Chapter 1

Introduction

1.1 Common terms in molecular biology.

DNA, RNA DNA is a polymer of four possible nucleotides denoted with A (adenine), C (cytosine), T (thymine), G (guanine). A nucleotide is a molecule connecting a phosphate group to a five carbon carbohydrate molecule (a sugar) which is connected to one of the four nitrogenous bases A,C,T,G. The phosphates connect to the sugar of the next nucleotide thereby forming a polymer of nucleotides called DNA. So a DNA molecule is a word written with a four letter alphabet $\{A, C, T, G\}$. In the coding regions of the DNA molecule, every three base pairs form the code for a amino acid. There exist only 20 amino acids. The DNA molecule is arranged as a double helix of two intertwined strands of deoxyribose molecules and phosphate groups, with complementary nitrogenous bases A,C,T,G extending out from the deoxyribose molecules toward one another. The complement of A is T and the complement of C is G.

A complete DNA molecule is found in the nucleus of each cell in an organism. For a human being a DNA molecule consists of 23 pairs of chromosomes, one chromosome from each parent. Each chromosome contains many functional regions, the so called **genes**, which encode the amino acids forming a protein. A human being has approximately 40,000 genes. A protein is a sequence of amino acids.

A gene consists of a coding region, a regularitory/operator region and a promotor region. The coding-regions in the gene which are used to form the mRNA (and the by the mRNA encoded protein) are called **exons** and the other regions are called **introns**. RNA contains uracil (U) as one of its four nitrogenous bases, whereas DNA has thymine (T) instead of uracil. RNA is single stranded, whereas DNA is a double stranded helix.

Upstream from the coding regions one finds first a regularitory region and then a promotor region, which are functional in the sense that they are involved in the process of transcription of the gene. The promotor region functions as a binding site for RNA polymerase enzyme, needed to start the transcription process. After binding this enzyme tries to find its way up to the start codon of the coding region to start the transcription process. However, before arriving there it needs to pass through the regularitory region which is located between the promotor region and the coding region. Binding of proteins to the regularitory region can positively (starting the uncoiling of the double stranded helix) and negatively (obstructing the RNA polymerase) control the transcription process. Genes can be recognized by a promotor region and start (TAC) and end codons (ATC). Most (99%

or so) of the complete DNA molecule consists of non-functional useless regions.

Intron An intervening non-coding section of mRNA that is removed before the production of the final mRNA molecule.

Exon A section of mRNA that specifies an amino acid sequence and that is retained during the production of the final mRNA molecule.

Amino Acid A chemical compound that contains at least one amino group (NH_3) and one acid group (CO_2) and is a building block of a protein. There are twenty amino acids.

Codon A three base sequence on an mRNA molecule that specifies a particular amino acid.

Operator and Promotor regions Structural genes are the coding regions which actually encode the structure of the protein it produces. Next to the structural genes is a series of nitrogenous bases responsible for binding mRNA to the ribosome, a process needed to translate mRNA into the encoded protein. Knows as the **ribosomal binding site**, this base sequence encodes the base sequence in mRNA that ensures the ability to unite correctly with the ribosome. This site contains no genetic message.

The next site encountered as we move further away from the structural genes is the recognition site for RNA polymerase. This site is called the **promotor** because binding to the site promotes transcription. RNA polymerase binds to the promotor then moves to the right until it encounters a special "start transcription" signal (TAC) at the beginning of the structural genes.

Lying between the promotor and the structural genes, we encounter the **operator or regulatory region**. This is the regulatory site for repression and activation of the gene. An important level of control takes place here because RNA polymerase must pass through the operator region to get to the structural genes. For instance, when the base sequence of the operator binds with repressor protein, the RNA polymerase cannot pass the blockage and the structural genes do not function.

RNA polymerase The enzyme that functions in transcription and synthesizes an RNA molecule with bases complementary to those in DNA. These RNA molecules, together with ribosomal proteins and enzymes, constitute a system that carries out the task of reading the genetic message and producing the protein that the genetic message specifies.

Enzyme A protein that catalyzes a chemical reaction of metabolism while itself remaining unchanged.

Transcription and protein synthesis The production of mRNA from DNA is called transcription. Transcription of the mRNA by RNA polymerase begins at a specific DNA site on the gene called the **promotor site**. The promotor site is a sequence of nitrogenous bases. For a given gene, the promotor site exists on one DNA strand but not on the other. The strand having the promotor site will transcribe its message to mRNA and is called the **sense strand**.

The process of transcription and thus protein synthesis is initiated by an uncoiling of the DNA double helix and an uncoupling of the two strands of DNA. A functional region of the DNA, the **gene**, is thereby exposed. Using the sequence of nitrogeneous bases along only one of the DNA strands, molecules of RNA are synthesized with complementary

bases. The enzyme **RNA polymerase** moves along one strand of the DNA molecule and synthesizes a complementary mRNA molecule, using the base code of the DNA strand as a guide. Component nucleotides stored in the region are used for the synthesis, and the enzyme RNA polymerase binds the nucleotides together to form the RNA molecule. This RNA molecule formed is called an RNA transcript.

Three types of RNA transcripts are constructed for use in protein synthesis. One type is called **messenger RNA** (mRNA). Each mRNA molecule is a long, single strand of RNA that passes out of the nucleus into the cell's cytoplasm carrying the genetic message. This molecule specifies which amoni acid is to occupy which position in the protein.

The second type of RNA transcript is **ribosomal RNA** (rRNA). Ribosomal RNA combines with protein to form ribosomes, the ultramicroscopic bodies existing along the cell's internal membranes. The ribosomes are sites on the rRNA molecule where the enzymes assemble amino acids to proteins according to the instructions delivered by mRNA molecules.

The third type of RNA transcript is **transfer RNA** (tRNA). A tRNA is a molecule of RNA that carries an amino acid molecule to the ribosome for use in protein synthesis. Transfer RNA molecules exist in dozens of different types and float freely in the cell's cytoplasm, where they bind to amino acids. Then they deliver the aminoacids to the ribosome.

Ribosome An ultramicroscopic cellular body where amino acids are enzymatically bonded to form a protein. Each ribosome has two subunits: a smaller one binds the ribosome to mRNA molecules and a large one where the enzymes for linking amino acids and the amino acids carried by tRNA (below) together are located.

Translation The ribosome moves along the mRNA molecule from codon to codon, as tRNA molecules bring their amino acids into position. Each amino acid is then attached to the growing protein chain, which is at this stage called a **polypeptide**. Once is has given up its amino acid molecule, the tRNA molecule moves back into the cytoplasm to unite with another amino acid molecule. Meanwhile the ribosome moves to the next codon and receives the next tRNA with its amino acid. Note that the codon in mRNA and the anticodon in tRNA contain complementary bases. This complementary pairing specifies which amino acids is slotted into which position in the growing protein chain. Also note that tRNA molecules with different anticodons unite with different amino acids.

Various ribosomes can work simultaneously on a single mRNA molecule. The number of ribosomes simultaneously attached to a gene-specific mRNA is a measure of the number of proteins formed from one mRNA.

Gene expression The production of a mRNA molecule as encoded by the gene. Gene expression can be quantified by the number of mRNA copies in the cell's cytoplasma.

Gene control All the genes are not operating all the time. Gene transcription only occurs for a specified period of time. In most cases, the control of gene expression involves control of transcription at the level of the gene. The site of control is usually a sequence of nucleotides called the **regularitory site**. In many cases, a specific regularitory protein, a so called **repressor protein**, will bind to the regularitory site and exert its controlling influence. This process is called **repression**. When a repressor protein reacts with a regularitory site to inhibit transcription, the control mechanism is referred to as a **negative control**.

Negative control often takes place between points where RNA polymerase binds and the gene for transcription begins. By binding to the site, the repressor protein prevents the movement of RNA polymerase toward the gene. Without RNA polymerase, the gene cannot be transcribed. Placing a log across a railroad track would have a similar effect.

Gene expression can also be controlled by a type of **positive control**. In this case, the regularitory protein encourages gene transcription. The regularitory protein is called an **activator protein** and the process is called **activation**. Activation takes place when the activator protein binds to the regulatory site and stimulates unwinding of the DNA helix to encourage mRNA formation. As usual, the mRNA formation is directed by RNA polymerase, but the enzyme operates more efficiently once the DNA has been unwound. Thus, the process has been activated.

Shape of regularitory protein matters The level of control is regulated by the shape of the regularitory protein. The change in shape can enhance or destroy the ability of a regularitory protein to bind to the regularitory site. For example, in its new shape the regularitory protein may recognize a binding site not recognized previously, or the newly configured protein may be unable to bind to a site where binding was previously possible.

Regulatory site A regularitory site is a sequence of nitrogenous bases where gene expression can be controlled by reaction with repressor or activator proteins.

Repressor protein A protein that reacts with a regulatory site and restricts expression of the gene by inhibiting transcription. If such a repressor is bound to a binding site in the promotor region, then a change in environment can disconnect the repressor protein. For example, the repressor protein unites with (e.g.) lactose molecules and therefore takes a new shape. Now, it is unable to hold onto the DNA sequence in the operator, and the repression is lifted. When the lactose is used up, the repressor protein changes back to its original shape. It then complexes to the operator and biochemically shuts off the gene again.

Enhancer, transcription factor A series of nitrogenous bases that encourages DNA activity at a distant promotor site. Enhancers can be thousands of bases away (upstream) on the DNA molecule. Proteins which attach to the enhancers are called **transcription factors**. They appear to induce the DNA to bend into a loop.

Regulation is complex Regulation of gene expression is not confined to control at the time of transcription. Regulation can occur at multiple levels of gene expression. It can occur, for instance, in how mRNA molecules are processed before leaving the nucleus, in the transport of mRNA molecules out of the nucleus, by the lifespan of mRNA molecules, by the number of ribosomes binding to mRNA molecules and in the activity and stability of the protein products of gene expression. These factors are as important for understanding gene activity as transcription and translation.

1.2 The microarray technology to measure gene expression profiles.

One important ingredient of the biotechnology is the collection of complementary DNA (cDNA) corresponding with each gene in the genome of the organism we study. This results in a cDNA

library which can then be used to construct cDNA gene chips covering the complete genome.

One carries this out by first identifying cells which are known to produce mRNA of large collection of genes. Given such a sample of cells, the task of isolating the mRNA begins. The cells are disrupted and the cellular contents are subjected to an exhaustive series of chemical and physical treatment to exclude all other proteins, fats, carbohydrates and nucleic acids. Then, the mRNA molecules are collected by zeroing in on the poly-A tails they all possess. These are chains of 150-200 nucleotides containing the base adenine. The mass of mRNA is combined with cellulose particles containing on their surfaces a series of nucleic acid segments having thymine. The poly-A section of the mRNA binds tightly to the poly-T molecules, and the remaining debris is washed away. Now, the mRNA molecules can be collected from the cellulose particles to yield concentrated mRNA.

Complementary DNA or cDNA is a DNA molecule that complements the base sequence in RNA and results from **reverse transcriptase activity**. Reverse transcriptase is an enzyme that uses the base sequence in an RNA molecule as a model for synthesizing a complementary DNA molecule. The use of reverse transcriptase to synthesize a DNA molecule requires a primer, or starting nucleotide sequence. The primer consists of a string of thymidine nucleotides, which binds to the poly-A tail of the mRNA and acts as an initiation site for DNA production. The reverse transcriptase then moves along the mRNA molecule, encoding a DNA molecule complementary to the mRNA. This new DNA molecule is referred to as complementary DNA. Then using degradation techniques in alkaline solutions, the cDNA molecule is cleaved away from the mRNA template molecule and isolated in pure form.

