is included or eliminated in the sequential gene selection process. To save time while computing the classification metrics we start with all the genes and eliminate them in exponentially decreasing chunks during gene selection. Specifically we consider 15600, 7800, 3900 and so on up to 1 gene (or δ-sequence) for the experiments.

Hypothesis testing results

The results of hypothesis testing on the raw linlog transformed data and the data normalized using the different methods are plotted in Fig. 3.12. Unlike the colon cancer data, the raw data does not seem to exhibit a significant support for the phenotypic classes. This could be due to various reasons including the noise and variability in the data introduced in the whole process of preparing the microarrays to recording the spot intensities. The hope is that post-normalization the evidence for phenotypic classes increases. Except for rank normalization, the others seem to achieve this goal to varying degrees with quantile normalization doing the best followed by δ-sequences, geometric and MIN-SS-CORR normalization. The TEST curve for quantile normalized data is above the RANDOM-5-percentile curve for

a certain number of genes at higher values of the absolute correlation coefficient threshold. A similar but slightly inferior behavior is observed for δ-sequences, geometric and MIN-SS-CORR normalized data.

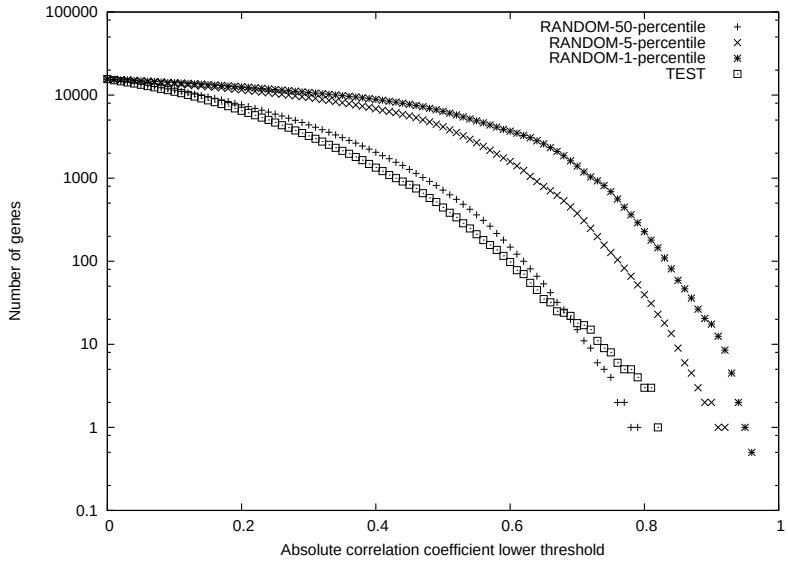
In general, this behavior is similar to that observed in the case of colon cancer dataset except for the δ-sequences, which is performing favorably in this case. Note that in order to save time not all the gene pairs were used for MIN-SS-CORR normalization. This could have contributed to its relatively inferior performance compared to the earlier experiments with colon cancer data but we are yet to investigate this.

Classification results using Golub classifier

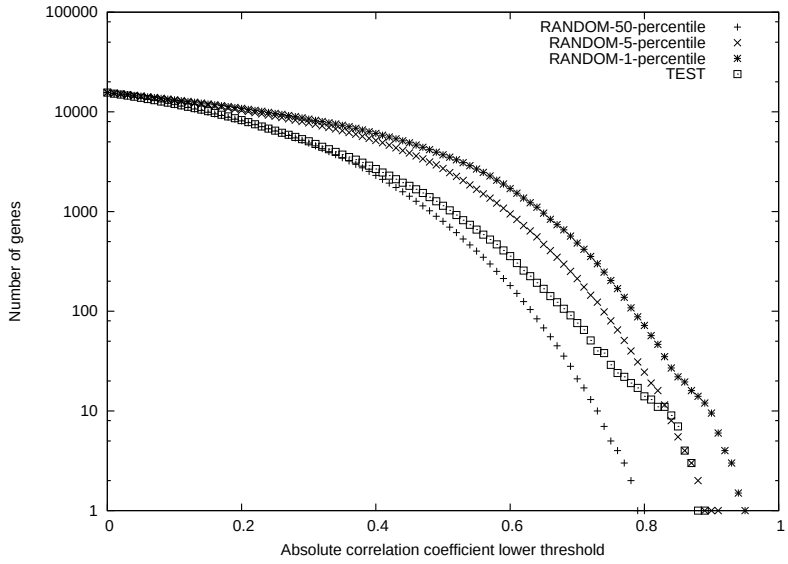
The results of classification using the Golub classifier are shown on Figs. 3.13 through 3.17. The results of using all the available samples for gene selection are shown in the plots of Figs. 3.13-3.14 and using only the training samples of the LOO divisions are shown in the plots of Figs. 3.15-3.17. All the normalization methods except rank normalization seem to help the raw data in terms of achieving higher prediction accuracy and confidence on the leave-one-out test samples. The degree to which they help roughly follows their order of performance in hypothesis testing with quantile normalization outperforming others. Quantile normalized data also exhibits lower variation in terms of the genes that are repeatedly selected across the different LOO sets as seen in the plot of Fig. 3.16. Unlike the colon cancer data, the Golub classifiers built using normalized data can achieve 100% accuracy with the exception of rank normalization and δ-sequences when gene selection uses LOO.

Classification results using SVM-RFE classifier

When using the SVM-RFE classifier with different input data the classification accuracy is higher compared to the corresponding Golub classifier. If all the available data is used for gene selection, then all the classifiers achieve 100% accuracy for most of the range of number of included genes (see Figs. 3.18 and 3.19). The prediction confidence is generally higher for normalized data than that for the raw data imply-

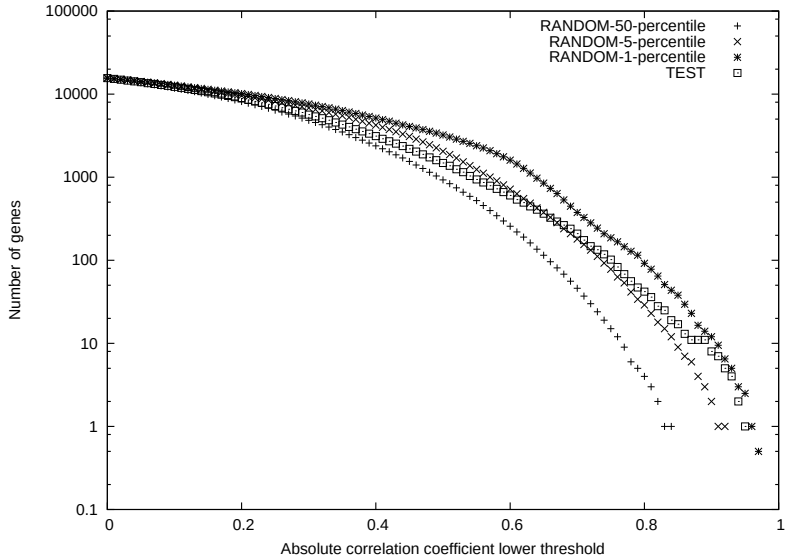


(a) Raw data

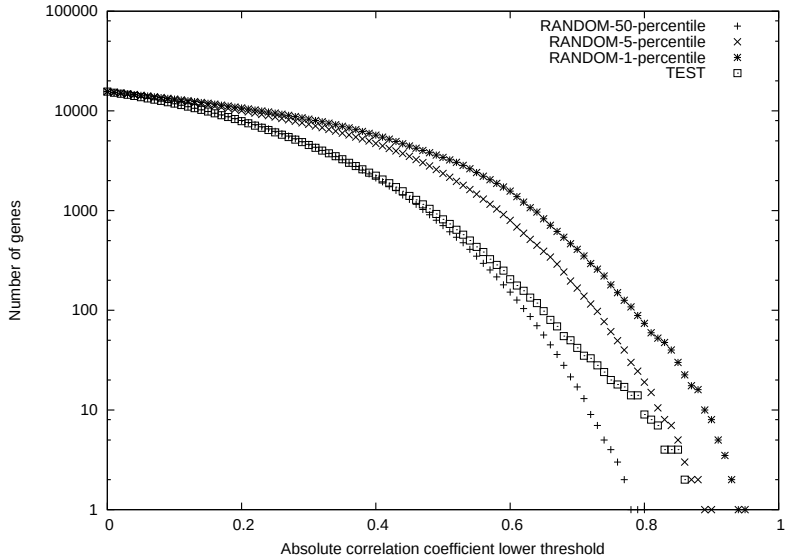


(b) Geometric normalization

Figure 3.12: Class label hypothesis testing of the Hoying angiogenesis dataset before and after normalization. TEST refers to the number of genes that have their absolute correlation coefficient value above a certain level with the correlation coefficient being computed with the phenotypic pattern of class labeling. See Section 3.0.6. RANDOM refers to statistics of the distribution obtained by considering 1000 different random binary labeling patterns on the tissue samples (null distribution). Plotting the median (50 percentile), 5-percentile and 1-percentile points of the null distribution at different minimum levels of the absolute correlation coefficient results in the three curves RANDOM-50-percentile, RANDOM-5-percentile and RANDOM-1-percentile respectively. The higher the TEST curve above the RANDOM curves the better is the statistical significance suggested by the data for the phenotypic classes.

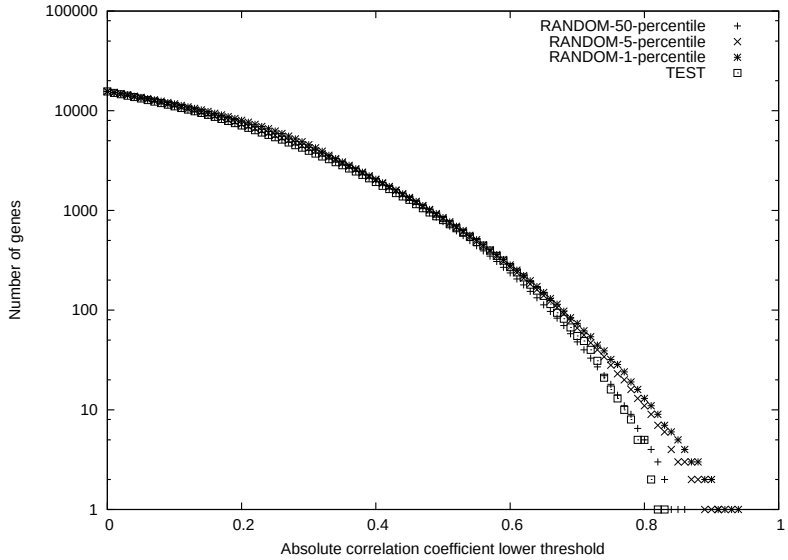


(c) Quantile normalization

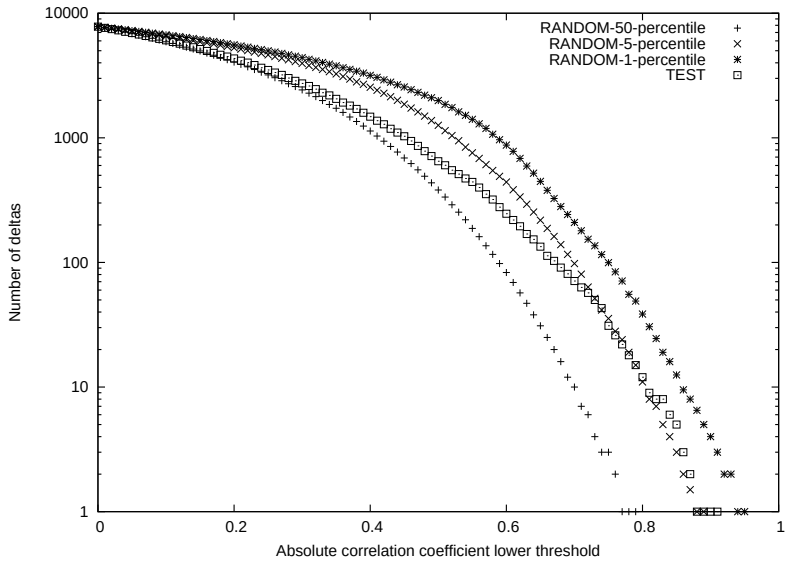


(d) MIN-SS-CORR normalization

Figure 3.12: continued.

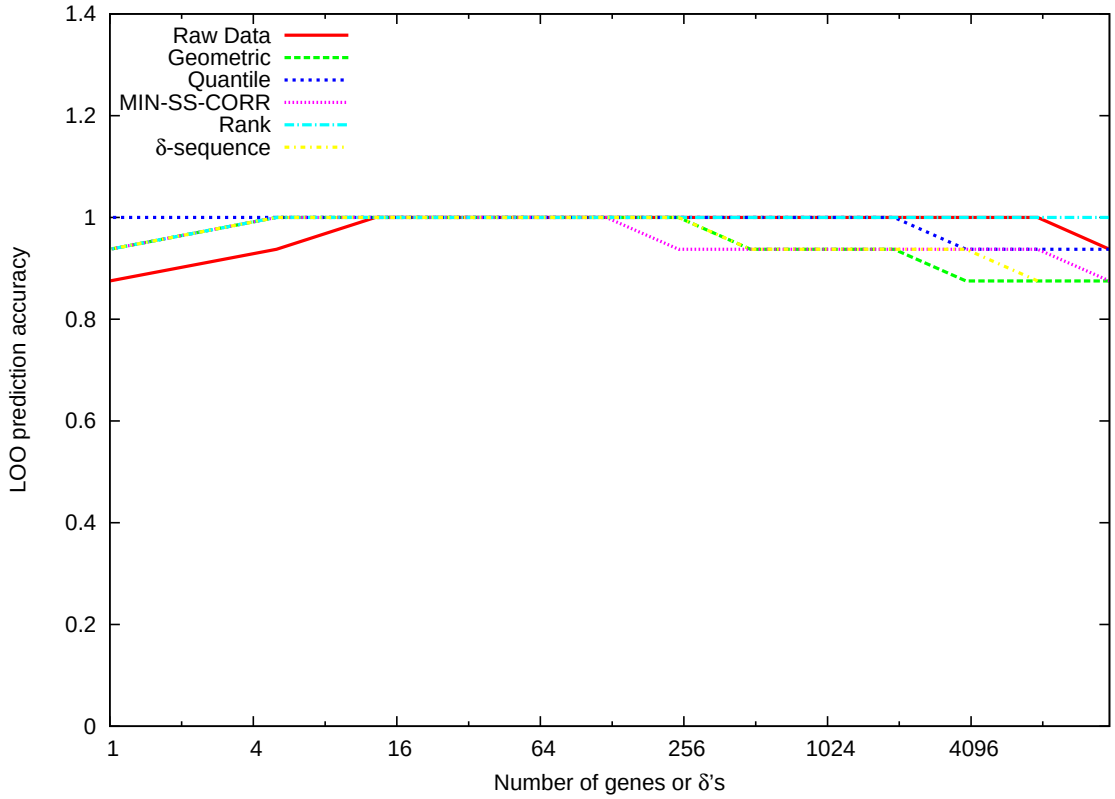


(e) Rank normalization



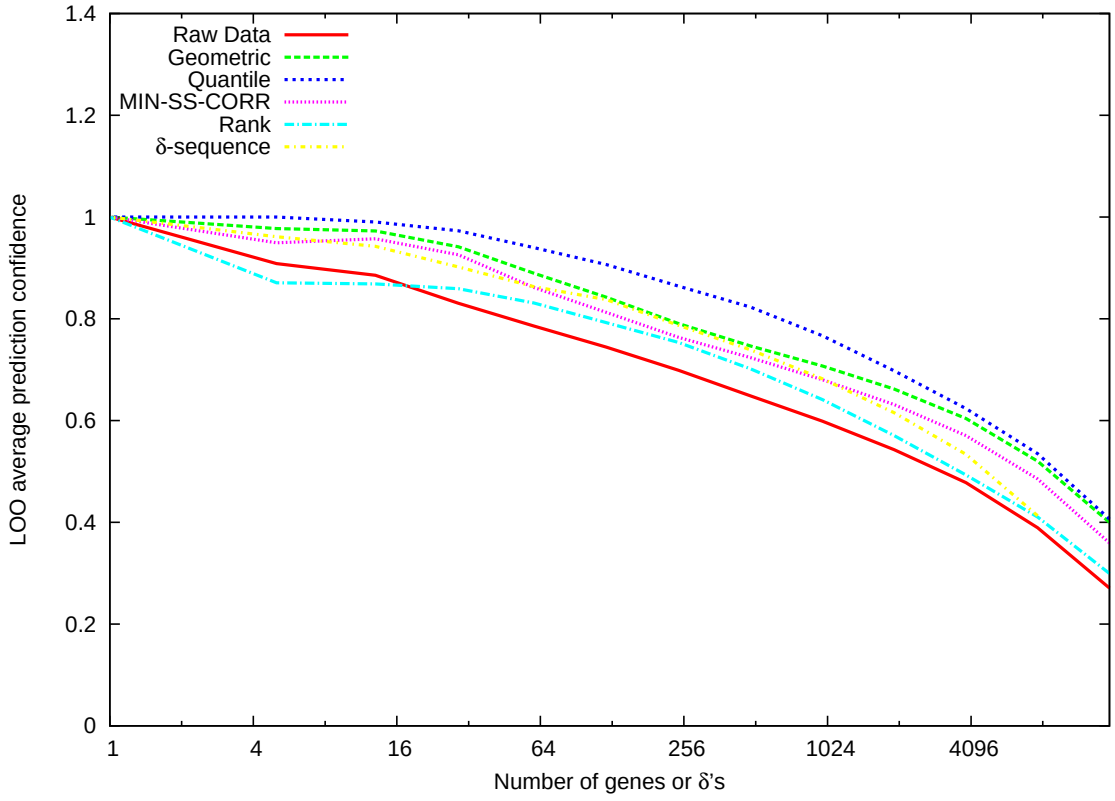
(f) δ-sequences

Figure 3.12: continued.



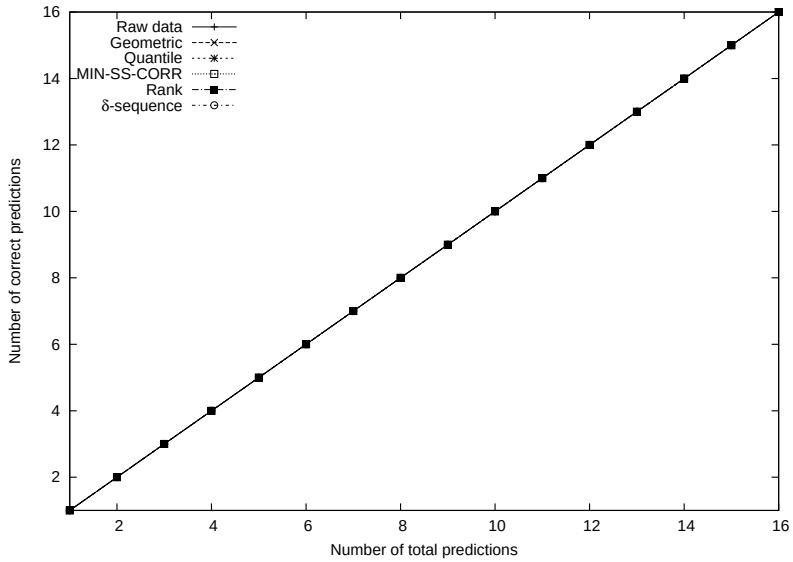
(a)

Figure 3.13: (GOLUB CLASSIFIER, HOYING DATA) Leave-one-out prediction accuracy as a function of the number of genes or δ's used for classification. Note that the maximum number of δ's available for this dataset is 7800. All the classifiers are able to achieve 100% prediction accuracy over a certain range of the number of included genes.

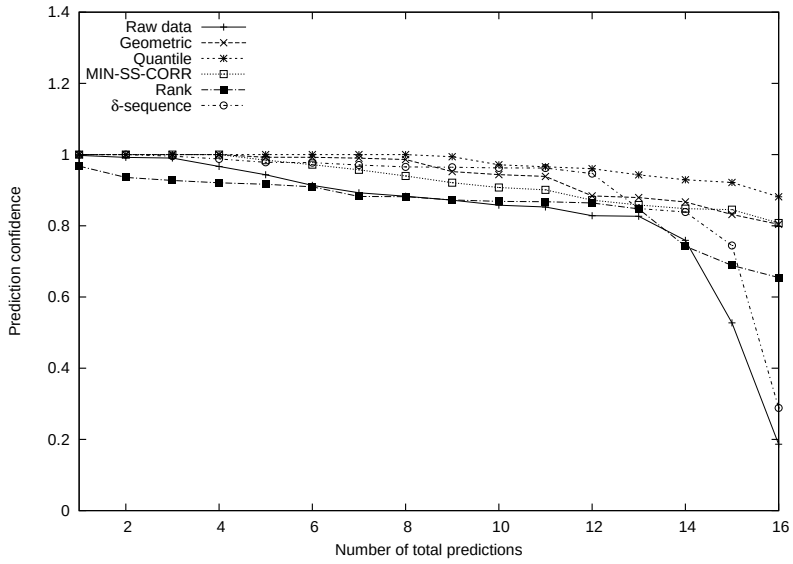


(b)

Figure 3.13: continued. Leave-one-out average prediction confidence as a function of the number of genes or δ's used for classification. Note that the maximum number of δ's available for this dataset is 7800. The average prediction confidence of the normalized data is generally higher than that of the raw data for number of genes greater than about 16. The classifier built using quantile normalized data seems to perform the best both in terms of prediction accuracy and average prediction confidence.

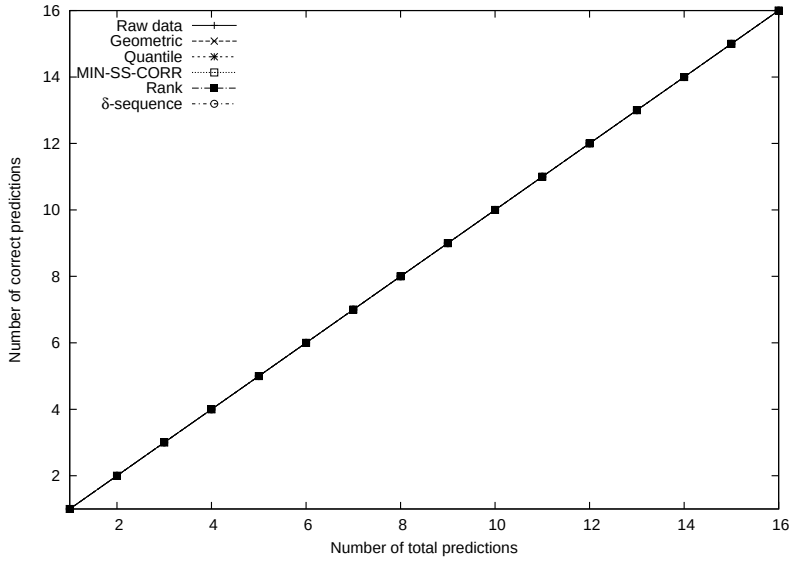


(a) 29 genes or δ's

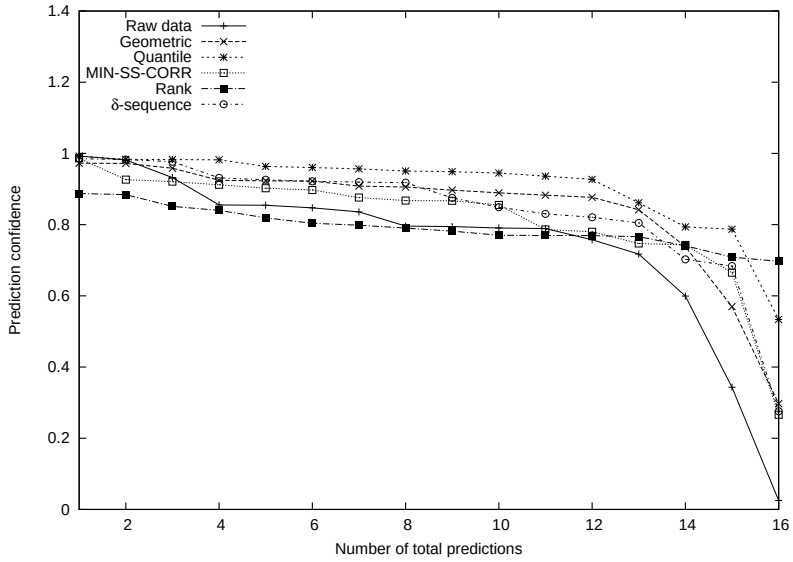


(b) 29 genes or δ's

Figure 3.14: (GOLUB CLASSIFIER, HOYING DATA) Leave-one-out prediction behavior for a fixed number of genes or δ's : 29, 121 and 487. The cumulative number of correct predictions as a function of the ordered LOO sample indices is plotted on the top panel and the respective prediction confidences are plotted on the bottom panel. The test samples are ordered in the decreasing order of their prediction confidences given by the corresponding classifiers. All the classifiers have almost 100% prediction accuracy for the number of genes or δ's considered. The classifier built using quantile normalized data seems to perform the best in terms of prediction confidences over the samples. Note that since all the methods have 100% prediction accuracy above, their plots overlap.

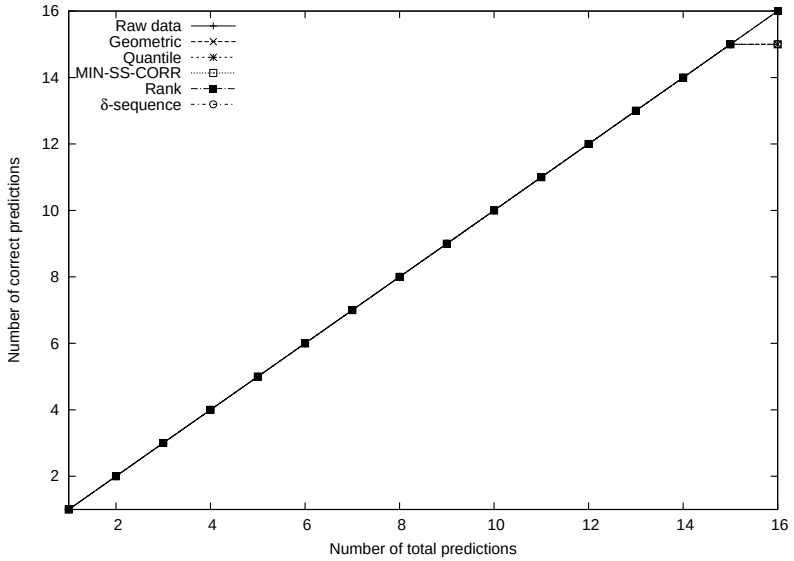


(c) 121 genes or δ's

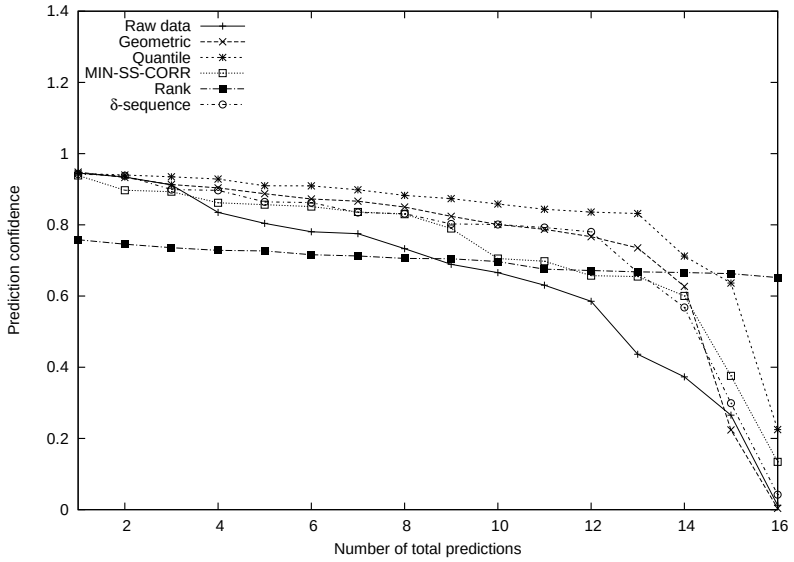


(d) 121 genes or δ's

Figure 3.14: continued. Note that since all the methods have 100% prediction accuracy above, their plots overlap.

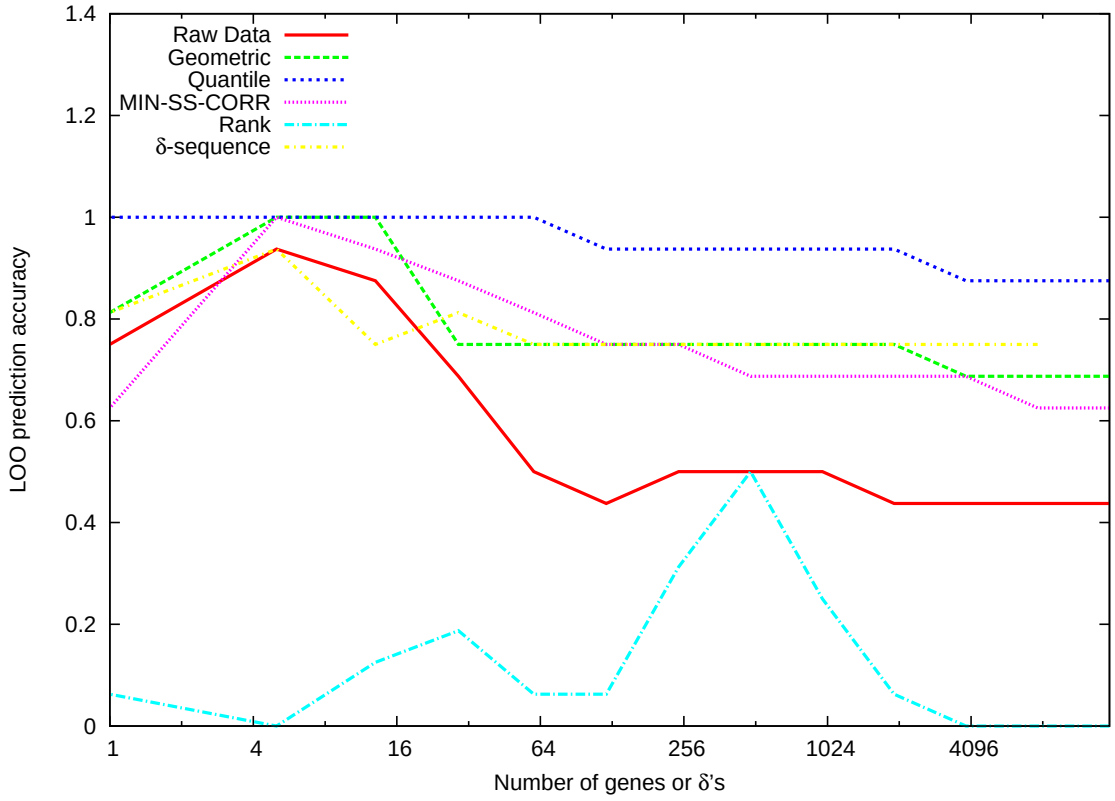


(e) 487 genes or δ's



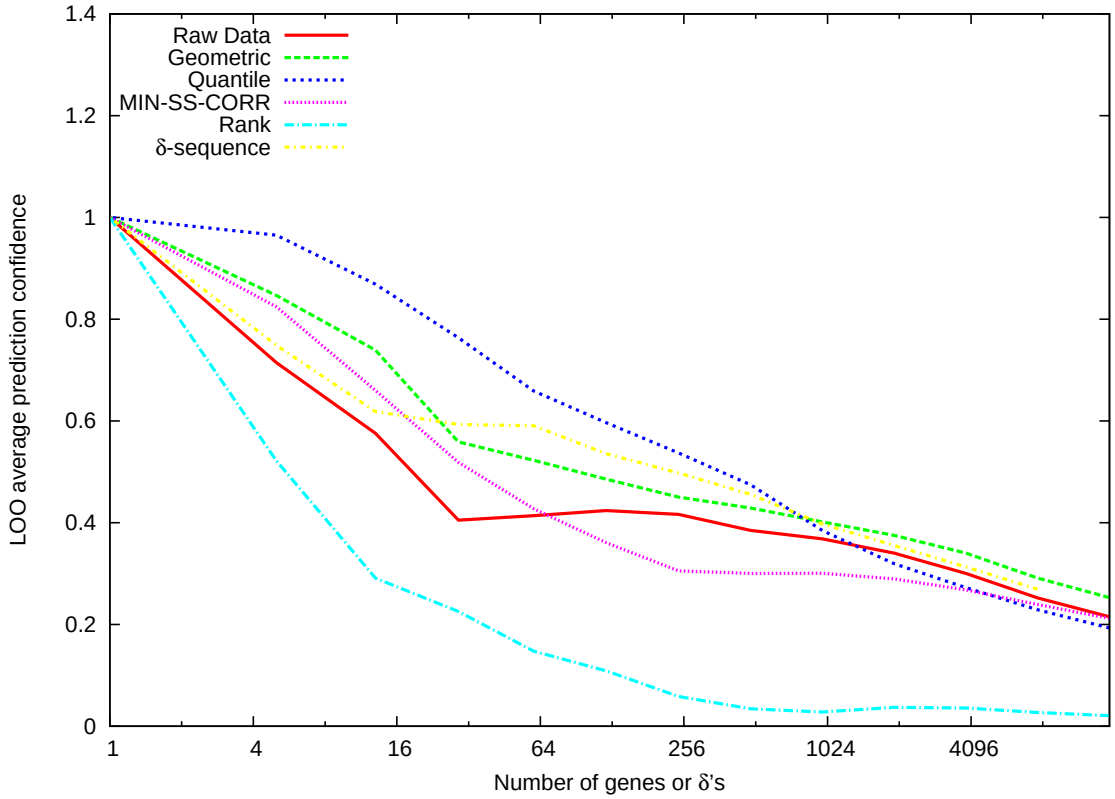
(f) 487 genes or δ's

Figure 3.14: continued.



(a)

Figure 3.15: (GOLUB CLASSIFIER, HOYING DATA: GENE SELECTION USES LOO) Leave-one-out prediction accuracy as a function of the number of genes or δ's used for classification. Note that the maximum number of δ's available for this dataset is 7800. All the normalization methods except δ-sequences and rank normalization help achieve 100% LOO prediction accuracy for at least one set of genes.



(b)

Figure 3.15: continued. Leave-one-out average prediction confidence as a function of the number of genes or δ's used for classification. Note that the maximum number of δ's available for this dataset is 7800. Quantile normalized data performs the best both in terms of prediction accuracy and average prediction confidence and rank normalization the worst.

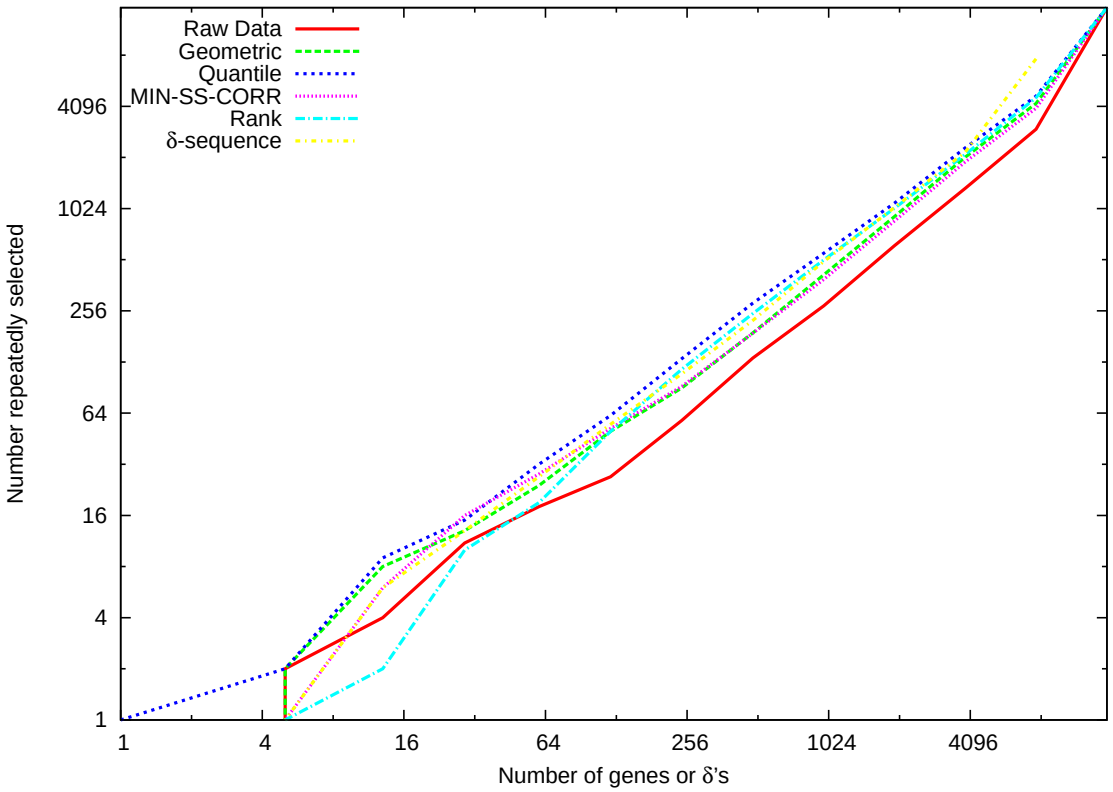
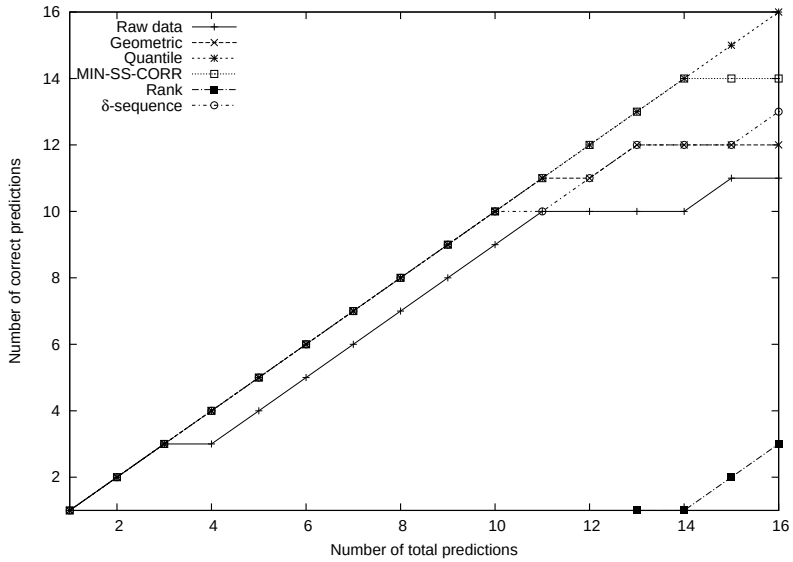
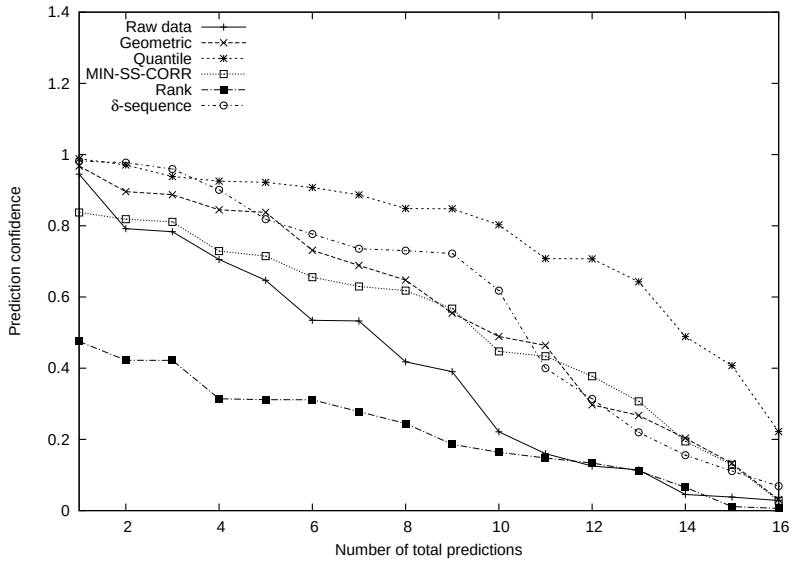


Figure 3.16: (GOLUB CLASSIFIER, HOYING DATA: GENE SELECTION USES LOO) Number of genes or δ's that are repeatedly selected across all the divisions of the available data into training and test sets of leave-one-out analysis. Note that the maximum number of δ's available for this dataset is 7800. The independent axis is the total number of genes used by the classifier. The higher the number on the dependent axis the lower the variation in the genes that are selected as the most useful for classification across different LOO training sets. Normalized data generally shows lower variation in the genes that are selected across the LOO divisions especially as the number of genes increases. Quantile normalization performs consistently better than other methods. Note that as the number of genes or δ's reach their maximum then all of them are repeatedly selected due to which the curve for the δ-sequence data crosses over the other curves at 7800. Similarly for the other datasets.