Extracting the gene-specific cDNA's is described in Alcamo (1999) "DNA Technology, the Awesome Skill" and is a delicate task. Once one has identified a complete cDNA library one can produce microarray slides which cDNA of all the genes spiked in. These microarray slides can be used to carry out the microarray experiment which maps two tissues or cell line samples into a relative gene expression profile for all genes simultaneously.

The basic microarray experiment. Two tissue or cell line samples are collected and the mRNA of each is labeled with red and green dye. Subsequently, equal amounts of the mRNA samples are combined and washed over microarray slides prepared with the cDNA of p genes. Say that each gene is represented by two spots on a slide. The labeled RNA of gene j will attach to the corresponding spots on the microarray, $j = 1, \ldots, p$. A scanner measures the red and green intensity at each of the spots. The red and green intensities can be normalized by 1) requiring that the total sum of red intensities equals the total sum of green intensities or 2) by including control genes on the chips which are spiked in equal amounts into two tissues and normalizing the other intensities on the chips in such a way that the control genes are empirically 1:1. Subsequently, for each spot we calculate the ratio of red and green intensity. A natural gene-specific summary measure R/G is the geometric mean (i.e. take the average of the log-ratios and exponentiate it) of these gene-specific ratios across the gene-specific spots. Due to the variability in the amount of cDNA in the wells and the strong correlation between red and green intensity at spots it is much better to take averages of spot-specific ratios than to take a ratio of an average of red and an average of green intensities. In order to adjust for dying bias one also carries out the same experiment but with reverse colors for the two samples. As final gene-specific summary measure one takes the average of the two gene-specific "ratios" R/G and G/R.

Typically, one of the samples is a control and the ratios are defined as test over control. When component $X_j > 1$, the DNA of gene j has a higher expression in the test sample than

in the control sample. We say that gene j is **overexpressed**. If component $X_j < 1$, then we say that gene j is **suppressed** and if $X_j \neq 1$, then one says that gene j is **differentially expressed**. Let $\mathbf{X}$ be the p-dimensional column vector of "ratios" representing the relative gene expression profile for a subject or cell line.

1.2.1 Measurement error and P-values.

For each experimental unit there exists a true gene expression profile X^*, so that $X = X^* + \epsilon$ for some error term ϵ, where X is the actual measured gene expression profile. This error term has various components, namely an actual measurement error corresponding with the microarray experiment and errors which can potentially occur *before* the actual microarray experiment in the collection of the two cell samples used in the microarray experiment.

The measurement error of the microarray experiment In order to measure the quality of the microarray experiment itself in measuring the relative gene expression profiles in the two cell samples at hand, one needs to carry out the microarray experiments with two identical samples, so that the mRNA ratios should be one for all genes. The results of such an experiment provide an estimate of a null distribution of the gene-specific ratios around the true ratio 1. In order to always have a sense of this measurement error one should spike in the two cell samples known concentrations of known genes and put these genes on each microarray slide. The measurement error will depend, in particular, on the surface of the slides and the concentration of the mRNA in the two samples. In general, the measurement error follows a U-parabolic shape as a function of concentration: If the mRNA has a low concentration in the sample, then the gene-specific "ratio" X_j will be more variable than if the mRNA occurs in reasonable concentrations. On the other hand, if the mRNA has a very high concentration in one of the samples, then the cDNA in the wells might be used up before the mRNA in the other sample might have a chance to bind. Note that if the cDNA is abundant in the wells relative to the samples, then this latter "saturation effect" would not occur.

Other sources of unwished variability The collection of the tissues and a possible amplification of mRNA from a small sample of mRNA are important sources of variability.

The measurement error of the microarray experiment itself is typically very small for high level gene-chips and technicians relative to the other sources of variability and is easy to establish.

To estimate the distribution of the full ϵ one will need to sample experimental units for which the true X^* is known and carry out the complete experiment resulting in the measured gene expression profile X. For example, suppose that X is the relative gene expression profile of cancerous tissue relative to healthy tissue of a randomly sampled subject. Then one can obtain an estimate of the distribution of ϵ (assuming it does not depend on the value of X^* itself) by sampling healthy patients and measuring the gene expression profile of healthy cells relative to healthy cells, thereby precisely imitating the complete experiment applied to the cancer patients. An estimate $\hat{P}_0$ of this error-distribution can be used as a null distribution to assign p-values $\hat{P}_{0j}(X_j > x_j)$ of an observed gene expression $X_j = x_j$. This p-value now truly measures the likelihood of seeing a gene expression as extreme as x_j, if in truth the subject was sampled from the null-distribution. Note that this null distribution now includes all sources of variability.

If the variance of ϵ is small relative to the variance of X^* across experimental units, then it is still appropriate to just focus on estimation of parameters, such as the mean and covariance

matrix, of the *measured* gene expression profile X and thus ignore the measurement error model $\log(X) = \log(X^*) + \epsilon$. This will be our approach, but we do recommend establishing an estimate of a null distribution of the measurement error ϵ is relatively small.

1.3 Data sets with gene expression.

Observational and experimental studies increasingly (over time) involve the collection of gene expression profiles at one or more time points on each experimental unit. The gene expressions can represent important outcomes, e.g. measuring how different a cell is relative to a control, and it can also be viewed as an important predictor of a clinical outcome such as survival. Therefore we decided that it is natural to distinguish 3 types of parameters. The first set of parameters are summaries of the "marginal" distribution of the gene expression profile(s). The second set of parameters quantify the effect of gene expression(s) on a future (i.e. post-expression) outcome such as survival. Finally, the third set of parameters quantify the effect of a pre-expression variable (such as medical treatment) on the gene-expression profile. The second and third type of parameter is typically defined as a regression parameter in a regression model. For each of these 3 types of parameters, we want to develop estimators of these parameters and corresponding estimates of variability. The second and third set of parameters might actually describe *causal* effects as defined in the causal inference literature. In the latter case these parameters will be defined as regression parameters in causal regression models such as the marginal structural models.

We note that a causal effect of a gene expression on (say) survival is particularly important because it predicts what one would see on a subject if one knocks-out the gene relative to if one does not knock out the gene. Note that such knock-out experiments are actually possible and are used by drug-development companies to develop gene-therapies for a variety of diseases.

Below, we provide a few examples of data sets involving the collection of gene expression. The reader should describe globally parameters of interest and classify them into these three groups.

1. A yeast Saccharomyces cerevisiae data set generated by Dr. M. D. Sollewijn Gelpke, Postdoc Molecular Biology Department, which measures for each gene the gene-specific distribution of the number of attached ribosomes to the mRNA-copies in a cell for a variety of mutant yeast strains.

This data set is important to determine how effective the mRNA copies for each gene in yeast are translated into protein copies and how much a mutant affects the distribution of ribosomes of genes. The results of this analysis might be used to improve understanding gene activity.

2. Yeast data (public domain): Cell cycle gene expression data on 6220 genes at 17 time points with 10 minute intervals such that two full cell cycles are covered in Cho et al. (2001).

Data (public domain): Rosetta Inpharmatics yeast data Hughes et al. (2000) contains expression of 6,220 yeast genes under 300 various experimental conditions (diverse mutations and chemical treatments). In particular, it contains experiments related with Copper and Iron intake methabolism.

In the analysis of yeast data the following data bases are important:

Data (public domain): Genome database: SGD, is a scientific database of the molecular biology and genetics of the yeast Cherry et al. (2001).

Data (public domain): SCPD: The Promoter Database of Saccharomyces cerevisiae (Zhu and Zhang, 1999). This contains known regulatory elements/transcription factors of yeast genome,

their location on upstream sequences etc.
Data (public domain): TRANSFAC: Transcription Factor Database of Heinemeyer et al. (1998) which is a more comprehensive database of transcription factors and it captures yeast as well as many other organisms.
3. The UCSF-cancer center, funded by two NIH-proposals, has data on $n \approx 2000$ breast cancer patients, including gene expression profiles at the primary tumor, histopathologic characteristics (e.g. tumor size, stage, grade, lymphnode involvement), molecular markers (e.g the well known tumor suppressors and oncogenes) and clinical outcomes such as survival, time till recurrence and the followed treatment regime.
4. Data (public domain): Molecular portraits of human breast tumors Perou et al. (2000), Perou et al. (1999).
Gene expression profile of breast tumors on time before and after chemotherapy and histopathologic characteristics of the tumor and baseline characteristics of the subject.
5. Data: Cell line data on 60 human cancers accompanying Ross et al. (public domain).
6. Gene expression profiles on a sample of AML and a sample of ALL Leukemia patients Golub et al. (1999).
7. Colon cancer patient data (Chiron/UCSF, confidential). Gene expression profiles on the primary tumors in 30 colon cancer patients, relative to healthy tissue taken from the colon. Gene expression profile on the metastasized tumors in 50 colon cancer patients. In addition, one collects medical treatment taken between the primary tumor and time at recurrence of a metastasized tumor, histopathologic characteristics of the primary tumor, baseline characteristics of the patients, and right-censored survival.
8. Fresno Asthma Children Environmental Study (FACES). The principal investigator of this study is Prof. Ira Tager, Epidemiology, School of Public Health, UC Berkeley. It is funded by the California Air Resource Board. A random sample of 450 clinically diagnosed asthmatic children ages 6-10 in the Fresno/Clovis area will be enrolled from a register of asthmatic children. The enrollment started in October 2000 and the PI has immediate access to the data base. This community has high asthma morbidity. Children are enrolled in groups of 50; membership in each group will be fixed. The study design consists of a longitudinal and a panel components with 4.5 years of follow up. In the longitudinal component, each subject will undergo detailed baseline and 6-monthly evaluations (medical history, house characteristics, medication use, lung function testing, prick skin testing somatic growth and a Biotech company assists in collecting a large number of biomarkers from blood-samples). For the panel component, each group of 50 subjects will be observed in ten 14-day panels (1 in each of 3 air pollution seasons over 4.5 years). During panel periods, daily data will be obtained on factors such as: twice daily forced expiratory volumes, symptoms, medication use, time-location activity patterns, etc.). Detailed, daily ambient air pollution data will be available from two special monitoring programs which will provide an unprecedented level of daily data on PM mass/constituents/particle number and gaseous pollutants. These will be supplemented by study monitoring of bioaerosol (e.g. fungi, pollens, endotoxin). A subset of each panel will undergo detailed personal monitoring to be used to develop microenvironmental models to assign personal exposures to all study subjects for all panel days.

Important features of the data set are 1) right censoring by time of analysis, 2) possibly informative right-censoring if subjects drop out before the end of the study, 3) (possibly informative) missing visits and missing variables will occur, 4) the joint effect of airpollution with other exposure variables (humidity, temperature) is confounded by the time-dependent variable "rescue-medication use", 5) airpollution and the biomarkers represent very high

dimensional variables and each component corresponds with parameters of interest.
9. Knock out (of candidate genes) experiments.
10. Two groups of identical mice, one group received diet I and the other group received diet II (e.g Prof. Vulpe, Department of Nutrition, UC Berkeley). One collects gene expression profile at one point in time (if it requires sacrificing the mouse) or more points in time (if the relevant tissue can be taken without sacrificing the mouse). One might also be interested in collecting data on time till recurrence of tumor, time till death, certain biomarkers, or other outcomes of interest such as weight loss.

Chapter 2

The Statistical Analysis of a Sample of Gene Expression Profiles

ABSTRACT

Recent developments in microarray technology make it possible to capture the gene expression profiles for thousands of genes at once. With this data researchers are tackling problems ranging from the identification of "cancer genes" to the formidable task of adding functional annotations to our rapidly-growing gene databases. Specific research questions suggest patterns of gene expression that are interesting and informative (e.g. genes with large variance or groups of genes that are highly correlated). Cluster analysis and related techniques are proving to be very useful. We add to this the visualisation of the clusters by visualizing an ordered distance matrix van der Laan and Pollard (2001). However, such exploratory methods alone do not provide the opportunity to engage in *statistical inference*. Given the high-dimensionality (thousands) and small sample sizes (< 30) encountered in these datasets, an honest assessment of sampling variability is crucial and can prevent the over-interpretation of spurious results. van der Laan, Bryan (2001) describe a statistical framework that encompasses many of the analytical goals in gene expression analysis; this framework is completely compatible with many of the current approaches and, in fact, can increase their utility. We propose the use of a deterministic rule, applied to the parameters of the gene expression distribution, to select a target subset of genes that are of biological interest. In addition to subset membership, the target subset can include information about relationships between genes, such as clustering. This target subset presents an interesting parameter that we can estimate by applying the rule to the sample statistics of microarray data. The parametric bootstrap based on a multivariate normal model, or the nonparametric bootstrap based on resampling from the observed data, is used to estimate the distribution of these estimated subsets and relevant summary measures of this sampling distribution are proposed. We focus, in particular, on rules that operate on the mean and covariance. Using Bernstein's Inequality, we obtain consistency of the subset estimates, under the assumption that the sample size converges faster to infinity than the logarithm of the number of genes. We also provide a conservative sample size formula guaranteeing that the sample mean and sample covariance matrix are *uniformly* within a distance $\epsilon > 0$ of the population mean and covariance. The practical performance of the method using a cluster-based subset rule is illustrated with a simulation study. The method is illustrated with an analysis of a publicly available leukemia data set.

2.1 Introduction

2.1.1 Microarray context

Microarray studies are swiftly becoming a very significant and prevalent tool in biomedical research. The microarray technology allows researchers to monitor the expression of thousands of genes simultaneously. A readable introduction to microarrays can be found in Marshall (1999) and a more technical overview is given in the "The Chipping Forecast" [1999].

By comparing gene expression profiles across cells that are at different stages in some process, in distinct pathological states, or under different experimental conditions, researchers gain insight into the roles and reactions of various genes. For example, one can compare healthy cells to cancerous cells within subjects in order to learn which genes tend to be over (or under) expressed in the diseased cells; regulation of such genes could produce effective cancer treatment and/or prophylaxis. DeRisi et al. (1996) suppressed the tumorigenic properties of human melanoma cells and compared gene expression profiles among "normal" and modified melanoma cells; this experiment allowed investigators to study the differential gene expression that is associated with tumor suppression. Data analysis methods appropriate for microarray data are surveyed by Claverie (1999), Eisen et al. (1998), and Herwig et al. (1999).

Recent microarray studies have relied heavily on clustering procedures. Eisen et al. (1998) apply a hierarchical cluster analysis algorithm to an empirical correlation matrix and Golub et al. (1999) use a neural network algorithm called self-organizing maps (SOM) which, like K-means clustering and the partitioning around medoids (PAM) of Kaufman and Rousseeuw (1990), places objects into a fixed number of clusters. We feel that such approaches suffer from two deficiencies, which we address in this chapter. First, since these techniques are used in a purely data exploratory manner, they lack important notions such as parameter, parameter estimate, consistency and confidence. Second, techniques that are purely descriptive and *ad hoc* make it difficult to design a study to meet particular goals.

2.1.2 Overview of the statistical method

For a randomly sampled subject or organism (from some population) we measure with the microarray experiment (as described previously) a relative gene expression profile for p genes in one cell sample relative to a control cell sample. Let $\mathbf{X}$ be the p-dimensional column vector of ratios representing the relative gene expression profile for a subject or cell line randomly drawn from a well-defined population. Suppose that we observe n i.i.d. copies $\mathbf{X}_1, \ldots, \mathbf{X}_n$ of this random vector $\mathbf{X}$, for example, one for each of n randomly sampled subjects.

In the dataset that originally motivated this work, the population of interest is human colon cancer patients and for each subject we have a sample of healthy colon tissue (control) and colon tumor tissue (test). From such data, we want to find a subset of genes for which differential expression is associated with cancer. Below, we will show that two sample data sets or paired sample data sets can be naturally transformed to a one sample data set. For example, in the dataset we analyze in section 2.9, the population of interest is human acute leukemia patients described by Golub et al. (1999). The data actually arise from a microarray technology slightly different than the cDNA arrays described above, namely, an oligonucleotide produced by Affymetrix. In any case, the dataset contains expression profiles for patients with two distinct types of leukemia, namely ALL and AML. One can now define $\mathbf{X}$ as the gene expression profile of a ALL patient relative to the mean profile among the AML patients. Our data analysis in section 2.9 focuses on finding genes whose expression best distinguishes the two tumor classes

and are, therefore, useful in diagnosis.

In light of the typical scientific goals, we generally wish to find (1) genes that are *differentially expressed*, e.g. expression is different in the test sample relative to the control, and (2) groups of genes which are *significantly correlated with each other*. We are interested in genes whose expression levels tend to vary together, because such genes might be part of the same causal mechanism.

Since k-fold over-expression represents the opposite of k-fold under-expression, it is natural to use a logarithmic transformation: let $Y_j^* = \log(X_j)$ $j = 1, \ldots, p$. In addition, to control the effect of outliers and to obtain nonparametric consistency results proved later in this paper, we also propose to truncate the log-ratios by a user-supplied constant M:

$$Y_j = \begin{cases} Y_j^* & \text{if } |Y_j^*| < M, \qquad j = 1, \ldots, p, \\ M \times \operatorname{sgn}(Y^*) & \text{if } |Y_j^*| \geq M. \qquad 0 < M < \infty. \end{cases}$$

Another additional truncation would be to examine centered data $Y_j^* - \widehat{\mu}_j^*$ and truncate all observations that were, for example, greater than 3 standard deviations in absolute value. Let $\mathbf{Y}$ be the column vector with component j being equal to Y_j, $j = 1, \ldots, p$. Denote the expectation, covariance, and correlation of $\mathbf{Y}$ by $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$, and $\boldsymbol{\rho}$, respectively.

We do not require that $\mathbf{Y}$ is a gene expression profile, but it can be any high dimensional vector. In particular, if one is interested in finding binding sites on the regularitory DNA-region of a gene which predict gene expression, then it is preferable to define $\mathbf{Y}$ as a transformation of a gene expression profile defined by the DNA-sequence of the regularitory region. For example, each component of $\mathbf{Y}$ might correspond with a word of bases $\{A, C, T, G\}$ of length 6 and measure its importance in predicting gene expression for the randomly sampled subject or organism. The latter approach is studied in chapter 4 (Keles, van der Laan, Eisen, 2001). For the sake of clarity, in this chapter we will treat $\mathbf{Y}$ as a vector or gene expression profiles.

Suppose that we know $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, and that subject matter experts believe that certain patterns of gene expression distinguish specific genes as important. Then a natural question is "How should we select a subset $(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \to \mathbf{S}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ of genes that merit special attention?" We might also wish to regard $\mathbf{S}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ as a set of genes that is subdivided into several groups labeled from 1 to K. We can identify such a subset $\mathbf{S}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ by a p-vector $\mathbf{S}$ whose components take values in $\{0, \ldots, K\}$. If $S_j = 0$, then gene j is excluded from the subset and if $S_j = k$, $k \in \{1, \ldots, K\}$, then gene j is included in the subset and carries label k. At times, we will also describe the subset as a set of gene indices j, $j \in \{1, \ldots, p\}$; this is equivalent to setting $S_j = 0$ for genes not in the subset and to some integer between 1 and K otherwise. Hereinafter $\mathcal{S} \equiv \mathbf{S}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ will represent the target subset of genes that we wish to distinguish as important.

As an example of a very simple rule, one could define $\mathbf{S}(\cdot, \cdot)$ as $\{j : \mu_j > C\}$, for some $C > 0$. A more sophisticated subset rule would be to (1) select those genes which are at least 3-fold differentially expressed w.r.t. the geometric mean (i.e. only include gene j if $|\mu_j| > \log 3$); and (2) construct a correlation-distance matrix for these differentially expressed genes from the appropriate elements of $\boldsymbol{\rho}$; and (3) apply a clustering algorithm to (some function of) this distance-matrix; and possibly (4) only include those genes in $\mathbf{S}$ that are closest to the cluster centers. In fact, most of the analytical techniques currently being applied to gene expression data (for example Eisen et al., 1998; Golub et al., 1999) operate on the mean and covariance and are, therefore, perfect candidates for the type of subset rule considered here. It is not necessary for the subset rule to eliminate genes at all, although it generally advantageous to do so. Even if the rule simply applies labels that have a stable meaning – for example, by employing a

supervised clustering technique that find clusters around pre-specified genes – the methods we propose would allow the analyst to assess the stability of these clusters.

Given a well-defined subset rule $\mathbf{S}(\cdot,\cdot)$, a natural estimate of the target subset $\mathcal{S}$ is $\widehat{\mathbf{S}}_n \equiv \mathbf{S}(\widehat{\boldsymbol{\mu}}_n, \widehat{\boldsymbol{\Sigma}}_n)$, where $\widehat{\boldsymbol{\mu}}_n$ and $\widehat{\boldsymbol{\Sigma}}_n$ are the sample mean and covariance, respectively, of the (truncated) data. We prove the consistency of $(\widehat{\boldsymbol{\mu}}_n, \widehat{\boldsymbol{\Sigma}}_n)$ and $\mathbf{S}(\widehat{\boldsymbol{\mu}}_n, \widehat{\boldsymbol{\Sigma}}_n)$ (see section 2.4) *non-parametrically* when $n/\log(p(n)) \to \infty$ and $M < \infty$. The case where $p = \infty$ and $p >> n$ is extremely relevant, as microarray experiments already produce data on 20000 genes and in the future we will encounter datasets with all human genes (estimated to be between 35000 and 140000). In stark contrast, sample sizes often fall below 30. We also provide a nonparametric sample size formula that guarantees with probability at least $0 < \gamma < 1$ that the maximal difference between $\widehat{\boldsymbol{\mu}}_n$ and $\boldsymbol{\mu}$ is smaller than ϵ and similarly for $\widehat{\boldsymbol{\Sigma}}_n$ and $\boldsymbol{\Sigma}$. If one is willing to assume that $\mathbf{Y} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ has a multivariate normal distribution, then the truncation is not needed, but we aim to be as nonparametric as possible.