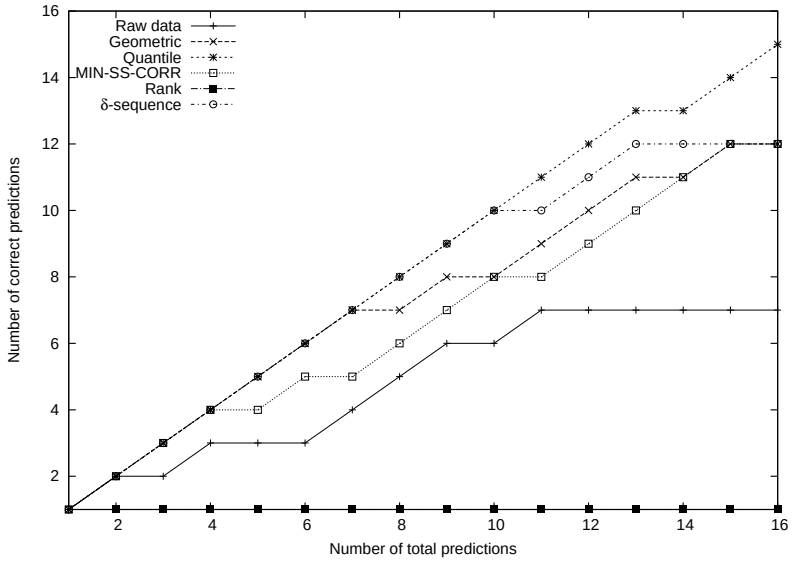


(a) 29 genes or δ's

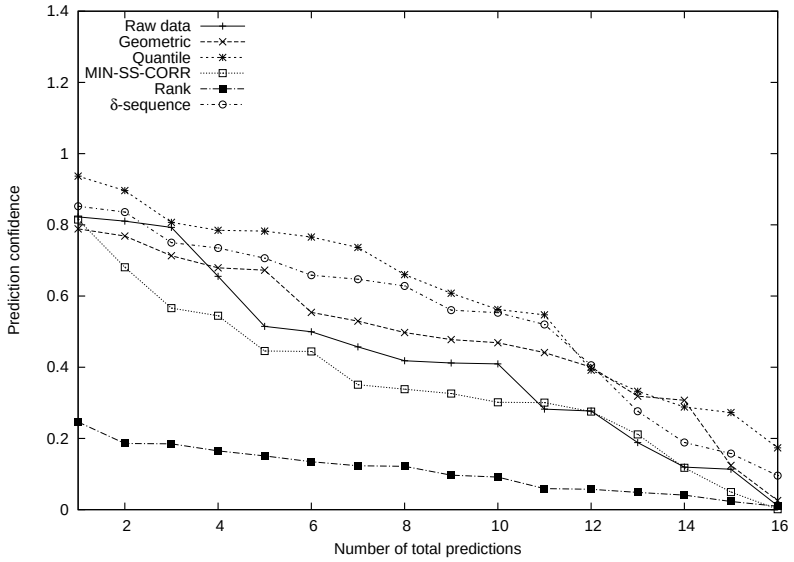


(b) 29 genes or δ's

Figure 3.17: (GOLUB CLASSIFIER, HOYING DATA: GENE SELECTION USES LOO) Leave-one-out prediction behavior for a fixed number of genes or δ's : 29, 121 and 487. The cumulative number of correct predictions as a function of the ordered LOO sample indices is plotted on the top panel and the respective prediction confidences are plotted on the bottom panel. The test samples are ordered in the decreasing order of their prediction confidences given by the corresponding classifiers. Using data normalized with the quantile and MIN-SS-CORR methods and the δ-sequence data helps classifiers achieve higher prediction accuracy. Classifier built using rank normalized data has the least prediction confidence across most of the samples irrespective of the number of genes or δ's considered.

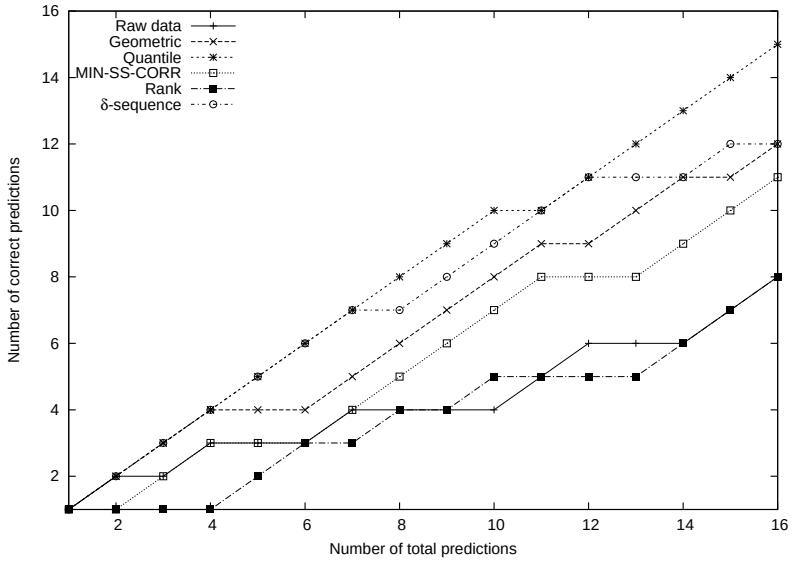


(c) 121 genes or δ's

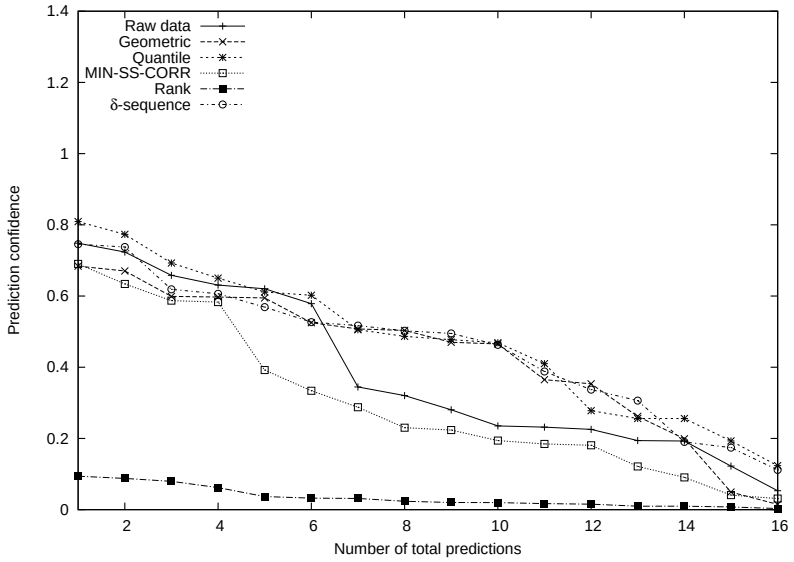


(d) 121 genes or δ's

Figure 3.17: continued.



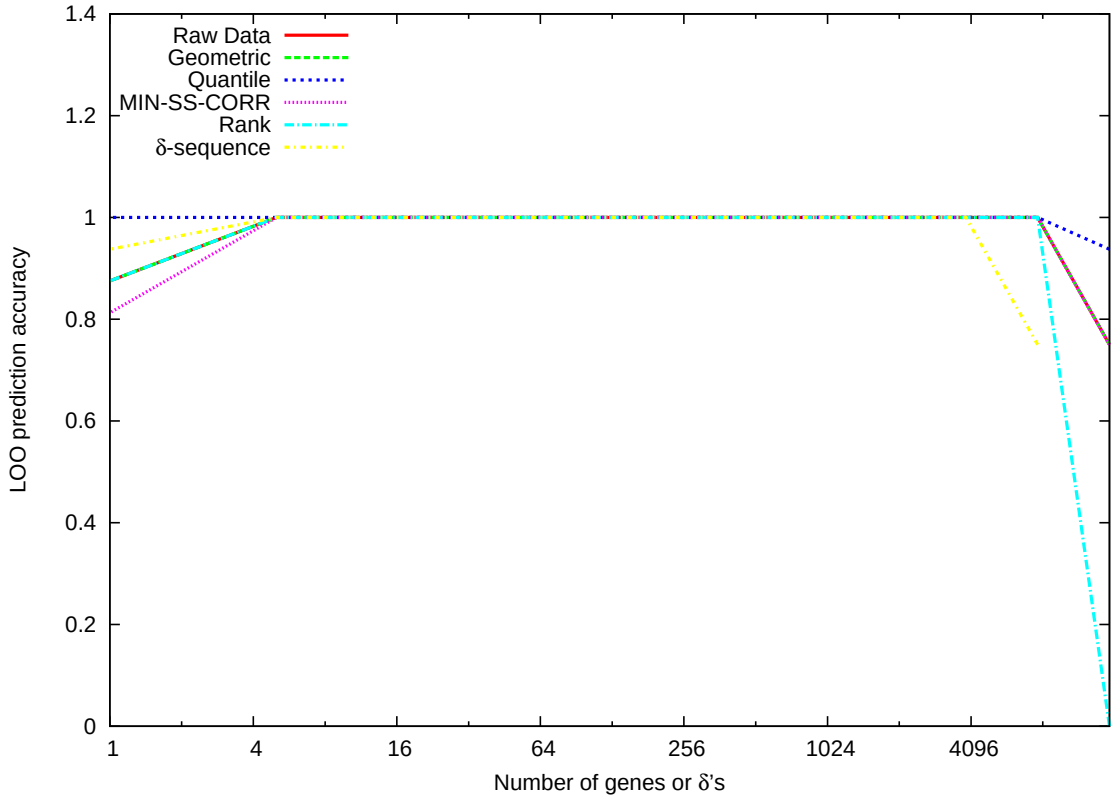
(e) 487 genes or δ's



(f) 487 genes or δ's

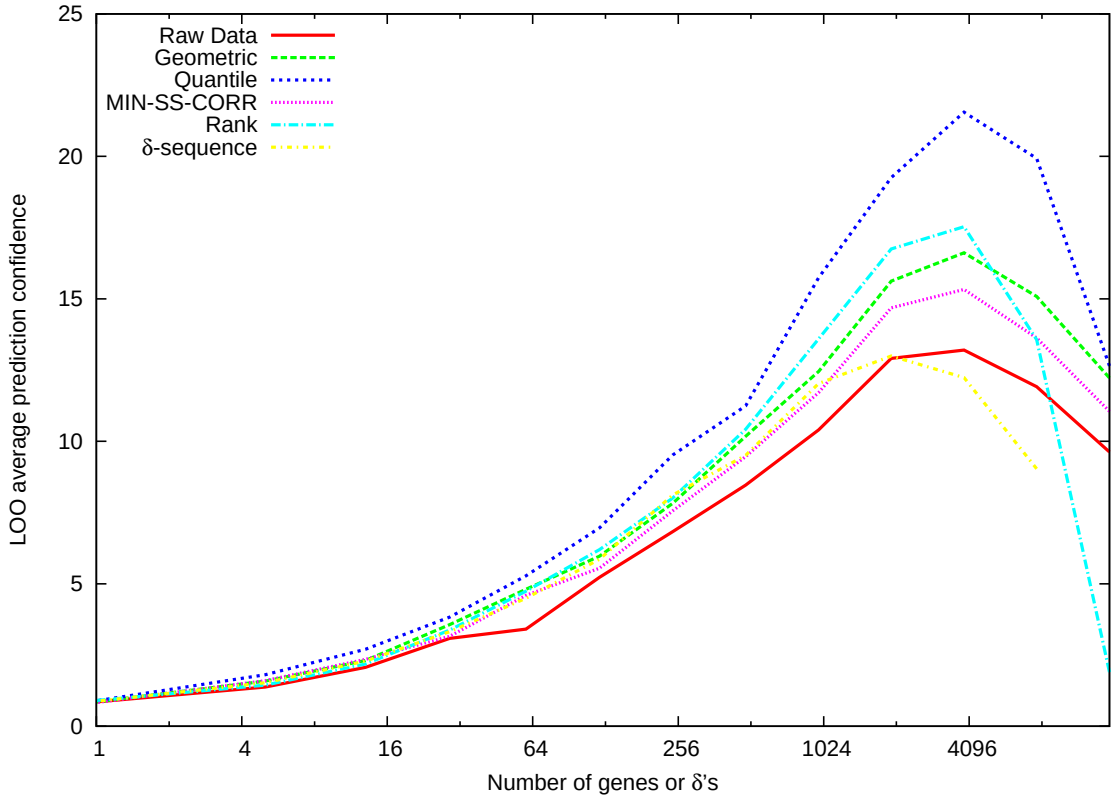
Figure 3.17: continued.

ing that any kind of normalization seems to help in this case. However if only the training samples of the LOO sets are used for gene selection then rank normalized data seems to degrade in classifiability. This has been true in general with other datasets and classifiers as well. The fact that there is a potential information leak when all the available data is used for feature selection seems to help rank normalization the most. As seen from Figs. 3.20 and 3.22, normalization methods other than the rank method improve the classifiability inherent in the raw data or at least preserve it. This probably has to do with the replacement of gene expression values with their discrete ranks, which appears detrimental to the inherent classifiability of the data. The results of the repeatability of gene selection in Fig. 3.21 are more variable. Perhaps a more comprehensive experiment using more number of samples will provide further insight.



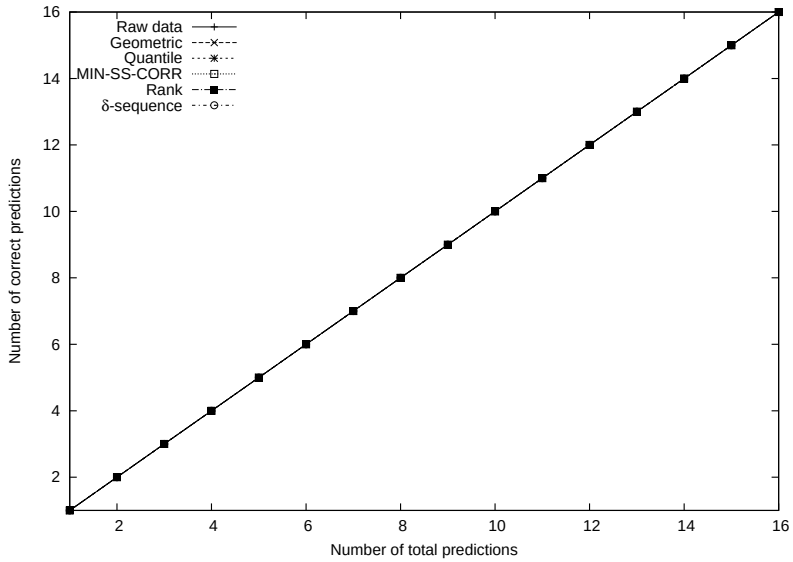
(a)

Figure 3.18: (SVM-RFE CLASSIFIER, HOYING DATA) Leave-one-out prediction accuracy as a function of the number of genes or δ's used for classification. Note that the maximum number of δ's available for this dataset is 7800. Both raw and normalized data achieve 100% prediction accuracy for most of the range of the number of included genes.

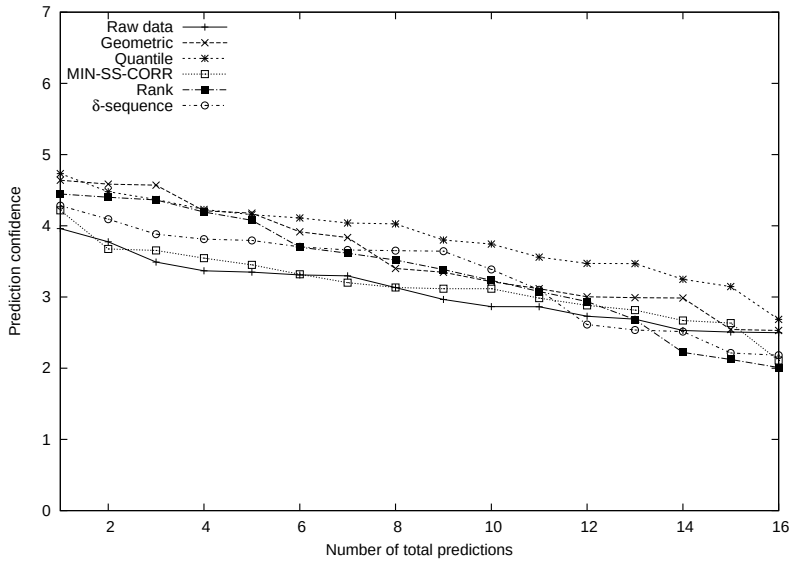


(b)

Figure 3.18: continued. Leave-one-out average prediction confidence as a function of the number of genes or δ's used for classification. Note that the maximum number of δ's available for this dataset is 7800. Normalization seems to improve the average prediction confidence achievable by a classifier over the one using just the raw data. Quantile normalization seems to help the most.

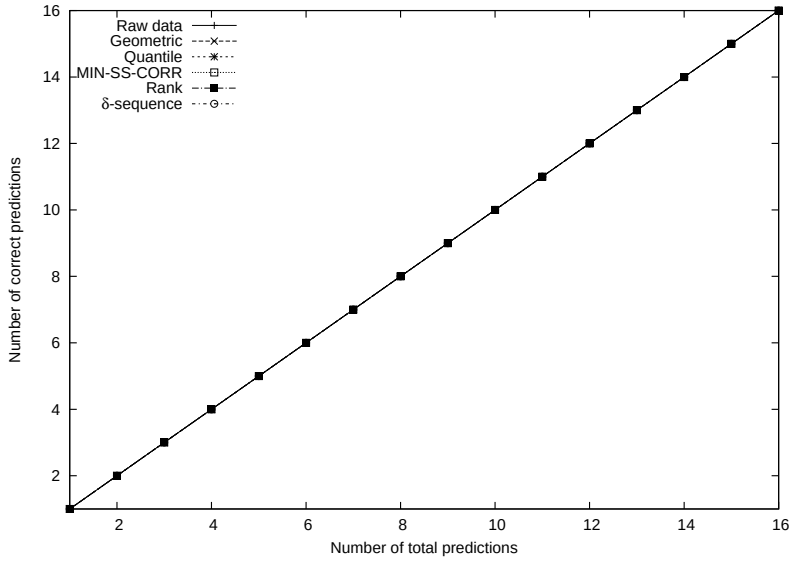


(a) 29 genes or δ's

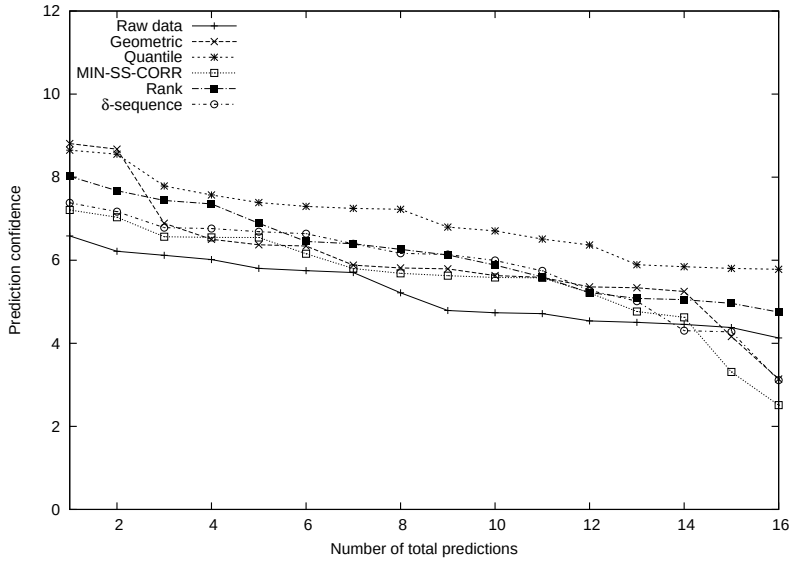


(b) 29 genes or δ's

Figure 3.19: (SVM-RFE CLASSIFIER, HOYING DATA) Leave-one-out prediction behavior for a fixed number of genes or δ's : 29, 121 and 487. The cumulative number of correct predictions as a function of the ordered LOO sample indices is plotted on the top panel and the respective prediction confidences are plotted on the bottom panel. The test samples are ordered in the decreasing order of their prediction confidences given by the corresponding classifiers. The classifiers built using normalized data exhibit at least as good or better prediction confidences for the test samples compared to the one using raw data. The classification is accuracy is 100% irrespective of the number of included genes. Note that since all the methods have 100% prediction accuracy above, their plots overlap.

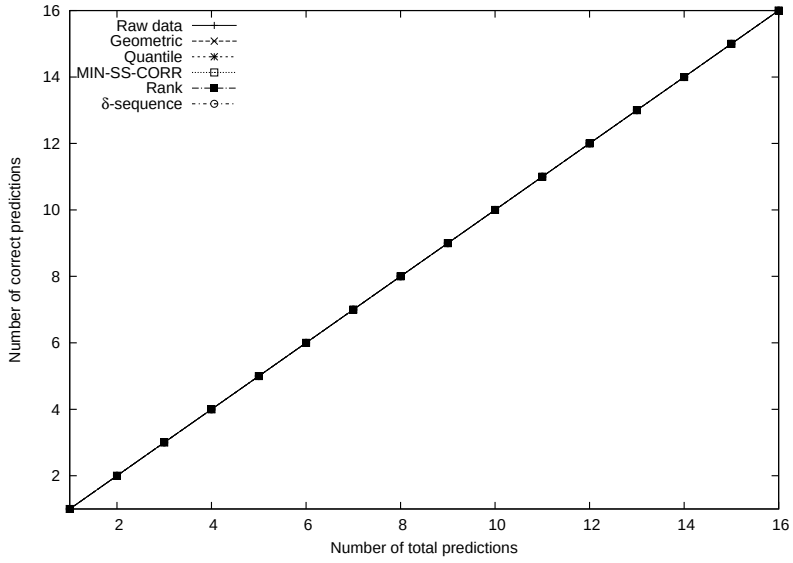


(c) 121 genes or δ's

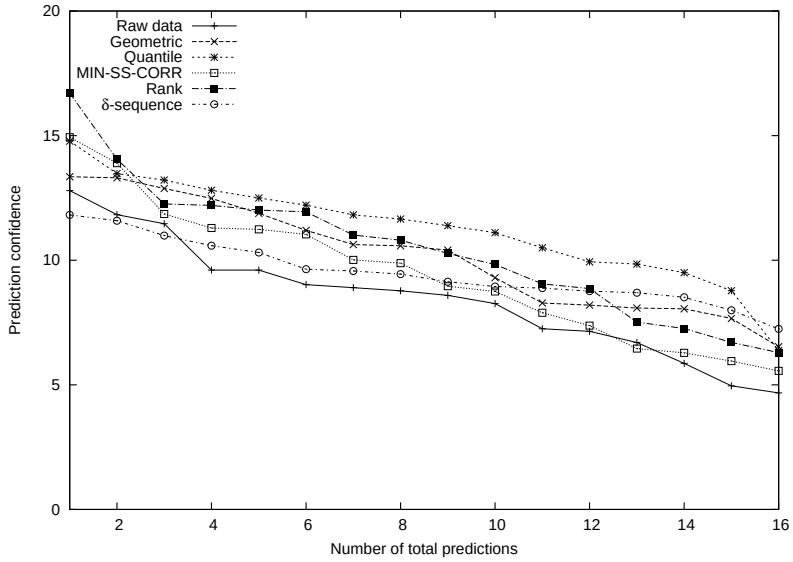


(d) 121 genes or δ's

Figure 3.19: continued. Note that since all the methods have 100% prediction accuracy above, their plots overlap.

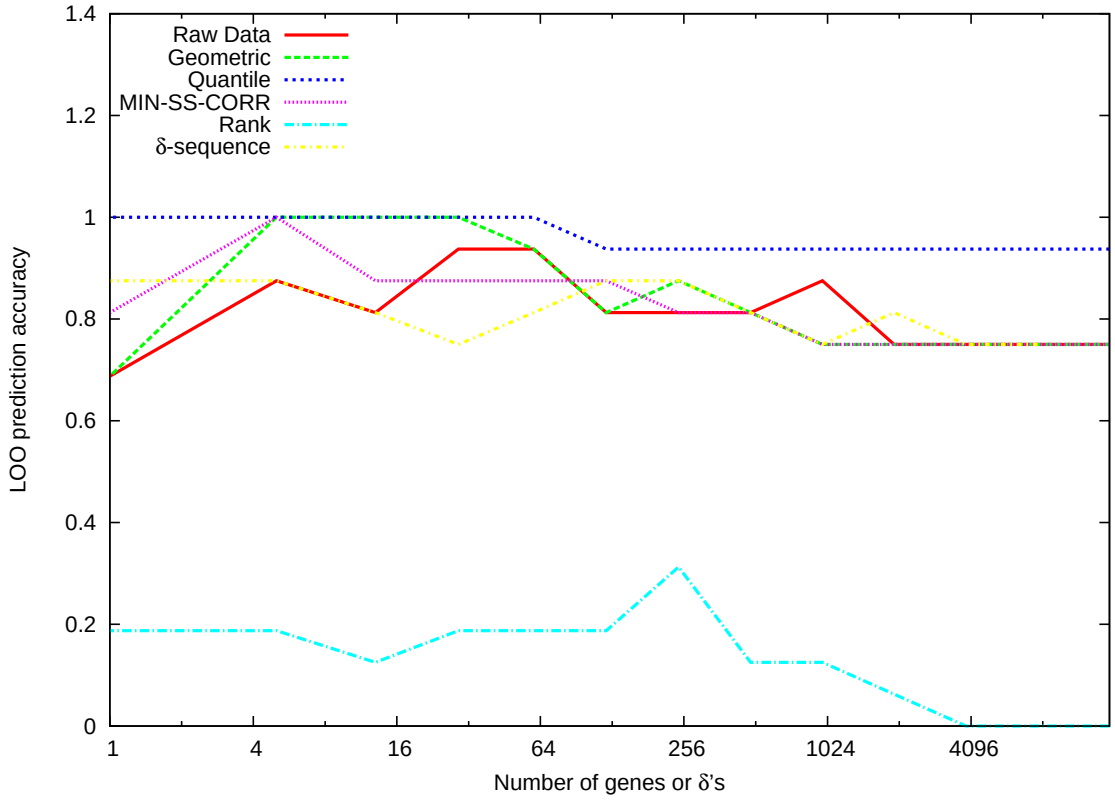


(e) 487 genes or δ's



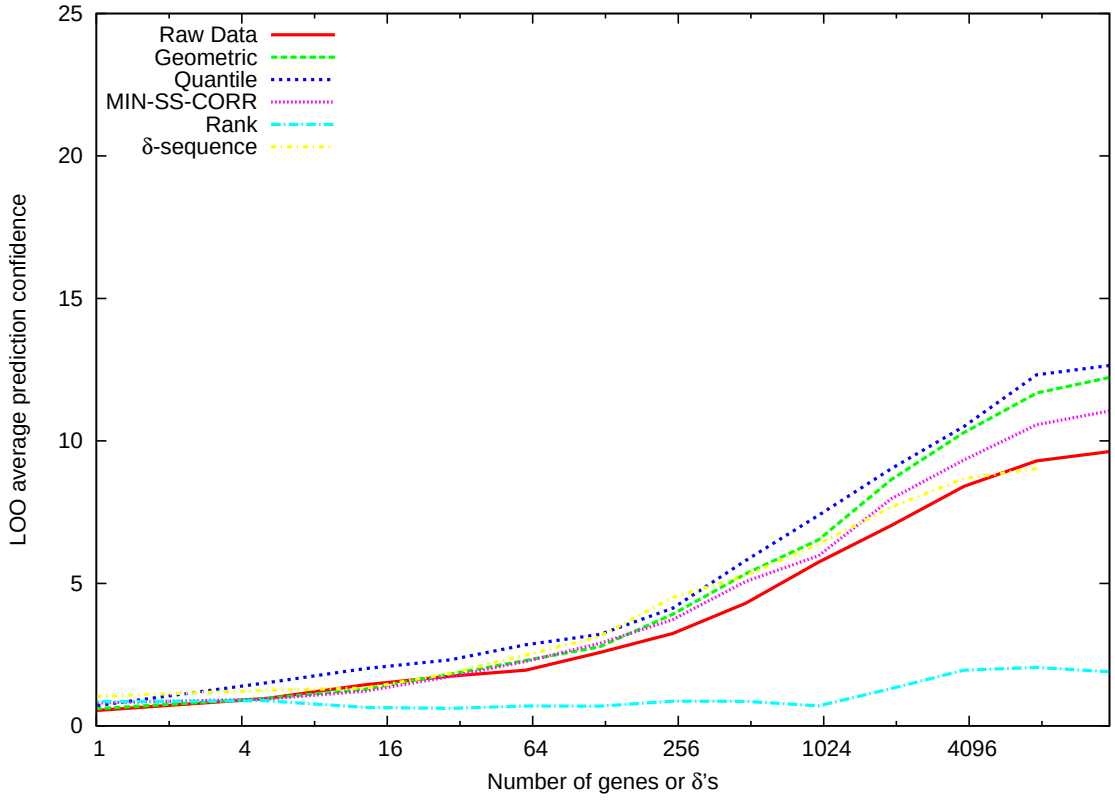
(f) 487 genes or δ's

Figure 3.19: continued. Note that since all the methods have 100% prediction accuracy above, their plots overlap.



(a)

Figure 3.20: (SVM-RFE CLASSIFIER, HOYING DATA: GENE SELECTION USES LOO) Leave-one-out prediction accuracy as a function of the number of genes or δ's used for classification. Note that the maximum number of δ's available for this dataset is 7800. All the normalization methods help achieve 100% prediction accuracy for at least one set of included genes with the exception of rank normalization and δ-sequences.



(b)

Figure 3.20: continued. Leave-one-out average prediction confidence as a function of the number of genes or δ's used for classification. Note that the maximum number of δ's available for this dataset is 7800. The fact that information from the LOO held out sample is not used for gene selection influences the classifier using rank normalized data the most causing it to degrade in performance. Quantile normalized data helps its classifier perform consistently better in terms of both prediction accuracy and average prediction confidence regardless of the number of included genes.

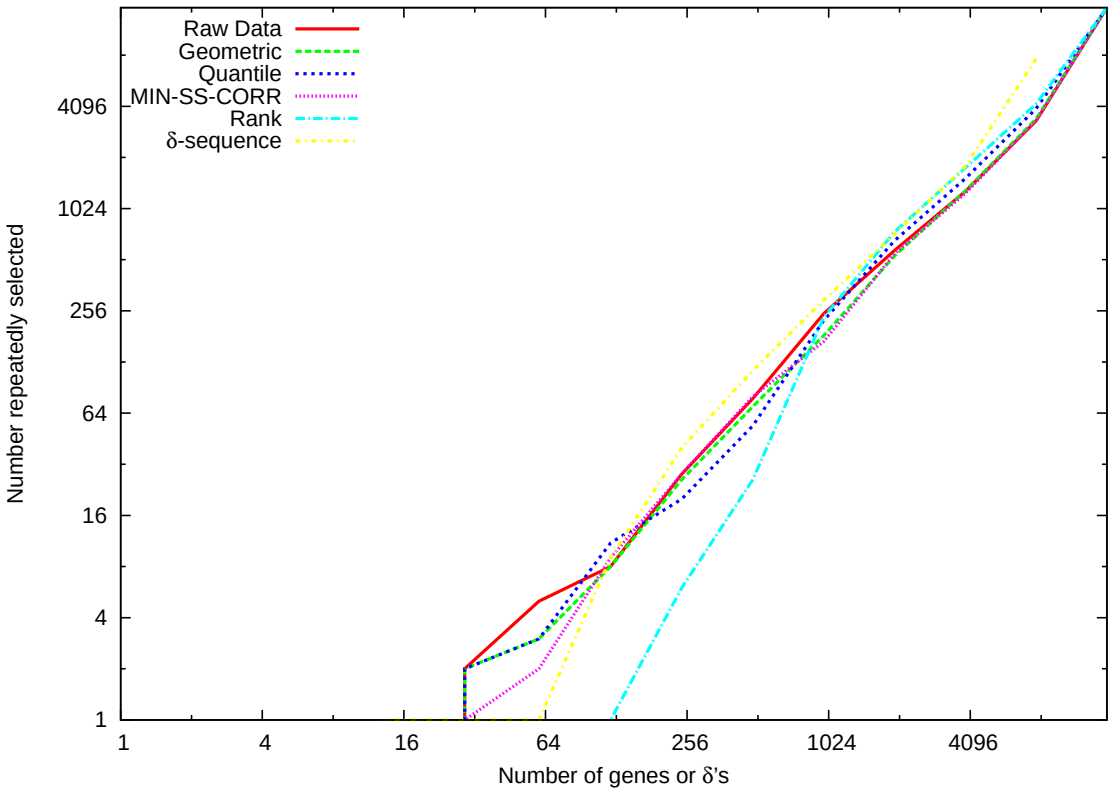
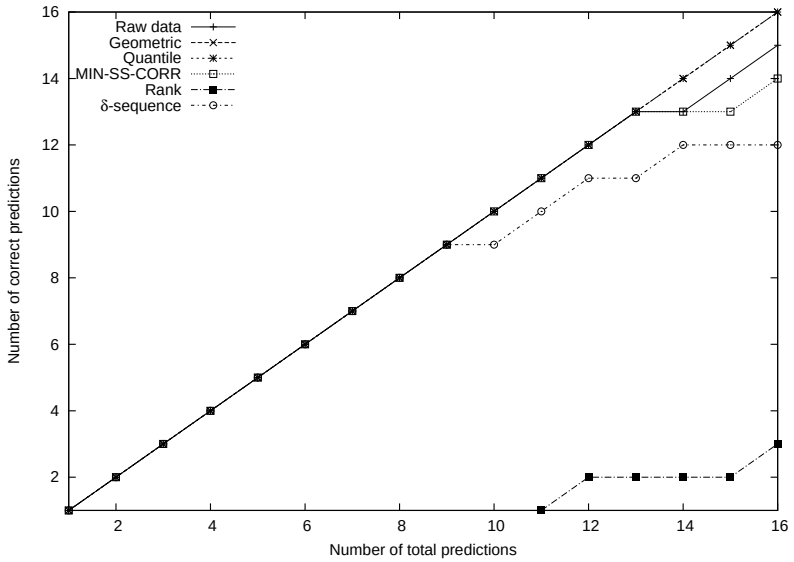
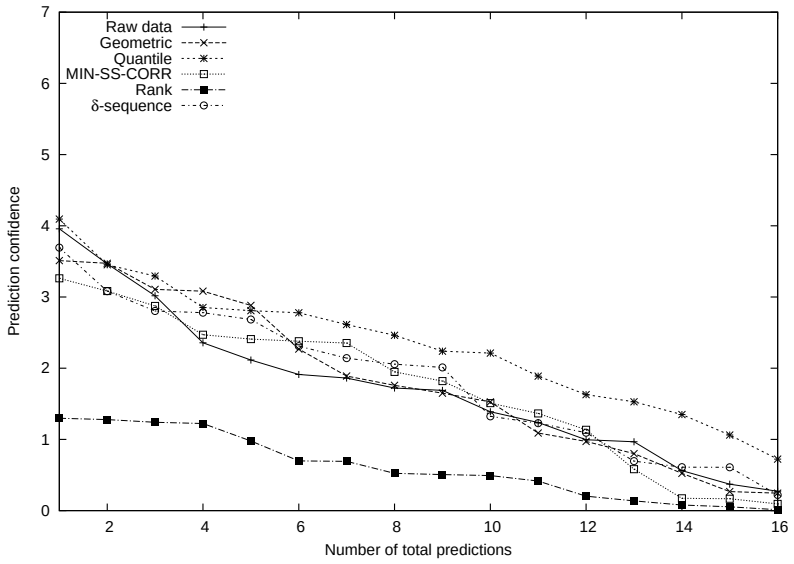


Figure 3.21: (SVM-RFE CLASSIFIER, HOYING DATA: GENE SELECTION USES LOO) Number of genes or δ's that are repeatedly selected across all the divisions of the available data into training and test sets of leave-one-out analysis. Note that the maximum number of δ's available for this dataset is 7800. The independent axis is the total number of genes used by the classifier. The higher the number on the dependent axis the lower the variation in the genes that are selected as the most useful for classification across different LOO training sets. Both raw and normalized data seem to exhibit similar variability in terms of genes that are repeatedly selected across LOO divisions. Rank normalized data seems to exhibit a larger variability in gene selection especially when the number of included genes is small. Note that as the number of genes or δ's reach their maximum then all of them are repeatedly selected due to which the curve for the δ-sequence data crosses over the other curves at 7800. Similarly for the other datasets.