The sampling distribution of the estimated subsets $\widehat{\mathbf{S}}_n$ provides valuable information for the analyst. One might wish to choose the sample size and/or subset rule in order to ensure the reproducibility of certain results or to realize some other performance measure. As an example of a feature we would hope to see reproduced in samples, consider a gene j that appears in $\mathcal{S}$. For a particular data-generating distribution, sample size n, and subset rule $\mathbf{S}(\cdot,\cdot)$, there is a probability p_j that gene j will appear in the estimated subset $\widehat{\mathbf{S}}_n$ produced by a randomly drawn sample; we will call such probabilities p_j "single-gene probabilities". If the single-gene probabilities are low for many of the genes in $\mathcal{S}$, we might choose to increase the sample size or select a subset rule that is easier to estimate. If the single-gene probabilities are generally high, we might proceed with the study and, when we observe estimates of p_j that are close to 1, feel confident that those genes are in $\mathcal{S}$.

Since we want to determine the membership of a specific set, it is natural to apply conventional measures of test quality, such as sensitivity and positive predictive value, to any procedure we devise. In this context, sensitivity is the proportion of the target subset that also falls in the estimated subset and positive predictive value is the proportion of the estimated subset that is also in the target subset.

Determining single-gene probabilities and the distribution of subset quality measures requires knowledge of the actual sampling distribution of $\widehat{\mathbf{S}}_n$. In order to estimate these quantities we use the parametric or nonparametric bootstrap. In general, the asymptotic validity of the parametric bootstrap requires that the chosen parametric model be correct. However, as long as we choose a parametric model that places no constraints on $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, even when it is incorrect, the parametric bootstrap will still consistently estimate the degenerate limit distribution of $S(\mu_n, Sigma_n)$ and $S(\mu_n)$. Specifically, we use as parametric model the multivariate normal model $\mathbf{Y} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and, based on the data we have seen, we believe this to be a reasonable choice after truncation.

The bootstrap (Efron and Tibshirani, 1993) was first used to investigate the reproducibility of certain features of phylogenetic trees by Felsenstein (1985). Efron and Tibshirani (1998) later took up this problem more generally and termed it the "problem of regions". They ask: given an interesting feature in an observed descriptive statistic, how confident can we be that this feature is present in the data-generating distribution? Efron and Tibshirani also link this confidence measure, in certain settings, to frequentist p-values and Bayesian a posteriori probabilities.

2.1.3 Application to paired and unpaired comparisons

Now suppose we have two sets of relative gene expression measurements $(\mathbf{X}, \mathbf{Y})$ on a common set of p genes that we wish to compare. Such data can arise under two different scenarios: paired and unpaired. In the paired scenario, we have two observations on each subject. For example, gene expression might be measured on a cell line at two different time points in the cell cycle relative to a baseline. Or we might observe the same subject before and after treatment. Perou et al. (2000), for example, analyzed gene expression in human breast cancer tumors before and after chemotherapy using a common reference sample. In the unpaired scenario, we have observations on subjects drawn from two subpopulations of subjects (possibly with different numbers of observations in each subsample). Golub et al., for example, used gene expression data to distinguish between acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML).

We might want to focus on genes that appear to be very differently expressed in the two data sets. One approach is to simply analyze the two data sets separately and compare the clustering patterns. Another approach is to combine the two data sets into one data set. The way we do this depends on the scenario that generated the data. In the paired scenario, we can form a p-dimensional vector of log ratios, $\log(\mathbf{X}_i/\mathbf{Y}_i)$, by dividing the relative expression for a subject at one time point by that at the other before taking the log. In the unpaired scenario, we can form a p-dimensional vector of log ratios by dividing the relative expression for a subject by the geometric mean relative expression for all subjects in the other subpopulation before taking the log so that we get $\log(\mathbf{X}_i/\hat{\mu_Y})$. For both scenarios, the empirical mean of the combined data set is the difference between the two sample means of the log ratios in the two separate data sets.

2.2 The estimated subset and the bootstrap

2.2.1 Subset rules

We propose several simple, but easily interpretable, subset rules and all are simply functions of the parameters $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. We have found it natural to divide the subset rule into three phases: (1) a pre-screen in which certain genes are eliminated; (2) a mid-rule in which inter-relationships between genes are sought; and (3) a post-screen in which even more genes are eliminated. We emphasize that it is not necessary to employ all three phases of the rule and, therefore, a clustering algorithm alone can be regarded as an example of such a rule, whenever the distance metric is a function of $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. For example, this is the case with Euclidean distance, correlation distance, the modified correlation distance proposed by Eisen et al. (1998), and principal component based metrics. From now on, we denote the distance between genes i and j by D_{ij}, the p by p symmetric matrix of such distances by $\mathbf{D}$, and we assume that $\mathbf{D}$ is determined by $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

Table 2.1 presents examples of the rules and metrics one can work with. A common requirement for inclusion in the subset is *differential expression* and we use the pre-screen to retain only those with sufficient evidence of differential expression. The table presents pre-screens that range from very simple cutoffs to those that determine whether a certain proportion p of the population exhibits a sufficient level $\log(\delta_1)$ of over and/or under (determines expression. We then seek groups of genes that tend to be coexpressed; a clustering algorithm, such as PAM (Kaufman and Rousseeuw, 1990, chap. 2), is a typical mid-rule, but many other clustering and neural network algorithms are also suitable. Finally, since clustering algorithms place all objects

Table 2.1: Subset rule examples.

Pre-screen	Distance metric	Mid-rule	Post-screen
$\mu_j > \log(\delta_1)$	Euclidean distance	PAM	$D_{ij} < \delta_2$,
$\lvert\mu_j\rvert > \log(\delta_1)$	$1 - \lvert\rho_{ij}\rvert, 1 - \rho_{ij}$	PAM, with fixed medoids	for some cluster center i
$\lvert\mu_j\rvert >$	1 - Eisen's modified	Self-organizing maps	$silhouette_j > \delta_2$
$\log(\delta_1) + \sigma_j\Phi^{-1}(p)$	correlation	Hierarchical clustering	(part of PAM output)
		K-means clustering	

into clusters, even if there is little evidence to favor one cluster assignment over another, we often use the post-screen to retain only those genes that appear to be well-matched to their cluster. One could use actual distances to cluster centers or members to make this determination or, as in the case of the "silhouettes" in PAM, there may be other useful output from the clustering procedure one can exploit.

2.2.2 Partitioning Around Medoids (PAM).

A particular subset rule $S(\mu, \Sigma)$ is based on the output of the clustering procedure PAM (Kaufman and Rousseeuw, 1990, chap. 2), which takes as input a dissimilarity matrix $\mathbf{D}$ based on any distance metric. Let D_{ij} denote the dissimilarity between genes i and j where each gene is represented by an n dimensional vector. Possible dissimilarities which are functions of Σ between these two n-dimensional vectors are:

$$\begin{aligned} D_{ij} &= 1 - \rho_{ij} \text{ correlation} \\ D_{ij} &= 1 - \mid \rho_{ij} \mid \text{ absolute correlation} \\ D_{ij} &= 1 - \rho^0_{ij} \text{ cosine-angle} \\ D_{ij} &= 1 - \mid \rho^0_{ij} \mid \text{ absolute-cosine-angle} \\ D_{ij} &= \sum_{l=1}^{n} (Y_{il} - Y_{jl})^2 \text{ euclidean,} \end{aligned}$$

where

$$\rho^0_{ij} \equiv \frac{\sum_{l=1}^n Y_{il} Y_{jl}}{\sqrt{\sum_{l=1}^n Y_{il}^2} \sqrt{\sum_{l=1}^n Y_{jl}^2}}.$$

It is of interest to note that the $1 - \rho^0_{ij}$ equals 2 times the squared euclidean distance of the two vectors standardized to have euclidean norm 1. This distance was used in Eisen et al. (1998), and it has been our experience that it is a sensible choice in many applications.

Let K be the number of clusters (*i.e.*: the number of causal mechanisms we believe to be operating). Given K, PAM selects K potential medoids, calculates for each gene its distance to the closest of these potential medoids and minimizes over the vector of K potential medoids the sum of these distances over all genes. The solution of this minimization problem is a vector of K medoids. Each medoid identifies a cluster, defined as the genes which are closer to this medoid

than to any of the other $K-1$ medoids. Like any clustering routine which solves a minimization problem, PAM often converges to one of the many local minima, which is not necessarily the global solution. For example, by randomly permutating the rows of the data matrix, we can produce different choices of medoids and possibly different clustering labels for some genes which lie between one or more clusters. As a solution, we recommend randomly permutating the data matrix a large number of times to produce different starting values, rerunning PAM each time, and selecting the medoids which give the smallest sum of distances.

One can consider K as given or it can be data-adaptively selected, for example, by maximizing the average silhouette as recommended by Kaufman and Rousseeuw. The silhouette for a gene is calculated as follows. For each gene j, calculate a_j which is the average dissimilarity of gene j with each other member of gene j's cluster. For each gene j and each cluster k that is not gene j's cluster, calculate b_{jk}, which is defined as the average dissimilarity of gene j with the members of cluster k. Let $b_j \equiv \min_k b_{jk}$, where the minimum is taken over all clusters k that are not gene j's cluster. Finally, the silhouette of gene j is defined by the formula:

$$silhouette_j = \frac{b_j - a_j}{\max(a_j, b_j)}.$$

Note that the largest this can be is 1, which occurs only if there is no dissimilarity within gene j's cluster (*i.e.*: $a_j = 0$). The other extreme is -1. Heuristically, the silhouette measures how well matched an object is to the other objects in its own cluster versus how well matched it would be if it were moved to another cluster.

The (minimal) output of PAM consists of two vectors: (1) a p-dimensional vector $\mathbf{c}$, where $c_j = k$ indicates that gene j belongs to cluster k, and (2) a K-dimensional vector $\mathbf{m}$, where $m_k = j$ indicates that the medoid of cluster k is gene j, where $j \in \{1, \ldots, p\}$ and $k \in \{1, \ldots, K\}$. An attractive property of PAM is that the clusters are identified by the medoids, which are genes themselves, and it has been our experience that the medoids are stable representations of the clusters.

Comparison with k-means One of the most well known partitioning methods is k-means. In the k-means algorithm the observations are classified as belonging to one of k groups. Group membership is determined by calculating the centroid for each group, the multidimensional version of the mean, and assigning each observation to the group with the closest centroid. The centroids are calculated by minimizing the sum over all elements of the squared-euclidean distance to its closest centroid. PAM has 2 crucial advantages relative to k-means. Firstly, k-means only allows clustering with respect to the euclidean distance, while PAM accepts any dissimilarity matrix as input. Secondly, PAM is more robust because it minimizes a sum of dissimilarities instead of a sum of *squared* euclidean distances. The latter is particularly important in the context of clustering genes when many genes do not really belong to any cluster so that most clusters contain many badly clustered genes. In this situation the centroids of k-means will be heavily affected by the badly clustered genes, while the medoids are much more robust elements of the clusters. It has been our experience that the medoid-genes typically represent the strongly clustered component of the cluster.

2.3 Visualisation of clusters.

We propose to visualize the clusters by 1) ordering the clusters 2) ordering the elements within the clusters and 3) visualising the ordered dissimilarity matrix with colors such as red (genes

are close) and green (genes are far apart). To be concrete, let's consider the visualisation of the clusters of genes as obtained with PAM. Let $PAM(data, k, d)$ represent the output of PAM when we give it the data set "data", number of clusters k and distance metric d. Since the clusters are defined by the medoids one can order the clusters by just ordering the corresponding medoids.

Ordering medoids. We propose to order the medoids by building a hierarchical tree from the medoids with PAM as follows. Let "medoids.data" be the k by n matrix containing the k medoids. Initially, we apply $PAM(medoids.data, 2, d)$ and label the two clusters with clust1 and clust2. For each of the two clusters we can now define the neighboring cluster "clust-next". Subsequently, at each node we apply PAM again with say $k = 2$ and we now order the k new clusters by their distance with respect to medoid of "clust-next" going from maximal distance to smallest distance if "clust-next" is to the right and from smallest distance to maximal distance if "clust-next" is to the left. In this way each level of the tree has an ordered list or clusters. By running down the tree until each cluster is of size one, we obtain a unique ordering of the k medoids. Note that this ordering is based on the same dissimilarity measure as we used to cluster the original data set.

We also implemented the following ordering based on minimizing a criteria. Consider a particular ordering of medoids. For component i and j in this ordered list we have a distance $d(i, j)$ between these two corresponding medoids. Compute now the empirical correlation between the distance $j - i$ in the list and the actual distance $d(i, j)$ over all pairs $(i, j), i < j$. We now compute the ordering of medoids which minimizes this empirical correlation. In all our data set examples the hierarchical PAM ordering of medoids corresponded with this optimized ordering of the medoids, but we do not claim they generally agree.

Ordering genes within cluster. Given the ordering of clusters, it remains to cluster the genes within the clusters. We choose to order the genes within each of cluster by either (i) their distance with respect to the medoid of that cluster so that the badly clustered genes end up at the edge of these clusters or (ii) their distance with respect to the medoid of the neighboring cluster.

2.3.1 The parametric and nonparametric bootstrap.

In order to establish the variability and reproducibility of the clustering output $S(\mu_n, \Sigma_n)$ (*e.g.* the clusters in level l^* of the tree), we propose to run the parametric or nonparametric bootstrap. This involves repeatedly sampling n observations $Y_1^\#, \ldots, Y_n^\#$ from a multivariate normal distribution $N(\mu_n, \Sigma_n)$ (van der Laan and Bryan (2001)) or from the empirical distribution which puts mass $1/n$ on each of the original observations $Y_1, \ldots, Y_n$. One estimates the distribution (and, in particular, the variance) of the clustering output $S(\mu_n, \Sigma_n)$, with the empirical distribution of $S(\mu_n^\#, \Sigma_n^\#)$.

Above we defined output $S(\mu_n, \Sigma_n)$ obtained by applying the PAM program to the empirical mean and covariance matrix μ_n, Σ_n. In order to carry out the bootstrap it is important that $S(\mu_n, \Sigma_n)$ is defined as a deterministic rule applied to the data or a summary of the data (μ_n, Σ_n): if the clustering output was based on visual inspection steps, then these need to be automated in order to satisfactorily carry out the bootstrap. Now, we carry out precisely the same procedure in each bootstrap sample.

Another clustering output is to simply apply PAM with fixed medoids. Since we have seen that the selection of medoids (but not so much the cluster assignments) may be dependent on the original order of the genes in the data set, it makes sense to select good medoids in the initial clustering as we have suggested above and then continue to use these in the bootstrap.

In order to establish the cluster variability when fixing the medoids, one fixes the medoids in the bootstrap. Note that this bootstrap avoids estimating the variability in the selection of the medoids. Nonetheless, it provides information about important components of the cluster variability. Since the medoids are the same in each bootstrap sample, we can keep track of the proportion of times a gene falls in each cluster. In other words, for each gene one keeps track of the proportion of times among the bootstrap samples the gene fell into each of the clusters. **?** propose a cluster-probability plot to summarize these statistics which provides a visual way to inspect the cluster reproducibility. These bootstrap cluster-specific probabilities can be used to order the genes within the clusters so that the badly clustered genes can be removed or end up at the edge of the clusters.

If the ordering of the clusters is not a parameter of interest, one might enforce an ordering of the bootstrapped clusters corresponding as close as possible to the ordering in $S(\mu_n, \Sigma_n)$ by comparing their medoids. In this way, one aims at measuring the variability of the actual clusters instead of the ordering. One can also plot the distance matrix "distance($k(l^*)$)" for a number of bootstrap samples and inspect the variability of the cluster structures visually.

Specifics on the parametric bootstrap. Our goal is to estimate the distribution of $\widehat{\mathbf{S}}_n \equiv \mathbf{S}(\widehat{\boldsymbol{\mu}}_n, \widehat{\boldsymbol{\Sigma}}_n) \in \{0, \ldots, K\}^p$, where $(\widehat{\boldsymbol{\mu}}_n, \widehat{\boldsymbol{\Sigma}}_n)$ are the observed mean and covariance matrix of a size n sample from a $N_{p(M)}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ distribution, where we will treat K as fixed. The parametric bootstrap described below could also be used to address uncertainty of a data adaptively determined K. The parametric bootstrap estimates the distribution of $\widehat{\mathbf{S}}_n$ with the distribution of $\widetilde{\mathbf{S}}_n \equiv \mathbf{S}(\widetilde{\boldsymbol{\mu}}_n, \widetilde{\boldsymbol{\Sigma}}_n)$, where $(\widetilde{\boldsymbol{\mu}}_n, \widetilde{\boldsymbol{\Sigma}}_n)$ are the observed mean and covariance matrix of a size n sample from a $N_p(\widehat{\boldsymbol{\mu}}_n, \widehat{\boldsymbol{\Sigma}}_n)$. From this point on, sample quantities (first-generation draws from the data-generating distribution) will be indicated by hats and bootstrap quantities (second-generation draws from statistics of an observed first-generation sample) with tildes.

When we draw from a $N_p(\widehat{\boldsymbol{\mu}}_n, \widehat{\boldsymbol{\Sigma}}_n)$, we will be faced with a singular covariance matrix $\widehat{\boldsymbol{\Sigma}}_n$ when n is smaller than p. In that case we add to the diagonal elements of $\widehat{\boldsymbol{\Sigma}}_n$ an arbitrarily small number $\lambda > 0$, which produces a nonsingular covariance matrix that is extremely close to $\widehat{\boldsymbol{\Sigma}}_n$. This ensures that we are sampling from a well-defined distribution.

So, for $b = 1, \ldots, B$ (B large), we draw n observations (i.e. microarrays) $\widetilde{\mathbf{Y}}_1^b, \ldots, \widetilde{\mathbf{Y}}_n^b$ from $N_p(\widehat{\boldsymbol{\mu}}_n, \widehat{\boldsymbol{\Sigma}}_n)$, compute the observed statistics $(\widetilde{\boldsymbol{\mu}}_n^b, \widetilde{\boldsymbol{\Sigma}}_n^b)$, and record the estimated bootstrap subset $\widetilde{\mathbf{S}}_n^b = \mathbf{S}(\widetilde{\boldsymbol{\mu}}_n^b, \widetilde{\boldsymbol{\Sigma}}_n^b)$. This provides us with B realizations $\widetilde{\mathbf{S}}_n^1, \ldots, \widetilde{\mathbf{S}}_n^B$ of $\widetilde{\mathbf{S}}_n = \mathbf{S}(\widetilde{\boldsymbol{\mu}}_n, \widetilde{\boldsymbol{\Sigma}}_n)$. We use this observed distribution as an estimate of the distribution of $\widetilde{\mathbf{S}}_n$ and, for n large enough, one can view the sampling distribution of $\widetilde{\mathbf{S}}_n$ as an estimate of the distribution of $\widehat{\mathbf{S}}_n = \mathbf{S}(\widehat{\boldsymbol{\mu}}_n, \widehat{\boldsymbol{\Sigma}}_n)$.

Important features of the sampling distribution.

To be specific, consider one of the cluster-based subset rules. Because of the high dimensionality of $\widehat{\mathbf{S}}_n$, we limit our focus to certain aspects of the empirical distribution of $\widehat{\mathbf{S}}_n$. Note that the values and/or distributions of the quantities defined in this section certainly depend on the sample size n; we will employ notation with and without the n, depending on the context.

Indicate the size of a set $\mathcal{A}$ by $|\mathcal{A}|$. Let

$$\begin{aligned} p_j = p_{j,n} &= P(\widehat{S}_j > 0) \\ P_{ij} = P_{ij,n} &= P(\widehat{S}_i > 0, \widehat{S}_j > 0) \\ Q_{ij} = Q_{ij,n} &= P(\widehat{S}_i = \widehat{S}_j > 0) \\ sens = sens_n &= |\mathcal{S} \cap \widehat{\mathbf{S}}|/|\mathcal{S}| \\ ppv = ppv_n &= |\mathcal{S} \cap \widehat{\mathbf{S}}|/|\widehat{\mathbf{S}}| \end{aligned}$$

The quantities p_j, P_{ij}, and Q_{ij} are referred to collectively as "feature-specific probabilities". The random variables *sens* and *ppv* are will be called "quality measures". It is important to note that the concepts of sensitivity and predictive value as employed here differ from their epidemiological counterparts, in that the p genes are not assumed to be i.i.d. Since these quantities are functions of the estimated subset, the distribution of sensitivity and positive predictive value are to be considered when evaluating a proposed subset rule.

The significance of sensitivity is rather obvious, but we would like to emphasize the importance of positive predictive value as well. If scientists use the estimated subset $\widehat{\mathbf{S}}_n$ as a means for selecting a relatively small set of genes for intensive study, it is crucial that the predictive value be high, since a great deal of time and money could be wasted otherwise. This is especially relevant when p is very large. Since an estimated subset for which 50% of the genes are false positives might not be considered usable, information on the predictive value of the estimated subset could alert researchers to the need to collect more data or to choose a different subset rule.

The bootstrap analogue of the above feature-specific probabilities is the empirical frequency in the bootstrap replicates of the appropriate event. Likewise the bootstrap estimate of the distribution of a quality measure will be based on the appropriate empirical proportions. For example, $\widehat{p}_j = \widehat{p}_{j,n} = \frac{1}{B}\sum_b I(\widetilde{S}_j^b > 0)$ and $\widehat{sens}^b = \widehat{sens}_n^b = |\widehat{\mathbf{S}} \cap \widetilde{\mathbf{S}}^b|/|\widehat{\mathbf{S}}|$. All of these quantities are retained in the bootstrap. For practical reasons, we focus on the single-gene probabilities, p_j, and sort the genes in descending order based on this. We report the top-ranked genes and ensure that all members of $\widehat{\mathbf{S}}_n$ are included. In such a list, the genes which fall "deep" into the target subset $\mathcal{S}$ will typically appear before the genes which barely qualified for inclusion into $\mathcal{S}$. The scientist can begin carefully investigating the top ranked genes.