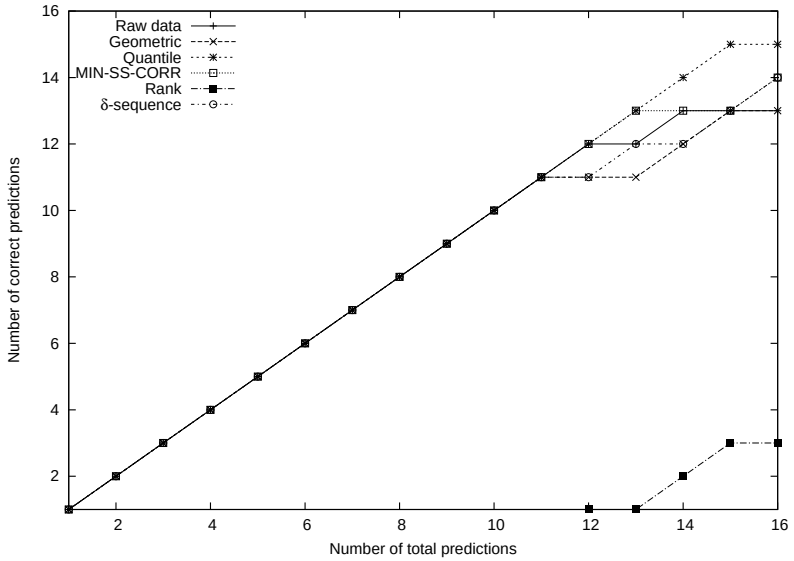


(a) 29 genes or δ's

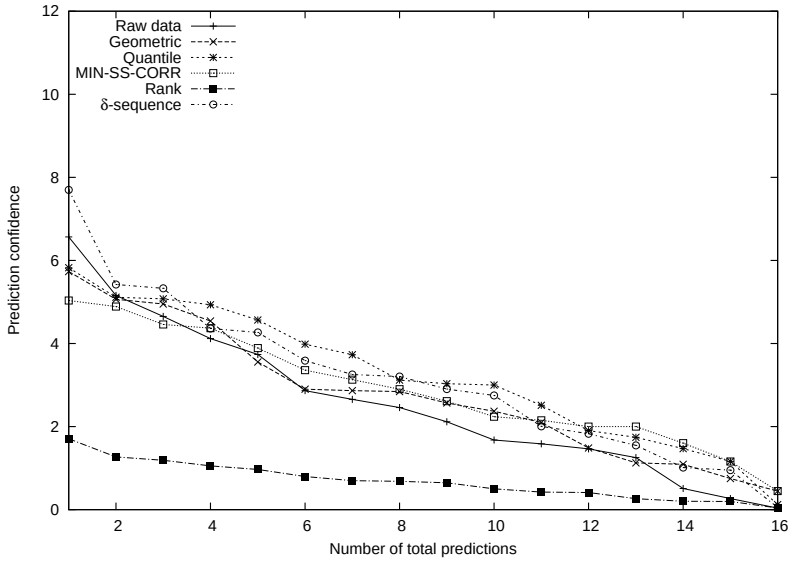


(b) 29 genes or δ's

Figure 3.22: (SVM-RFE CLASSIFIER, HOYING DATA: GENE SELECTION USES LOO) Leave-one-out prediction behavior for a fixed number of genes or δ's : 29, 121 and 487. The cumulative number of correct predictions as a function of the ordered LOO sample indices is plotted on the top panel and the respective prediction confidences are plotted on the bottom panel. The test samples are ordered in the decreasing order of their prediction confidences given by the corresponding classifiers. For the gene subsets considered the prediction accuracies of both raw data and the normalized data seem comparable with the exception of rank normalization. Similarly the prediction confidences of the classifiers using normalized data are at least as good or better than the raw data over LOO test samples except for rank normalization. Quantile normalization seems to help the most.

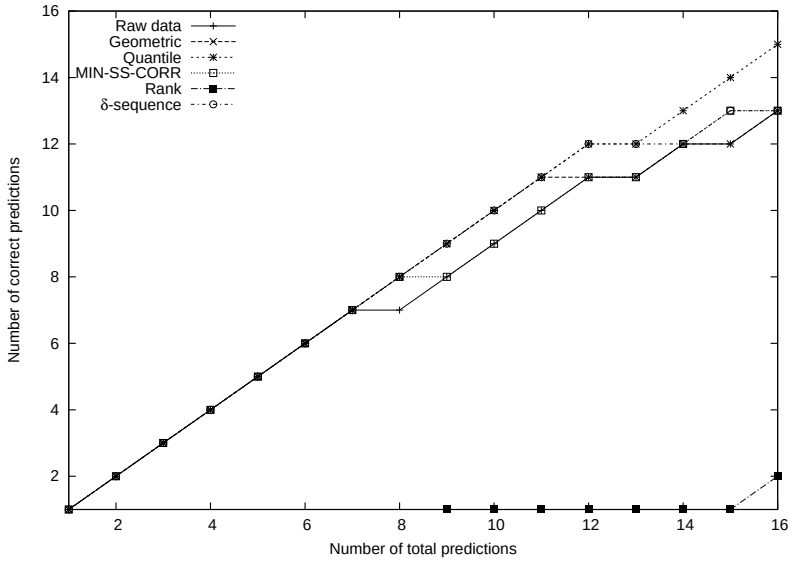


(c) 121 genes or δ's

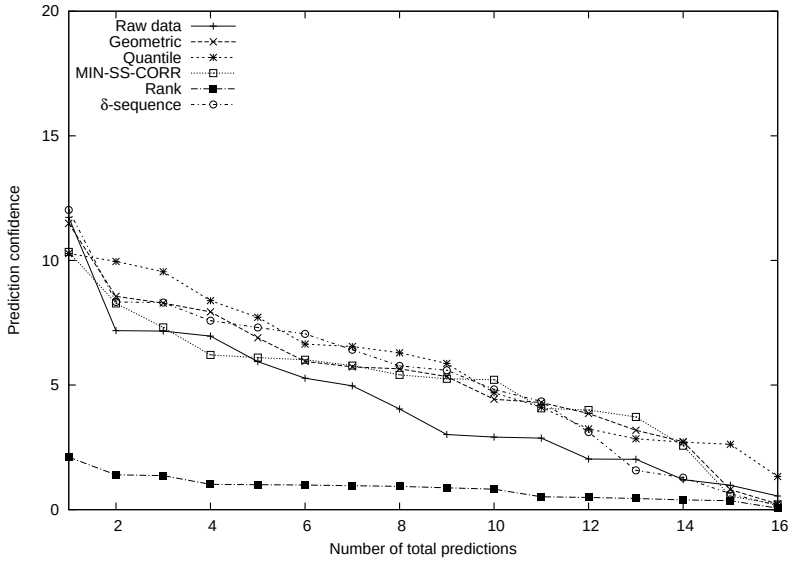


(d) 121 genes or δ's

Figure 3.22: continued.



(e) 487 genes or δ's



(f) 487 genes or δ's

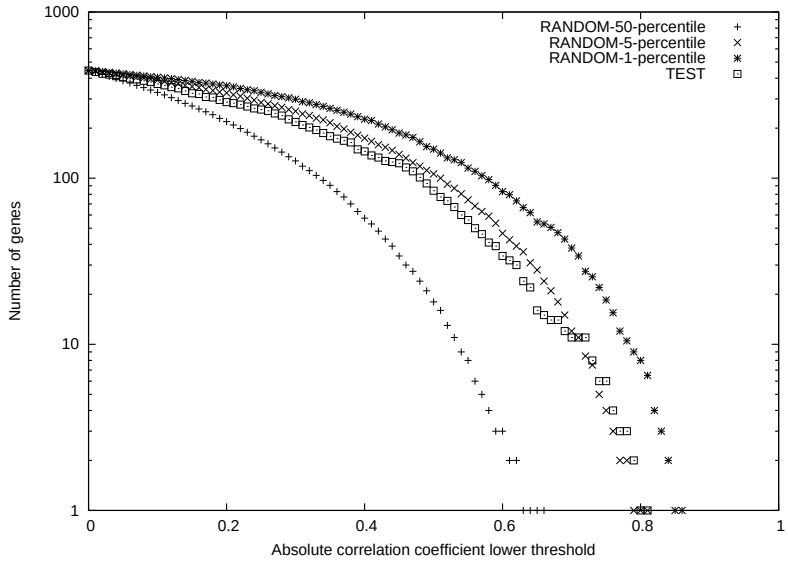
Figure 3.22: continued.

3.1.5 Experiments on angiogenesis dataset using genes selected by CARMA

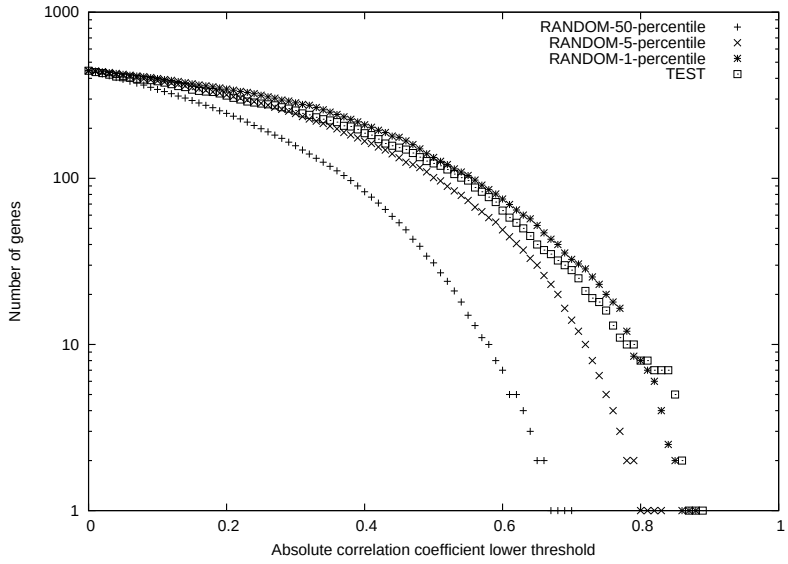
We conducted an additional set of experiments on the angiogenesis dataset using a reduced number of genes. These 444 genes were determined to be the most differentially expressed using a recently developed normalization and gene selection software called CARMA (Greer et al., 2006) . The software assumes an experimental design in which tissue samples have repeated measurements with different array and dye combinations. CARMA corrects the linlog transformed measurements for spatial location and intensity variations using lowess regression. Then a gene-by-gene ANOVA is performed to adjust for the gene specific variations introduced by the experimental factors namely the Array (A), Dye (D) and Variety (V) effects. The variety effect in this case corresponds to time. The main reason for doing a separate ANOVA for each gene as against a possible global ANOVA using all the gene measurements is the resulting substantial reduction in computer memory requirements. A gene is declared differentially expressed if the time effect's (V) F-statistic in the ANOVA analysis has p-value less than or equal to 0.05 with adjustment for a false discovery rate of 10%. Starting with the raw data and the data normalized by the techniques described in section 2.1 but using the measurements for only the differentially expressed genes, we conduct the same set of experiments as before. The results are reported below.

Hypothesis testing results

The hypothesis testing results of Fig. 3.23 suggest that the raw data consisting of only the subset of CARMA selected genes has better evidence for the phenotypic classes than that with all the genes included (compare with Fig. 3.12a). This is improved further by the quantile, geometric and MIN-SS-CORR normalization techniques with quantile providing the largest improvement. Note that the 500 genes used for MIN-SS-CORR optimization did not necessarily include the ones selected by CARMA. Rank normalization and δ-sequences do not seem to be particularly helpful in terms of improving the classifiability of the data.

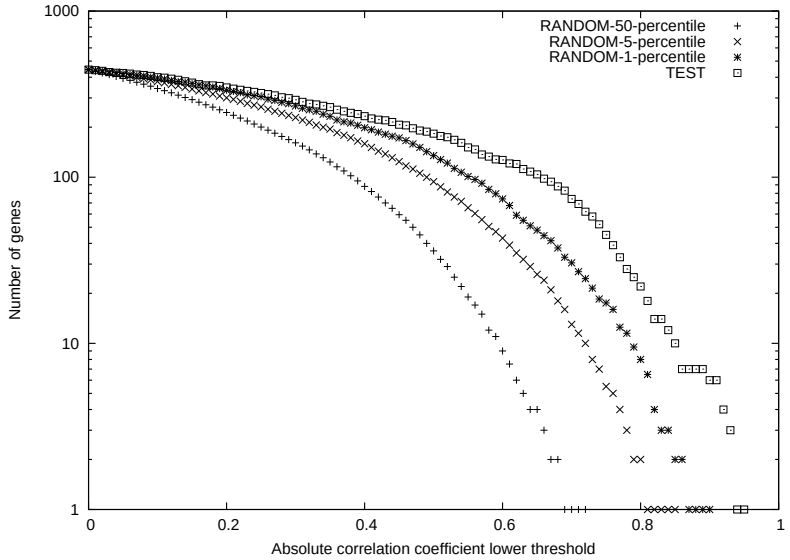


(a) Raw data

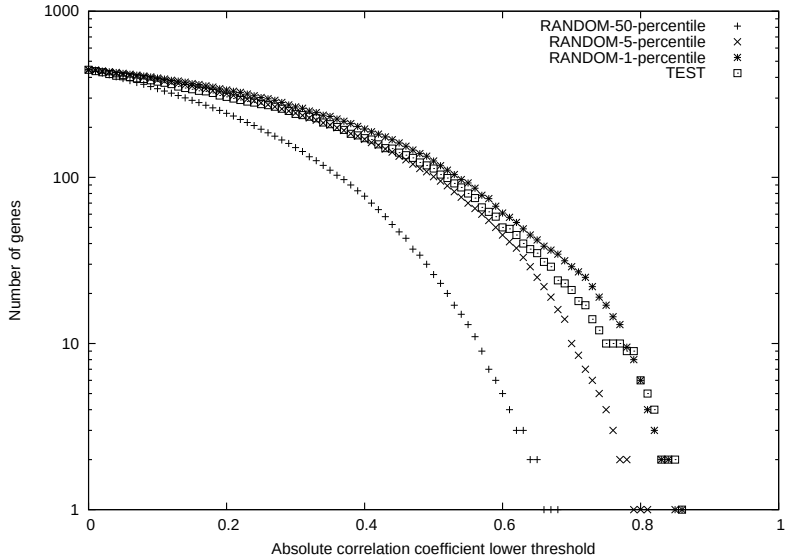


(b) Geometric normalization

Figure 3.23: Class label hypothesis testing of the Hoying angiogenesis dataset before and after normalization using the 444 genes selected by CARMA. TEST refers to the number of genes that have their absolute correlation coefficient value above a certain level with the correlation coefficient being computed with the phenotypic pattern of class labeling. See Section 3.0.6. RANDOM refers to statistics of the distribution obtained by considering 1000 different random binary labeling patterns on the tissue samples (null distribution). Plotting the median (50 percentile), 5-percentile and 1-percentile points of the null distribution at different minimum levels of the absolute correlation coefficient results in the three curves RANDOM-50-percentile, RANDOM-5-percentile and RANDOM-1-percentile respectively. The higher the TEST curve above the RANDOM curves the better is the statistical significance suggested by the data for the phenotypic classes.

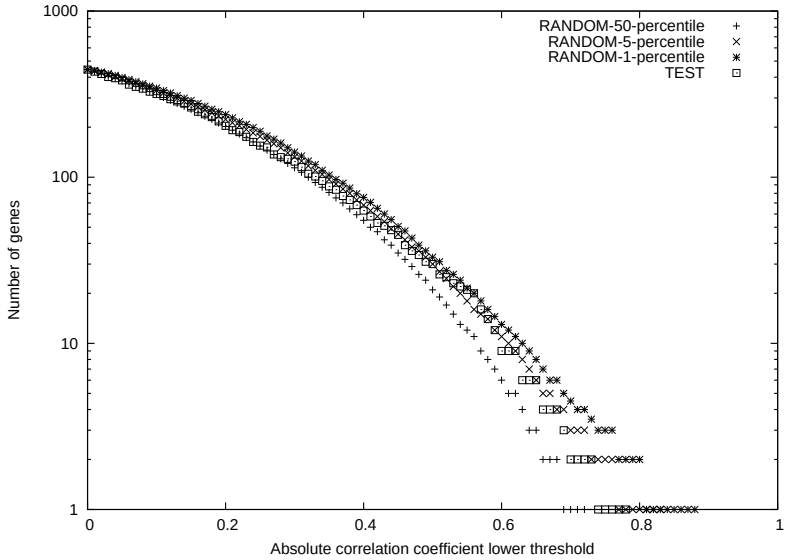


(c) Quantile normalization

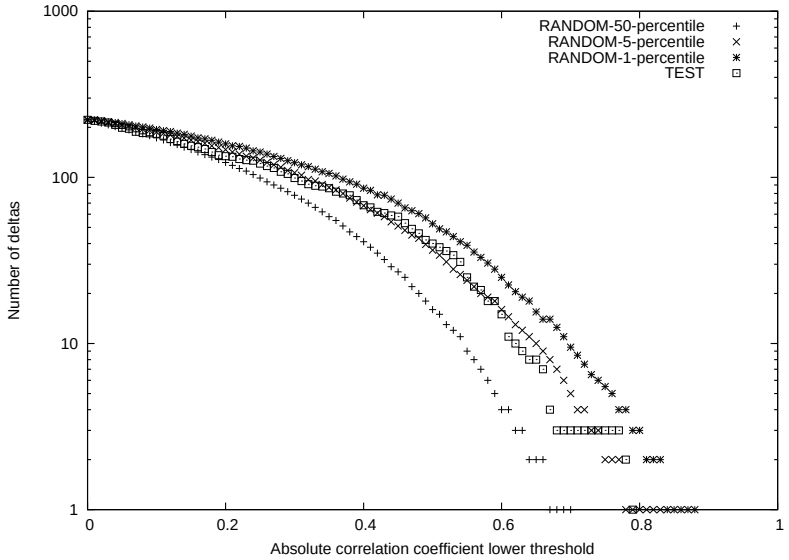


(d) MIN-SS-CORR normalization

Figure 3.23: continued.



(e) Rank normalization

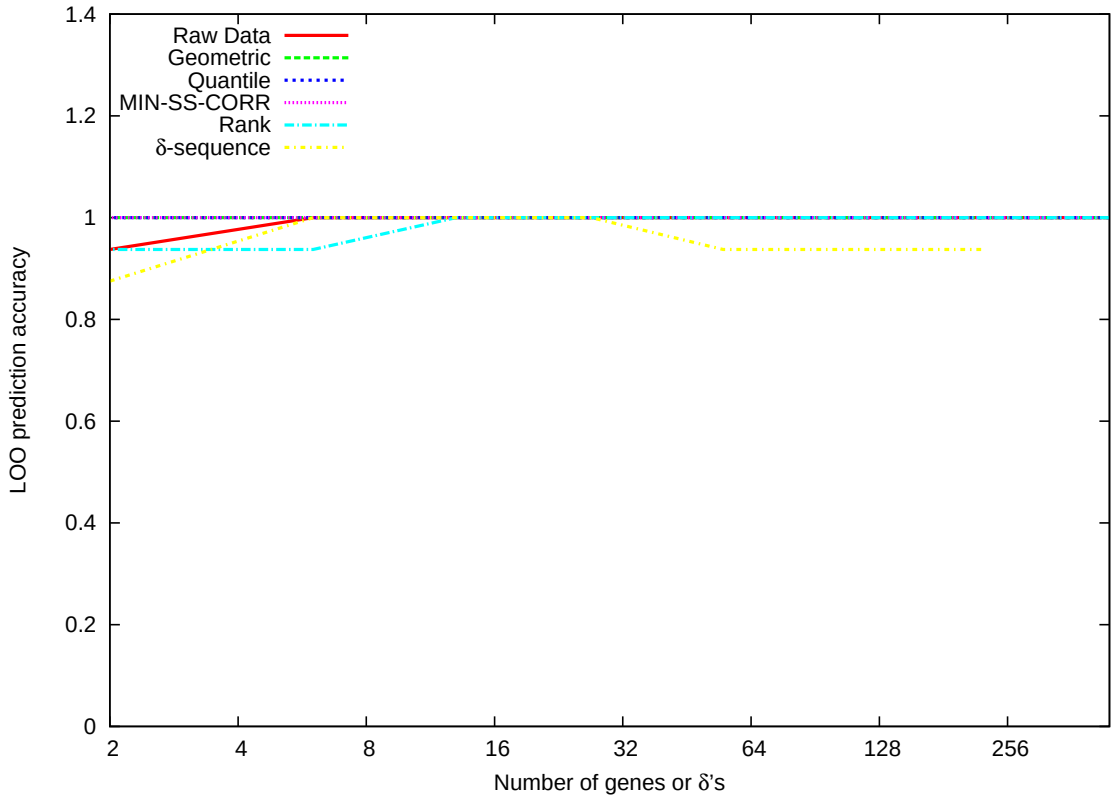


(f) δ-sequences

Figure 3.23: continued.

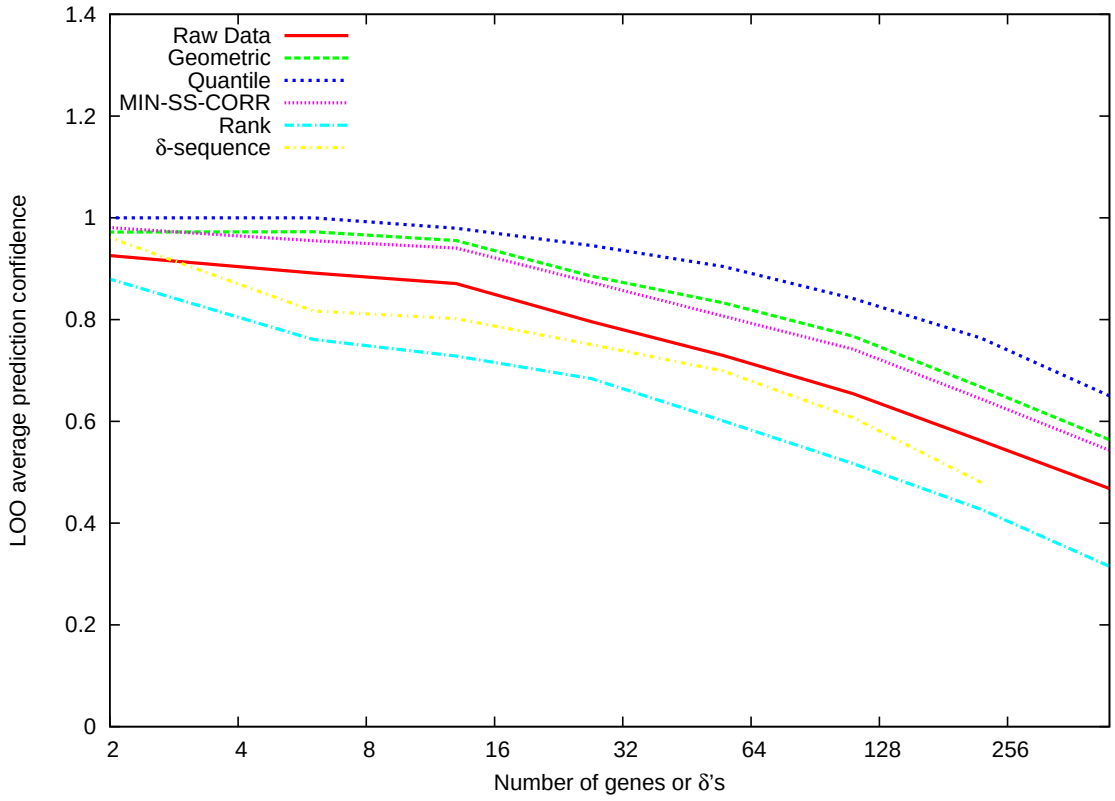
Classification results using Golub and SVM-RFE classifiers

The prediction accuracy and average confidence of the Golub classifiers using the raw and normalized data with gene selection using all the available data and only the training data of the LOO sets are shown in Figs. 3.24-3.25 and Figs. 3.26-3.28 respectively. The corresponding results for the SVM-RFE classifiers are shown in Figs. 3.29-3.30 and Figs. 3.31-3.33 respectively. The relative performances induced by the raw and the normalized data on the corresponding classifiers are similar in both the cases. Quantile, geometric and MIN-SS-CORR normalization methods lead to their respective classifiers showing improvements over the raw data. Classifiers using the rank normalized and δ-sequence data perform slightly worse than the raw data. This trend resonates with the one in the hypothesis testing framework.



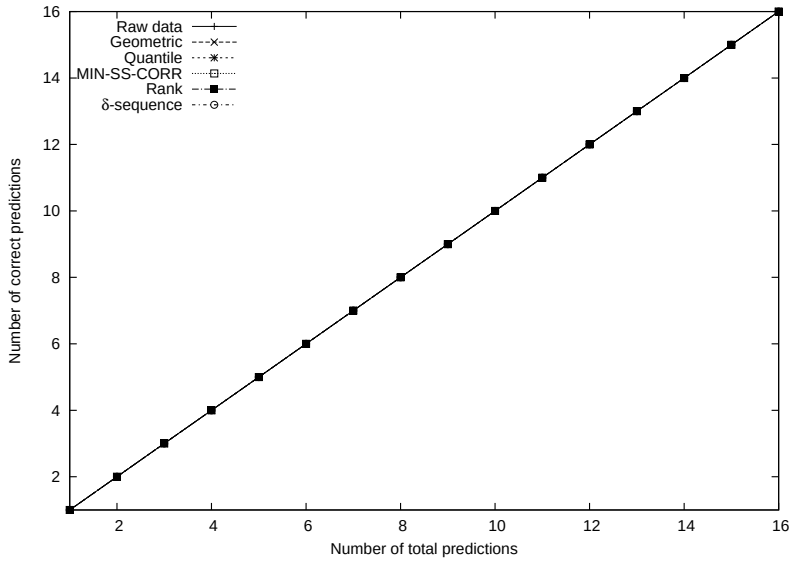
(a)

Figure 3.24: (GOLUB CLASSIFIER, HOYING DATA, CARMA GENES) Leave-one-out prediction accuracy as a function of the number of genes or δ's used for classification. Note that the maximum number of δ's available for this dataset is 222.

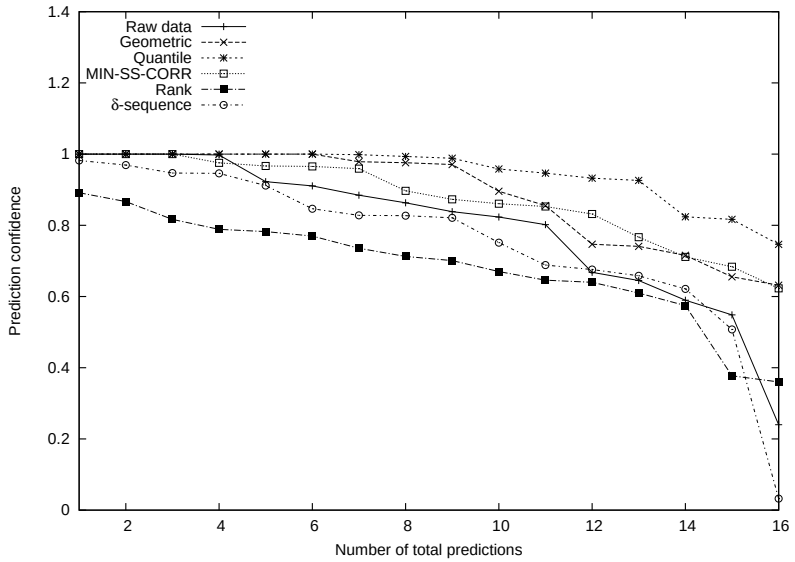


(b)

Figure 3.24: continued. Leave-one-out average prediction confidence as a function of the number of genes or δ's used for classification. Note that the maximum number of δ's available for this dataset is 222.

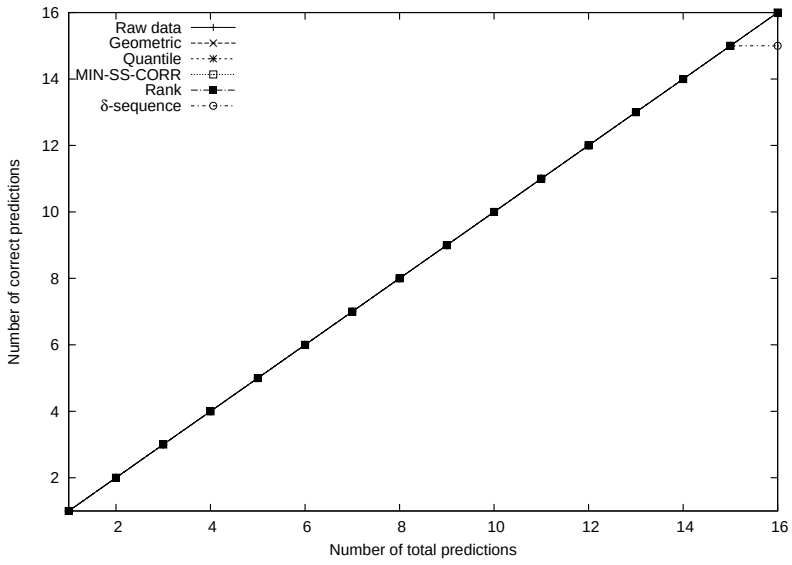


(a) 27 genes or δ's

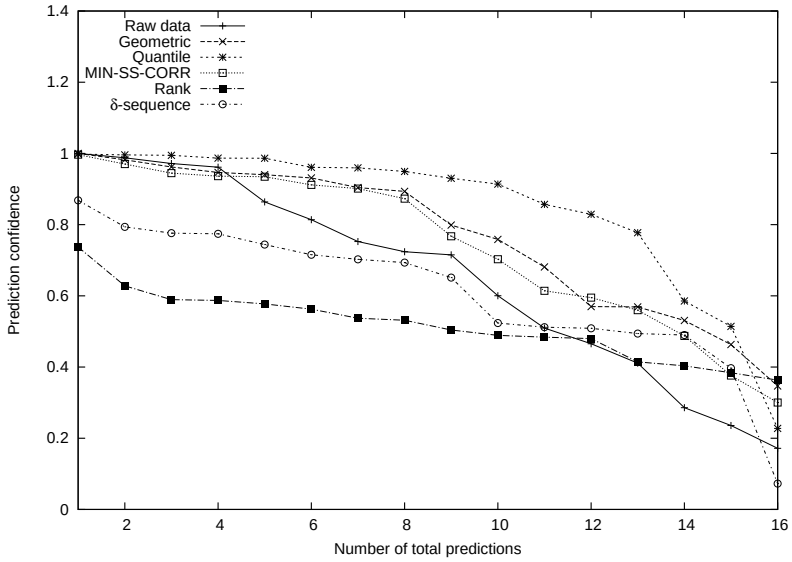


(b) 27 genes or δ's

Figure 3.25: (GOLUB CLASSIFIER, HOYING DATA, CARMA GENES) Leave-one-out prediction behavior for a fixed number of genes or δ's : 27, 111 and 222. The cumulative number of correct predictions as a function of the ordered LOO sample indices is plotted on the top panel and the respective prediction confidences are plotted on the bottom panel. The test samples are ordered in the decreasing order of their prediction confidences given by the corresponding classifiers. Note that since all the methods have 100% prediction accuracy above, their plots overlap.

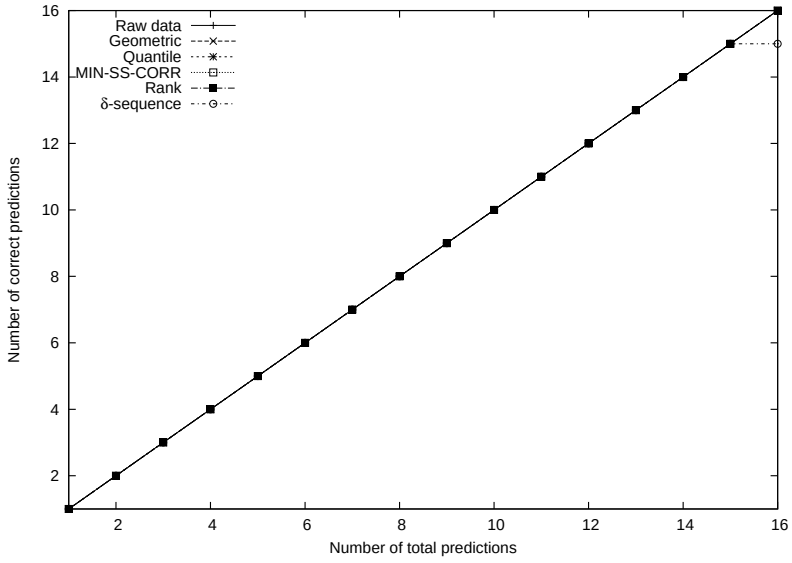


(c) 111 genes or δ's

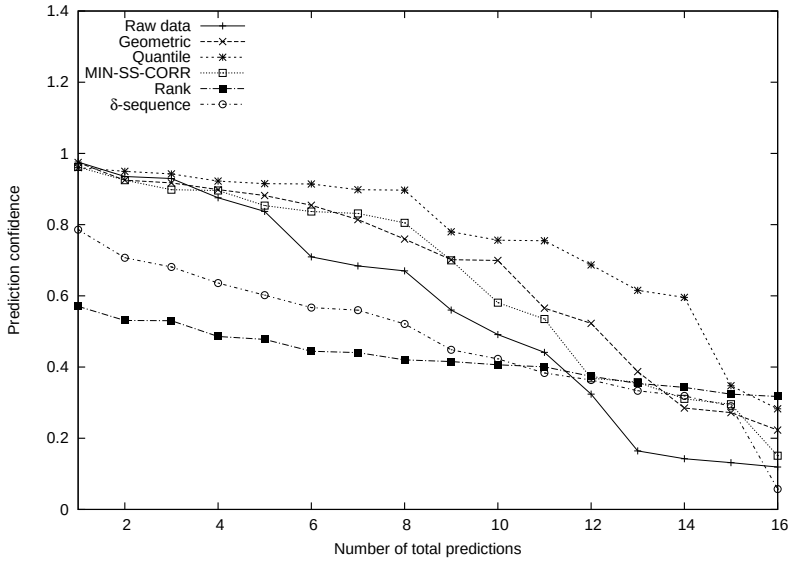


(d) 111 genes or δ's

Figure 3.25: continued.Note that since all the methods have 100% prediction accuracy above, their plots overlap.

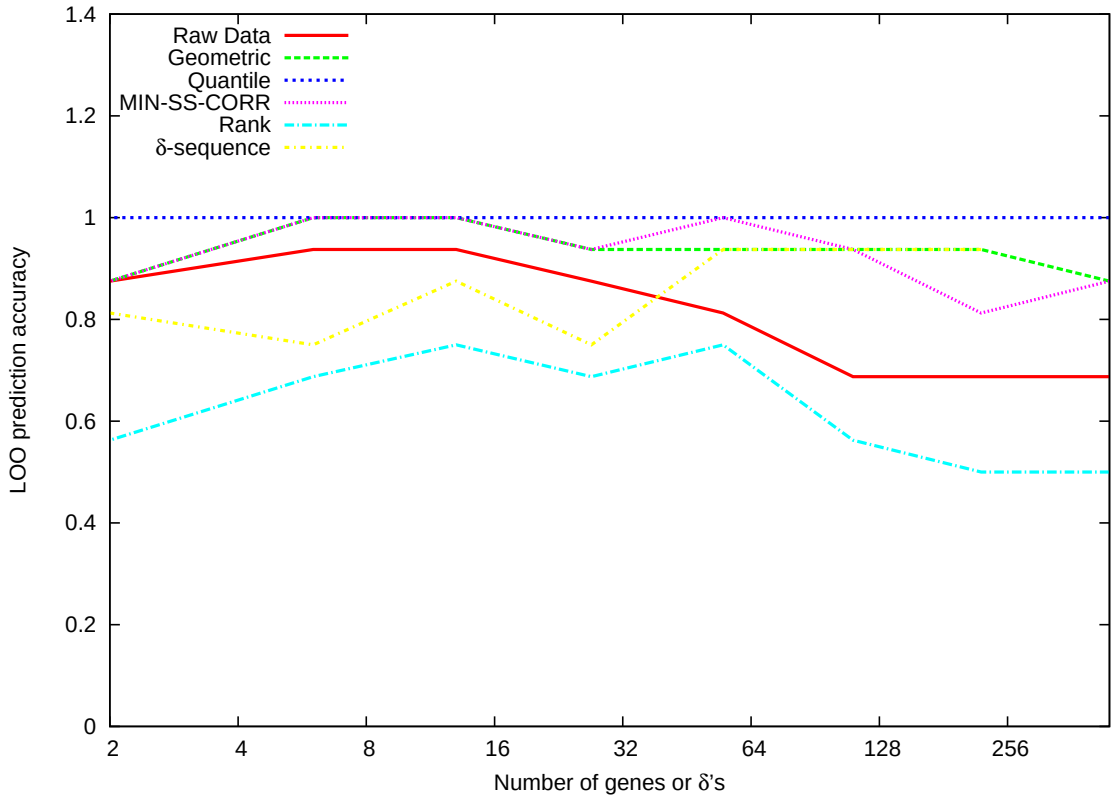


(e) 222 genes or δ's



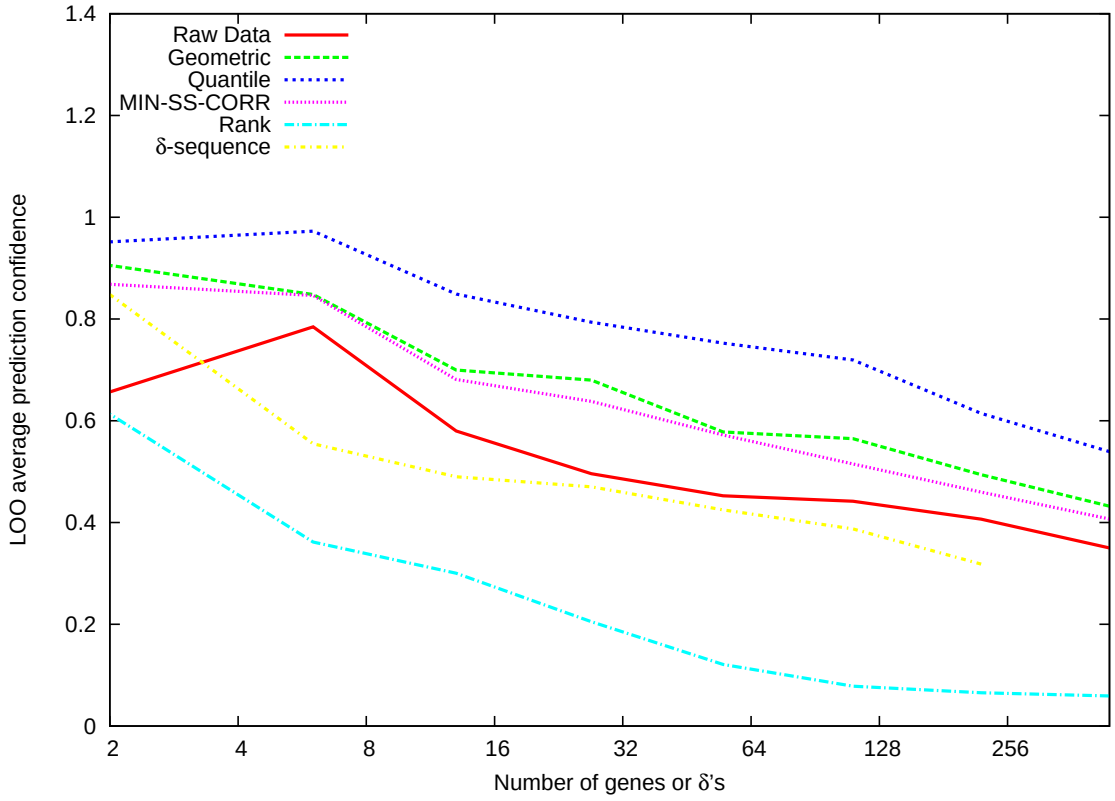
(f) 222 genes or δ's

Figure 3.25: continued. Note that since all the methods have almost 100% prediction accuracy above, their plots overlap for most part of the range.



(a)

Figure 3.26: (GOLUB CLASSIFIER, HOYING DATA, CARMA GENES: GENE SELECTION USES LOO) Leave-one-out prediction accuracy as a function of the number of genes or δ's used for classification. Note that the maximum number of δ's available for this dataset is 222.