In some settings, there may be a subset of genes that are not only excluded from the target subset $\mathcal{S}$, but are regarded as particularly unsuitable for further study. The definition of such genes is completely up to the user, but will generally correspond to genes that lie *far* outside the target subset. We will denote this subset by $\mathcal{L}$, which is a subset of $\mathcal{S}^c$, the complement of $\mathcal{S}$. For example, one might regard the set $\mathcal{L} = \{j : |\mu_j| < D_\mu\}$, where $D_\mu < C_\mu$, as particularly inappropriate for further study. The proportion of such genes in the estimated subset is of great interest; we will refer to this quantity as the "proportion of extremely false positives" or *pefp*. If this proportion is always very low, one can be reasonably confident that the top ranked genes of the reported subset contain no extremely false positives. We define $\xi_j = \xi_j(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = 1$ if gene j is in $\mathcal{L}$ and $\xi_j = 0$ otherwise, $j = 1, \ldots, p$. Now, the proportion of extremely false positives (*pefp*) for the estimated subset $\widehat{\mathbf{S}}_n$ can be defined as:

$$pefp = pefp_n = \frac{1}{|\widehat{\mathbf{S}}_n|}\sum_{j=1}^{p} I(\xi_j = 1, \widehat{S}_j > 0).$$

As with sensitivity and predictive value, we can use the parametric bootstrap to estimate the

expectation $E(pefp)$ and variance $\mathrm{Var}(pefp)$. Note that if $\mathcal{L} = \mathcal{S}^c$, $pefp$ is simply one minus the positive predictive value. Therefore, $pefp$ becomes interesting only when $\mathcal{L}$ is considerably smaller than $\mathcal{S}^c$.

Other important quantities of interest are the 0.95-quantiles of $|\widehat{\boldsymbol{\mu}}_n - \boldsymbol{\mu}| \equiv \max_j |\widehat{\mu}_{nj} - \mu_j|$ and $\max_{i,j} |\widehat{\boldsymbol{\rho}}_n - \boldsymbol{\rho}|$. It should also be noted that one could replace the PAM procedure described above with some "supervised" clustering method that allows the analyst to specify the cluster centers. If the centers were fixed at genes of known function, the clusters have a coherent meaning throughout the bootstrap. In that case, we can also track how often each gene appears in each fixed-center cluster and, thereby, obtain information on cluster stability.

Interpretation of the output of the parametric bootstrap.

By relying on asymptotic properties established in Section 2.4, we can view the relative frequencies $(\widehat{p}_j, \widehat{P}_{ij}, \widehat{Q}_{ij})$ as estimates of the probabilities (p_j, P_{ij}, Q_{ij}). Consider now the situation in which n is too small to reasonably assume that $(\widehat{\boldsymbol{\mu}}_n, \widehat{\boldsymbol{\Sigma}}_n)$ is close to $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. In this case, it does not follow that the distribution of $\widetilde{\mathbf{S}}_n$ is close to the distribution of $\widehat{\mathbf{S}}_n$. It is our experience that the results of the parametric bootstrap are still valuable. One can simply interpret the results as a simulation study for estimation of $\widehat{\mathbf{S}}_n$ when sampling from $N(\widehat{\boldsymbol{\mu}}_n, \widehat{\boldsymbol{\Sigma}}_n)$. Findings of such a simulation study (such as a low predictive value) will demonstrate the difficulty of estimating $\mathcal{S}$ with the given sample size n. In particular, one can run the parametric bootstrap for several subset rules and thereby determine which types of subsets can be reasonably estimated with the available sample size.

2.4 Asymptotic theory.

Our proposed method for analyzing gene expression data reports an estimated subset $\widehat{\mathbf{S}}_n \equiv \mathbf{S}(\widehat{\boldsymbol{\mu}}_n, \widehat{\boldsymbol{\Sigma}}_n)$ and bootstrap estimates of the feature-specific probabilities and other quantities, such as the distribution of sensitivity and predictive value. In this section we prove the consistency of $\widehat{\mathbf{S}}_n$ and thereby the consistency of the bootstrap estimate of its distribution under appropriate conditions. Additionally, we provide a sample size formula that controls the probability of the estimated subset containing any extremely false positives.

Because p is typically much larger than n, we are interested in the performance of $\widehat{\mathbf{S}}_n$ and the parametric bootstrap when $p(n) \gg n$. Clearly, if p is fixed at some finite value, our method will be valid for some sufficiently large n; but we are concerned with the case where p is essentially infinite. If $\mathcal{S}$ is a fixed (in p) finite subset, which is a reasonable assumption, we have that $P(\mathcal{S} \subseteq \widehat{\mathbf{S}}_n)$ (or, alternatively, the sensitivity) converges to 1 as the sample size converges to infinity. This is true regardless of the rate at which $p(n)$ converges to infinity. However, the positive predictive value still may not converge to one (i.e. the number of false positives may not converge to zero). It is not enough for the target subset to be merely *contained* in the sample subset; we want the two sets to be identical with probability tending to one. To achieve this convergence, we require uniform consistency of $(\widehat{\boldsymbol{\mu}}_n, \widehat{\boldsymbol{\Sigma}}_n)$ for $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. In summary, for a typical subset rule $\mathbf{S}(\cdot, \cdot)$ and under the assumption that there are no subset elements on the boundary, uniform consistency of $(\widehat{\boldsymbol{\mu}}_n, \widehat{\boldsymbol{\Sigma}}_n)$ for $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ implies that $P(\widehat{\mathbf{S}}_n = \mathcal{S}) \to 1$ as $n \to \infty$.

In order to control the error in $\widehat{\mathbf{S}}_n$ as an estimate of $\mathcal{S}$ one needs to control the uniform distance between $(\widehat{\boldsymbol{\mu}}_n, \widehat{\boldsymbol{\Sigma}}_n)$ and $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. In particular, if one wants to control the probability of finding extremely false positives in $\widehat{\mathbf{S}}_n$, then one needs $(\widehat{\boldsymbol{\mu}}_n, \widehat{\boldsymbol{\Sigma}}_n)$ to be within a specified distance ϵ from the true $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where ϵ will depend on the definition of an extremely false positive. For

example, consider the simple subset rule $\mathbf{S}(\boldsymbol{\mu}) = \{j : \mu_j > \log 3\}$. Then one might define an extremely false positive as a gene j with $\mu_j < \log 2$. In this case, given a small user-supplied number $\delta > 0$, one wants to choose the sample size n such that the probability that the uniform distance between $\widehat{\boldsymbol{\mu}}_n$ and $\boldsymbol{\mu}$ is smaller than $\epsilon = \log 3 - \log 2 \approx 0.41$ is larger than $1 - \delta$.

The next two theorems establish the uniform consistency of $(\widehat{\boldsymbol{\mu}}_n, \widehat{\boldsymbol{\Sigma}}_n)$ (and, therefore, $\widehat{\mathbf{S}}$) and provide a sample size formula, respectively; both proofs rely on Bernstein's Inequality for sums of independent mean-zero random variables. Recall that (see van der Vaart and Wellner, 1996, page 102): If $Z_1, \ldots, Z_n$ are independent, have range within $[-W, W]$, and have zero means, then

$$P(|Z_1 + \ldots + Z_n| > x) \leq 2 \exp\left(\frac{-x^2/2}{v + Wx/3}\right) \tag{2.1}$$

for $v \geq \text{var}(Z_1 + \ldots + Z_n)$.

Theorem 2.4.1 (Consistency) *Let $p = p(n)$ be such that $n/\log(p(n)) \to \infty$ as $n \to \infty$ and $M < \infty$ (recall that M bounds the absolute value of the underlying data). As $n \to \infty$, then*

$$\max_j |\widehat{\mu}_j - \mu_j| \to 0 \text{ in probability}$$

and

$$\max_{ij} |\widehat{\Sigma}_{ij} - \Sigma_{ij}| \to 0 \text{ in probability.}$$

This implies the following: Suppose that the subset rule $(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \to \mathbf{S}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is continuous in the sense that if, for the sequence $(\widehat{\boldsymbol{\mu}}_n, \widehat{\boldsymbol{\Sigma}}_n)$ ($p(n)$ vectors and $p(n) \times p(n)$ matrices, respectively),

$$\|(\widehat{\boldsymbol{\mu}}_n, \widehat{\boldsymbol{\Sigma}}_n) - (\boldsymbol{\mu}, \boldsymbol{\Sigma})\|_{max} \to 0, \text{ then } \|\mathbf{S}(\widehat{\boldsymbol{\mu}}_n, \widehat{\boldsymbol{\Sigma}}_n) - \mathbf{S}(\boldsymbol{\mu}, \boldsymbol{\Sigma})\|_{max} \to 0. \tag{2.2}$$

Then for any $\epsilon > 0$,

$$P(\|\widehat{\mathbf{S}}_n - \mathcal{S}\|_{max} > \epsilon) \to 0 \text{ as } n \to \infty.$$

For example, consider subset rule 2.5, defined in section 2.2, indexed by user-supplied C_μ and C_ρ. If there exists an $\epsilon > 0$ such that

$$\begin{aligned} \{j : \mu_j \in (C_\mu - \epsilon, C_\mu + \epsilon)\} &= \emptyset \\ \{(i,j) : \rho_{ij} \in (C_\rho - \epsilon, C_\rho + \epsilon)\} &= \emptyset, \end{aligned} \tag{$*$}$$

then, for such a subset rule, we have

$$P(\widehat{\mathbf{S}}_n = \mathcal{S}) \to 1 \text{ as } n \to \infty.$$

Theorem 2.4.2 (Sample Size Formula) *Let $\epsilon > 0, 1 > \delta > 0$, and the number of genes p be given. Let σ^2 be an upper bound of $\max_j \sigma_j^2$ and let $W > 0$ be a constant such that $P(Y_j - \mu_j \in [-W, W]) = 1$, for all j. Define $n^*(p, \epsilon, \delta, W, \sigma^2)$ as follows:*

$$n^*(p, \epsilon, \delta, W, \sigma^2) = \frac{1}{c}(\log p + \log \frac{2}{\delta}), \text{ where } c = c(\epsilon, \sigma^2, W) = \frac{\epsilon^2}{2\sigma^2 + 2W\epsilon/3}.$$

If $n > n^*$, then

$$P(\max_j |\widehat{\mu}_j - \mu_j| > \epsilon) < \delta.$$

Similarly, if $n > n^*(p^2, \epsilon, \delta, W^2, \sigma_\Sigma^2)$, where σ_Σ^2 is an upper bound of the variance of $Y_i Y_j$, then

$$P(\max_{ij} |\widehat{\Sigma}_{ij} - \Sigma_{ij}| > \epsilon) < \delta.$$

The consistency of $\mathbf{S}(\widehat{\boldsymbol{\mu}}_n, \widehat{\boldsymbol{\Sigma}}_n)$ is a direct consequence of the uniform consistency of $(\widehat{\boldsymbol{\mu}}_n, \widehat{\boldsymbol{\Sigma}}_n)$. We will prove the uniform consistency of $\widehat{\boldsymbol{\mu}}_n$. For a particular component of $\widehat{\boldsymbol{\mu}}_n$, application of Bernstein's Inequality gives:

$$P(|\widehat{\mu}_j - \mu_j| > \epsilon) \leq 2\exp\left(\frac{-n\epsilon^2}{2\sigma^2 + 2W\epsilon/3}\right).$$

Since $P(\cup_j A_j) \leq \sum_j P(A_j)$, we have an upper bound on the probability that the uniform distance from $\widehat{\boldsymbol{\mu}}_n$ to $\boldsymbol{\mu}$ exceeds $\epsilon > 0$:

$$P(\max_j |\widehat{\mu}_j - \mu_j| > \epsilon) \leq \sum_j P(|\widehat{\mu}_j - \mu_j| > \epsilon) \leq 2p\exp\left(\frac{-n\epsilon^2}{2\sigma^2 + 2W\epsilon/3}\right). \tag{2.3}$$

The expression on the right converges to zero if $n/\log(p(n)) \to \infty$ as $n \to \infty$. A similar argument holds for $\widehat{\boldsymbol{\Sigma}}_n$.

The n^* in theorem 2.4.2 is precisely the solution obtained when we set the right-hand expression of (2.3) equal to $0 < \delta < 1$ and solve for n. If one assumes independence of genes, then $P(\max_j |\widehat{\mu}_j - \mu_j| \leq \epsilon) = \prod_j P(|\widehat{\mu}_j - \mu_j| \leq \epsilon)$. By applying Bernstein's Inequality to each term of the product, one obtains the same sample size formula as above, to first order approximation. Therefore, this sample size formula is as sharp as Bernstein's Inequality. In a similar fashion as for the mean, one obtains such a sample size formula for $P(\max_{ij} |\widehat{E}Y_iY_j - EY_iY_j| > \epsilon) < \delta$ and thus for $P(\max_{ij} |\widehat{\Sigma}_{ij} - \Sigma_{ij}| > \epsilon) < \delta$. In this case the summation is over $p(p-1)/2$ elements and Y_iY_j is bounded by W^2.□

If p increases from p_1 to p_2, then $n^*(p)$ increases by a magnitude $\log(p_2/p_1)/c$ and, if p is large, then the derivative $\frac{d}{dp}n^*(p) = 1/(cp)$ actually converges to zero. Therefore the sample size is heavily driven by the factor $1/c$. Consider the example given above and suppose we want the probability of including any extremely false positives to be less than 0.1. Suppose that $p = 5000$, that the maximal variance is 0.5 and that the truncation level W is 1.4 (twice the maximal standard deviation). Application of theorem 2.4.2 says that, if $n > n^*(5000, 0.41, 0.1, 1.4, 0.5) \approx 95$, then $P(\sup_j |\widehat{\mu}_j - \mu_j| > 0.41) < 0.1)$. Note that the effect of a huge increase in p on the required sample size is minor: e.g. $n^*(100000, 0.41, 0.1, 1.4, 0.5) = 120$.

To convey a general sense of the implications of this sample size formula, we provide a few examples:

$$\begin{aligned}
n^*(p = 5000, \epsilon = 0.1, \delta = 0.10, M = 2, \sigma^2 = 0.5) &\approx 1304 \\
n^*(p = 5000, \epsilon = 0.5, \delta = 0.10, M = 2, \sigma^2 = 0.5) &\approx 77 \\
n^*(p = 5000, \epsilon = 0.5, \delta = 0.01, M = 2, \sigma^2 = 0.5) &\approx 92 \\
n^*(p = 5000, \epsilon = 1.0, \delta = 0.05, M = 2, \sigma^2 = 0.5) &\approx 28
\end{aligned}$$

If we set the right-hand expression of (2.3) equal to δ and solve for ϵ, we obtain a $1-\delta$-*uniform confidence band for* μ with radius ϵ for each component:

$$\widehat{\mu}_j \pm \epsilon(p, n, \delta, W, \sigma^2), \text{ where}$$
$$\epsilon = \frac{1}{2n}\left[\frac{2W}{3}(\log p + \log\frac{2}{\delta}) + \sqrt{\left((\frac{2W}{3}(\log p + \log\frac{2}{\delta}))^2 + 8n\sigma^2(\log p + \log\frac{2}{\delta})\right)}\right] \tag{2.4}$$

It is better to construct a confidence band by first scaling the data to have variance one (*i.e.* apply the formula to Y_j/σ_j and use $\sigma^2 = 1$ in (2.4)) and then returning to the original scale by multiplying the radius ϵ with σ_j for each gene:

$$\widehat{\mu}_j \pm \sigma_j \epsilon(p, n, \delta, W, \sigma^2 = 1).$$

In the above σ_j can be estimated with its empirical counterpart $\widehat{\sigma}_{jn}$, $j = 1, \ldots, p$.

Figure 2.1 illustrates visually the implications of this sample size formula for several realistic scenarios. In the top panel we have set $\epsilon = 1$ and in the bottom $\epsilon = 0.58$. Decreasing the tolerable distance ϵ, holding all other constants fixed, increases the required sample size. The noise levels implied by the values of σ in the range [0.5, 1.0] are typical for the data sets we have seen. We note that the sample size formula is actually quite conservative. It makes no assumptions about the correlation between the p genes and, when there is a significant amount of correlation, the true dimension of the problem can be much smaller than p. In practice, given the highly correlated microarray data, we see that the sample size formula produces extremely false positive rates much lower than the nominal rate of δ.

Bonferoni simultaneous confidence interval. Suppose one is concerned with setting ϵ such that $P(\max_j \mid \mu_{jn} - \mu_j \mid /(\sigma_j/\sqrt{n}) > \epsilon) < \delta$ for a small number δ such as 0.05. Once this number is obtained then that yields a simultanous confidence band $\mu_{jn} \pm \epsilon\sigma_j/\sqrt{n}$ which contains with probability $1-\delta$ all μ_j simultaneously. It is easy to show that if all μ_{jn}, $j = 1, \ldots, p$, are pairwise independent, then $\epsilon \approx q_{1-\delta/(2p)}$, where $q_r = \Phi^{-1}(r)$ is the r-th quantile of the standard normal cumulative distribution function Φ. This choice of ϵ is referred to as the Bonferoni adjustment, which is thus conservative if the μ_{jn} happen to be dependent.

Bootstrapped simultaneous confidence interval. Let q be chosen so that the distribution of the resampled $\mu_n^{\#}$ is such that $P_n^{\#}(\max_j \frac{|\mu_{jn}^{\#} - \mu_{jn}|}{\sigma_{jn}/\sqrt{n}} > q) = 0.05$, i.e. the proportion of times across all bootstrapped samples that $\max_j \frac{|\mu_{jn}^{\#} - \mu_{jn}|}{\sigma_{jn}/\sqrt{n}} > q$ is smaller than or equal to 0.05. Now, a simultaneous 0.95-confidence interval for μ is given by $\mu_{jn} \pm q * \sigma_{jn}/\sqrt{n}$. The validity of this confidence interval relies on the consistency of the bootstrap and therefore it will be of interest to test its validity in a simulation study. The advantage of this simultaneous confidence interval is that it exploits the dependencies between the components of μ_n and will therefore be less conservative than the Bonferoni simultaneous confidence interval.

2.4.1 Consistency of the bootstrap.

Given this consistency of $\widehat{\mathbf{S}}_n$, we are now concerned with the asymptotic behavior of the feature-specific probabilities as $n \to \infty$. To be able to prove such a theorem, we need to consider a specified simple subset rule. For simplicity, we consider

$$\mathbf{S}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \{j : \mu_j, \mu_i > C_\mu, \rho_{ij} > C_\rho, \text{ for some } j \neq i\}, \tag{2.5}$$

This rule seeks genes that are over expressed and that have a large correlation with at least one other over expressed gene. Theorem 2.4.3 demonstrates that these probabilities converge to one (zero) when the appropriate feature is present (absent) in the target subset. For example, for genes j such that $\mathcal{S}_j > 0$, we have that $p_j \to 1$.

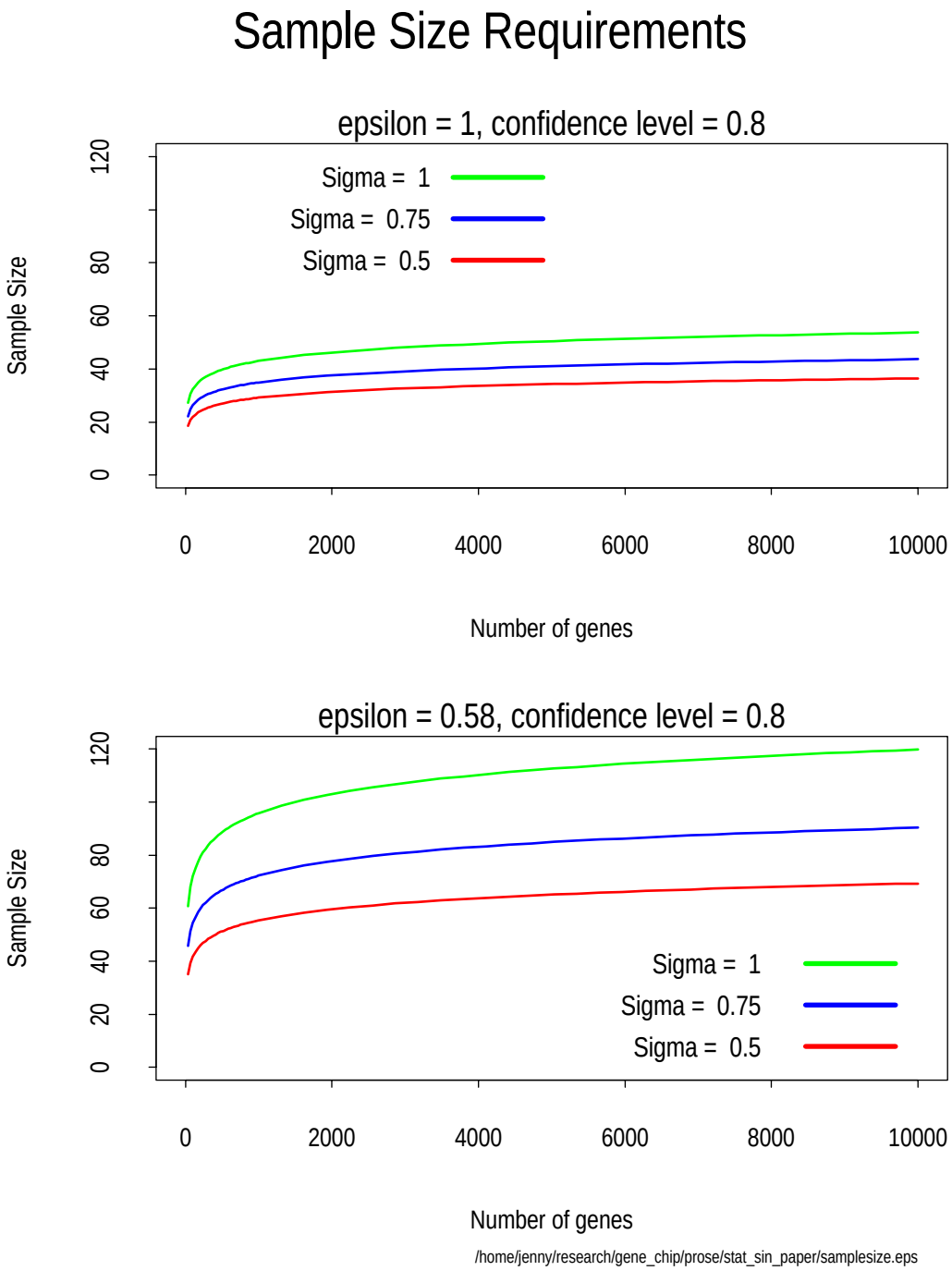

Figure 2.1: Sample size requirements for different situations.