(b)

Figure 3.26: continued. Leave-one-out average prediction confidence as a function of the number of genes or δ's used for classification. Note that the maximum number of δ's available for this dataset is 222. The performance of the classifiers using rank normalized data degrades the most both in terms of prediction accuracy and confidence when the LOO sample is ignored for gene selection as part of classifier training.

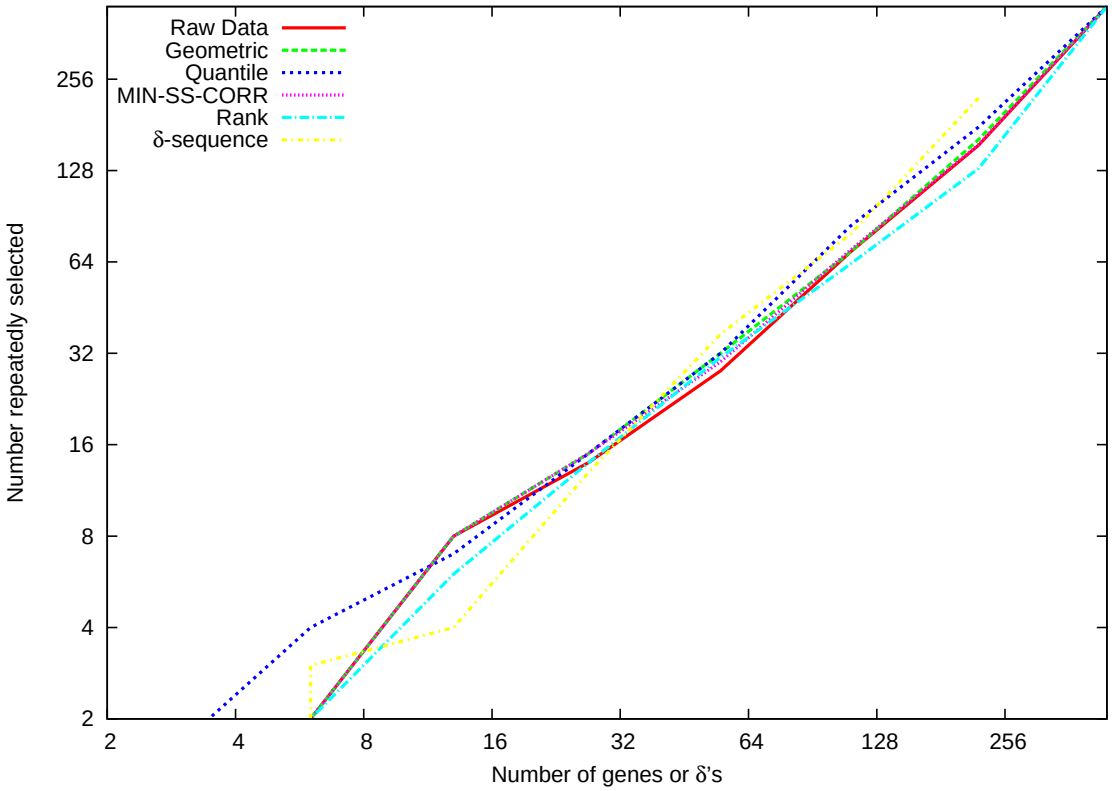
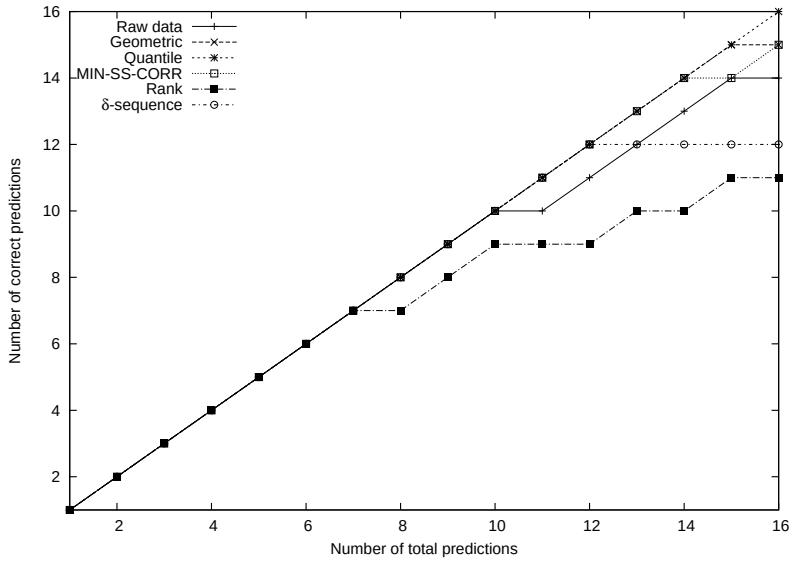
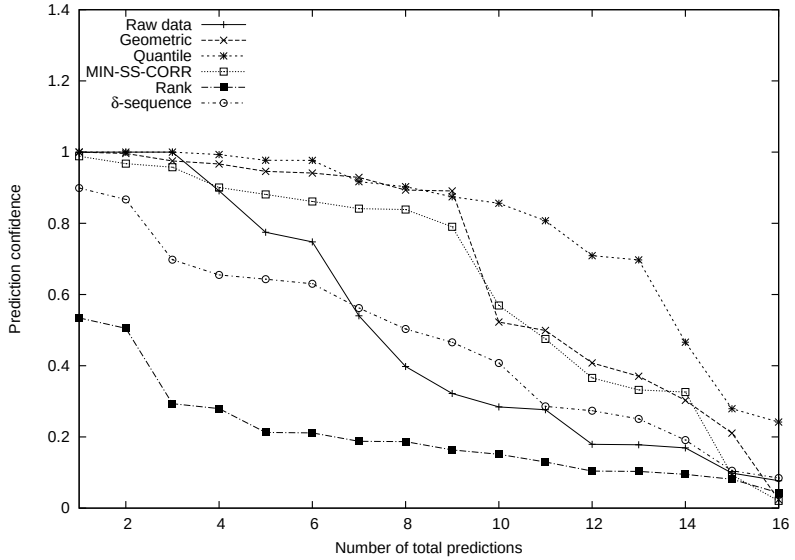


Figure 3.27: (GOLUB CLASSIFIER, HOYING DATA, CARMA GENES: GENE SELECTION USES LOO) Number of genes or δ's that are repeatedly selected across all the divisions of the available data into training and test sets of leave-one-out analysis. Note that the maximum number of δ's available for this dataset is 222. The independent axis is the total number of genes used by the classifier. The higher the number on the dependent axis the lower the variation in the genes that are selected as the most useful for classification across different LOO training sets. Note that as the number of genes or δ's reach their maximum then all of them are repeatedly selected due to which the curve for the δ-sequence data crosses over the other curves at 222. Similarly for the other datasets.

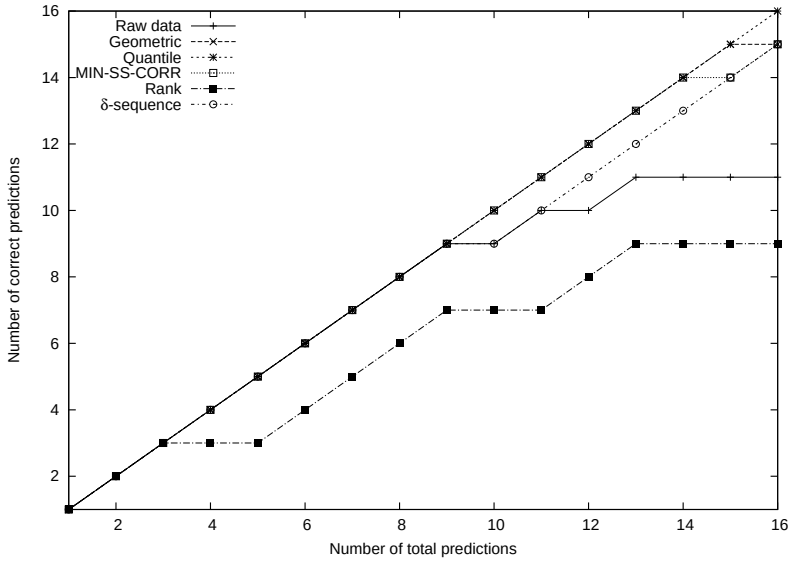


(a) 27 genes or δ's

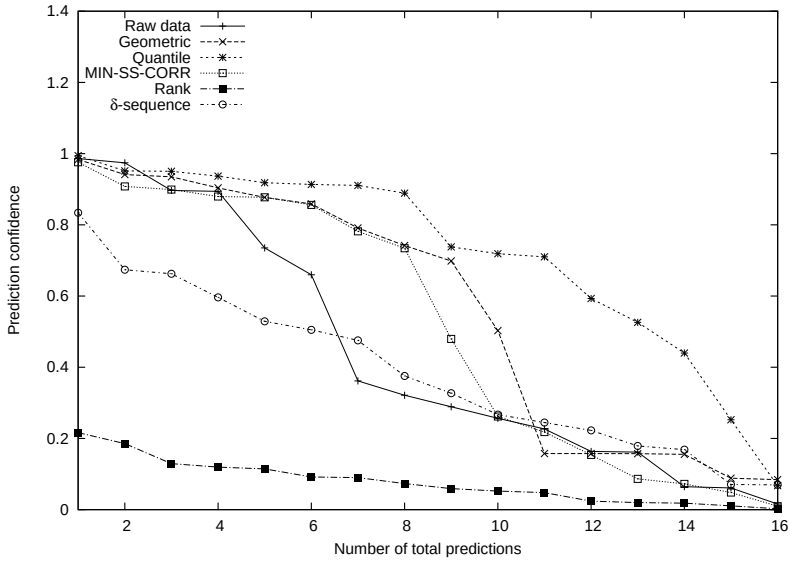


(b) 27 genes or δ's

Figure 3.28: (GOLUB CLASSIFIER, HOYING DATA, CARMA GENES: GENE SELECTION USES LOO) Leave-one-out prediction behavior for a fixed number of genes or δ's : 27, 111 and 222.The cumulative number of correct predictions as a function of the ordered LOO sample indices is plotted on the top panel and the respective prediction confidences are plotted on the bottom panel. The test samples are ordered in the decreasing order of their prediction confidences given by the corresponding classifiers.

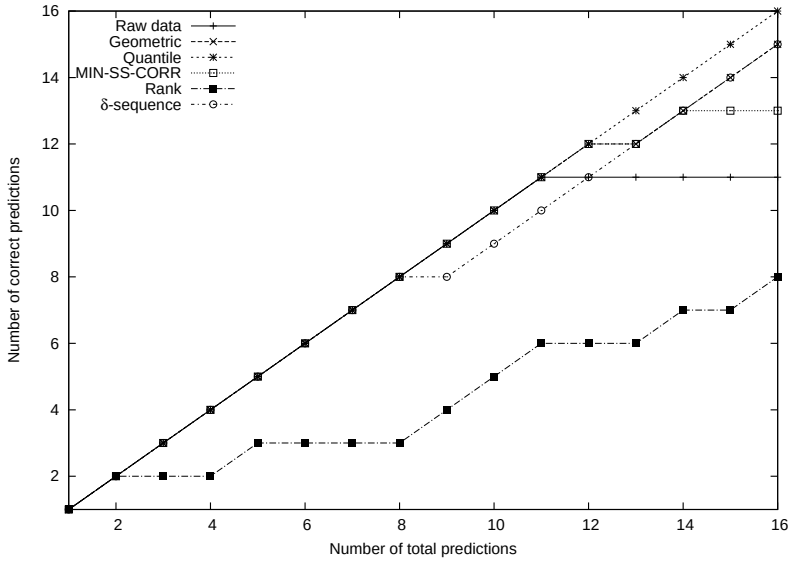


(c) 111 genes or δ's

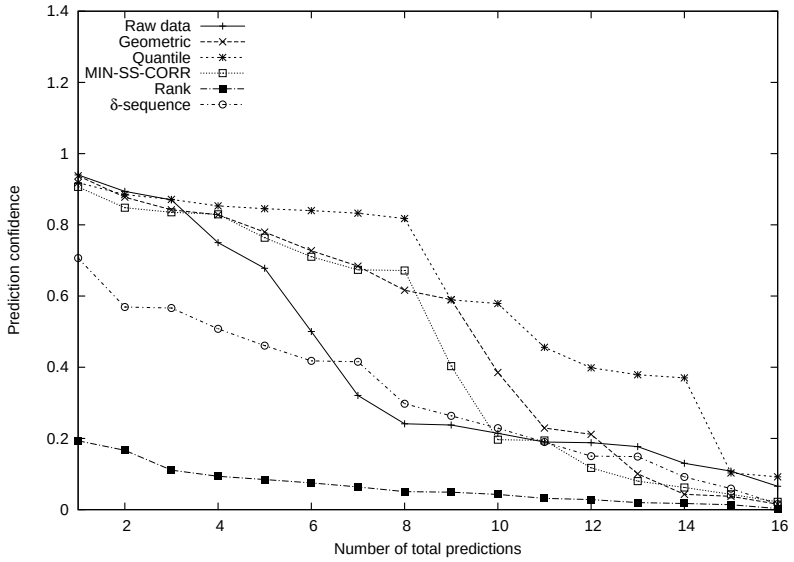


(d) 111 genes or δ's

Figure 3.28: continued.

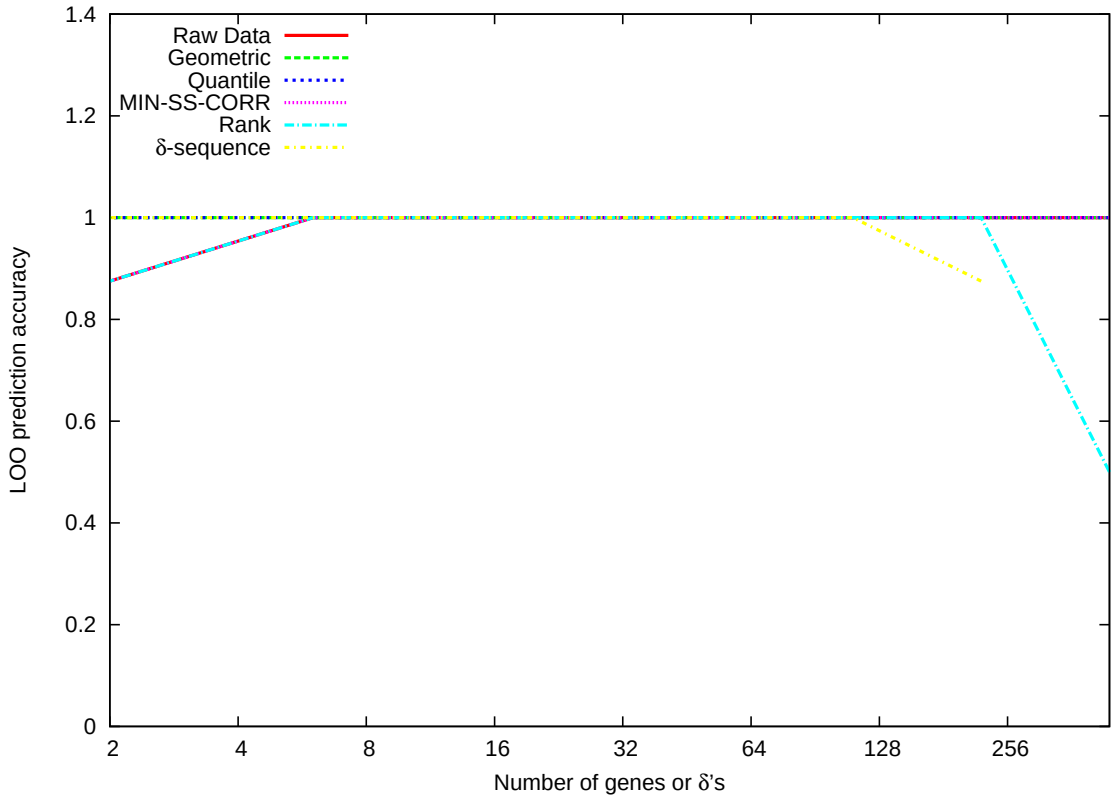


(e) 222 genes or δ's



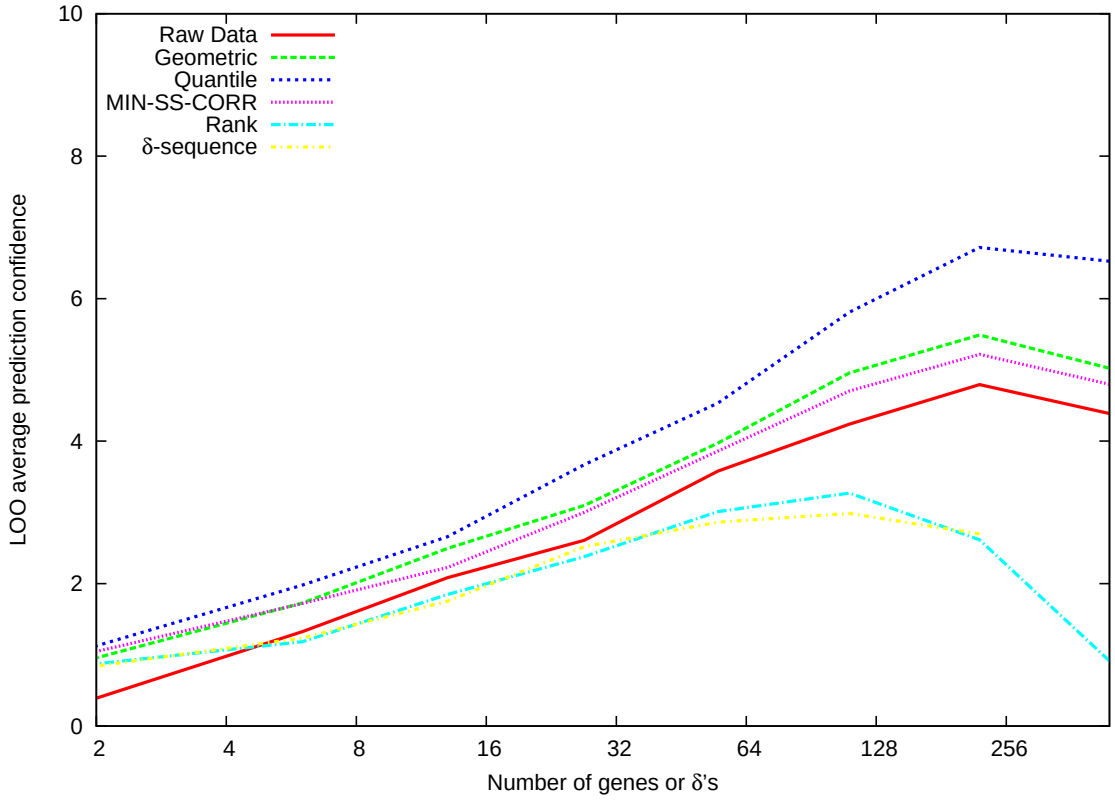
(f) 222 genes or δ's

Figure 3.28: continued.



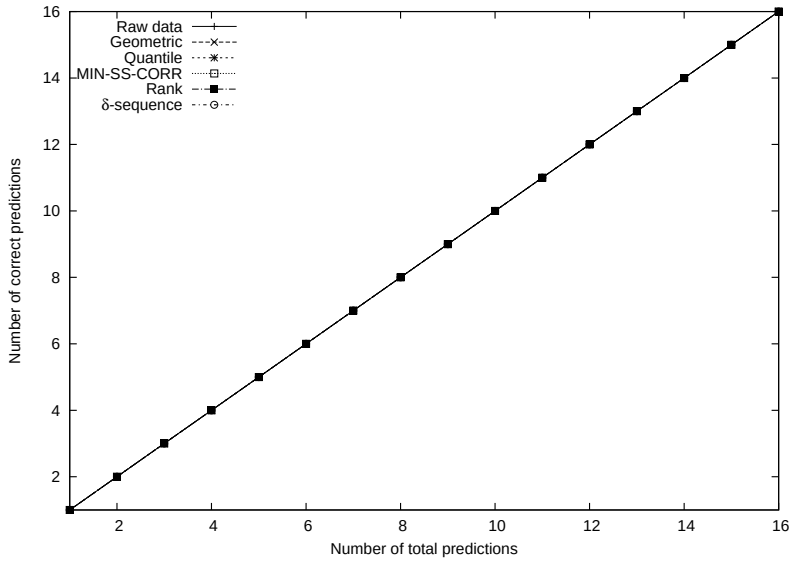
(a)

Figure 3.29: (SVM-RFE CLASSIFIER, HOYING DATA, CARMA GENES) Leave-one-out prediction accuracy as a function of the number of genes or δ's used for classification. Note that the maximum number of δ's available for this dataset is 222.

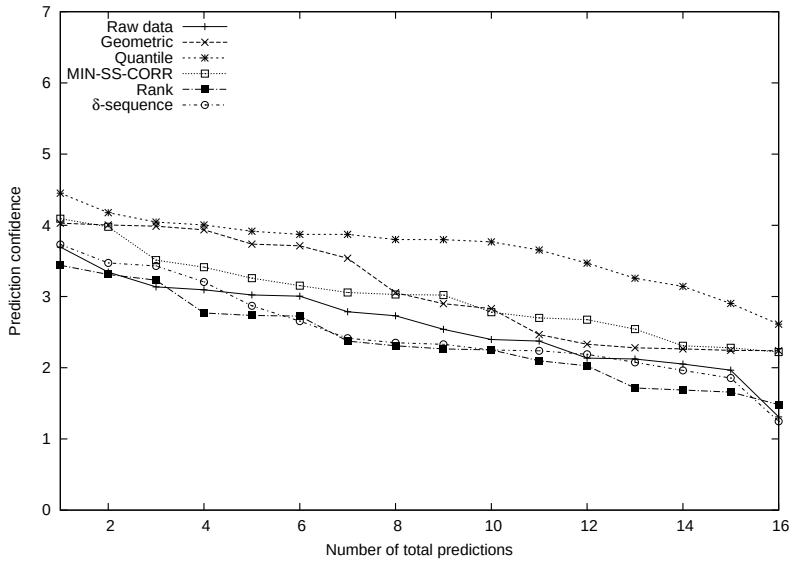


(b)

Figure 3.29: continued. Leave-one-out average prediction confidence as a function of the number of genes or δ's used for classification. Note that the maximum number of δ's available for this dataset is 222. For most part of the range of the included number of genes, the data normalized using quantile, geometric and MIN-SS-CORR methods results in classifiers performing better than those using the raw data. The converse is true with the data normalized using rank and δ-sequence methods.

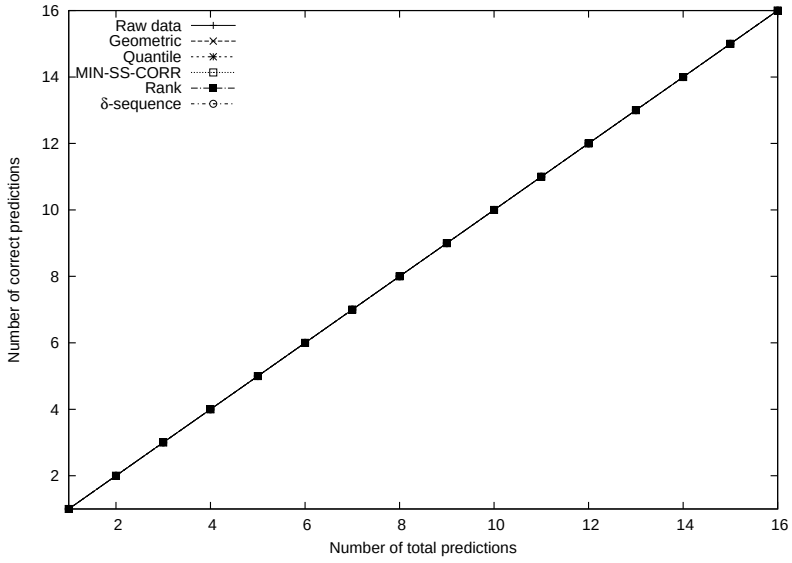


(a) 27 genes or δ's

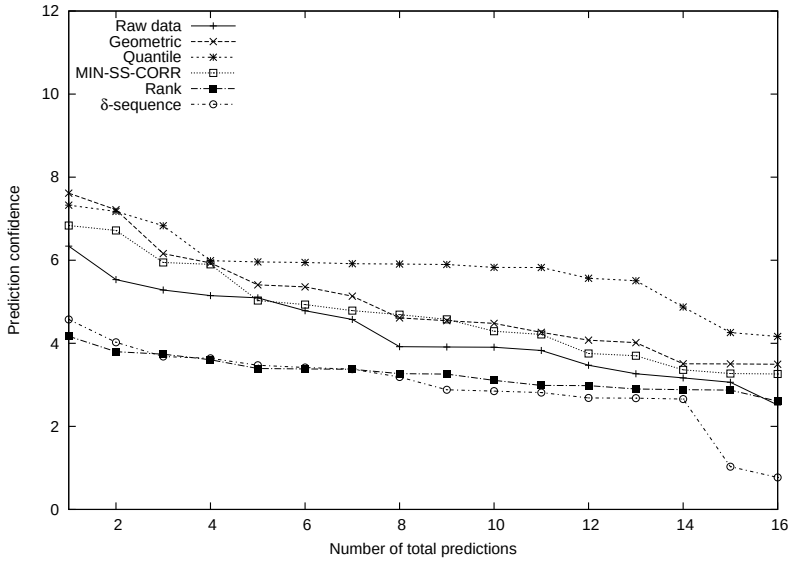


(b) 27 genes or δ's

Figure 3.30: (SVM-RFE CLASSIFIER, HOYING DATA, CARMA GENES) Leave-one-out prediction behavior for a fixed number of genes or δ's : 27, 111 and 222. The cumulative number of correct predictions as a function of the ordered LOO sample indices is plotted on the top panel and the respective prediction confidences are plotted on the bottom panel. The test samples are ordered in the decreasing order of their prediction confidences given by the corresponding classifiers. Note that since all the methods have 100% prediction accuracy above, their plots overlap.

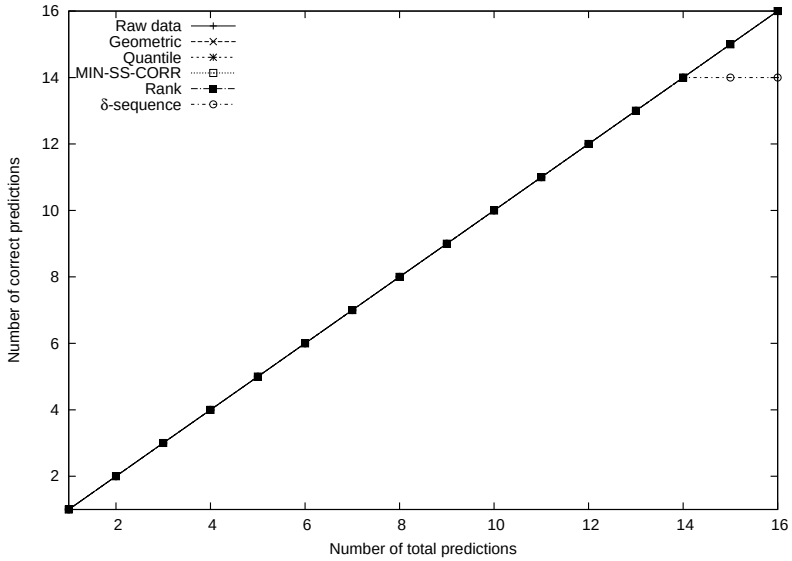


(c) 111 genes or δ's

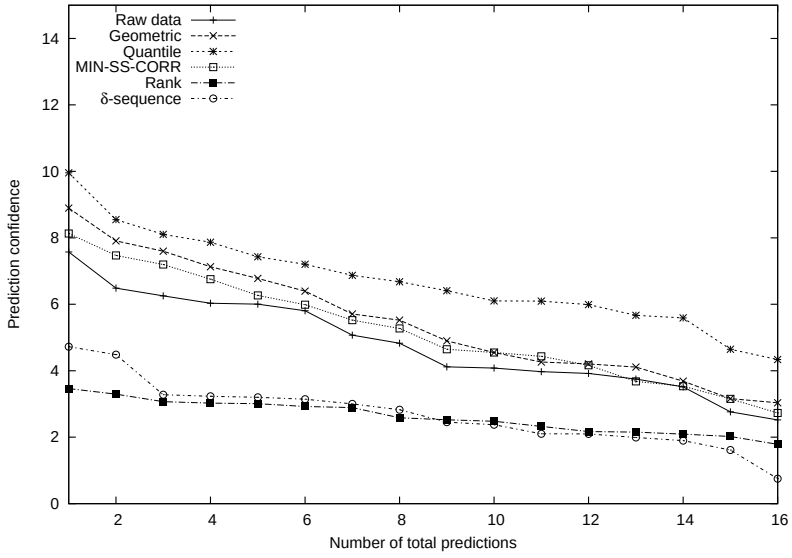


(d) 111 genes or δ's

Figure 3.30: continued. Note that since all the methods have 100% prediction accuracy above, their plots overlap.

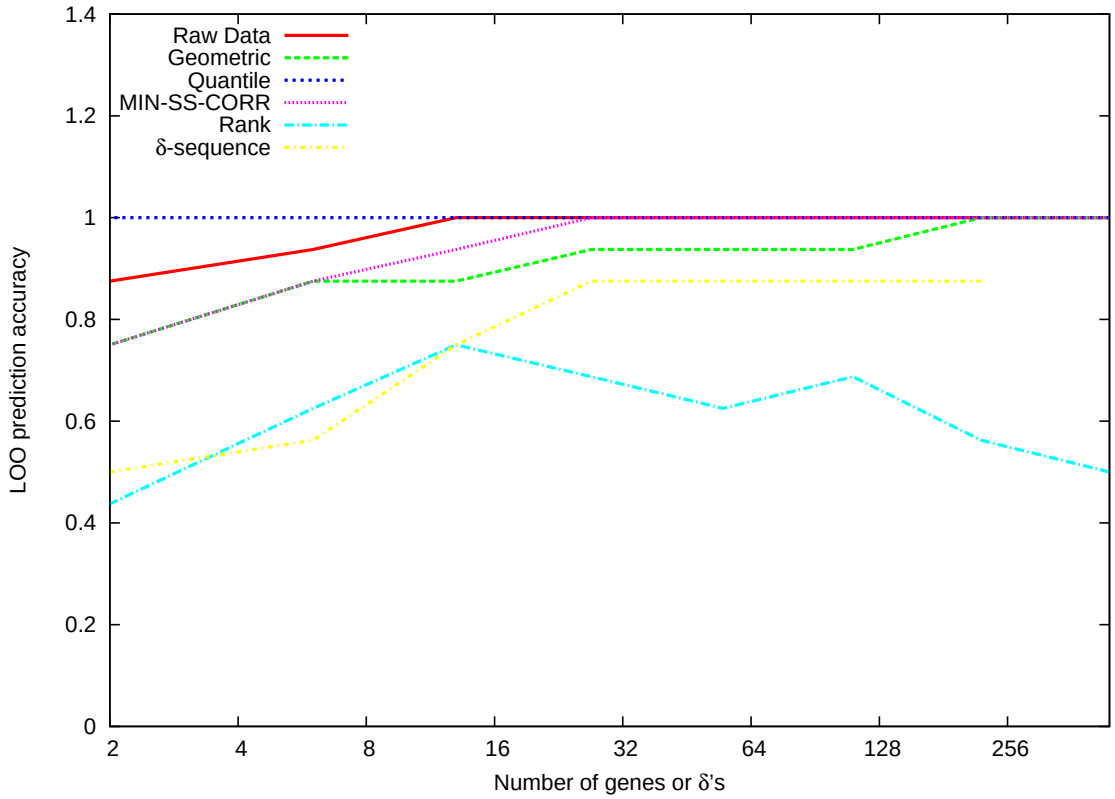


(e) 222 genes or δ's



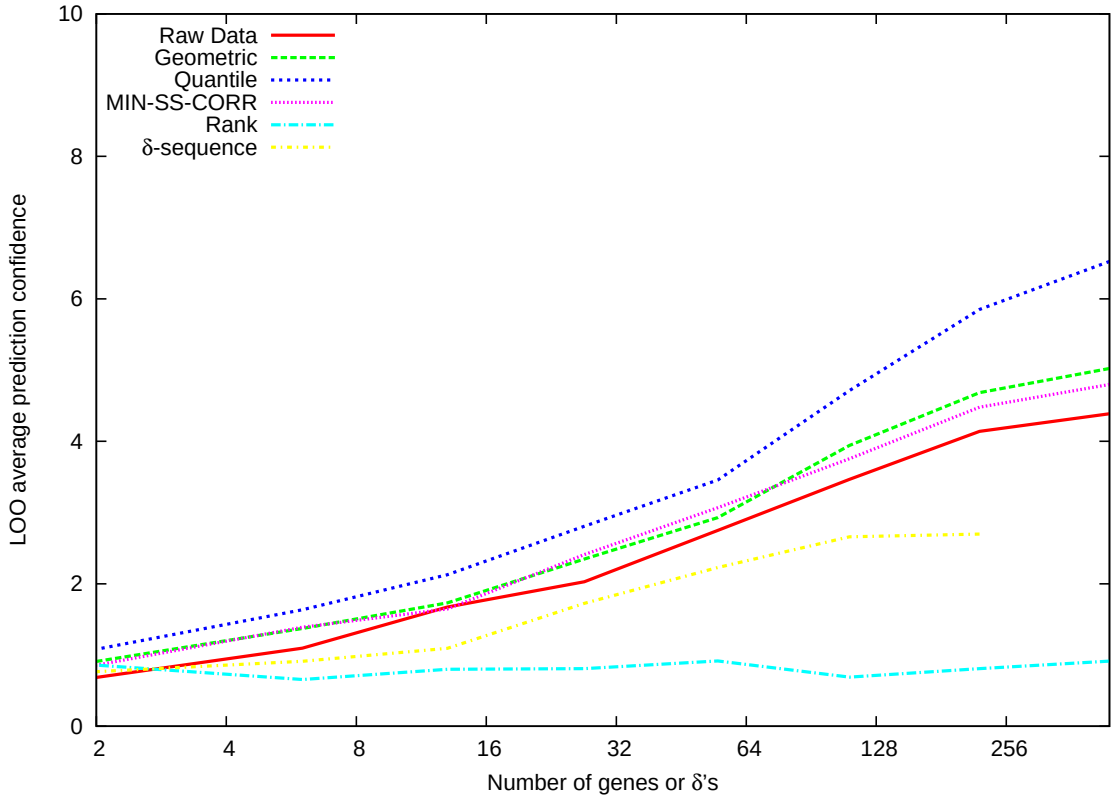
(f) 222 genes or δ's

Figure 3.30: continued.



(a)

Figure 3.31: (SVM-RFE CLASSIFIER, HOYING DATA, CARMA GENES: GENE SELECTION USES LOO) Leave-one-out prediction accuracy as a function of the number of genes or δ's used for classification. Note that the maximum number of δ's available for this dataset is 222.



(b)

Figure 3.31: continued. Leave-one-out average prediction confidence as a function of the number of genes or δ's used for classification. Note that the maximum number of δ's available for this dataset is 222. The relative performances of the classifiers using normalized data relative to raw data are the same as before where gene selection is done using all samples (Fig. 3.29b).

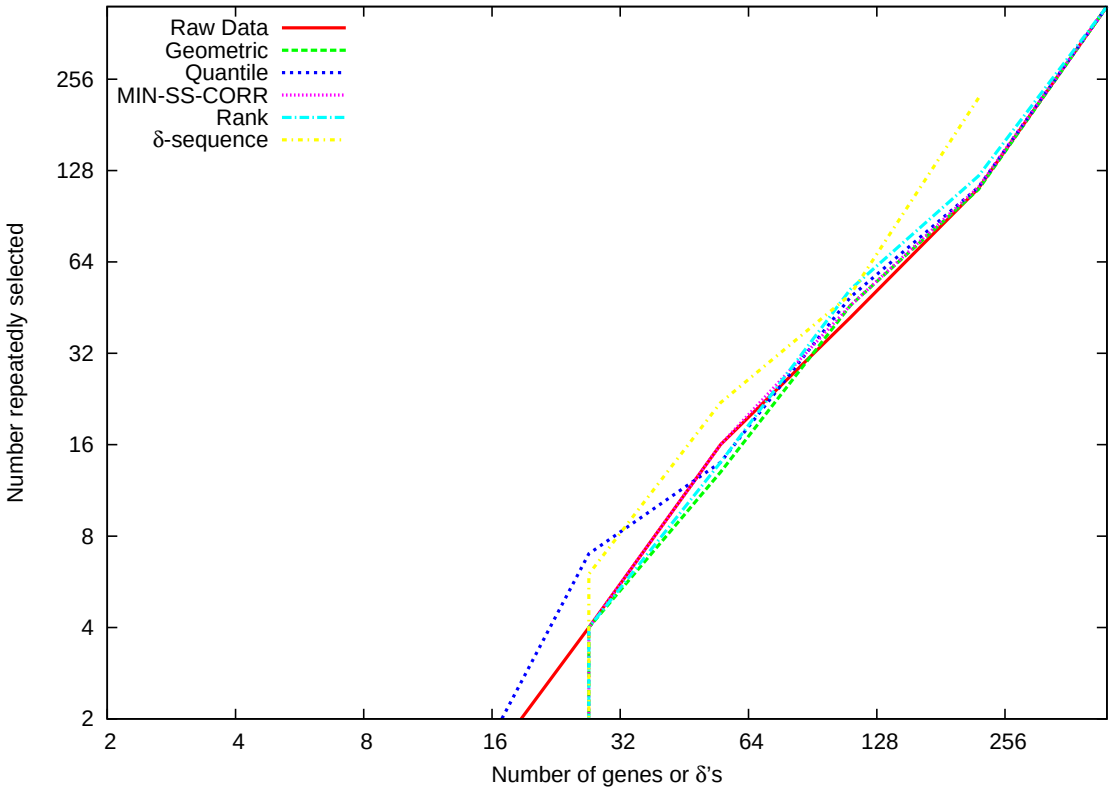
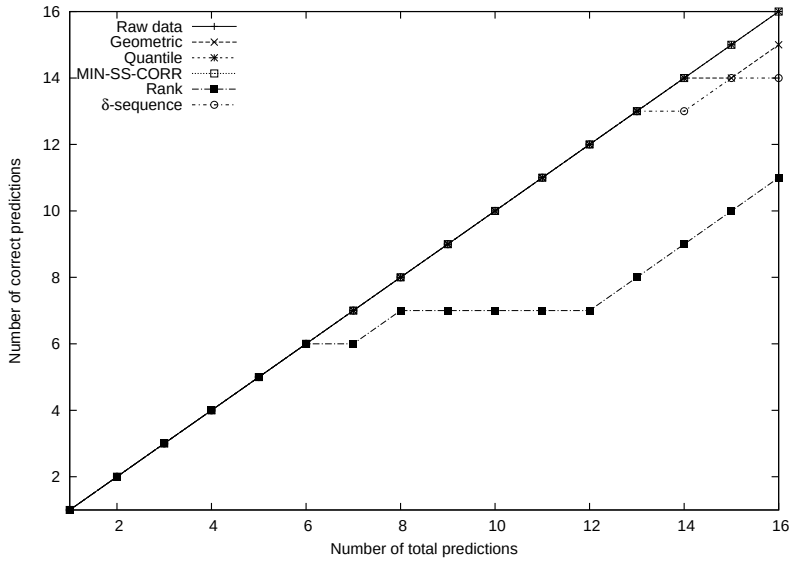
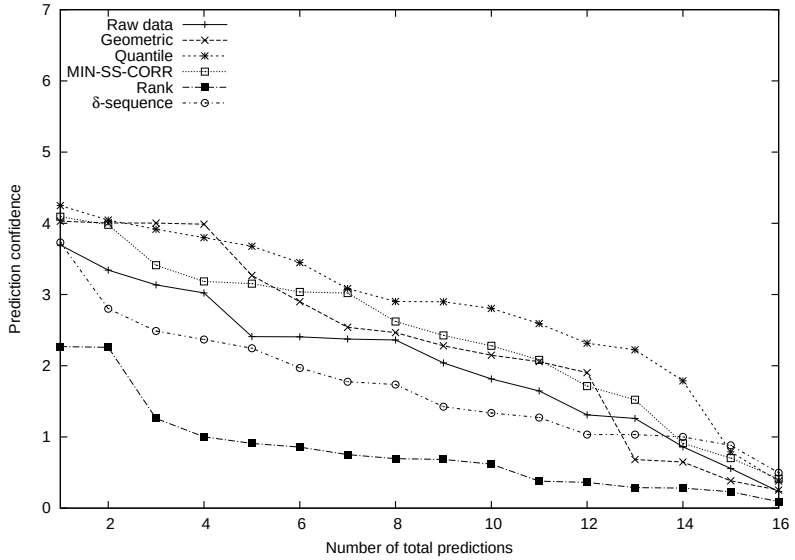


Figure 3.32: (SVM-RFE CLASSIFIER, HOYING DATA, CARMA GENES: GENE SELECTION USES LOO) Number of genes or δ's that are repeatedly selected across all the divisions of the available data into training and test sets of leave-one-out analysis. Note that the maximum number of δ's available for this dataset is 222. The independent axis is the total number of genes used by the classifier. The higher the number on the dependent axis the lower the variation in the genes that are selected as the most useful for classification across different LOO training sets. Note that as the number of genes or δ's reach their maximum then all of them are repeatedly selected due to which the curve for the δ-sequence data crosses over the other curves at 222. Similarly for the other datasets.

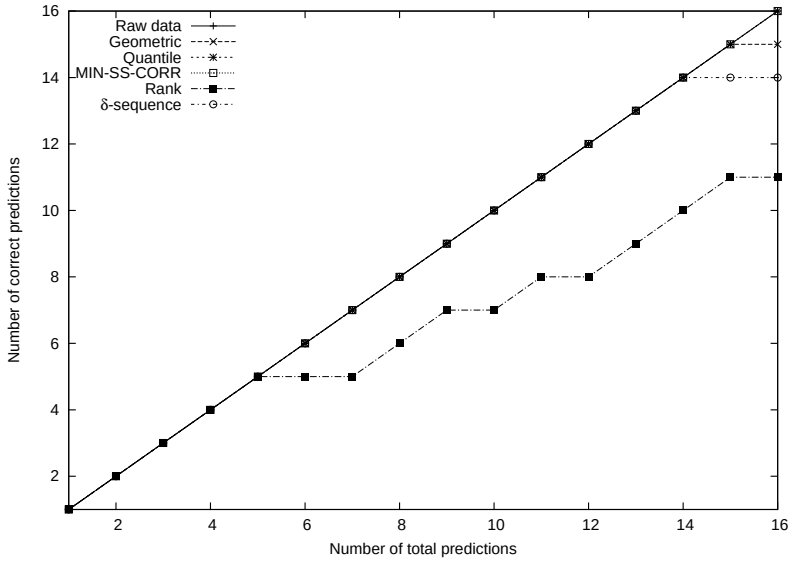


(a) 27 genes or δ's

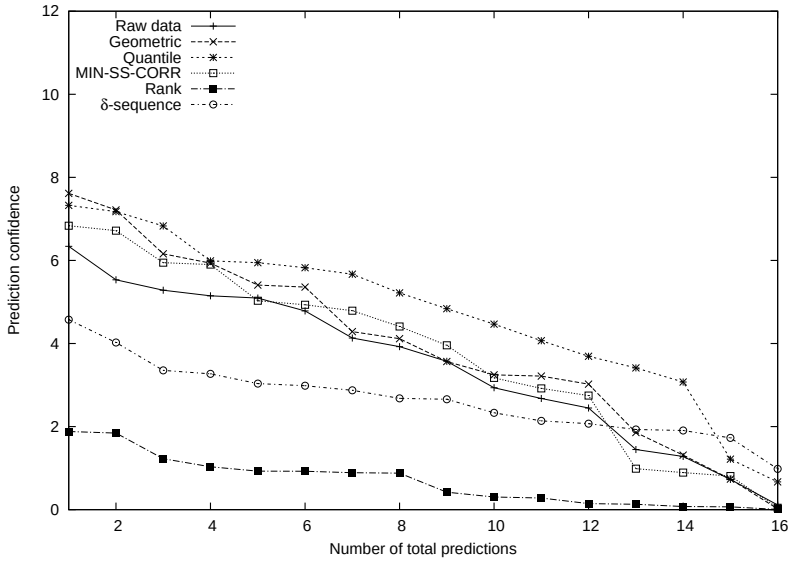


(b) 27 genes or δ's

Figure 3.33: (SVM-RFE CLASSIFIER, HOYING DATA, CARMA GENES: GENE SELECTION USES LOO) Leave-one-out prediction behavior for a fixed number of genes or δ's : 27, 111 and 222. The cumulative number of correct predictions as a function of the ordered LOO sample indices is plotted on the top panel and the respective prediction confidences are plotted on the bottom panel. The test samples are ordered in the decreasing order of their prediction confidences given by the corresponding classifiers.

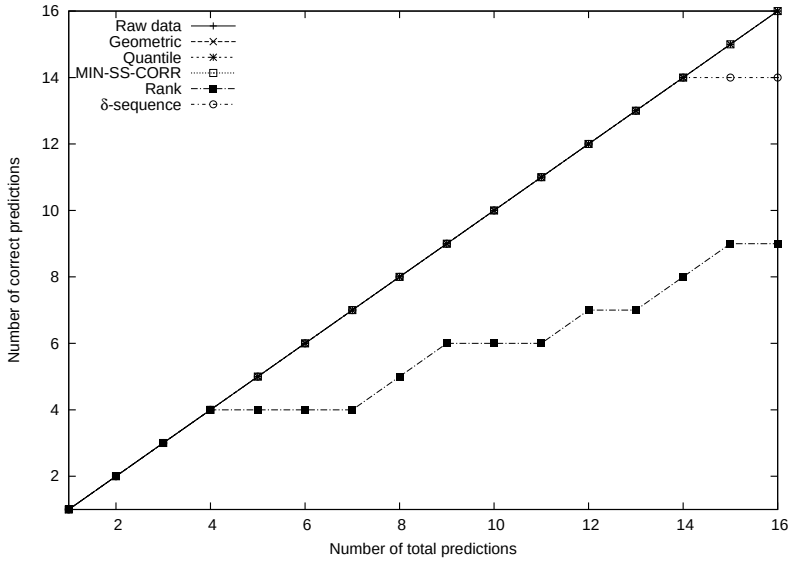


(c) 111 genes or δ's

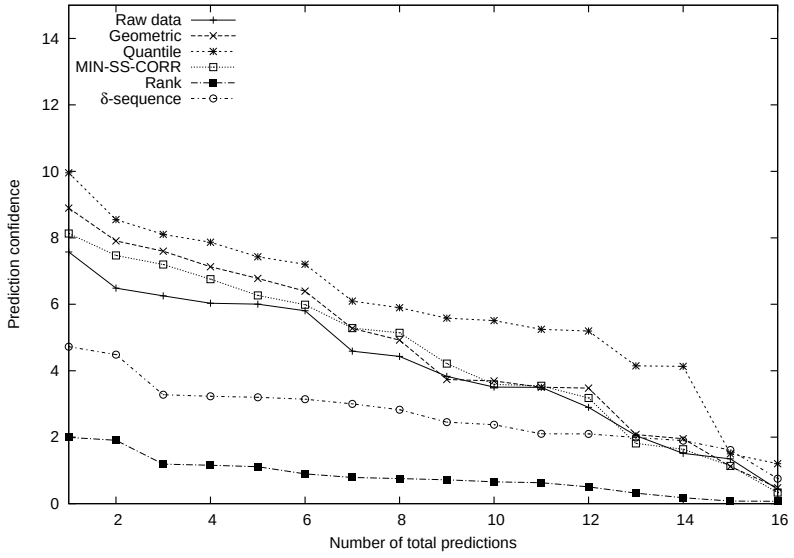


(d) 111 genes or δ's

Figure 3.33: continued.



(e) 222 genes or δ's



(f) 222 genes or δ's

Figure 3.33: continued.

3.2 Conclusions

Classification is the main goal in a number of microarray data analysis experiments. In order to account for experimental variations, normalization is usually performed on the raw microarray data. We proposed a new normalization technique based on minimizing the sum of squared correlation coefficients (MIN-SS-CORR) between pairs of genes. We then evaluated the influences of the proposed method and a few other often used microarray data normalization methods on the inherent classifiability of the raw gene expression measurements. In total, five normalization methods were considered for the experiments: geometric, quantile, MIN-SS-CORR, rank and δ-sequences. The evaluation was based on two methodologies: hypothesis testing and leave-one-out classification using the Golub classifier and the SVM-RFE classifier. The microarray datasets used were the colon cancer dataset of Alon et al. and the angiogenesis dataset of Hoying et al. Different normalization methods influenced the classifiability of the gene expression data to different degrees in terms of enhancing or degrading it.

The three normalization methods: quantile, geometric and MIN-SS-CORR performed favorably by improving the classifiability of the raw data, with quantile normalization being more consistent on both the datasets. The degree to which MIN-SS-CORR helped was a bit less in the case of the angiogenesis dataset but this was probably due to using only a subset of the available genes in the optimization process. δ-sequences were detrimental in the case of the colon cancer dataset but useful in the case of the angiogenesis dataset. Rank normalization proved to be unfavorable for data classification in both the datasets given the results of a true leave-one-out evaluation. However it helped in terms of improved prediction accuracy when leave-one-out was not used for gene selection. More experiments with other datasets are needed to confirm if this is true in general. The two evaluation methodologies agreed with each other in characterizing the effects of the different normalization methods. Better or worse performance in one usually implied the same in the other.

CHAPTER 4

USE OF GENE ONTOLOGIES (GO): A PROBABILISTIC GENERATIVE MODELING FRAMEWORK FOR GENE EXPRESSION LEVELS AND GO TAGS

The Gene Ontology (GO) Consortium (http://www.geneontology.org) maintains annotations of genes and gene products of eukaryotic cells using a standardized controlled vocabulary (Ashburner, 2000; Yon Rhee et al., 2008) . The GO is organized as three sub-ontologies, each being a directed acyclic graph (DAG), describing a particular gene attribute: Biological Process (BP), Molecular Function (MF), or Cellular Component (CC). As new knowledge is gained about the specific roles of genes or their products, the ontology is updated to reflect the new findings. Since eukaryotic organisms share a significant number of genes, i.e. similar nucleotide sequences with similar functions, the ontology helps biologists working on different model organisms share useful knowledge between them. The GO annotations are also useful to statisticians and computational biologists working with high throughput gene expression data (e.g., microarrays) to evaluate their methods and draw biological conclusions from them. Uses of the GO tags include biological analysis of differentially expressed genes and enrichment analysis of gene clusters (Khatri and Drăghici, 2005) . A number of methods have been proposed to do these kinds of analyses (Grossman et al., 2007; Alexa et al., 2006; Lu et al., 2008) . Previous research uses GO annotations after a quantitative analysis of expression data. Instead of treating the two as disjoint sources of information, much can be gained by utilizing the two in conjunction. For example, the information in GO tags can help deal with the biological and experimental noise in expression data. The reverse can be true as high throughput expression measurements can alleviate the effects of noise in GO annotations arising mostly from missing or wrong annotations for some genes.

Probabilistic models for multimodal data provide a way to combine information from multiple modalities. They have been used extensively in other areas of research such as linking words and image regions (Barnard et al., 2003; Barnard and Forsyth, 2001; Lavrenko et al., 2003; Carbonetto and de Freitas, 2003) , to fuse information from different sources and draw useful inferences from them. In this work, we use similar probabilistic generative models to cluster GO terms associated with genes in conjunction with their expression profiles measured using microarrays. This helps to exploit the information available in GO annotations in accounting for some of the noise in the expression measurements. For example, the inclusion of GO terms in clustering increases the likelihood of genes with similar GO annotations to lie within the same cluster even though their expression profiles may differ significantly due to noise.

A few different generative models are proposed to describe the behavior of genes over samples based on their expression levels and GO annotation terms. Two of them, Multimodal Mixture Model (MMM) and its sample-specific counterpart (SS-MMM) assume independence between samples. The other models capture sample correlations by pooling data across samples within the same phenotypic group. Two such models are the Pooled-Sample Multimodal Mixture Model (PS-MMM) and the Multimodal Mixture of Pooled-Sample Models (MM-PSM) for analyzing static gene expression data. The rest are based on multimodal versions of Hidden Markov Models (MHMM) and their mixture (MM-HMM) for analyzing time-course gene expression data. Each pooled-sample model is equivalent to a corresponding multimodal HMM with certain constraints imposed on the latter. Specifically, the PS-MMM is equivalent to a Multimodal Hidden Markov Model (MHMM) and the MM-PSM is equivalent to a Multimodal Mixture of Hidden Markov Models (MM-HMM). Without the constraints, the multimodal HMMs allow more flexibility and power in modeling and making inferences on time-course datasets.

We use phenotype or biological state prediction performance as a measure of how well the proposed models capture the underlying biological process behavior. A number of analyses of gene expression data aim at utilizing the expression mea-

surements to predict the state of a tissue, e.g. diseased or not. So phenotypic prediction performance, in this case state of a biological process, is a sensible measure to gauge any analysis method. To this end, we propose a state prediction framework for each of our models. For the MMM and SS-MMM models, where there is no state information learnt during training, state prediction is done using sample based marginal likelihoods. However for the MHMM (PS-MMM) and MM-HMM (MM-PSM) models, a Bayesian approach is adopted utilizing state information learnt as part of model training. Models with and without using GO tags are compared to assess the usefulness of using GO tags in the joint modeling approach. We posit that our framework provides an objective way to evaluate various methods to model expression data by being able to quantify their performance on unseen data. Our results suggest that the use of GO tags is helpful for phenotype prediction.

4.1 Notation

The following description of the various generative models uses a particular terminology and notation. More specifically, a sample refers to the collection of measurements on a particular microarray. Usually a microarray dataset is a collection of samples, where each sample is obtained by measuring a tissue under a certain biological condition. The vector of expression measurements for a gene g across all the arrays in the dataset is denoted by $\boldsymbol{e}_g$ and the set of GO tags associated with it is denoted by O_g. A particular sample in the data is indexed by t, which corresponds to a time point in the case of time-course experiments. The gene's expression value for that sample or time point is denoted by $\boldsymbol{e}_{gt}$. A pool p or state s represents a collection of samples under a particular phenotypic condition in the case of static and time-course data respectively. The static dataset (Alon colon cancer data) used for our experiments has two conditions corresponding to cancer and normal. The time course datasets have distinct biological stages, each one containing a number of samples. In the case of the angiogenesis data, these stages are the angiogenesis and maturation stages. The cell cycle datasets either have all of the S, G1, M and

G2 phases during the cell division cycle or a subset of them.

4.2 A probabilistic generative modeling framework for gene expression profiles and GO terms

In this section a generic clustering framework for gene expression profiles and GO terms is discussed, which will be useful in understanding the formulation of such an approach and its merits. Let $\boldsymbol{e}_g$ denote the expression vector of a gene g measured under different experimental conditions and O_g be the set of GO tags associated with it. In a generative framework, we assume that the gene's expression vector and its associated GO tags jointly arise from a certain cluster, which is one of a set of possible clusters. The sampling of a particular cluster is governed by a prior over clusters. Let $\boldsymbol{z}_{gC}$ be a vector of indicator variables, which specifies the cluster associated with gene g. In other words, if g comes from the mth cluster, then only the mth component of the vector $\boldsymbol{z}_{gC}$ would be 1 and all the others 0. The vector $\boldsymbol{z}_{gC}$ is missing information, which can be estimated using an Expectation Maximization (EM) (Dempster et al., 1977) procedure. EM is an iterative technique that maximizes the expected value of a complete data likelihood function given data and parameters, at each iteration. There are many possible ways of specifying the complete data likelihood function. The following is one such likelihood function

which is both generic and simple enough for our purposes.

$$L_C = \prod_g p(\boldsymbol{e}_g, O_g, \boldsymbol{z}_{gC}) \tag{4.1}$$

$$= \prod_g \prod_c (P(c)p(\boldsymbol{e}_g, O_g|c))^{z_{gc}} \tag{4.2}$$

$$= \prod_g \prod_c (P(c)p(\boldsymbol{e}_g|c)P(O_g|c))^{z_{gc}} \tag{4.3}$$

$$= \prod_g \prod_c \left(P(c)p(\boldsymbol{e}_g|c) \prod_{o \in O_g} (P(o|c))^{\frac{1}{|O_g|}} \right)^{z_{gc}} \tag{4.4}$$

$$= \prod_g \prod_c \left(\prod_{o \in O_g} (p(\boldsymbol{e}_g|c)P(o,c))^{\frac{1}{|O_g|}} \right)^{z_{gc}} \tag{4.5}$$

where z_{gc} is the cth component of the vector $\boldsymbol{z}_{gC}$, o is a particular GO term and $|O_g|$ is the total number of GO terms associated with gene g. This might include all ancestors of the gene's GO terms in the Directed Acyclic Graph (DAG) (Ashburner, 2000) . In the above, it is assumed that all GO terms are independent of each other and the expression vector, given the cluster. The exponent $\frac{1}{|O_g|}$ adjusts for different numbers of GO tags associated with different genes and lets each gene contribute the same towards the likelihood function.

By specifying parametric forms for $p(\boldsymbol{e}_g|c)$ and $P(o,c)$, it is possible to estimate the underlying parameters using EM. These parameters locally maximize the

corresponding incomplete data likelihood function L_I given by

$$L_I = \prod_g p(\boldsymbol{e}_g, O_g) \tag{4.6}$$

$$= \prod_g \sum_c P(c) p(\boldsymbol{e}_g, O_g|c) \tag{4.7}$$

$$= \prod_g \sum_c P(c) p(\boldsymbol{e}_g|c) P(O_g|c) \tag{4.8}$$

$$= \prod_g \sum_c P(c) p(\boldsymbol{e}_g|c) \left(\prod_{o \in O_g} (P(o|c))^{\frac{1}{|O_g|}} \right) \tag{4.9}$$

$$= \prod_g \sum_c \left(\prod_{o \in O_g} (p(\boldsymbol{e}_g|c) P(o,c))^{\frac{1}{|O_g|}} \right) \tag{4.10}$$

The form of $p(\boldsymbol{e}_g|c)$ depends on the experiment. For example, a time-series experiment might model dependencies between successive components of the vector $\boldsymbol{e}_g$. The estimated $P(o,c)$ and $p(\boldsymbol{e}_g|c)$ are useful for various inferences:

- The parameters estimated using EM imply a density function over the joint space of gene expressions of different samples or time points (time-series experiments) and ontology terms. This density function is proportional to the incomplete data likelihood that is maximized in the EM procedure. It can be used to predict the class label of a new sample. One way to do this is by computing the marginal density of the new sample under all possible sample index assignments and choosing the assignment resulting in the highest likelihood.

- $P(o|c) \propto P(o,c)$ can be used to examine biological category enrichment in gene clusters or to evaluate clustering methods.

- Selection of gene markers is a goal of microarray data analysis methods. Clustering helps reduce the combinatorially exhaustive search space of putative gene marker groups. The most significant genes for each cluster, for example those closest to the mean in terms of Mahalanobis distance (assuming Gaussian clusters), might be chosen as the putative markers. This might help establish

what biological sub-processes or molecular functions are of most relevance to the higher level biological process being studied, using GO tags associated with the markers or using the estimated $P(o|c)$ for the clusters.

- Let $p(\boldsymbol{e}_g, O_g|c) = p(\boldsymbol{e}_g|c) \prod_{o \in O_g} \alpha_{g|o} P(o|c)$, where $\alpha_{g|o}$ is the probability that gene g is annotated with the term o. If we estimate $\alpha_{g|o}$ under certain constraints, using the EM framework, it might imply a useful noise model for GO annotations themselves, i.e., errors and missing annotations.

In this work, we propose generative models for the joint likelihood $p(\boldsymbol{e}_g, O_g)$ of observing gene g's expression vector and tags. The estimated joint likelihood over all genes is then used to predict the state label for a new expression vector. Each one of our models achieves clustering of genes based on their expression levels and GO tags. This clustering is through a latent or hidden variable, which generates expression levels and GO tags conditionally independent of each other, similar to the above formulation. So the simple clustering model above forms an integral component of all our models except that the cluster variable may generate the entire gene expression vector, its subset or a single expression value depending on the model.

4.2.1 Multimodal Mixture Model (MMM)

Here, the gene g's expression vector and its associated GO tags jointly arise from a certain cluster c, which is one of a set of possible clusters. The sampling of a particular cluster c is governed by a prior over clusters $P(c)$.

$$p_{MMM}(\boldsymbol{e}_g, O_g) = \sum_c P(c) p(\boldsymbol{e}_g|c) \left(\prod_{o \in O_g} (P(o|c))^{\frac{1}{|O_g|}} \right) \tag{4.11}$$

where o is a particular GO term and $|O_g|$ is the total number of GO terms associated with gene g (includes all ancestors in the GO DAG). In the above, it is assumed that all GO terms are independent of each other and the expression vector, given the cluster. This is a simplification assumption that will be used for all the models

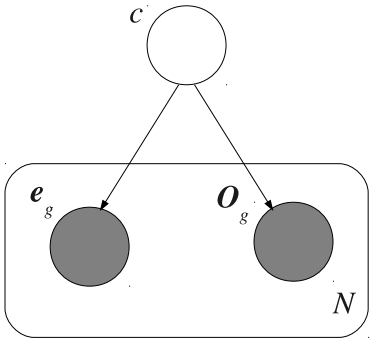


Figure 4.1: Graphical representation of the Multimodal Mixture Model (MMM). The shaded nodes represent random vectors that are observed, which in this case are the gene expression profile ($\boldsymbol{e}_g$) and the GO tags (O_g) of each gene g. It is assumed that there are N such genes and that their data is independently generated. The hidden node represents the latent cluster variable c responsible for generating the expression profile and the GO tags.

proposed in this work. The exponent $\frac{1}{|O_g|}$ adjusts for different numbers of GO tags associated with different genes and lets each gene contribute the same towards the likelihood function over all the genes. The form of $p(\boldsymbol{e}_g|c)$ is assumed to be a multivariate Gaussian with a diagonal covariance matrix and $P(o|c)$ is a discrete distribution learnt during training.

A graphical representation of the above model is shown in Fig. 4.1. The graph depicts a generative model for the expression profile $\boldsymbol{e}_g$ of a gene g and its associated GO tags O_g through the hidden or latent variable c corresponding to a cluster. It can be verified from the form of the corresponding joint distribution that the expression profile and GO tags are independent given a cluster. This cluster conditional independence assumption applies to all the other models proposed in this work.

4.2.2 Sample Specific Multimodal Mixture Model (SS-MMM)

The model of Section 4.2.1 leads to a clustering of genes into groups where the genes within a group have similar expression profiles across all the samples (arrays) and have similar GO tags. It is possible that two genes annotated with the same GO terms might exhibit different behaviors across states or even samples due to biological reasons. We address this possibility by considering alternate models which jointly cluster gene expression values and GO terms within each sample or state separately. The first one among these assuming the extreme case of sample-specific clusters is the Sample Specific Multimodal Mixture Model (SS-MMM). It is described below and the other models which assume pool or state specific clusters are described later. The SS-MMM allows for a gene to cluster with different sets of genes for different samples, which possibly have shared GO terms with the gene. For such a model, the gene likelihood is given by

$$p_{SS-MMM}(\boldsymbol{e}_g, O_g) = \prod_t \sum_{c_t} P(c_t) p(\boldsymbol{e}_{gt}|c_t) \left(\prod_{o \in O_g} (P(o|c_t))^{\frac{1}{|O_g|}} \right) \tag{4.12}$$

where t indexes samples and c_t denotes one of a possible set of clusters for sample t. $p(\boldsymbol{e}_{gt}|c_t)$ is now a univariate Gaussian to model the expression value $\boldsymbol{e}_{gt}$ for sample t.

A graphical model corresponding to the SS-MMM is shown in Fig. 4.2. There is a separate hidden cluster variable c_t specific to the sample index t. The expression value $\boldsymbol{e}_{gt}$ for the gene g for sample indexed by t is generated independently of its associated GO tags O_g given c_t.

4.2.3 Pooling data across samples

The MMM and SS-MMM models of sections 4.2.1 and 4.2.2 respectively do not pool data across samples that belong to a particular phenotype or biological state. This is due to the independence assumption between the expression levels measured for different samples. In general, the expression measures for different samples are correlated. For instance, in a time-course experiment the relative mRNA level of a

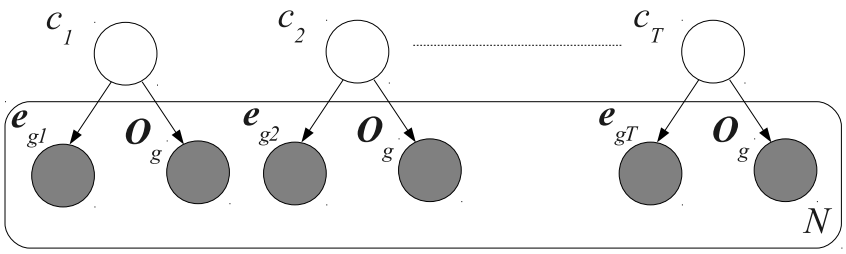


Figure 4.2: Graphical representation of the Sample-Specific Multimodal Mixture Model (SS-MMM). The gene expression measurement $\boldsymbol{e}_{gt}$ for each gene g for the sample indexed by t is generated through the corresponding cluster variable c_t. c_t also generates the GO tags O_g for the gene independent of the expression level. There are N such genes whose observations are assumed to be independently generated according to the model.

gene at a particular time point influences the relative mRNA level at a later time point. A gene that is in relative abundance initially is bound to remain so even though it is not specifically expressed at a later point in time during a biological process. In a dataset with multiple samples per phenotype (e.g. Alon colon cancer dataset), a given gene might have similar expression levels across all the samples for the phenotype except for noise. Such a dataset exhibits correlations among expressions of a gene within a group of samples corresponding to the phenotype.

4.2.4 Pooled-Sample Multimodal Mixture Model (PS-MMM)

In this model, the gene expression measurements for all the samples belonging to a particular biological state or phenotype are pooled together by assuming that they arise from a pool-specific cluster. The joint likelihood $p(\boldsymbol{e}_g, O_g)$ is then given by

$$p_{PS-MMM}(\boldsymbol{e}_g, O_g) = \prod_p \prod_{t \in p} \sum_{c_p} P(c_p) p(\boldsymbol{e}_{gt} | c_p) \left(\prod_{o \in O_g} (P(o|c_p))^{\frac{1}{|O_g|}} \right) \tag{4.13}$$

where p refers to a pool or group of samples representing a biological state or phenotype. c_p represents the latent cluster variable for pool p. A graphical model for the PS-MMM is given in Fig. 4.3. In the figure, p_i corresponds to a particular pool that spans a subset of the available samples. The clusters for the samples belonging to a pool share the same set of parameters. The shaded nodes $\boldsymbol{e}_{gt}$ represent the observed gene expression measures for a gene g for sample t. The nodes O_g represent the set of GO tags for gene g.

4.2.5 Multimodal Mixture of Pooled-Sample Models (MM-PSM)

Although the PS-MMM model pools together measurements belonging to a particular biological state or phenotype, a given gene does not have a single cluster identity across all the samples. In general the clustering of genes is different for different pools, which is similar to the SS-MMM model of Section 4.2.2 except that the clusters are pool-specific rather than sample-specific. It is desirable to have a model where genes have a single cluster identity across the entire set of samples even

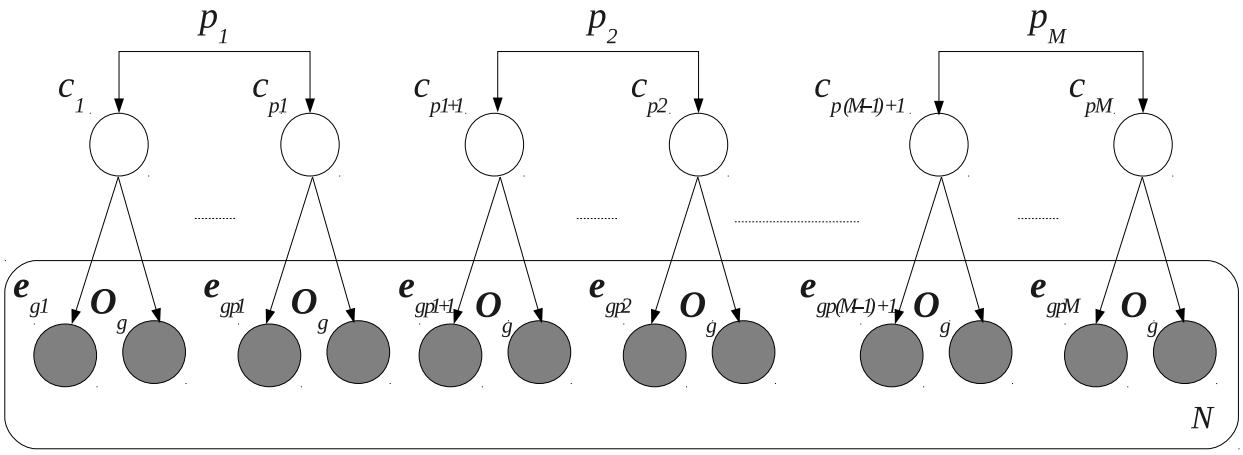


Figure 4.3: Graphical representation of the Pooled-Sample Multimodal Mixture Model (PS-MMM). The gene expression measurements $\boldsymbol{e}_{gt}$ for all samples t belonging to a particular pool p_i are generated from the pool specific clusters. The grouping of variables, e.g. $(c_1, \ldots, c_{p_1})$ under p_1, suggests that the parameters of the underlying clusters are the same for all the samples within a pool. The GO tags O_g are generated independently of the expression levels by the same process. It is assumed that there are M such pools and N genes in the microarray dataset.

though the samples may belong to different phenotype groups. This helps make biological inferences about gene clusters (see Section 4.2) based on their behavior over the entire biological process rather than sub-processes. In order to be able to do this, we propose a multimodal mixture of pooled-sample models (MM-PSM). According to this model the joint density $p(\boldsymbol{e}_g, O_g)$ is given by

$$p_{MM-PSM}(\boldsymbol{e}_g, O_g) = \sum_l P(l) \left(\prod_{p_l} \prod_{t \in p_l} \sum_{c_{p_l}} P(c_{p_l}) p(\boldsymbol{e}_{gt} | c_{p_l}) \right) \left(\prod_{o \in O_g} (P(o|l))^{\frac{1}{|O_g|}} \right) \tag{4.14}$$

where l denotes a mixture component each of which is a pooled-sample model. The number and set of poolings p_l could be the same or different for the mixture components. The pool specific clusters c_{p_l} now model distributions only in the continuous space of gene expression values, i.e., they are not multimodal. However the mixture is multimodal due to the joint modeling of GO tags by the mixture component specific distribution $P(O_g|l)$. As with the PS-MMM model, pooling of samples is achieved by having the same set of clusters c_{p_l} for each pool p_l of each mixture component l.

Figure 4.4 shows a graphical model representation of the MM-PSM model. l is the latent variable corresponding to a mixture component of the multimodal mixture, each of which is a pooled-sample model. The expression measures are pooled into groups, which may be the same or different for the various components. Each component also generates the GO tags O_g associated with a gene g according to the conditional distribution $P(O_g|l)$, the form of which is given in equation 4.14. It can be verified that the GO tags are conditionally independent of the expression levels given a mixture component. The pool specific clusters c_{p_l} now model continuous distributions for the expression levels for all the samples within a pool p_l. These are modeled using a mixture of Gaussians here.

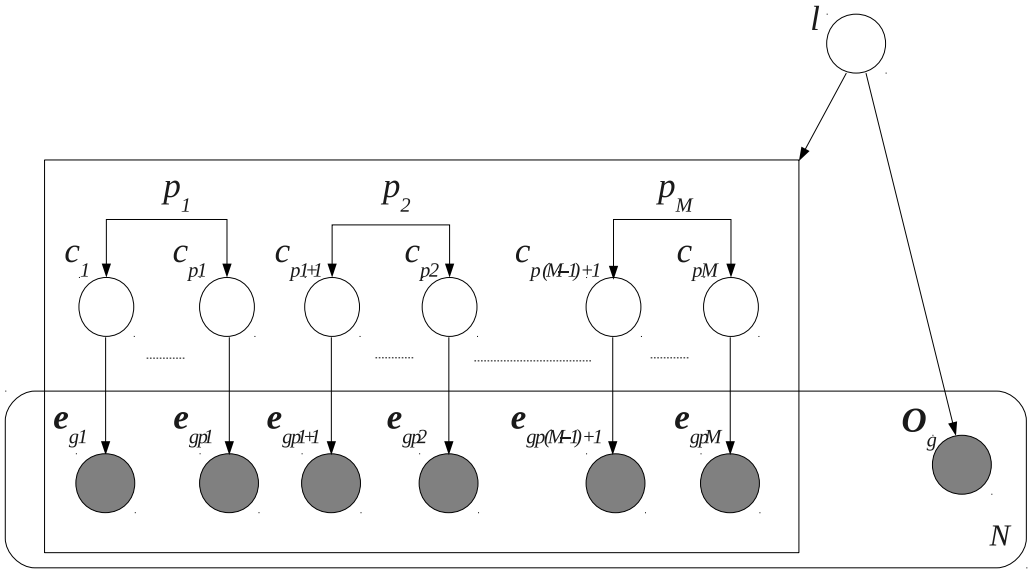


Figure 4.4: Graphical representation of the Multimodal Mixture of Pooled-Sample Models (MM-PSM). Each component of the mixture denoted by the latent variable l generates the gene expression vector $\boldsymbol{e}_g$ and the GO tags O_g associated with a gene g independently. The expression values $\boldsymbol{e}_{gt}$ for various samples t are pooled into groups p_i corresponding to phenotypes or biological states. The number and set of poolings can be different for different components in general. The gene expression values for samples within a pool are generated according to a pool-specific mixture distribution, which is a mixture of Gaussians here. The grouping of variables, e.g. $(c_1, \ldots, c_{p_1})$ under p_1, suggests that the parameters of the underlying clusters are the same for all the samples within a pool. It is assumed that the data has M such pools and N genes.

4.3 A Time-course Perspective

The discussion of the models above is very generic so that the methods are applicable to any kind of dataset including those recorded in time-course experiments. In a time-course experiment, each sample represents a snapshot of gene expression at a particular time instant within a duration of interest. Treating the samples as independent would then be equivalent to assuming that measurements at different time instants as statistically uncorrelated. In many time-course experiments, especially the ones considered in this work, measurements are taken at multiple time points within a particular stage of the biological process being studied. Each stage is usually demarcated from the other by observations that are physical in nature. For example, the various stages in a yeast cell cycle data, which will be discussed in a later chapter (Chapter 5), are marked by observable events such as the size of the buds and the cellular position of the nucleus (Cho et al., 1998) . In such datasets, the assignment of time points to stages is known beforehand. The pools in the pooled-sample models are then the groups of time points within a stage. However such grouping is sometimes not known beforehand. In such cases, models that learn this stage-wise grouping of time points are more desirable.

In what follows, we discuss models that are directly applicable to time-course datasets where the grouping of time points into stages or pools is known. At the same time they have the potential to treat this grouping as unknown and learn it from the data, where needed. They are based on Hidden Markov Models (HMM) and reduce to pooled-sample models when the hidden states (corresponding to pools) are known. If the grouping of time-points into stages is unknown, it can be learned by making inferences on the hidden states as will be discussed in Section 4.4.3.

4.3.1 Hidden Markov Models

Hidden Markov Models (HMMs) (Rabiner, 1989) provide an alternate way to model such correlations. They have been applied before for analyzing time-course gene expression data (Schliep et al., 2005; Bar-Joseph, 2004; Yuan and Kendziorski, 2006)

but none of these have used expression and ontology data together, as we do here. HMMs make use of a hidden or latent variable for each sample or time-point. The hidden variables exhibit first order Markovian property between successive samples and are solely responsible for the generation of observations for the corresponding samples. These properties render tractability in model training and inference and at the same time capture correlations and pool measurements. Each latent variable assumes one of a possible set of values or states. By limiting the number of such states to less than the total number of samples, it is possible to model a microarray dataset as consisting of multiple sub-groups (phenotypic), each one possibly covering multiple samples. It is also possible to introduce additional constraints regarding the initial state and the transitions between successive states. For example, a HMM can be constrained to always start in a nominal first state and switch only to the consecutive state at any sample index within the data and never come back later. This is useful to model a process such as blood vessel growth or one mitotic cycle of the yeast cell because genes usually start in a particular phase and do not come back after switching to the next phase. Further, the transitions can be made to occur simultaneously for all the genes at predetermined sample boundaries. This is useful when there is prior knowledge about biological phase changes in time-series data or even to infer when such changes occur in the data. The learning algorithms for HMMs are tolerant to missing values for one or more genes for one or more samples.

In the case of multiple sample microarray data, the observations are the gene expression profiles and the associated GO tags. There are a few considerations that naturally lead to the particular type of HMMs we propose. One is that the expression levels of all genes under a given biological state, possibly corresponding to a hidden state of the HMM, can be thought of as coming from a finite set of groups (e.g., low, low-medium, medium, medium-high, high) and can be modeled using a mixture distribution. We make use of the associated GO tags in the process of clustering, which makes the densities multimodal. As in the case of sample-independent models of sections 4.2.1 and 4.2.2, we group genes and their GO tags based on their behavior either over the entire sample set or a subset of it. This leads

to two HMM-based models. These models are equivalent to a corresponding pooled-sample model proposed in Sections 4.2.4 and 4.2.5 subject to certain constraints imposed on the state switches.

4.3.2 Multimodal Hidden Markov Model with Constrained Switches (MHMM-CS)

This is a standard HMM with emission densities for the hidden states being multimodal mixture distributions. Each component of the mixture is a product of a parametrized continuous distribution (Gaussian) for the expression level and a discrete distribution for the GO tags. In other words, each hidden state is assumed to emit gene expression levels and GO tags jointly according to the state specific mixture distribution. Let K denote the total number of hidden states of the HMM. The likelihood of observing a gene g's expression vector $\boldsymbol{e}_g$ and its associated GO tags O_g is given by

$$\begin{aligned} p_{MHMM}(\boldsymbol{e}_g, O_g) &= \sum_{i=1}^{K} P(s_1 = i) p(\boldsymbol{e}_{g1}, O_g | s_1) \\ &\quad \prod_{t=2}^{T} \sum_{j=1}^{K} \sum_{k=1}^{K} P(s_t = k | s_{t-1} = j) p(\boldsymbol{e}_{gt}, O_g | s_t) \end{aligned} \tag{4.15}$$

where s_t denotes the hidden variable for sample or time (in case of time-course data) t with the total number of samples or time points being T. The probabilities $P(s_1)$ and $P(s_t|s_{t-1})$ govern the initial state and transition probabilities of the latent variables between samples $t-1$ and t respectively. The HMM is constrained so that state switches occur at predetermined sample boundaries by choosing appropriate fixed values for $P(s_t|s_{t-1})$ as a function of sample index. The emission density for a particular state $s_t = i$ is a multimodal mixture density given by

$$p(\boldsymbol{e}_{gt}, O_g | s_t = i) = \sum_{c_i} P(c_i) p(\boldsymbol{e}_{gt} | c_i) \left(\prod_{o \in O_g} (P(o|c_i))^{\frac{1}{|O_g|}} \right) \tag{4.16}$$

where c_i indexes over all the clusters for state i. The above model assumes separate clusters for each state of the HMM, thereby allowing genes with similar GO tags be-

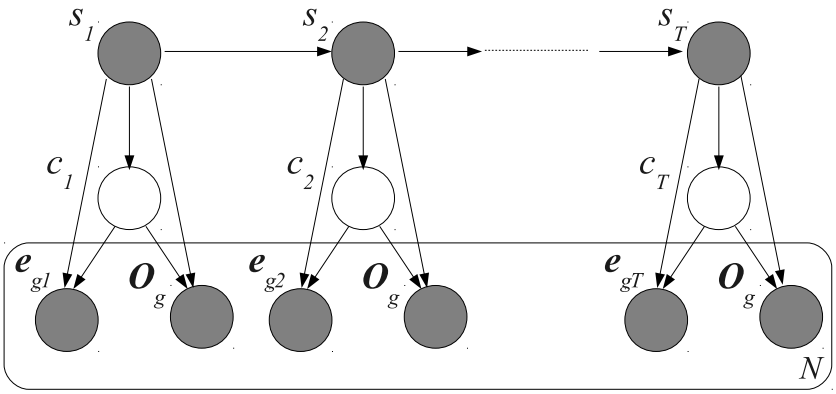


Figure 4.5: Graphical representation of the Multimodal Hidden Markov Model with Constrained Switches (MHMM-CS). A node s_t denotes the HMM state at sample index t, which becomes observed (shaded) given that the sample boundaries at which state switches occur are known. Each state's emission distribution is modeled by a multimodal mixture. The node c_t represents the hidden cluster variable for the sample indexed by t. A gene g's expression value $\boldsymbol{e}_{gt}$ for a sample t and its GO tags O_g are assumed to be independent given a cluster. There are N such genes in the dataset. Note that with the nodes s_t being observed this model is the same as the one in Fig. 4.3 corresponding to a PS-MMM.

have and hence cluster differently for different biological states under consideration. This is addressed by the model of section 4.2.2 in the sample independent case.

By specifying the sample boundaries at which state switches occur, the hidden states become observed and this model becomes equivalent to the PS-MMM model of section 4.2.4. This can be verified from the graphical representation of this model shown in Fig. 4.5 with each discrete HMM state corresponding to a particular pool in the pooled-sample model.

4.3.3 Multimodal Mixture of Hidden Markov Models with Constrained Switches (MM-HMM-CS)

This is a mixture of HMMs to allow for clustering of genes based on their expression profiles over the entire sample set and their associated GO tags. Each gene g's expression profile $\boldsymbol{e}_g$ and its associated GO tags O_g are assumed to be sampled from a multimodal mixture distribution. Each component of the mixture distribution has two factors. The first is a continuous distribution, modeled by a HMM, for the expression profile and the second is a discrete distribution for the GO tags. This model is the time-correlated or pooled data counterpart of the model described in section 4.2.1 with application to time-course gene expression datasets. The gene likelihood according to this model is given by

$$p_{MM-HMM}(\boldsymbol{e}_g, O_g) = \sum_l P(l) p_{HMM_l}(\boldsymbol{e}_g|l) \left(\prod_{o \in O_g} (P(o|l))^{\frac{1}{|O_g|}} \right) \tag{4.17}$$

where l indexes clusters and the usual assumption of cluster conditional independence between GO tags and gene expression levels is made. The continuous part of the lth component density, p_{HMM_l}, pools gene expression levels with the help of a HMM by expressing it as

$$\begin{aligned} p_{HMM_l}(\boldsymbol{e}_g|l) &= \sum_{i=1}^{K_l} P_l(s_1 = i) p_l(\boldsymbol{e}_{g1}|s_1) \\ &\quad \prod_{t=2}^{T} \sum_{j=1}^{K_l} \sum_{k=1}^{K_l} P_l(s_t = k|s_{t-1} = j) p_l(\boldsymbol{e}_{gt}|s_t) \end{aligned} \tag{4.18}$$

As before, s_t denotes the hidden variable for sample t, which can assume one of K_l states for the lth HMM. Note that the state emission densities, $p_l(\boldsymbol{e}_{gt}|s_t)$, of the HMMs are not multimodal. They are continuous distributions modeled using a parametric distribution such as a mixture of Gaussians. Similar to the MHMM-CS model, the state transitions of the component HMMs are designed to occur at predetermined sample boundaries resulting in the constrained MM-HMM-CS model. The number of states is kept the same for each component HMM but they can have

different state transition points. This allows for lags for certain genes in switching stages in the case of modeling a time-course experiment. This model also has the property of assigning genes to clusters over entire sample set as opposed to over subsets of it as is done by the MHMM-CS (PS-MMM) model.

A graphical model representation of MM-HMM-CS is given in Fig. 4.6. Similar to the MHMM-CS model, the hidden nodes of the component HMMs in the mixture become observed due to specifying the state switch sample boundaries. Also for the same reason this model is equivalent to the MM-PSM model of Section 4.2.5. This can be verified from the graphical representations of the two models with each HMM state corresponding to a pool in the pooled-sample model.

4.4 Model training and inference

Each of the above proposed models is trained using a suitable version of the EM algorithm (Dempster et al., 1977; Rabiner, 1989; Bilmes, 1998) . EM locally maximizes the likelihood over all the genes assuming that the data for each gene is independently generated. That is it maximizes the likelihood function $\prod_g p(\boldsymbol{e}_g, O_g)$. The estimated parameters include the Gaussian mixture means, variances, cluster priors and cluster conditional ontology term distributions.

4.4.1 Sample-based state prediction

The models of sections 4.2.1 and 4.2.2 do not learn about biological states and hence cannot be used directly to make inferences about state for a new sample. We propose a framework for these models to do state prediction based on marginal sample likelihoods using the estimated model parameters. The posterior distribution over states s given the new measurement sample $\boldsymbol{m}^{new}$ and the set of GO tag

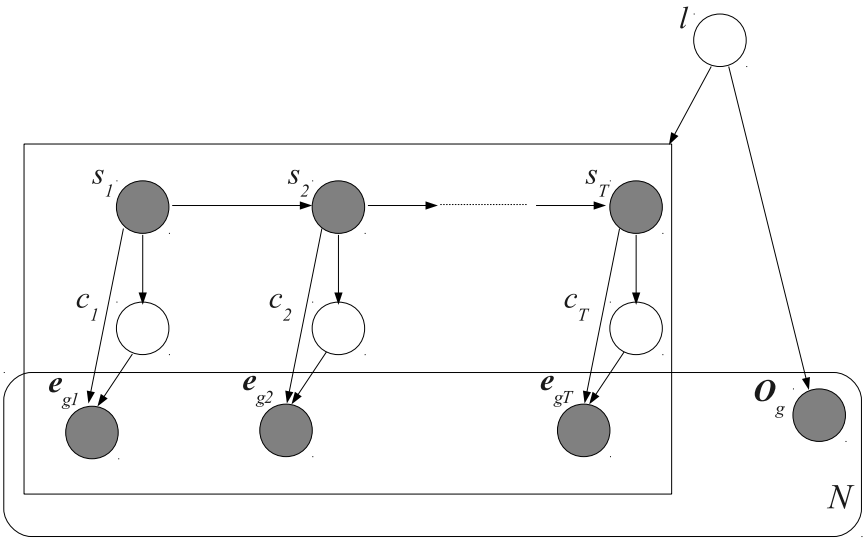


Figure 4.6: Graphical representation of the Multimodal Mixture of Hidden Markov Models with Constrained Switches (MM-HMM-CS). The node l is the hidden node denoting a mixture component of the HMM mixture. The part of the graph within the square box represents a HMM (see Fig. 4.5), which is one of the components in the mixture. The states s_t of the individual HMMs become observed (shaded) due to specifying the state switch sample boundaries. The GO tags O_g for a gene g are assumed to be generated independently of the corresponding expression vector $\boldsymbol{e}_g$ through the node l. The expression value $\boldsymbol{e}_{gt}$ for a particular sample t is modeled by a mixture of Gaussians governed by the cluster variable c_t. It is assumed that there are a total number of N genes in the dataset.

assignments O for all genes can be written as

$$P(s|\boldsymbol{m}^{new}, O) \propto p(s, \boldsymbol{m}^{new}, O) \tag{4.19}$$

$$= \sum_t p(t, s, \boldsymbol{m}^{new}, O) \tag{4.20}$$

$$= \sum_t p(\boldsymbol{m}^{new}, O|s, t)P(s, t) \tag{4.21}$$

$$= \sum_t p(\boldsymbol{m}_t^{new}, O)P(s, t) \tag{4.22}$$

where a marginalization over all samples t is performed. The term $p(\boldsymbol{m}^{new}, O|s, t)$ denotes the conditional likelihood of the new data observed as sample indexed by t within training data and belonging to state s. This is nothing but the marginal likelihood $p(\boldsymbol{m}_t^{new}, O)$ of the observed sample assuming it is assigned the sample index t. The quantity $P(s, t)$ can be set to 1 or 0 depending on whether the sample t in the training data arises from state s or not followed by appropriate normalization of the posterior.

Marginal Sample Likelihoods

The maximum likelihood parameters estimated in the EM procedure imply a joint probability density function over the space of gene expression levels of different samples and GO terms. This density function is proportional to the maximized likelihood L_I itself. Assuming there are a total of T samples, the joint density function is given by

$$p(\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots \boldsymbol{m}_T, O) \propto L_I(\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots \boldsymbol{m}_T, O) \tag{4.23}$$

where $\boldsymbol{m}_i$ is the vector of measurement of all genes for sample i and O is the set of GO term assignments to genes. Given a new gene expression measurement sample $\boldsymbol{m}^{new}$, its marginal density, assuming it is assigned the tth sample label under the above joint density model, is given by

$$\begin{aligned} p(\boldsymbol{m}_t^{new}, O) \quad &= \textstyle\int_{M_{-t}} p(M, O) dM_{-t} \\ &\propto \textstyle\int_{M_{-t}} L_I(M, O) dM_{-t} \end{aligned} \tag{4.24}$$

In the above, M is used to denote the collection of gene expression vectors over all samples, i.e., $M \equiv (\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_T)$ and M_{-t} is used to denote the same collection with the tth sample removed, i.e., $M_{-t} \equiv (\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_{t-1}, \boldsymbol{m}_{t+1}, \ldots, \boldsymbol{m}_T)$. The marginal sample likelihoods are useful for state prediction for models that do not explicitly learn about biological states. One way of using these likelihood values for state prediction is to sum them up over all the sample labels corresponding to the state and normalize to obtain state probabilities.

We first develop the framework for marginal likelihood computation that applies to the two stateless models (MMM and SS-MMM) and then describe the specific details for each model. In order to compute the marginal, first note that the likelihood function L_I is a product of likelihoods for individual genes

$$L_I(\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_N, O) = \prod_{g=1}^{N} L_I(\boldsymbol{e}_g, O_g) \tag{4.25}$$

where N is the total number of genes used in the study. The gene likelihood $L_I(\boldsymbol{e}_g, O_g)$ is given by the form of $p(\boldsymbol{e}_g, O_g)$ for the model of interest (see equations 4.11 and 4.12). With this notation, the measurement vector for sample t is nothing but $\boldsymbol{m}_t = (\boldsymbol{e}_{1t}, \boldsymbol{e}_{2t}, \ldots, \boldsymbol{e}_{Nt})$. The marginal of equation 4.24 can be computed by factorizing the integral over all genes into integrals for individual genes

$$\begin{aligned} p(\boldsymbol{m}_t^{new}, O) &\propto \int_{E_{-t}} L_I(E, O) dE_{-t} \\ &= \prod_{g=1}^{N} \left(\int_{\boldsymbol{e}_{g,-t}} L_I(\boldsymbol{e}_g, O_g) d\boldsymbol{e}_{g,-t} \right) \\ &= \prod_{g=1}^{N} M(g) \end{aligned} \tag{4.26}$$

where $M(g)$ can be thought of as the gene-wise marginal. E is the set of expression vectors for all genes, i.e., $E \equiv (\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_N)$. Similarly, E_{-t} is the set of expression vectors for all genes with the values for sample t removed for each gene, i.e., $E_{-t} \equiv (\boldsymbol{e}_{1,-t}, \boldsymbol{e}_{2,-t}, \ldots, \boldsymbol{e}_{N,-t})$, where $\boldsymbol{e}_{g,-t}$ represents the expression vector of gene g with its tth component removed. The form of $M(g)$ for the different models can be derived analytically based on the linearity and factorization properties

of integration and the cluster conditional independence assumption between gene expression levels and GO tags.

MMM

The gene-wise marginal in the case of MMM can be shown to be

$$M(g) = \sum_c P(c)p(\boldsymbol{e}_{gt}^{new}|c)P(O_g|c) \tag{4.27}$$

where the usual assumption of cluster conditional independence of GO terms applies for $P(O_g|c)$ (see equation 4.11). Similar arguments apply to the marginal for the SS-MMM model.

SS-MMM

The form of the gene-wise marginal $M(g)$ for gene g in this case is

$$\begin{aligned} M(g) \;=\; & \left(\prod_{s=1}^{t-1}\sum_{c_s} P(c_s)P(O_g|c_s)\right) \\ & \left(\sum_{c_t} P(c_t)p(\boldsymbol{e}_{gt}^{new}|c_t)P(O_g|c_t)\right) \\ & \left(\prod_{s=t+1}^{T}\sum_{c_s} P(c_s)P(O_g|c_s)\right) \end{aligned} \tag{4.28}$$

4.4.2 Bayesian state prediction for pooled-sample and state models

Unlike the models of sections 4.2.1 and 4.2.2, the models of Sections 4.2.4, 4.2.5, 4.3.2 and 4.3.3 are designed to learn about biological states during training. The pools in the case of pooled-sample models or the latent states in the case of HMMs can be thought of as modeling various phenotypic states of the biological process under study. A Bayesian approach is taken to compute a probability distribution over states for a new sample. Given $\boldsymbol{m}^{new}$ and the set of GO tags O, the posterior distribution over states $P(s|\boldsymbol{m}^{new}, O)$ can be written using Bayes' rule as:

$$P(s|\boldsymbol{m}^{new}, O) \propto p(s, \boldsymbol{m}^{new}, O) \tag{4.29}$$

Note that s is used to denote a phenotype or biological state as applied to both pooled-sample and HMM-based models. This corresponds to the variable p in the case of pooled-sample models. The form of $p(s, \boldsymbol{m}^{new}, O)$ depends on the particular model. For the MHMM-CS (PS-MMM) model of section 4.3.2 (4.2.4), it is proportional to the pool or state specific likelihood, i.e.,

$$p(s, \boldsymbol{m}^{new}, O) \propto p(\boldsymbol{m}^{new}, O|s) \tag{4.30}$$

$$= \prod_g p(\boldsymbol{e}_g^{new}, O_g|s) \tag{4.31}$$

The form of $p(\boldsymbol{e}_g^{new}, O_g|s)$ is given in sections 4.2.4 and 4.3.2.

A similar approach is adopted for the MM-HMM-CS (MM-PSM) models of Sections 4.3.3 (4.2.5) except that the state specific likelihoods are marginalized out over all the mixture components

$$p(s, \boldsymbol{m}^{new}, O) = \prod_g \sum_l p(\boldsymbol{e}_g^{new}, s|l) P(O_g|l) P(l) \tag{4.32}$$

where $p(\boldsymbol{e}_g^{new}, s|l)$ is computed using the parameters of the lth mixture component.

4.4.3 Maximum likelihood state sequence estimation using multimodal HMMs

The constrained multimodal HMM-based models discussed earlier assume that the time points at which stage switches occur are known a priori. They use this information to pool data within a particular biological stage for all the genes. This is done by assigning all the samples within a stage to a corresponding (pseudo) hidden state. However, such information is not always available but it can be inferred within the framework of HMMs. A possible way to do this is to estimate the most likely hidden state sequence for a given data sequence and then figure out the switch point from the resulting state sequence. Efficient algorithms exist for HMMs to do such inferences, e.g. the Viterbi algorithm (Rabiner, 1989) .Given that we impose other constraints on the models such as always starting in a nominal first state and switching states in a left-to-right fashion ensures that the resulting sequence agrees with the biology of the data. In what follows we propose two counterparts to

the earlier HMM-based models that do not assume anything about the time points where state switches occur. The property of all genes switching stages at the same point is preserved by letting the states of the HMM emit the expression levels of all the genes on the microarray at once. The state switch points are then inferred from the learnt models in a maximum likelihood framework using either the Viterbi algorithm, where applicable, or a brute force search.

Let E denote the matrix of gene expression levels over all the microarrays and let E_t denote the vector of expressions for a sample or time point indexed by t. The joint likelihood of the complete microarray data E and the GO tags O is modeled using a multimodal HMM by

$$\begin{aligned} p(E,O) &= \sum_{i=1}^{K} P(s_1 = i) p(E_1, O|s_1) \\ & \quad \prod_{t=2}^{T} \sum_{j=1}^{K} \sum_{k=1}^{K} P(s_t = k|s_{t-1} = j) p(E_t, O|s_t) \end{aligned} \tag{4.33}$$

where s_t is the HMM state at time t taking on one of K possible values. The probabilities $P(s_1)$ and $P(s_t|s_{t-1})$ govern the initial state and transition probabilities of the latent variables between time points $t-1$ and t respectively. The state emission densities are multimodal in order to incorporate gene expression levels and GO tags. Given a state at a time point t, each gene g's expression value $\boldsymbol{e}_{gt}$ and its associated GO tags O_g are emitted according to the state-specific multimodal mixture distribution.

$$p(\boldsymbol{e}_{gt}, O_g|s_t = i) = \sum_{c_i} P(c_i) p(\boldsymbol{e}_{gt}|c_i) \left(\prod_{o \in O_g} (P(o|c_i))^{\frac{1}{|O_g|}} \right) \tag{4.34}$$

where c_i indexes over all the clusters for state i. Note the conditional independence assumption between gene expression levels and GO tags given a cluster. The state emission likelihood is given by

$$p(E_t, O|s_t) = \left(\prod_{g} p(\boldsymbol{e}_{gt}, O_g|s_t) \right)^{\frac{1}{N}} \tag{4.35}$$

where N denotes the total number of genes measured on the microarray. The genes are assumed to be independent given a state and the likelihood $p(E_t, O|s_t)$ can be viewed as the average (geometric) per-gene likelihood at time point t.

In addition to a single multimodal HMM above, we model the gene expression data and GO tags using a mixture of them. Each mixture component, indexed by l, is a multimodal HMM modeling the joint likelihood of the matrix of expression levels E and GO tags O using the same formalism as in equation 4.33. The likelihood according to the mixture is then given by

$$p(E, O) = \sum_l P(l) p_{HMM_l}(E, O) \tag{4.36}$$

where $P(l)$ is a prior over the mixture components and $p_{HMM_l}(E, O)$ denotes the likelihood of the lth component. The form of $p_{HMM_l}(E, O)$ for any l is given in equation 4.33 except that the parameters are different for the different components. This model allows clustering of genes over the entire sample set with cluster membership of a gene in a particular component proportional to the joint likelihood of its expression profile and GO tags under the corresponding HMM component.

Using the parameters of the learnt model in each case, the maximum likelihood (ML) state sequence is estimated. In the case of a single multimodal HMM this can be done efficiently using the Viterbi algorithm (Rabiner, 1989; Bishop, 2006) . However the Viterbi algorithm is not directly applicable to a mixture of HMMs because the path elimination strategy (Bishop, 2006) does not generalize from a single HMM to a mixture of them. We choose to estimate the ML state sequence in this case by brute force search. That is we compute the likelihood of all the possible state sequences under the learnt model and choose the one with the maximum. In order to compute the likelihood for a given state sequence, we assume the same sequence for each of the component HMMs in the mixture.

CHAPTER 5

EXPERIMENTS

The models proposed in Chapter 4 are general enough to describe a number of microarray datasets. Broadly, the datasets recorded from microarray experiments can be categorized into two types: dynamic (time-course) and static (non time-course). In a dynamic or time-course experiment, snapshots of gene expression of the same set(s) of tissue are taken over a period of time. Usually, the goal is to study a biological process behavior over the period of interest. Examples of such experiments include studies of progression of certain types of cancer, prognosis after treatment, and cyclic biological phenomena such as the cell division cycle or the circadian rhythm. Static experiments are conducted irrespective of time to study molecular differences between certain biological conditions, e.g. between diseased and normal. Datasets measuring gene expression in diseased tissues collected from different people suffering from a particular type of cancer paired with measurements from their normal tissues serve as good examples of static datasets. Gene expression measures recorded over a few cell cycle periods in a number of organisms (fungus, mouse, humans) are examples of time-course datasets.

We use the modeling framework described in the previous chapter on both dynamic as well as static datasets. This is possible because our approach treats microarray samples as being collected into groups corresponding to biological states. In the case of time-course data, these groups represent particular phases of the biological process being studied, e.g. a cell cycle phase. In the case of static data, the groups correspond to a particular phenotype, e.g. disease. The prediction framework employing the models is then useful for predicting a label for a new sample. This label represents a stage for a time-course experimental design and a phenotype for a static design. A brief description of the datasets used for our experiments is given below with the corresponding experimental protocol followed by the results.

5.1 Datasets and experimental protocol

5.1.1 Static

The static dataset used for our experiments is the colon cancer data of Alon et al. (Alon et al., 1999) . It consists of 62 tissue samples of which 40 are tumor samples and 22 are normal. The samples were obtained from different patients and from some of them both normal and cancerous tissues were collected. Affymetrix arrays containing more than 6500 oligonucleotide sequences complementary to human genes were used to measure the gene expression profiles across the samples. After a filtering process, 2000 genes with the highest minimal intensity across all the samples were retained. These expression measures and the annotations for the 2000 genes are publicly available (http://microarray.princeton.edu/oncology/affydata/index.html).

The raw data was normalized using the Quantile normalization method. The data was split into a training and a held-out set. The training set consists of 40 tissue samples of which 14 are normal and 26 are cancerous tissues. Similarly, the test set has 8 normal and 14 cancerous tissues. Extracting the GO tags by tracing the Biological Process (BP) and Molecular Function (MF) GO hierarchies up to the root node for each of the 2000 genes resulted in a total of 3360 and 1294 tags respectively. Models with and without using GO tags were learnt using samples from the training set. The models that pool data across samples do so within a particular phenotypic group. In this case, the two phenotypic groups correspond to normal and cancer.

Two sets of models were trained using GO tags from the BP and MF GO hierarchies respectively and one without any tags (No GO). The prediction task was to label a test sample as being either normal or cancerous. For all the experiments reported here, models were trained using 40 iterations of EM with 10 different initial points (E-step) and the phenotype or stage label prediction accuracies were averaged over all the different initializations. Averaging helps to account for the local maximum likelihood estimation property of EM. Phenotype or stage label prediction was done on samples from both training and test sets and average prediction

accuracies were computed.

5.1.2 Time-course

We used data from three time-course experiments to train our models and then predict the biological stage labels using the estimated model parameters: angiogenesis data of Hoying et al. (Greer et al., 2006) , yeast cell cycle data of Cho et al. (Cho et al., 1998) , and human cell cycle data of Whitfield et al. (Whitfield et al., 2002) .

Hoying angiogenesis data

The Hoying angiogenesis data came from an experimental model (in vivo) of tissue vascularization in SCID mice with the implants obtained from either tie2:GFP mouse or rat adipose. Tissue samples were extracted from the implanted constructs at discrete time points – days 3, 7, 14, 21 and 28. In a unique experimental design these samples and a day 0 sample (implant source) were hybridized using two channel microarrays to obtain measurements of gene expression. Care was taken so that biological variations were averaged out in the measurements. There were 4 measurements per gene per time point. The intensity measurements were background subtracted and lin-log transformed (Greer et al., 2006) for variance stabilization and only those measurements that were consistently well above the background level were retained. The transformed measurements were corrected for spatial location and intensity variations using *lowess* regression in a custom statistical software called CARMA (Greer et al., 2006) . Then a gene-by-gene ANOVA was performed to adjust for the gene specific variations introduced by the experimental factors namely the Array (A), Dye (D) and Variety (V) effects. The V effect corresponded to time and was the measurement of interest. Data from two such hybridization runs (run-1 and run-2) was used for the experiments.

Measurements from both the hybridization runs were used (run-1, run-2). To evaluate prediction accuracies, we trained models using measurements from each run

separately. Accuracies on training data were computed by predicting stage labels for samples from the same run used for training. Held-out accuracies were computed by predicting labels for samples from the other run. Only differentially expressed genes selected by CARMA were used. Since the differentially expressed genes that were selected were different for different runs we included only the ones that were common to both. Starting from 1282 genes that were selected as differentially expressed in run-1, we searched for their corresponding normalized expression levels in run-2. Only 706 of them had any valid measurements in run-2 based on ANOVA and so only these were used for one set of experiments. Similarly, starting from the 978 differentially expressed genes of run-2, we found 932 of them having any valid normalized expression values in run-1 based on ANOVA.

Two sets of models were trained using GO tags from the Biological Process (BP) and Molecular Function (MF) GO hierarchies respectively and one without any tags (No GO). For data with 706 genes, this resulted in 480 and 363 total GO tags from the BP and MF hierarchies respectively. The total number of GO tags for the other data with 932 genes were 569 and 400 from the BP and MF hierarchies respectively. Note that we included all the ancestors of the GO annotations for the genes by tracing the GO hierarchies starting from each annotation all the way to the root of the corresponding hierarchy.

We considered stage prediction corresponding to our hypothesis of the two stages of blood vessel growth (angiogenesis and maturation). We hypothesized that the **first two time points** correspond to angiogenesis and the **next four to maturation**. This is based on the experimental results described in section 5.3. For the stageless models of sections 4.2.1 and 4.2.2, stage probabilities were computed from marginal time probabilities by adding up the marginals over all time points corresponding to the stage. The predicted stage was the one with the highest probability.

Cho yeast cell cycle data

The yeast cell cycle data has been used to identify and study the periodic fluctuations of mRNA levels of cell cycle regulated genes. Synchronized cells derived

Table 5.1: Assignment of time points to various stages of the yeast cell cycle in the Cho dataset. The data has two complete cell cycles with the transition from the first to second cycle occurring at 90 mins from start. The assignment of time samples from the two cycles are listed in separate columns.

Phase	Time points (Cycle-1)	Time points (Cycle-2)
G1	0, 10, 20	90, 100
S	30, 40	110, 120
G2	50, 60	130, 140
M	70, 80	150, 160

from cultures of *Saccharomyces cerevisiae* and arrested at time 0 were allowed to go through mitotic cell division over a period of 160 minutes. The method of synchronization was based on the temperature sensitive *cdc28-13* allele. Expression levels of all the genes in the yeast genome were sampled at 10 min intervals resulting in a total of 17 measurements including time 0. The cells underwent almost two complete cell divisions within this period and hence the data represents measurements over two cell cycles. The transition from the first to second cycle is approximately at about 90 mins from the start. Each cell cycle is divided into 4 phases: G1, S, G2, and M based on bud size, cellular position of nucleus and standardization to previously known cell cycle regulated genes (Cho et al., 1998) . The S phase corresponds to genome duplication, M phase to nuclear division, which are separated from each other by the two gap phases G1 and G2.

The raw yeast cell cycle data was preprocessed by log transformation followed by geometric normalization. Measurements from both the cell cycles were used and prediction accuracies were computed by training models on data from each cell cycle separately. Stage prediction accuracies were computed in a similar fashion as the Hoying data. Table 5.1 lists the time samples corresponding to the various phases of cell cycle for both the cycles. A previous work using this dataset has identified 421 (Cho et al., 1998) genes as cell cycle regulated. We used only the measurements for these genes in our experiments. This resulted in a total of 1275 and 491 tags including the ancestors in the BP and MF GO hierarchies respectively. Models

were trained with (BP or MF) and without (No GO) GO tags followed by stage prediction.

Whitfield human cell cycle data

Similar to the yeast, genes regulated periodically during the human cell cycle have been identified using the HeLa cancer cell line (Whitfield et al., 2002) . Three different synchronization methods were used to arrest cells in the S phase (double thymidine block) and M phase (thymidine-nocodazole block and mitotic shake-off) of the cell division cycle. Data from the third double thymidine block study was used for the experiments here. After release from the double thymidine block, when the cells entered S phase, gene expressions were monitored every hour up to 46 hours during which the cells went through three successive cell division cycles. The cell divisions occurred after approximately 13h, 27h and 44h relative to the 0 time point (release from the block). This yielded 14, 14 and 17 measurements for the first, second and third cycles respectively. cDNA arrays probing nearly 30000 human genes were used to measure gene expression levels.

In the human cell cycle data the time points corresponding to the S phase are clearly delineated from that of the other phases but it is not the case with the other remaining phases. There seems to be an overlap between the time points corresponding to the M, G1 and G2 phases. So we split the time points over one cell cycle into two phases: S and Non-S, where the Non-S phase is the superset of the remaining phases. This leads to the following correspondence between samples of the three cell cycles and the two phases (Table 5.2). We use a subset of the 1134 genes identified as cell cycle regulated for our experiments. This is the set of 1099 genes for which the publicly available data table has non-empty rows (http://genome-www.stanford.edu/Human-CellCycle/HeLa/). The total number of GO tags for the included genes is 567 and 222 in the BP and MF hierarchies respectively.

Table 5.2: Assignment of time points to the S and Non-S phases of the human cell cycle in the Whitfield dataset. This dataset has 3 complete cell cycles with samples recorded every hour over a period of 46 hours, with a different number (14, 14 and 17) and assignment of samples to the two phases of each cycle. Each column shows the phase assignment for the samples, denoted by the measurement hour, within a cycle.

Phase	Time points (Cycle-1)	Time points (Cycle-2)	Time points (Cycle-3)
S	0-3	14-18	28-32
Non-S	4-13	19-27	33-44

5.2 Prediction results

Using the protocol for each dataset described above we computed the mean of prediction accuracies over the 10 different EM runs for each of the models. We then averaged the resulting means separately over training and held-out sets and computed the standard error of the means. The choice of the number of clusters for MMM and SS-MMM models and the number of pools or states and clusters for the pooled-sample or HMM-based models is described below. Note that each pooled-sample model is equivalent to a corresponding constrained HMM-based model (Chapter 4) in the case of time-course data. So we describe results for static and time-course data using suitable notation for the models that pool data across samples or time-points respectively.

5.2.1 MMM

The number of clusters was chosen to be 14. Note that the choice of number of clusters is rather arbitrary for this set of experiments. It is possible to choose it based on prior knowledge or using a model selection technique such as AIC or BIC. We do not address this issue in this section since the main goal is to evaluate the relative usefulness of GO tags for prediction. However, the prediction performance as a function of different number of clusters for this model will be plotted in Section 5.5.

5.2.2 SS-MMM

The number of clusters at each sample or time point was chosen to be 14.

For the pooled-sample or HMM-based models, Gaussian distribution was assumed for the pool or state specific emission densities or the mixture components thereof for describing the expression levels. All the HMMs were constrained to start in the first state and allowed to switch only to the next state during each time point and never allowed to switch back.

5.2.3 MHMM-CS (PS-MMM)

For the angiogenesis data, the number of hidden states was set to 2 and each state's emission density had a mixture with 14 components. The choice of the number of states is based on the two hypothesized stages of blood vessel growth. For the yeast cell cycle data, the number of hidden states was set to 4 based on the four phases of the cell cycle. The number of components in each state's emission mixture was set to 14. Since we are considering only two phases (S or Non-S) for the human cell cycle data, the number of hidden states was correspondingly set to 2 with each state's emission density being a mixture of 14 components.

Since there are two phenotype labels for the samples in the colon cancer data (cancer and normal), the number of pools in the PS-MMM was set to 2. The first pool corresponded to cancer and the second to normal tissues. Each pool's multimodal mixture was modeled using 14 mixture components with a Gaussian distribution assumption for the continuous part.

5.2.4 MM-HMM-CS (MM-PSM)

The number of HMMs in the mixture was chosen to be 15 for all the time-course datasets. The component HMMs were divided into three groups with five components in each group. The five components in the first group were constrained to have state switch points corresponding to the true or estimated biological stage switch points for the data being modeled. The state switches in the components of the

other two groups were delayed and advanced respectively by one time point relative to the true biological stage switches. This is so that genes that tend to switch stages either later or earlier were accounted for. The number of hidden states in each component HMM was set to 2, 4 and 2 for the angiogenesis, yeast cell cycle and human cell cycle data respectively. Further, each state's emission density was modeled using a mixture density with just one component (univariate Gaussian).

In the case of static data, there is no notion of a lag or advance in switching biological states because there is no time involved. We used 14 mixture components in the MM-PSM model with each component using 2 pools, one for each of the two phenotypes. The pool-specific expression density for each component was modeled using a univariate Gaussian.

5.2.5 Phenotype prediction results on static data

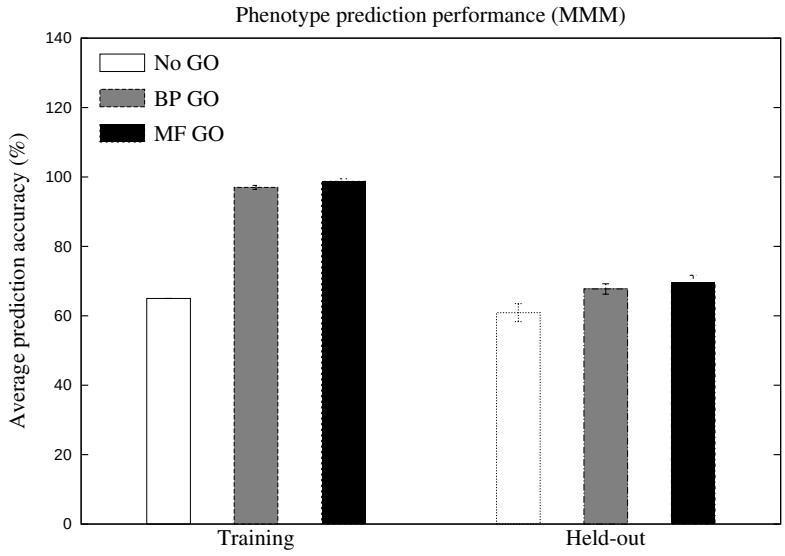
The phenotype prediction results for the Alon colon cancer data are plotted in Fig. 5.1.

5.2.6 Stage prediction results on time-course data

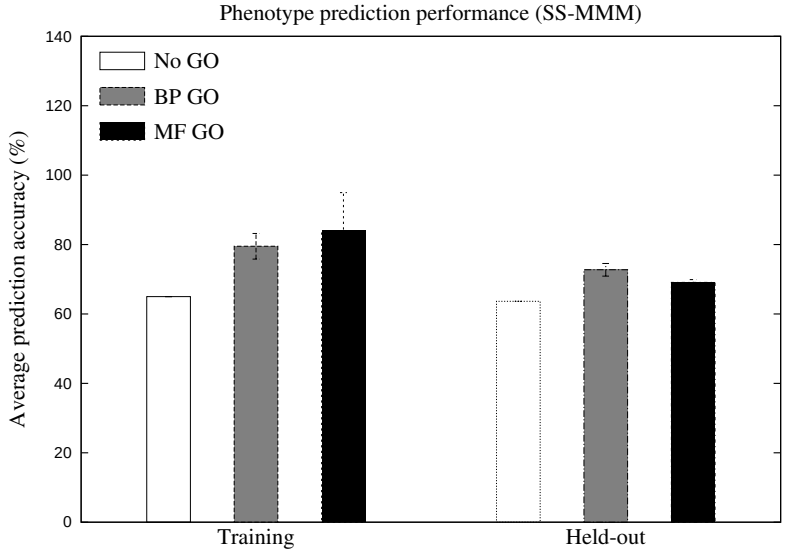
The stage prediction results for the three time-course datasets are shown in Figs. 5.2 through 5.4.

5.3 Estimation of hypothesized biological stages for the angiogenesis data

The angiogenesis data was recorded to study the process of blood vessel growth in a microvascular construct. Blood vessel growth can be hypothesized to involve roughly two stages. During the first stage (*angiogenesis*), relevant microvessel segments relax their normal vessel structure leading to the expansion of the microvasculature via the addition of new vessel segments. Subsequently, the newly formed vessels differentiate into the varied elements of a normal vasculature including arterioles, venules and capillaries finally leading to a mature vascular network (*maturation*) through vessel adaptation. In our data, we do not know beforehand as to where the switch from

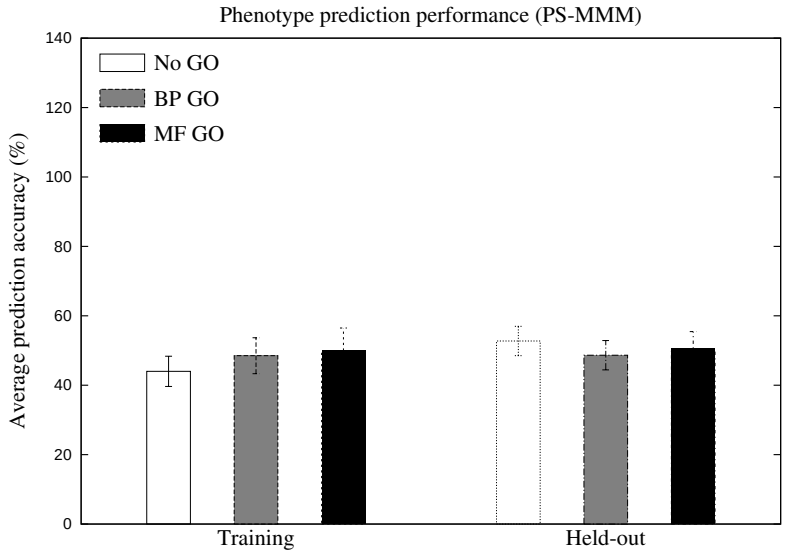


(a) MMM

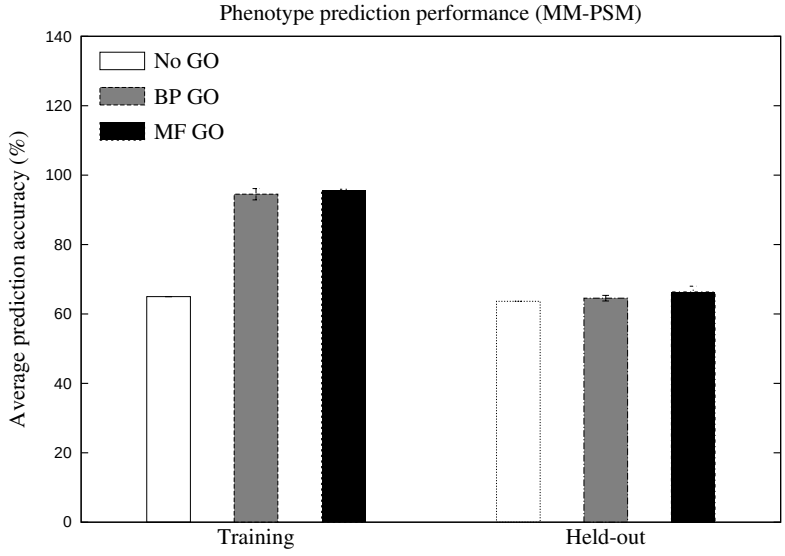


(b) SS-MMM

Figure 5.1: Averaged phenotype prediction accuracies over training and held out samples using the Alon colon cancer dataset. Figs. 5.1a and 5.1b show the results for the sample-independent models (MMM and SS-MMM). These models use a sample-based phenotype prediction scheme. A prediction strategy based on random guess would have a phenotype prediction accuracy of 50%.

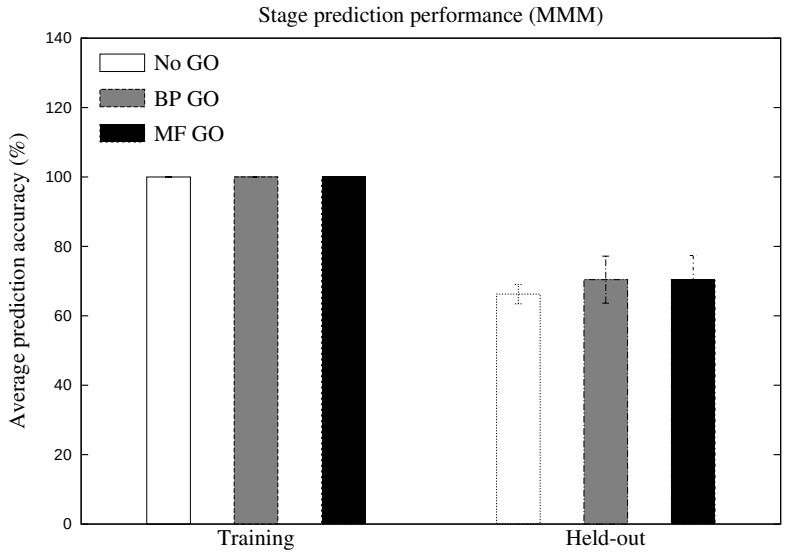


(c) PS-MMM

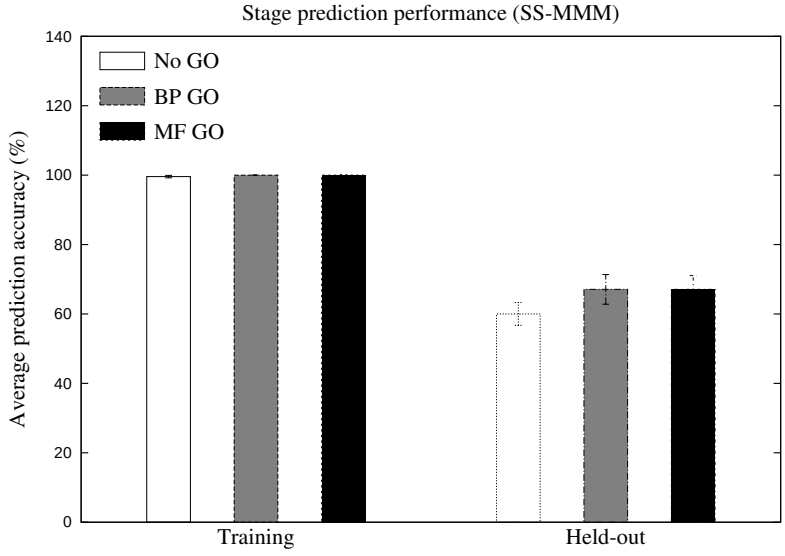


(d) MM-PSM

Figure 5.1: continued. Figs. 5.1c and 5.1d show the results for pooled-sample models (PS-MMM and MM-PSM). These models do phenotype prediction based on pool prediction probabilities. The pool prediction probabilities are computed in a Bayesian framework. A prediction strategy based on random guess would have a prediction accuracy of 50%.

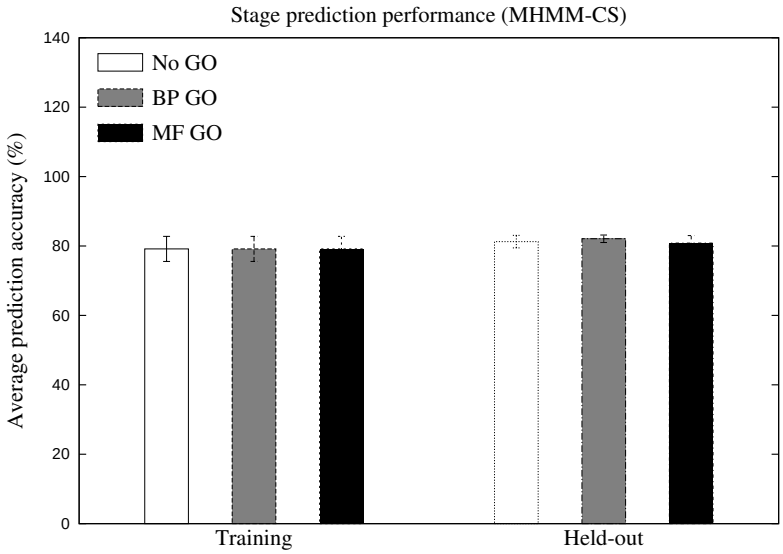


(a) MMM

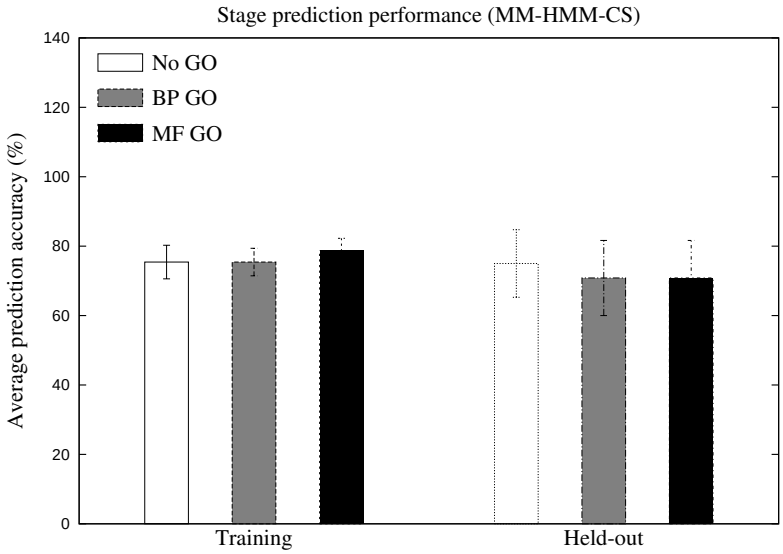


(b) SS-MMM

Figure 5.2: Averaged stage prediction accuracies over training and held out samples using the Hoying angiogenesis dataset. Figs. 5.2a and 5.2b show the results for the sample-independent models (MMM and SS-MMM). These models use a time-based stage prediction scheme. A prediction strategy based on random guess would have a stage prediction accuracy of 50%.

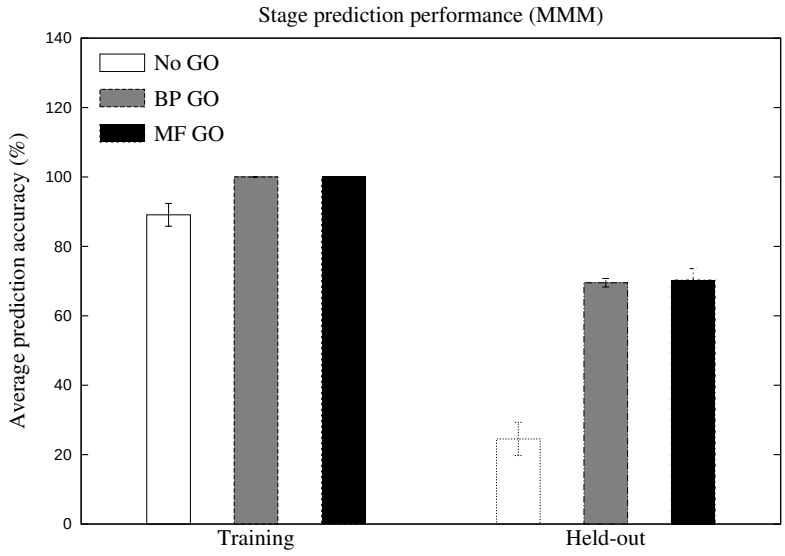


(c) MHMM-CS

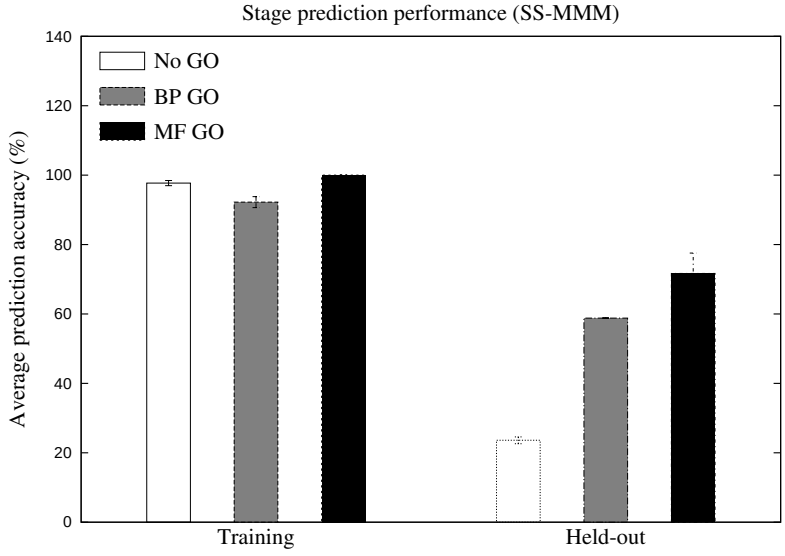


(d) MM-HMM-CS

Figure 5.2: continued. Figs. 5.2c and 5.2d show the results for state-based models (MHMM-CS and MM-HMM-CS). These models do stage prediction based on state prediction probabilities. The state prediction probabilities are computed in a Bayesian framework. As described in section 5.3, it is assumed that stage switch occurs at the **day 7** sample. This switch point is used for both constraining the HMM-based models as well as evaluating all the models. A prediction strategy based on random guess would have a prediction accuracy of 50%.

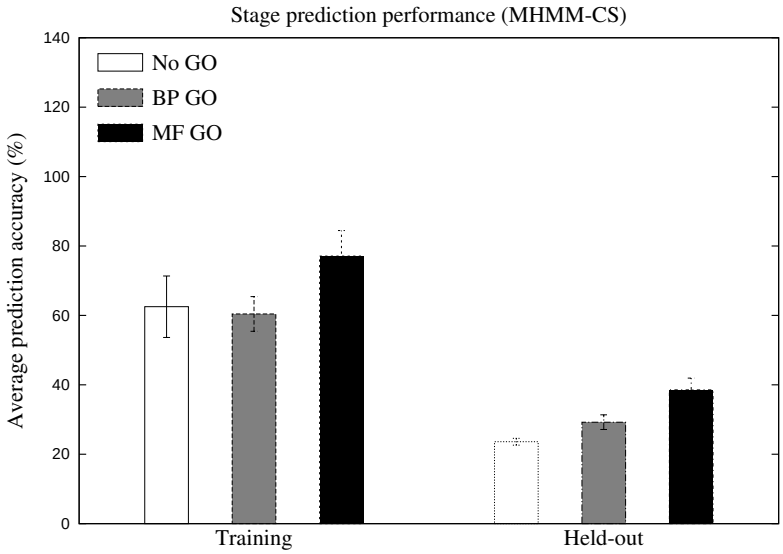


(a) MMM

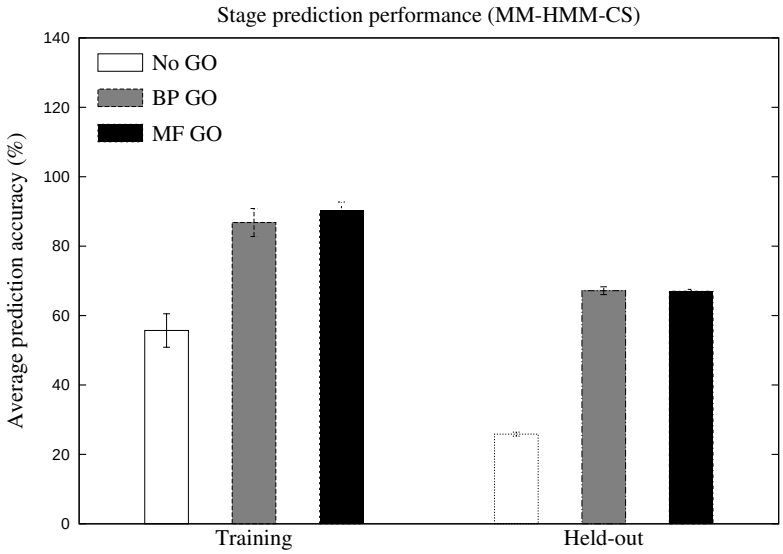


(b) SS-MMM

Figure 5.3: Averaged stage prediction accuracies over training and held out samples using the Cho yeast cell cycle dataset. Figs. 5.3a and 5.3b show the results for the sample independent models (MMM and SS-MMM). These models use a time-based stage prediction scheme. A prediction strategy based on random guess would have a stage prediction accuracy of 25%.

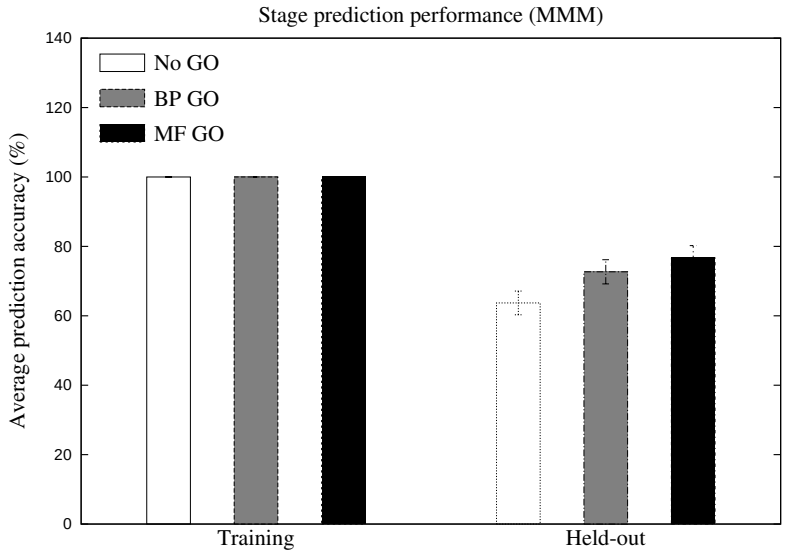


(c) MHMM-CS

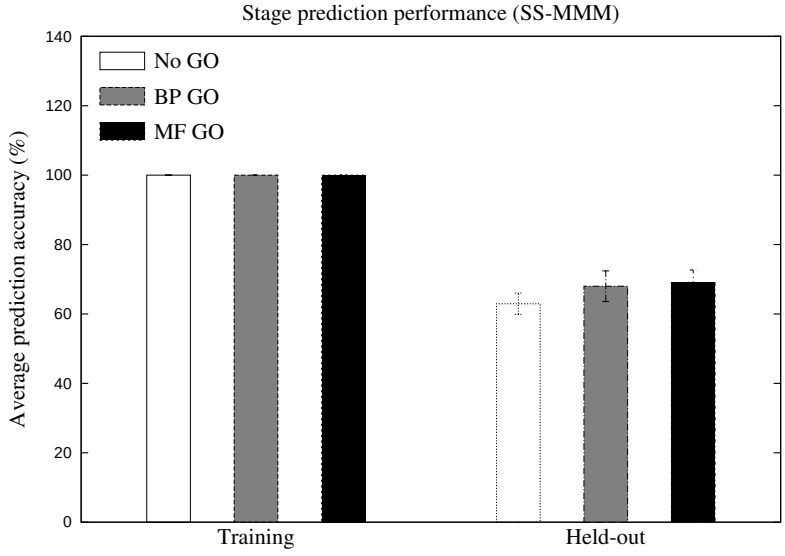


(d) MM-HMM-CS

Figure 5.3: continued. Figs. 5.3c and 5.3d show the results for state-based models (MHMM-CS and MM-HMM-CS). These models do stage prediction based on state prediction probabilities. The state prediction probabilities are computed in a Bayesian framework. The time points at which stage switches occur are listed in Table 5.1. These switch points are used for both constraining the HMM-based models as well as evaluating all the models. A prediction strategy based on random guess would have a prediction accuracy of 25%.

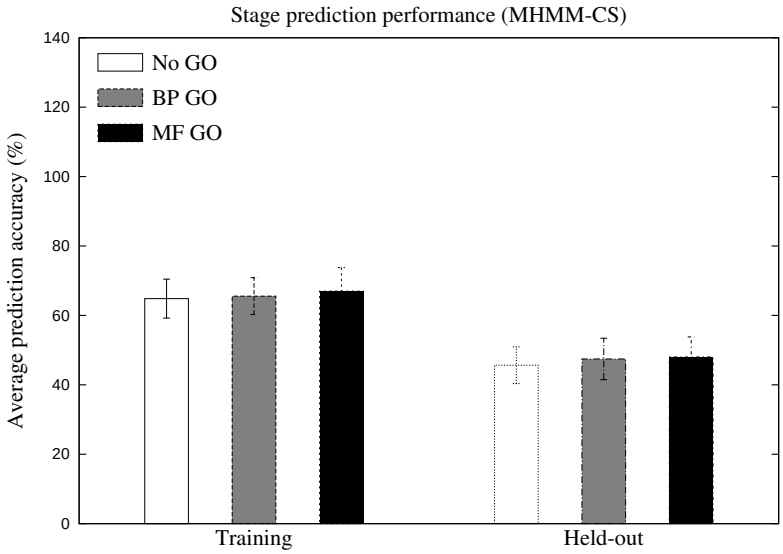


(a) MMM

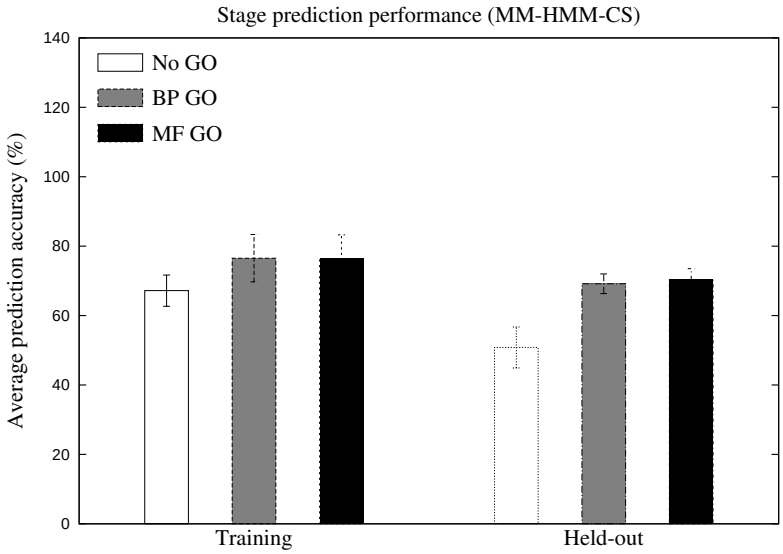


(b) SS-MMM

Figure 5.4: Averaged stage prediction accuracies over training and held out samples using the Whitfield human cell cycle dataset. Figs. 5.4a and 5.4b show the results for the sample independent models (MMM and SS-MMM). These models use a time-based stage prediction scheme. A prediction strategy based on random guess would have a stage prediction accuracy of 50%.



(c) MHMM-CS



(d) MM-HMM-CS

Figure 5.4: continued. Figs. 5.4c and 5.4d show the results for state-based models (MHMM-CS and MM-HMM-CS). These models do stage prediction based on state prediction probabilities. The state prediction probabilities are computed in a Bayesian framework. The time points at which stage switches occur are listed in Table 5.2. These switch points are used for both constraining the HMM-based models as well as evaluating all the models. A prediction strategy based on random guess would have a prediction accuracy of 50%.

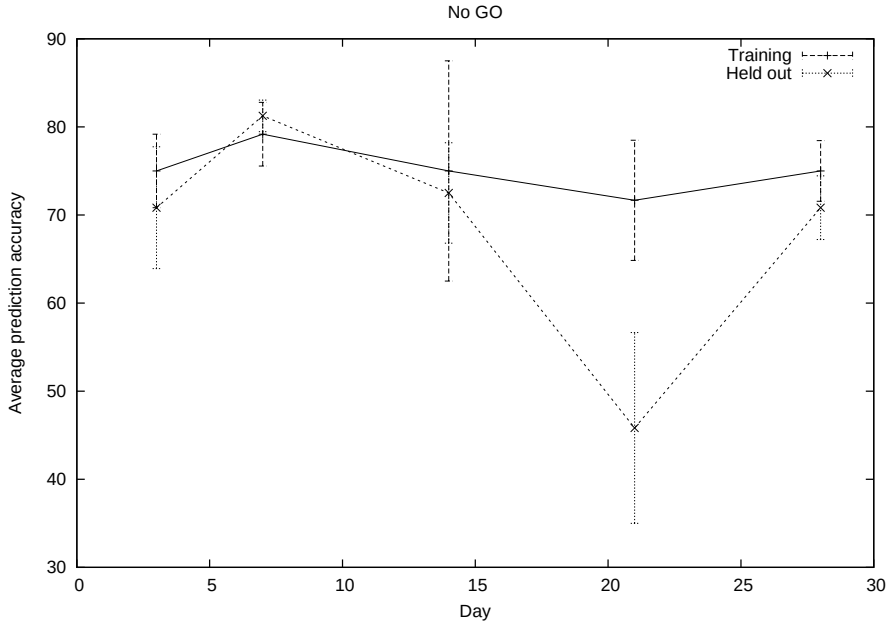
one stage to the other occurs within the recorded time series. We estimate it using two independent methods. The first is based on stage prediction accuracy using the MHMM-CS model. And the second is by doing maximum likelihood inference on the state sequence using Viterbi algorithm on HMM-based model(s) for gene expressions and GO tags.

Using MHMM-CS stage prediction accuracy

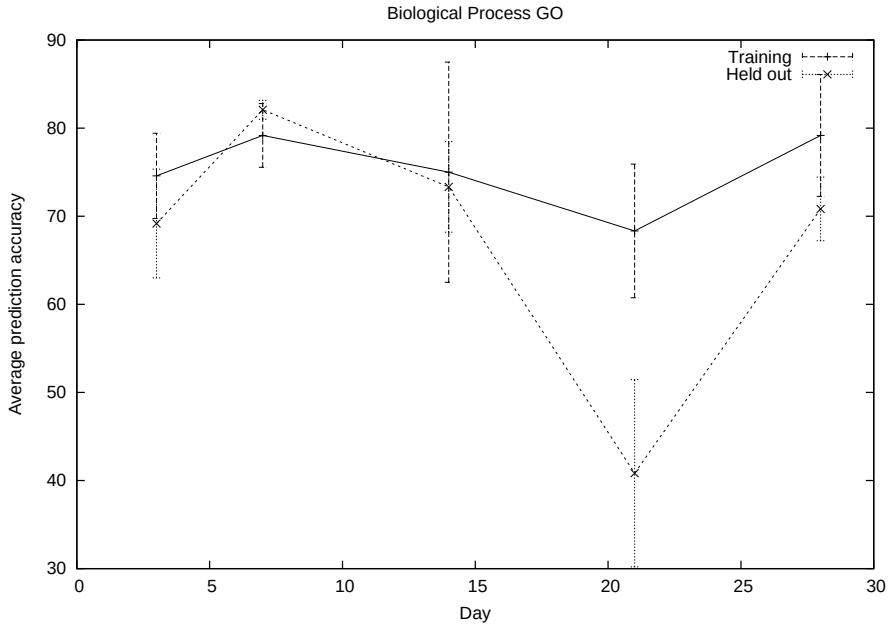
Separate MHMM-CS (Section 4.3.2) models are trained, where each one is constrained by one of all possible switch points (day 3, 7, 14, 21 and 28). Stage prediction is performed for each of these models using the same protocol as in section 5.1, except that the stage switch points for training and evaluation vary over the models. The average prediction accuracies as a function of the switch point for both training and held out data are plotted in figure 5.5. We hypothesize that the switch point leading to the most accurate prediction on held out samples is the best estimate (day 7) of the true switch between the two stages of blood vessel growth. Interestingly, the same switch point gives the best accuracy for the training samples also regardless of whether the models use GO tags or not.

Using maximum likelihood state sequence inference

Estimation of the switch point based on prediction accuracy over held-out data can be thought of as a brute force method where all possible switch points are tried. This is feasible if the number of stages and the total number of time points in the data set is small. With a moderately large number either for the number of stages or for the number of time points, the number of combinations to try becomes exhaustive. To address this issue, we use the HMM framework for estimating the state switch point where it is not known a priori (see Section 4.4.3). Specifically, we train HMM-based multimodal models without imposing the switch constraint as opposed to the previous models (Chapter 4) and infer the mostly likely switch point based on the maximum likelihood (ML) state sequence.

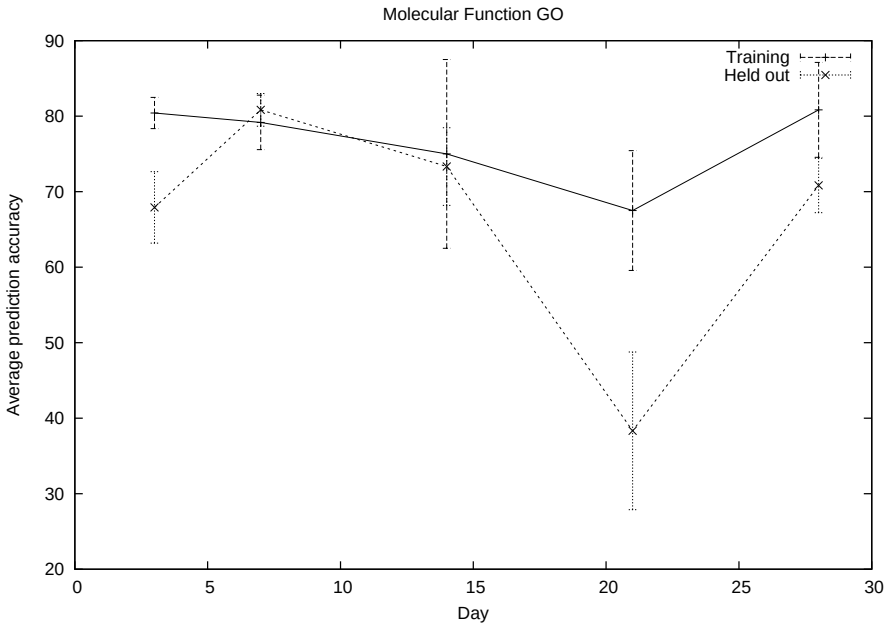


(a)



(b)

Figure 5.5: Estimation of the switch point between the angiogenesis and maturation stages for the Hoying dataset. MHMM-CS models with and without GO tags and each with a different switch point constraint are trained and evaluated on stage prediction. All possible switch points (day 3, 7, 14, 21 and 28) are tried and the average prediction accuracy as a function of the switch point is plotted. The two plots correspond to models without using GO tags and using tags from the Biological Process or Molecular Function GO hierarchies. Based on average stage prediction accuracy over held-out data, day 7 is the optimal estimate of the time of stage switch.



(c)

Figure 5.5: Continued. The above plot corresponds to the model using GO tags from the Molecular Function GO hierarchy. Again, day 7 is the optimal estimate of the time of stage switch.

The ML state sequence is estimated using the Viterbi algorithm for the single component multimodal HMM and a brute force search for the mixture of HMMs. Note that a brute force search for ML state sequence is feasible here because we have only 6 time points and 2 states leading to a total of 64 possibilities. For a higher number of time points and states, this method might get expensive. However it is still more feasible than the held-out prediction method because the model is trained only once and not for each possible state sequence.

Four different sets of data available for the angiogenesis experiments (2 runs, 2 gene sets: see Section 5.1) were used. The two models were trained using each set as the input data and the ML state sequence was inferred on the same set. The HMM(s) were constrained to start in a nominal first state and proceed in the forward direction, i.e., left-to-right, with the total number of states being two. The two states correspond to the two hypothesized stages of blood vessel growth. The number of clusters in the multimodal HMM was chosen to be 14. For the mixture of HMMs, the number of component HMMs was set to 14 with each state of each HMM having a single cluster. This kept the number of groupings the same in the two models. The one-cluster-per-state assumption for the mixture of HMMs leads to each gene contributing to a single cluster corresponding to a component HMM.

Training was done using the EM algorithm. To address the local maximum likelihood problem of the EM algorithm, 10 different random initial points were tried for each set. This resulted in 40 different state sequence estimates over the 6 time points. The state for each time point was chosen to be the one with the most repeats over the 40 trials (majority vote). The resulting state sequence was found to be (1, 1, 2, 2, 2, 2) corresponding to the time points (day 0, day 3, day 7, day 14, day 21, day 28). This was true with both the single multimodal HMM as well as the mixture of HMMs model. So the switch point at day 7 agrees with the previous estimate using held-out stage prediction.

5.4 Comparison between models

The above choice of the number of clusters for the various models leads to grouping of the genes' measurements (expression levels and GO tags) into almost the same number (approx. 14) of groups. This is regardless of whether grouping is done considering expression profiles across the entire sample set (MMM, MM-HMM-CS and MM-PSM), individual time points (SS-MMM) or pooling expression levels across a subset of samples (MHMM-CS and PS-MMM). Our notion is that clustering is a mere discretization of the joint expression and GO tag space and subsequent smoothing of the empirical distribution implied by the training data. The hope is that the smoothing leads to better generalizability on any inference task using the learnt model. We think that keeping the number of clusters almost the same in all the groupings enables a fair comparison across models that are inherently quite different in how they group the data.

Although the number of groups is almost the same for all the models, the actual number of parameters to achieve the same differs across models. Since the MMM and SS-MMM are sample based models, they learn the distribution in the joint space using a larger number of parameters compared to the state-based models (pooled-sample and HMM-based). This is because they learn a mean and variance for each cluster at each time point. However the state-based models learn a mean and variance for each cluster at each state, which spans a subset of samples or time points. Therefore we expect a better performance of MMM and SS-MMM models on training data. The results confirm this on all our datasets. However the performance on held out data depends on how well the learnt distribution generalizes on novel data on the task of stage prediction.

We argue that all the proposed models learn a good deal about the underlying distribution. This is because the stage prediction performance on held out data is almost always above chance, **except for the MMM and MHMM-CS models on the yeast cell cycle data and the PS-MMM model on colon cancer data**. Further, the models that use GO tags almost always perform better than

their corresponding models that do not make use of GO tags. The only exception is the MM-HMM-CS model on the angiogenesis data. Note that this data has only very few samples (6) and so an anomalous result is quite likely.

The above is clear evidence that GO tags provide independent and useful information that can augment measured expression levels of genes. In addition to providing independent information, GO tags also serve as place-holders for the genes. Each of the proposed models has a generative mechanism for the expression levels of genes through the parameters of the Gaussians involved. However they do not distinguish between the genes' identities. In other words, if two genes' expression levels were interchanged in a microarray sample, then the resulting sample would still have the same likelihood for a particular biological state when GO tags are not used. But this is not the case when GO tags are used because more often than not the tags would be different for the two different genes. This serves to distinguish between the two scenarios thereby giving more discriminative power to the predictive mechanism.

The merit to using models that pool data over the ones that assume sample independence, on the task of learning about biological stages, is a function of the dataset. The samples in the colon cancer data come from different patients even within the same phenotype. So pooling does not seem to help in this case perhaps due to the lack of correlation resulting from the independent nature of data collection. On the angiogenesis data, the HMM-based models seem to help the task of stage prediction. But this is not the case with the yeast and human cell cycle data. Note that we are using constrained state-based models to capture time correlations. Some of these constraints, e.g. all the genes within a group making phase transitions together, might agree more with some datasets than others. Models without such constraints, for example letting all the genes switch stages independently, are easy to construct but they have a tendency to fit noise more than the underlying biological phenomena. Due to our lack of a complete understanding of such issues with microarray datasets, we defer such considerations to a future work.

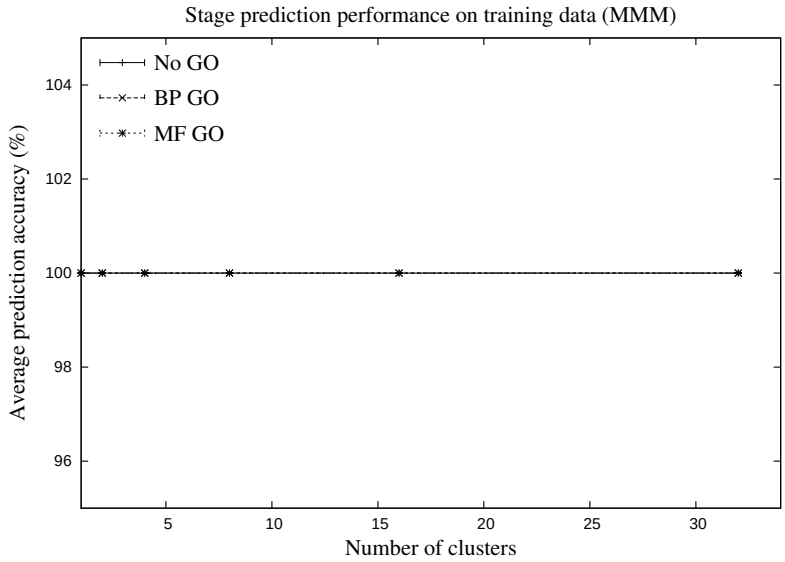
5.5 Varying the number of clusters

The experiments in Sections 5.2.5 and 5.2.6 used a fixed number of clusters (14). The idea was to keep the number of groupings almost the same in the experiments to make a fair comparison between the models with and without using GO tags. Different models might perform optimally using different number of groupings. It is interesting to study the prediction behavior as a function of the number of clusters in the model. We perform this study for the simplest multimodal mixture model (MMM) by varying the number of clusters from 1 to 32 exponentially, i.e. doubling each time. The stage prediction results on the three time course datasets for training as well as held-out samples are plotted in Figs. 5.6-5.8. The average stage prediction accuracy on training samples is almost 100% over all the number of clusters on the angiogenesis and the human cell cycle data. It increases with the number of clusters in the case of yeast cell cycle data. This is not surprising given that the resulting increase in the number of model parameters leads to a better fit to the training data.

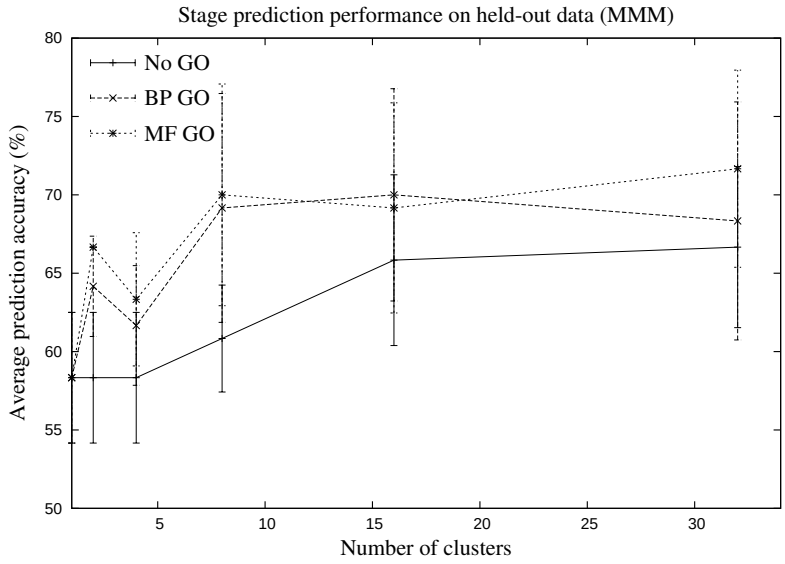
The held-out prediction behavior on all the datasets improves with increasing number of clusters when GO tags are used. This is not necessarily true with models not using GO tags. This implies that when more patterns (clusters) of gene behavior are allowed, GO tags can help delineate these patterns in a way useful for phenotype prediction. From Fig. 5.7, it can be seen that by increasing the number of clusters from 1 to 16 the average held-out prediction accuracy improves from around 30% to 70% when GO tags are used with the yeast cell cycle data. A similar improvement from around 63% to around 75% is observed with the human cell cycle data in Fig. 5.8. Note that with models not using GO tags in the two cases the average prediction accuracy remains almost the same over the range of number of clusters. It ranges between 15% to 30% in the case of yeast cell cycle data and between 60% to 65% in the case of human cell cycle data.

The above improvement using GO tags relative to not using them is more significant in the two cell cycle datasets than the angiogenesis dataset. This is perhaps due to two possible reasons. One is that the number of samples in the Hoying an-

giogenesis data is small (6 per run, see Section 5.1) leading to estimates of standard errors that are relatively large. The second is the quality of GO tags for the genes chosen for the experiment. For the two cell cycle datasets, the chosen genes have been very well studied by other research groups and are known to be involved in the process of cell cycle regulation. Yeast and human genes have been thoroughly studied and annotated. The genes used for the Hoying data are chosen based on differential expression. More work is needed to ascertain that the selected genes are all involved in the blood vessel growth process and have reliable GO annotations.

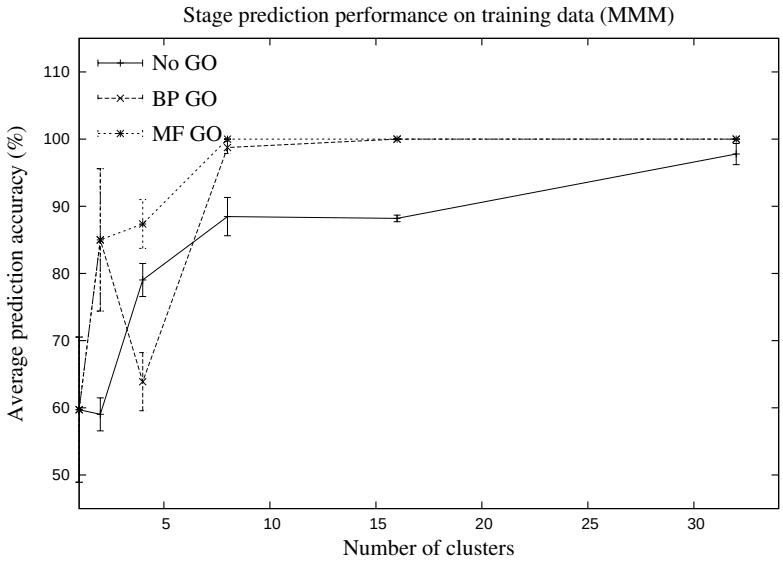


(a) MMM(train)

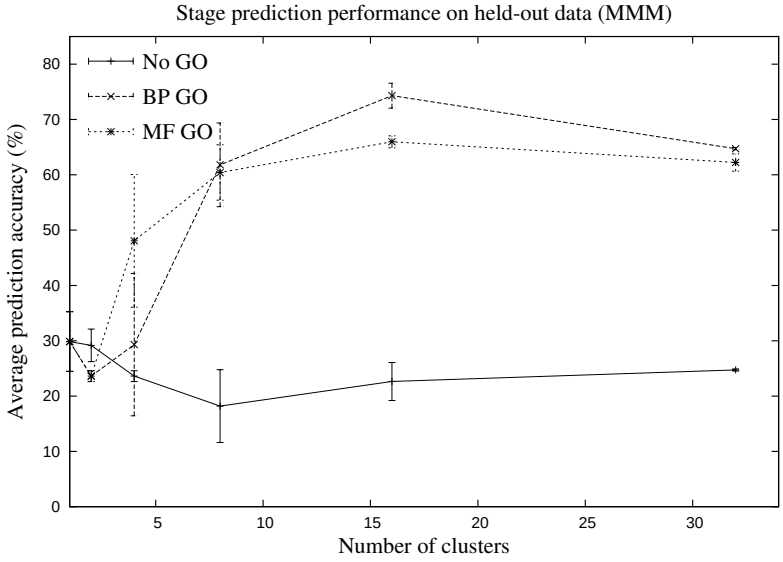


(b) MMM(held-out)

Figure 5.6: Averaged stage prediction accuracies over training and held-out samples of the Hoying angiogenesis dataset as a function of the number of clusters used in the MMM model. Note that all the models, with or without using GO tags, have a 100% accuracy on training samples for all cluster numbers. Therefore the curves overlap in the corresponding plot above. A prediction strategy based on random guess would have a stage prediction accuracy of 50%.

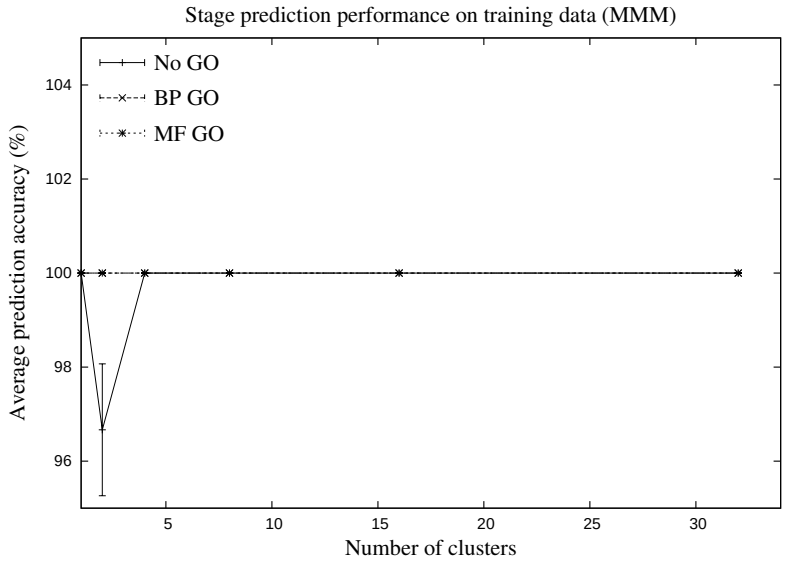


(a) MMM(train)

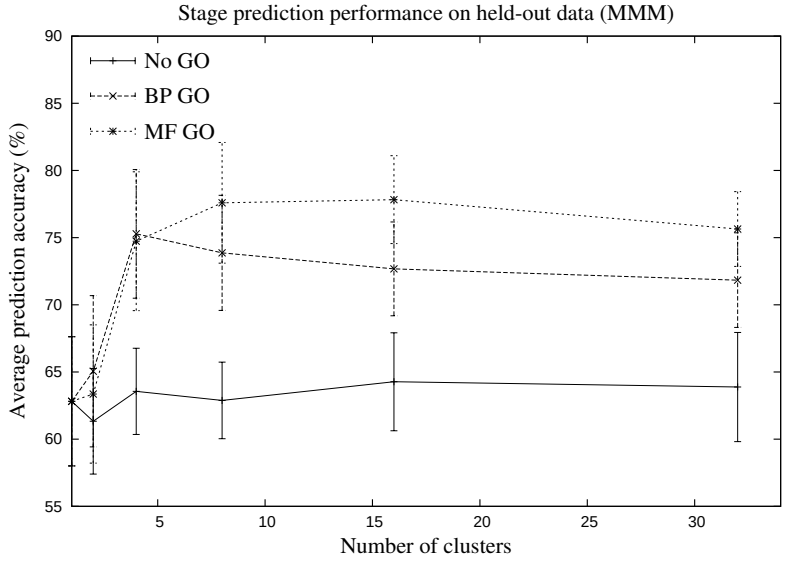


(b) MMM(held-out)

Figure 5.7: Averaged stage prediction accuracies over training and held-out samples of the Cho yeast cell cycle dataset as a function of the number of clusters used in the MMM model. A prediction strategy based on random guess would have a stage prediction accuracy of 25%.



(a) MMM(train)



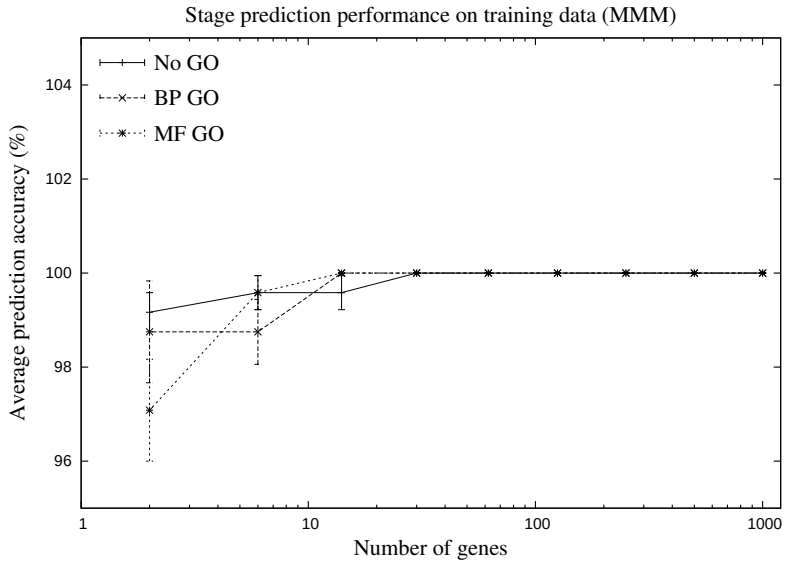
(b) MMM(held-out)

Figure 5.8: Averaged stage prediction accuracies over training and held-out samples of the Whitfield human cell cycle dataset as a function of the number of clusters used in the MMM model. Note that all the models, with or without using GO tags, have a 100% accuracy on training samples for all cluster numbers. Therefore the curves overlap in the corresponding plot above. A prediction strategy based on random guess would have a stage prediction accuracy of 50%.

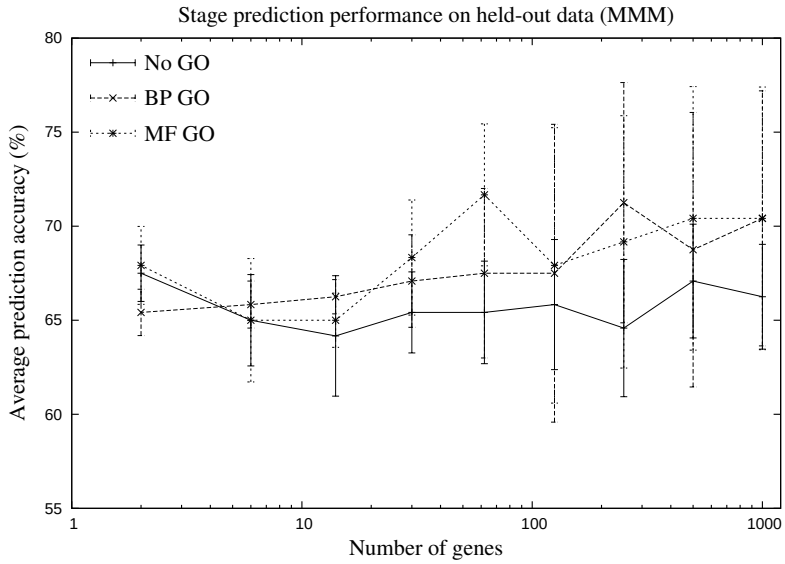
5.6 Varying the number of genes

As described in Section 4.2, the generative models yield a systematic way to select a subset of genes from the entire set. This can be done by choosing a certain number of genes closest to the cluster centers in the learnt models. Since the underlying distributions are Gaussian, the distance to cluster centers is measured in terms of the Mahalanobis distance (Bishop, 2006) . The reduced set of genes can be used to design smaller microarrays for screening or diagnostic purposes, making the technology more affordable. This makes sense only if the reduced set can serve as a reliable indicator of the underlying biological state. In order to quantify the usefulness of our gene selection procedure, we study the stage prediction performance using different numbers of genes for prediction. This is done using the MMM model on the three time course datasets. Note that all available genes in the dataset are used for training but only a given number of them are selected based on their closeness to the centers of each of the clusters. The number of clusters is fixed for these experiments (14).

Starting from all the genes, the number of closest genes to each cluster is halved successively up to an order of one or two closest genes. The stage prediction behavior as a function of these different number of selected genes is plotted in Figs. 5.9-5.11. The results confirm that there is a good potential in reducing the number of genes without affecting the stage prediction performance too much using our gene selection procedure.

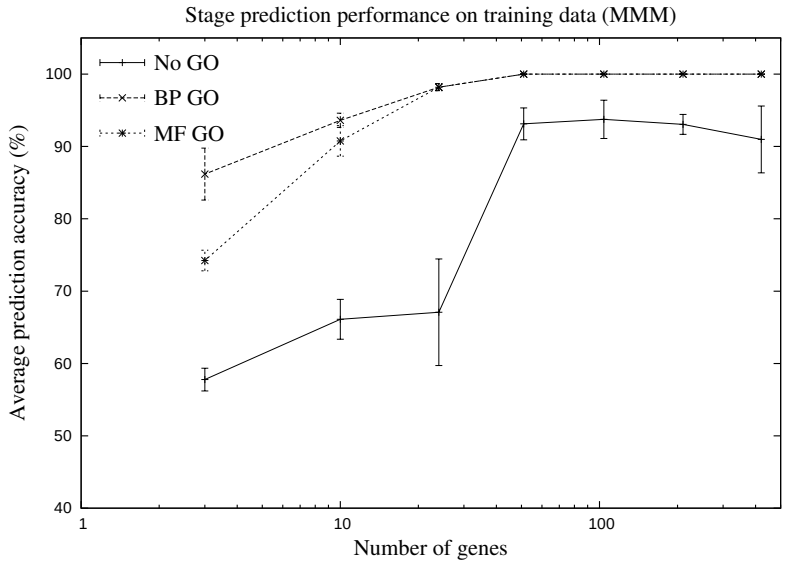


(a) MMM(train)

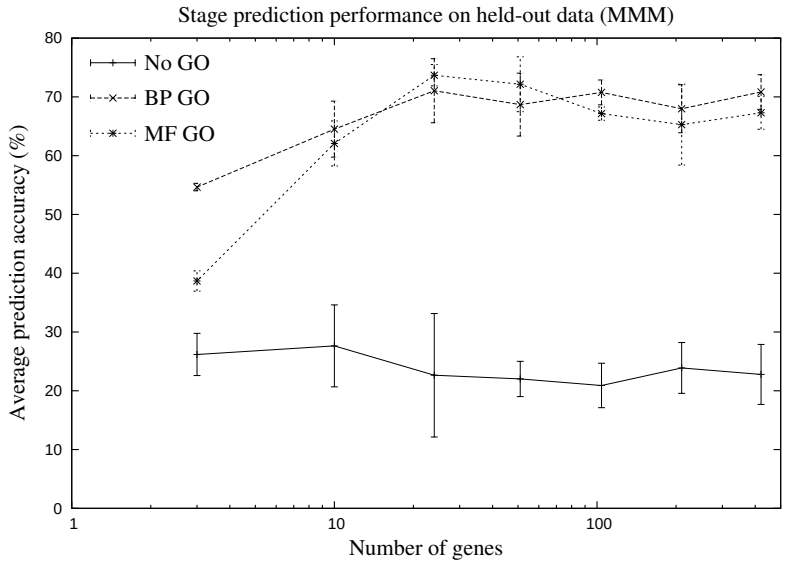


(b) MMM(held-out)

Figure 5.9: Averaged stage prediction accuracies over training and held-out samples of the Hoying angiogenesis dataset as a function of the number of genes used for prediction in the MMM model. Model training is performed using all the available genes but prediction uses only a given number of genes (X axis) that are closest to the cluster means in terms of Mahalanobis distance. A prediction strategy based on random guess would have a stage prediction accuracy of 50%.

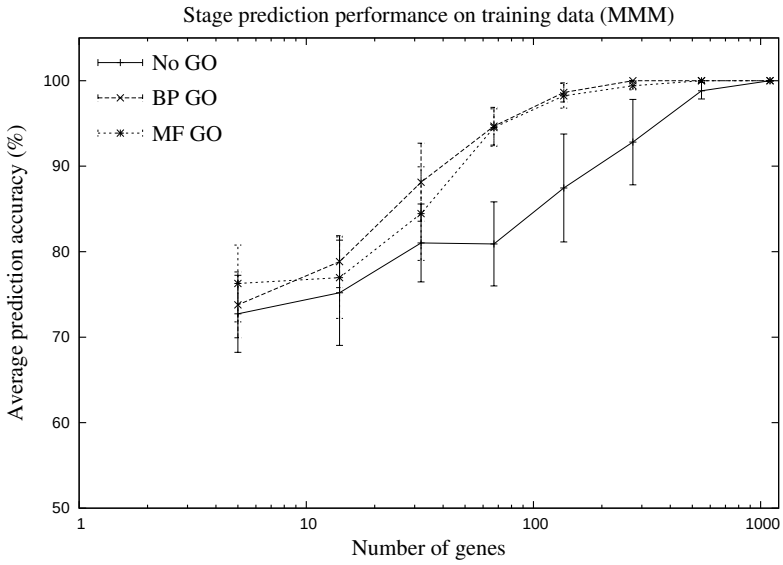


(a) MMM(train)

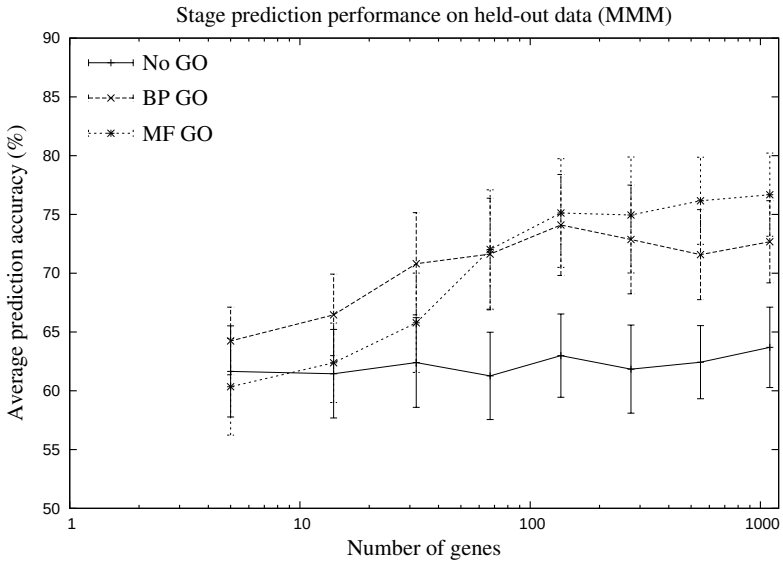


(b) MMM(held-out)

Figure 5.10: Averaged stage prediction accuracies over training and held-out samples of the Cho yeast cell cycle dataset as a function of the number of genes used for prediction in the MMM model. Model training is performed using all the available genes but prediction uses only a given number of genes (X axis) that are closest to the cluster means in terms of Mahalanobis distance. A prediction strategy based on random guess would have a stage prediction accuracy of 25%.



(a) MMM(train)



(b) MMM(held-out)

Figure 5.11: Averaged stage prediction accuracies over training and held-out samples of the Whitfield human cell cycle dataset as a function of the number of genes used for prediction in the MMM model. Model training is performed using all the available genes but prediction uses only a given number of genes (X axis) that are closest to the cluster means in terms of Mahalanobis distance. A prediction strategy based on random guess would have a stage prediction accuracy of 50%.

CHAPTER 6

DISCUSSION AND CONCLUSIONS

Microarrays enable highly parallel measurement of gene expression levels within tissues and cell lines. At the same time, the technology poses a number of data analysis challenges. Useful information in raw microarray data is mired in noise arising from various sources, which are both biological and non-biological in nature. A main source of non-biological noise is the systematic experimental noise introduced into the microarray measurements due to experimental differences between multiple measurements using the same type of array. These arise from differing sample mRNA concentrations, dye labeling efficiencies, dye bias or even the gain settings of the imaging equipment. Since the technology is expensive there are usually very few samples recorded within a particular study. From a statistical perspective, this leads to data with very large feature dimensionality relative to the number of available samples. A few problems with such data include the curse of dimensionality and the large number of false positives from statistical tests such as the t-test. Usually microarray experiments are aimed at finding a set of genes that can be used as markers for certain biological conditions, e.g. disease genes. A brute force search for the best set of genes within a larger set measured by a microarray is combinatorially exhaustive. This calls for some kind of feature dimensionality reduction before the search.

In this work we have tried to address most of the above problems. We first consider the problem of accounting for systematic experimental noise in the microarray data. Motivated by the fact that these systematic effects introduce spurious correlations between gene expression profiles, we propose a new normalization method based on minimizing these spurious correlations. This method (MIN-SS-CORR) is based on minimizing the sum of squared correlation coefficients between all pairs of genes within a set having the same systematic effect. A gradient descent procedure

in the space of effects leads to an estimate of the offsets that can be used to normalize the data. Most normalization methods that have been proposed before reduce these spurious correlations but only indirectly.

We evaluate a few previously proposed normalization methods (Geometric, Quantile, Rank and δ-sequences) along with the MIN-SS-CORR method on the phenotype prediction task. Microarray datasets usually have a number of gene expression measurement samples each one taken under a particular biological condition representing a phenotype, e.g. cancer, normal, or one of the stages (S, G1, M, G2) of the mitotic cell-cycle. There is a genotype-phenotype relationship in such data, which can be quantified by the mutual information or correlation structure between the phenotypic classes and the corresponding gene expression levels. It is our notion that systematic experimental effects degrade some of this correlation structure and an ideal normalization method should either preserve or enhance this genotype-phenotype relationship. We evaluate the effectiveness of the various normalization methods in preserving, enhancing or degrading this relationship in the data after transforming the raw data using the particular method.

Two evaluation approaches are considered for comparing the various normalization methods. The first one is based on class label hypothesis testing for seeking out the support for phenotypic classes in the data normalized using various methods in comparison to the raw data. The level of statistically significant support for the phenotypic classes relative to randomly assigned classes for samples quantifies the support for the true classes. The second one is based on learning phenotype class boundaries in the gene expression space followed by phenotype prediction on held-out samples in a leave-one-out (LOO) framework. LOO prediction accuracy and confidence provide quantitative measures for comparing the various normalization methods. The classifiers used are based on learning a linear-discriminant function in the gene expression space with an inherent gene selection mechanism.

Both of the above evaluation approaches lead to the same conclusions on the relative performances of the different normalization methods on two datasets: Alon colon cancer data and Hoying angiogenesis data. Three normalization methods con-

sistently help enhance the phenotype-genotype relationship in the raw data. These are the Quantile, Geometric and MIN-SS-CORR methods with Quantile normalization having the best overall performance. δ-sequences are helpful with the angiogenesis data but not with the colon cancer data. Rank normalization seems to degrade the genotype-phenotype correlation structure present in the raw data using a true LOO gene selection and evaluation framework. We think that this has to do with the quantization of gene expression values that results from replacing continuous values with discrete ranks. To our knowledge, this is a first effort in performing such a quantitative evaluation on real microarray datasets using a held-out phenotype prediction framework. More work on these lines is helpful in quantifying the effects of normalization methods on the genotype-phenotype relationship of raw microarray data. This might entail using different classifiers, gene selection mechanisms and other normalization methods on different microarray datasets.

The second contribution of this work is in using a multimodal probabilistic generative framework for modeling a number of microarray datasets. Multimodal models enable us to incorporate an independent source of information in the form of Gene Ontology (GO) tags for analyzing microarray gene expression data. These are terms from an evolving controlled vocabulary to describe genes and gene products. The most common approach to use these terms is to examine gene clusters or candidate gene groups by doing an enrichment analysis. We deviate from this approach and use these terms in clustering genes in a continuous-discrete space of gene expression levels and GO terms. This probabilistic clustering framework addresses a number of problems. It achieves dimensionality reduction by grouping genes into clusters thereby reducing the search space for gene markers by working with clusters rather than individual genes. Clustering also leads to a systematic gene selection method by considering only genes that are closest to the cluster centers. Although we use Gaussians in this work, different sources of noise can be easily modeled with appropriate parametric probability distributions in the generative framework. A number of inferences including phenotype or biological state inference can be done employing a Bayesian framework.

We propose four generic multimodal models to cluster genes differently based on their behavior over the entire microarray sample set, phenotypic groups or individual samples within the set. These are the Multimodal Mixture Model (MMM), Sample-Specific Multimodal Mixture Model (SS-MMM), Pooled Sample Multimodal Mixture Model (PS-MMM) and a Multimodal Mixture of Pooled Sample Models (MM-PSM). In order to account for dependencies and infer unknown stage switches in time-course datasets, two HMM based models are also proposed: Multimodal Hidden Markov Model with Constrained Switches (MHMM-CS) and Multimodal Mixture of Hidden Markov Models with Constrained Switches (MM-HMM-CS). The above models are applied to three time-course datasets (Hoying angiogenesis data, Cho yeast cell cycle data, Whitfield human cell cycle data) and one static dataset (Alon colon cancer data). Models trained on a set of samples are used to predict phenotypes for new samples in a Bayesian maximum likelihood framework. The HMM based models are also used to do maximum likelihood state sequence inference for data with unknown biological state switches (Hoying angiogenesis data).

Our results based on phenotype or biological state prediction for various datasets suggest that GO tags provide useful information to obtain better gene clusters for the task of prediction. However the GO tags themselves are continuously evolving in that there are a number of wrong or missing annotations for genes of model organisms. We observed that for organisms such as yeast and humans whose genes have been thoroughly studied and annotated, the use of GO tags in conjunction with gene expression data leads to a significant improvement in phenotype prediction performance. We did not observe such an improvement with mouse (Hoying) data. There are two possible reasons for this: small size of the dataset or the relatively poor quality of GO annotations. In some sense, our approach also provides a way to assess the quality of annotations for a model organism based on prediction performance. The proposed models do not assume anything about how the gene expression data were measured. Expression data measured using other technologies is easily handled within our modeling framework.

APPENDIX A

OPTIMIZING NORMALIZATION FOR CLASSIFICATION

The criteria for evaluation of various normalization methods is the class prediction accuracy on novel data, i.e., the leave-one-out samples. It is then natural to ask if a normalization method can be designed to maximize this objective. We attempt to do this in the maximum margin framework of linear SVMs. This is because a SVM maximizes the margin between the class boundary and the nearest training sample, thereby providing a convenient objective function to represent classifier generalizability (Vapnik, 1998; Bishop, 2006) . By introducing variables that account for systematic experimental variations of microarray gene expression data, it is possible to optimize them with classifier generalization as the objective.

As an example, we consider a simple and widely used model to account for the systematic experimental effects introduced into microarray gene expression measurements. This is the array-specific multiplicative model. Both geometric and MIN-SS-CORRCF normalization methods discussed above are based on this model. The multiplicative effect is equivalent to a global additive effect in the log gene expression measurement space. If $\boldsymbol{x}_i$ is the vector of measured gene expression values (log transformed) on the ith array, then $(\boldsymbol{x}_i - \boldsymbol{o}_i)$ represents the true gene expression values for a certain offset vector $\boldsymbol{o}_i$. Note that $\boldsymbol{o}_i = o_i\mathbf{1}$, where o_i is the scalar array-specific offset and $\mathbf{1}$ is the vector of all ones. We estimate the offsets o_i by maximizing the margin of a linear SVM using the offset points $(\boldsymbol{x}_i - \boldsymbol{o}_i)$ as training data points.

More formally, we work with a two class linear SVM assuming separable data. The results for the non-separable case are the same. Let $\boldsymbol{x}_n$ represent the measured log gene expression vector of array $n \in \{1, 2, \ldots N\}$ with the associated phenotypic class label $y_n \in \{-1, +1\}$. Using the offset data points $(\boldsymbol{x}_n - \boldsymbol{o}_n)$, the linear SVM optimization function takes the form

$$L(\boldsymbol{w}\,, b, \boldsymbol{o}\,, \boldsymbol{a}\,) = \frac{1}{2}\boldsymbol{w}^{\,T}\boldsymbol{w} \; - \sum_n a_n \left[y_n \left\{ (\boldsymbol{x}_{\,n} - \boldsymbol{o}_{\,n})^T \boldsymbol{w} \; + b \right\} - 1 \right] \tag{A.1}$$

The above Lagrangian is minimized with respect to the weight vector $\boldsymbol{w}$, bias b, offsets o_n and maximized with respect to the Lagrange multipliers a_n. The Lagrange multipliers are introduced to account for the constraints $y_n \left\{ (\boldsymbol{x}_{\,n} - \boldsymbol{o}_{\,n})^T \boldsymbol{w} \; + b \right\} \geq 1$ in the SVM formulation. Due to the negative sign in the above Lagrangian, the multipliers a_n must be non-negative

$$a_n \geq 0$$

The Wolfe dual of the above primal formulation is obtained by setting the derivatives of L with respect to $\boldsymbol{w}$, b and $\boldsymbol{o}$ to zero and substituting the results. Doing this for the $\boldsymbol{w}$ and b parameters leads to the following equations

$$\boldsymbol{w} \; = \sum_n a_n y_n (\boldsymbol{x}_{\,n} - \boldsymbol{o}_{\,n}) \tag{A.2}$$

$$\sum_n a_n y_n = 0 \tag{A.3}$$

Setting the derivative of L with respect to the offsets o_i to zero leads to an additional constraint that the sum of weights be equal to zero. That is

$$\boldsymbol{w}^{\,T} \mathbf{1} \; = 0 \tag{A.4}$$

Substituting the expression for optimum $\boldsymbol{w}$ into the above

$$\sum_n a_n y_n (\boldsymbol{x}_{\,n} - \boldsymbol{o}_{\,n})^T \mathbf{1} \; = 0$$

$$\sum_n a_n y_n (\boldsymbol{x}_{\,n}^{\,T} \mathbf{1} \; - o_n \mathbf{1}^{\,T} \mathbf{1}\,) = 0$$

$$\sum_n a_n y_n \left(\sum_m \boldsymbol{x}_{\,nm} - M o_n \right) = 0$$

where M is the dimensionality of the data points, which is equal to the total number of genes on a single microarray and $\boldsymbol{x}_{nm}$ represents the mth gene expression measurement on the nth array. The above expression leads to the following optimum values for the offsets o_n

$$o_n = \frac{1}{M}\sum_m \boldsymbol{x}_{nm} + K \tag{A.5}$$

where K is any arbitrary constant. Substituting the optimal values for the weight vector and bias leads to the following objective in the dual space

$$\tilde{L}(\boldsymbol{a}) = -\frac{1}{2}\sum_{l,m} a_l a_m y_l y_m (\boldsymbol{x}_l - \boldsymbol{o}_l)^T (\boldsymbol{x}_m - \boldsymbol{o}_m) + \sum_n a_n \tag{A.6}$$

with the constraints

$$a_n \geq 0$$
$$\sum_n a_n y_n = 0$$

This is the standard linear-SVM quadratic optimization problem with linear constraints except that the offsets o_n are substituted from equation A.5.

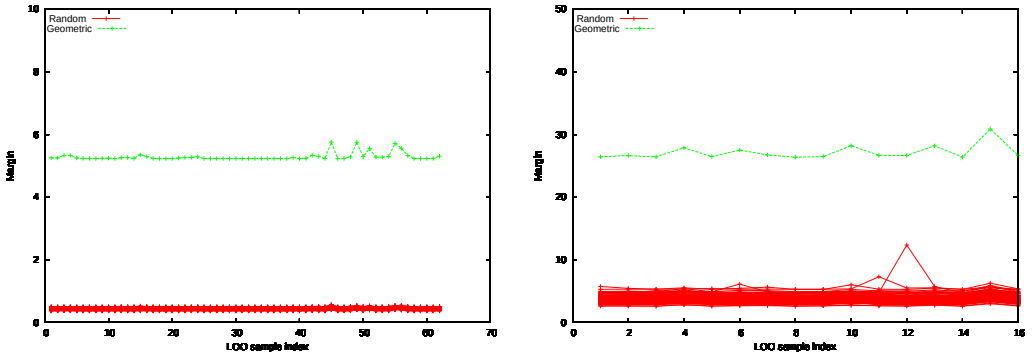
It is clear from equation A.5 that, assuming an array specific offset model, the offsets that maximize the margin between the adjusted data points are the same as the ones obtained from geometric normalization. This gives an interesting alternate way to look at geometric normalization.

Experiments

In order to validate the maximum margin property of geometric normalization, we performed experiments using the two microarray datasets considered before. Each dataset was normalized using the corresponding offsets obtained by geometric normalization, which are the array-specific means of gene expression measurements in the log space. In addition, 100 other array-specific random offsets were generated

as (pseudo) adjustments for experimental effects in each dataset. The array-specific offsets were drawn independently from a uniform distribution symmetric around zero with standard deviation of the order of the offsets given by geometric normalization. This generated 100 additional (pseudo) normalized measurements for each microarray dataset.

Using the associated phenotypic labels, we trained linear SVM classifiers using geometrically normalized data and the 100 additional randomly normalized data and compared the resulting margins. Adopting the LOO framework, experiments were repeated as many times as the number of available samples by leaving out one sample each time. Since we are interested in only the relative margins of the SVM classifiers, the genes used for classifier training do not matter as they remain the same for all the classifiers. All the genes in the respective datasets were used for training. Each gene's expression profile across samples was shifted and scaled to have zero mean and unit variance. The plots of Fig. A.1 show the resulting margins. Clearly the margins of the SVM classifiers trained using geometrically normalized data are consistently higher than the corresponding margins of the 100 other classifiers trained using the randomly offset data. This is a validation of the result obtained above.



(a) Alon data (b) Hoying data

Figure A.1: Comparison of margins of linear SVM classifiers trained using geometrically normalized data and data normalized using random sets of array-specific offsets. The plots on the left and right show results using Alon (Alon et al., 1999) and Hoying (Greer et al., 2006) microarray datasets respectively. The number of classifiers trained using a particular normalized data is equal to the number of available samples due to leaving out one sample following the LOO framework. The classifiers trained using geometrically normalized data consistently result in the largest margin relative to the others.

REFERENCES

Affymetrix (2001). *Affymetrix Microarray Suite Users Guide*. Affymetrix, Santa Clara, CA, version 5.0 edition.

Alexa, A., J. Rahnenführer, and T. Lengauer (2006). Improved scoring of functional groups from gene expression data by decorrelating GO graph structure. *Bioinformatics*, **22**(13), pp. 1600–1607.

Alon, U., N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, **96**(12), pp. 6745–6750.

Ashburner, M. (2000). Gene Ontology: Tool for the unification of biology. *Nature Genetics*, **25**, pp. 25–29.

Bar-Joseph, Z. (2004). Analyzing time series gene expression data. *Bioinformatics*, **20**(16), pp. 2493–2503.

Barnard, K., P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, and M. Jordan (2003). Matching Words and Pictures. *Journal of Machine Learning Research*, **3**, pp. 1107–1135.

Barnard, K., P. Duygulu, and D. Forsyth (2001). Clustering Art. In *2001 IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, pp. 434–441.

Barnard, K. and D. Forsyth (2001). Learning the Semantics of Words and Pictures. In *International Conference on Computer Vision*, pp. II:408–415.

Bilmes, J. (1998). A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. Technical report, ICSI (U. C. Berkeley).

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Bolstad, B. (2007). *Fundamentals of Data Mining in Genomics and Proteomics*, pp. 51–78. Springer US.

Bolstad, B., R. Irizarry, M. Astrand, and T. Speed (2003). A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *BMC Bioinformatics*, **19**(2), pp. 185–193.

Carbonetto, P. and N. de Freitas (2003). Why can't José read?: the problem of learning semantic associations in a robot environment. In *Proceedings of the HLT-NAACL 2003 workshop on Learning word meaning from non-linguistic data*, pp. 54–61.

Cho, R. J., M. J. Campbell, E. A. Winzeler, L. Steinmetz, A. Conway, L. Wodicka, T. G. Wolfsberg, A. E. Gabrielian, D. Landsman, D. J. Lockhart, and R. W. Davis (1998). A Genome-wide Transcriptional Analysis of the Mitotic Cell Cycle. *Molecular Cell*, **2**(1), pp. 65–73.

Cun, Y. L., J. S. Denker, and S. A. Solla (1990). Optimal brain damage. In *Advances in neural information processing systems 2*, pp. 598–605. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. ISBN 1-55860-100-7.

Dempster, A., N. Laird, and D. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, **39**(1), pp. 1–38.

Duda, R. O., P. E. Hart, and D. G. Stork (2000). *Pattern Classification*. Wiley Interscience.

Eisen, M. and P. Brown (1999). DNA arrays for analysis of gene expression. *Methods Enzymology*, **303**, pp. 179–205.

Fodor, S., J. Read, M. Pirrung, L. Stryer, A. Lu, and D. Solas (1991). Light-directed, spatially addressable parallel chemical synthesis. *Science*, **251**(4995), pp. 767–773.

Furey, T. S., N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, **16**(10), pp. 906–914.

Golub, T. R., D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander (1999). Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. *Science*, **286**(5439), pp. 531–537.

Greer, K. A., M. R. McReynolds, H. L. Brooks, and J. B. Hoying (2006). CARMA: A platform for analyzing microarray datasets that incorporate replicate measures. *BMC Bioinformatics*, **7**, p. 149.

Grossman, S., S. Bauer, P. N. Robinson, and M. Vingron (2007). Improved detection of overrepresentation of Gene-Ontology annotations with parent child analysis. *Bioinformatics*, **23**(22), pp. 3024–3031.

Guyon, I., J. Weston, S. Barnhill, and V. Vapnik (2002). Gene Selection for Cancer Classification using Support Vector Machines. *Machine Learning*, **46**(1-3), pp. 389–422. ISSN 0885-6125.

Hua, J., Y. Balagurunathan, Y. Chen, J. Lowey, M. L. Bittner, Z. Xiong, E. Suh, and E. R. Dougherty (2006). Normalization benefits microarray-based classification. *EURASIP Journal on Bioinformatics and Systems Biology*, **2006**(1), pp. 1–13.

Khatri, P. and S. Drăghici (2005). Ontological analysis of gene expression data: current tools, limitations, and open problems. *Bioinformatics*, **21**(18), pp. 3587–3595.

Klebanov, L. and A. Yakovlev (2007). Diverse correlation structures in gene expression data and their utility in improving statistical inference. *The Annals of Applied Statistics*, **1**(2), pp. 538–559.

Lavrenko, V., R. Manmatha, and J. Jeon (2003). A Model for Learning the Semantics of Pictures. In *Proceedings of the Seventeenth Annual Conference on Neural Information Processing Systems*, volume 16, pp. 553–560.

Lenoir, T. and E. Giannella (2006). The emergence and diffusion of DNA microarray technology. *Journal of Biomedical Discovery and Collaboration*, **1**(11).

Lu, Y., R. Rosenfeld, I. Simon, G. J. Nau, and Z. Bar-Joseph (2008). A probabilistic generative model for GO enrichment analysis. *Nucleic Acids Research*, pp. gkn434+.

Pavlidis, P., J. Weston, J. Cai, and W. N. Grundy (2001). Gene functional classification from heterogeneous data. In *RECOMB '01: Proceedings of the fifth annual international conference on Computational biology*, pp. 249–255. ACM.

Qiu, X., A. I. Brooks, L. Klebanov, and A. Yakovlev (2005). The effects of normalization on the correlation structure of microarray data. *BMC Bioinformatics*, **6**, p. 120.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, **77**(2), pp. 257–286.

Schliep, A., I. G. Costa, C. Steinhoff, and A. Schonhuth (2005). Analyzing Gene Expression Time-Courses. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **2**(3), pp. 179–193.

Szabo, A., K. Boucher, W. Carroll, L. B. Klebanov, A. D. Tsodikov, and A. Y. Yakovlev (2002). Variable selection and pattern recognition with gene expression data generated by the microarray technology. *Mathematical Biosciences*, **176**, pp. 71–98. doi:doi:10.1016/S0025-5564(01)00103-1.

Tsodikov, A., A. Szabo, and D. Jones (2002). Adjustments and measures of differential expression for microarray data. *BMC Bioinformatics*, **18**(2), pp. 251–260.

Vapnik, V. (1998). *Statistical learning theory*. Wiley Interscience.

Whitfield, M. L., A. J. Sherlock, G. Saldanha, C. A. Murray, J. I. ane Ball, K. E. Alexander, J. C. Matese, C. M. Perou, M. M. Hurt, and P. O. Brown (2002). Identification of Genes Periodically Expressed in the Human Cell Cycle and Their Expression in Tumors. *Molecular Biology of the Cell*, **13**(6), pp. 1977–2000.

Wu, Z., R. A. Irizarry, R. Gentleman, F. Martinez-Murillo, and F. Spencer (2004). A Model-Based Background Adjustment for Oligonucleotide Expression Arrays. *Journal of the American Statistical Association*, **99**(468), pp. 909–917.

Yang, Y. H., M. J. Buckley, S. Dudoit, and T. P. Speed (2002a). Comparison of Methods for Image Analysis on cDNA Microarray Data. *Journal of Computational and Graphical Statistics*, **11**(1), pp. 108–136.

Yang, Y. H., S. Dudoit, P. Luu, D. M. Lin, V. Peng, J. Ngai, and T. P. Speed (2002b). Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Research*, **30**(4), p. E15.

Yon Rhee, S., V. Wood, K. Dolinski, and S. Draghici (2008). Use and misuse of the gene ontology annotations. *Nature Reviews Genetics*, **9**(7), pp. 509–515.

Yuan, M. and C. Kendziorski (2006). Hidden Markov Models for Microarray Time Course Data in Multiple Biological Conditions. *Journal of the American Statistical Association*, **101**(476), pp. 1323–1332.