Theorem 2.4.3 *Consider the simple subset rule 2.5. Let $p = \infty$ and $M < \infty$. Assume that C_μ and C_ρ are chosen so that the boundary condition $(*)$ of theorem 2.4.1 holds. Then the feature specific probabilities $p_{j,n}$ and $P_{ij,n}$ converge uniformly in i, j to the corresponding feature-indicators $I(j \in \mathcal{S})$ and $I((i,j) \in \mathcal{S})$, as $n \to \infty$.*

The following theorem proves consistency of the bootstrap estimates of the feature specific probabilities under the condition that $n/\log(p(n)) \to \infty$.

Theorem 2.4.4 *Consider the simple subset rule (2.5). Let $M < \infty$. Assume that C_μ and C_ρ are chosen so that the boundary condition $(*)$ of theorem 2.4.1 holds. If $n/\log(p(n)$ converges to infinity, then $\widehat{p}_{j,n}$ and $\widehat{P}_{ij,n}$ converge in probability uniformly in (i,j) to the feature-indicators $I(j \in \mathcal{S})$ and $I((i,j) \in \mathcal{S})$.*

In order to establish asymptotic consistency of $(\widehat{\boldsymbol{\mu}}_n, \widehat{\boldsymbol{\Sigma}}_n)$ and validity of the bootstrap at a non-degenerate level, we have proven an infinite dimensional central limit theorem for $\sqrt{n}(\mu_n - \mu)Var$, in which the latter are treated as elements of an infinite dimensional Hilbert space with a weighted Euclidean norm . Subsequently, we prove nonparametric asymptotic validity of the parametric bootstrap for the purpose of estimating the limiting distribution of $\sqrt{n}(\mu_n - \mu)$. Similarly, this can be proved for $\sqrt{n}(\Sigma_n - \Sigma)$ if one assumes the multivariate normal model. These proofs can be found in van der Laan and Bryan (2001).

In order to show formally that the parametric bootstrap is consistent for $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, one first establishes that $\sqrt{n}(\mu_n - \mu)Var$ converges in distribution to a Gaussian process as random elements of some Banach (or Hilbert) space endowed with the Borel sigma algebra. Subsequently, one shows that the bootstrap empirical process (sampling from $N_\infty(\widehat{\boldsymbol{\mu}}_n, \widehat{\boldsymbol{\Sigma}}_n)$) $\sqrt{n}(\mu_n - \mu)VarBoot$ converges in distribution to the same Gaussian process. If the latter holds, then one says that the bootstrap method is asymptotically valid for estimation of the distribution of $(\widehat{\boldsymbol{\mu}}_n, \widehat{\boldsymbol{\Sigma}}_n)$, considered as random elements of the Banach space.

Let $\mathbf{R}^\infty(\lambda_1)$ be the Hilbert space of infinite-dimensional vectors with inner-product $\langle x, y\rangle_1 = \sum_j x_j y_j \lambda_{1j}$, where it is assumed that $\sum_j \lambda_{1j} < \infty$. Then we can view $\sqrt{n}(\mu_n - \mu)$ as a random element of the Hilbert space $\mathbf{R}^\infty(\lambda)$. Similarly, let $\mathbf{R}^\infty(\lambda_2)$ be the Hilbert space of infinite dimensional vectors with inner-product $\langle x, y\rangle_2 = \sum_{ij} x_{ij} y_{ij} \lambda_{2,ij}$, where it is assumed that $\sum_{ij} \lambda_{2ij} < \infty$. Then we can view $\sqrt{n}(\Sigma_n - \Sigma)$ as a random element of $\mathbf{R}^\infty(\lambda_2)$. The potential limiting distribution of $\sqrt{n}(\mu_n - \mu)$ is determined by the multivariate CLT for any finite dimensional sub-vector of $\sqrt{n}(\mu_n - \mu)$ and is thus a Gaussian process $Z_1 = (Z_1(j) : j = 1, \ldots,)$. Similarly, the potential limiting distribution of $\sqrt{n}(\Sigma_n - \Sigma)$ is a Gaussian process $Z_2 = (Z_2(ij) : i, j)$. In Hilbert spaces an infinite dimensional central limit theorem follows from the point-wise central limit theorem and a rather weak tightness condition (see Appendix).

We have the following functional central limit theorem for $\widehat{\boldsymbol{\mu}}_n$ and $\widehat{\boldsymbol{\Sigma}}_n$.

Theorem 2.4.5 *Let $M < \infty$. We have that $\sqrt{n}(\mu_n - \mu)Var$ converges in distribution to the Gaussian process (Z_1, Z_2) as random elements in $\mathbf{R}^\infty(\lambda_1) \times \mathbf{R}^\infty(\lambda_2)$ endowed with the Borel sigma algebra.*

The following theorem proves that the bootstrap estimate of the distribution of $\sqrt{n}(\mu_n - \mu)$ is asymptotically consistent.

Theorem 2.4.6 *Let $M < \infty$ and let $\widetilde{\boldsymbol{\mu}}_n$ be the empirical mean vector based on sampling from $N_\infty(\widehat{\boldsymbol{\mu}}_n, \widehat{\boldsymbol{\Sigma}}_n)$. We have that $\sqrt{n}(\widetilde{\boldsymbol{\mu}}_n - \widehat{\boldsymbol{\mu}}_n)$ converges in distribution to the Gaussian process Z_1 (of theorem 2.4.5 above) as random elements in $\mathbf{R}^\infty(\lambda_1)$.*

We can also prove asymptotic validity of the parametric bootstrap for $\sqrt{n}(\Sigma_n - \Sigma)$, but that requires assuming that $Y \sim N_\infty(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

2.5 Data analysis

We examine a data set which is an example of an unpaired comparison with observations from two subpopulations. We extracted a publicly available data set from the data base accompanying Ross et al. (2000). The authors performed microarray experiments on 60 human cancer cell lines (the NCI60) derived from tumors from a variety of tissues and organs by researchers from the National Cancer Institute's Developmental Therapeutics Program. The data set includes gene expression measurements for 9,703 cDNAs representing approximately 8,000 unique transcripts. Each tumor sample was cohybridized with a reference sample consisting of an equal mixture of twelve of the cell lines chosen to maximize diversity. We used the normalized tumor:reference ratios, as in Ross et al. (2000). These were transformed to a log10 scale and truncated above and below, so that any ratio representing greater than 20-fold over- or under-expression was set to log10(20).

For this comparative analysis, we selected two very different types of cancer from those included in the NCI60: melanoma and breast. We created a data set with all samples from these two types of cancer, which included seven breast and eight melanoma cell lines. Next, we applied an initial subset rule in order to reduce the size of the data set for computational reasons only. We retained those genes where at least 30% of all cell lines had a ratio corresponding with greater than 2-fold over- or under-expression. Using a 30% cut-off, a gene differentially expressed in one type of cancer and not the other would still be included. There were 3500 genes in the resulting data set. This data set was divided into two smaller data sets consisting of the cell lines from each type of cancer. These data sets were analyzed separately and also combined into one data set by dividing the melanoma ratios by the geometric mean breast ratios before taking the log. Unless otherwise noted, we are working with the single, combined data set containing 3500 genes and eight observations.

One goal of the analysis is to identify genes differently expressed in melanoma relative to breast cancer; such genes help us to understand the biological characterization of different cancers and may lead to new cancer-specific treatments. Another goal revisited in chapter 3 is to study clustering patterns in the data set in order to discover information about how the genes involved in tumors work together.

2.5.1 Selecting differently expressed genes

A common approach to selecting differently expressed genes is to retain those genes whose absolute mean log ratios are greater than some cut-off value. In order to account for variance as well as mean expression, one can standardize the log ratios by dividing them by their gene-specific standard errors before taking the mean. These standardized means can be compared to the quantiles of a standard normal distribution on an individual basis. For the combined data set, $p^* = 1731$ genes were significantly differently expressed at the $\alpha = 0.05$ level (cut-off value$= z_{1-\frac{\alpha}{2}}/\sqrt{n} = 0.69$). Since we are in a multiple comparisons setting, it is advisable to adjust the cut-off value. The Bonferoni adjusted cut-off value was 1.53 and produced a much smaller subset of $p^* = 605$ differently expressed genes. As mentioned above, the Bonferoni adjustment is sharp if the genes are independent, but is conservative otherwise.

An alternative, less conservative approach is to derive a cut-off value from an appropriate null distribution with zero means and the true covariance structure. A parametric method is to generate a large number of samples from a multivariate normal distribution $N(0, \rho)$, where ρ is the correlation matrix, and select a cut-off value such that no more than $1 - \frac{\alpha}{2}$ of samples have any differently expressed genes. The correlation matrix ρ can be estimated by the empirical correlation matrix. A non-parametric method is to standardize the observed data so that each gene has mean zero and variance one, then generate a large number of bootstrap samples from this data (resampling cell lines with replacement), and use these to compute the cut-off value such that no more than $1 - \frac{\alpha}{2}$ of samples have any differently expressed genes. For both the parametric and non-parametric methods, a less stringent approach is to choose the cut-off value such that on average any sample is expected to have no more than $1 - \frac{\alpha}{2}$ of genes differently expressed. We used the nonparametric bootstrap with the more stringent criteria and obtained a subset of $p^* = 889$ genes. The cut-off value was 1.17, which lies between the value which ignores the multiple comparisons and the too strict Bonferoni adjusted value.

2.6 Simulation to assess variability of empirical relative gene expression and empirical pairwise distance between genes.

Let $\mathbf{X}$ be the p-dimensional column vector of gene-specific relative gene expressions representing the relative gene expression profile for a randomly drawn subject. Thus we observe n i.i.d. copies $\mathbf{X}_1, \ldots, \mathbf{X}_n$ of this random vector $\mathbf{X}$. The total data set "Gene-ID", $\mathbf{X}_1, \ldots, \mathbf{X}_n$, can be represented by an array with p rows and n columns.

Let $Y = \log(\mathbf{X})$ be the vector of truncated log-ratios and $Y_1, \ldots, Y_n$ are the i.i.d observations of the p-dimensional vector Y. In these experiments we are particularly concerned with estimation of

$$\begin{aligned} \mu &= EY \text{ the vector of gene-specific population means} \\ \Sigma &= E(Y-\mu)(Y-\mu)\top \text{the p by p matrix of covariances} \end{aligned}$$

and the corresponding p by p correlation matrix ρ which one can compute from Σ:

$$\rho_{ij} = \frac{\Sigma_{ij}}{\sigma_i \sigma_j}, i, j \in \{1, \ldots, p\}.$$

In particular, our subset rules $S(\mu, \Sigma)$ maps these unknown parameters into a subset of genes which is believed to be of interest for drug-development.

Let μ_n, Σ_n, ρ_n be the empirical counterpart of μ, Σ, ρ. In practice the subset of genes we are going to select is $S(\mu_n, \Sigma_n)$. Therefore this estimated subset will only be close to the wished subset if μ_n and Σ_n are close to μ and Σ-respectively. For example, if μ_{nj} deviates more than $\log(3) - \log(2)$ from the true μ_j, then the fact that $\mu_{nj} > \log(3)$ (i.e. it is 3-fold differentially expressed on average among the selected patients) does not imply that gene j is in truth 2-fold differentially expressed. Therefore we are interested in determining a sample size so that we can trust that the true means and true correlation are within a reasonable distance from the observed means and correlations for each of the p genes.

A subset rule can still have a good performance when there are some highly wrong empirical correlations. In particular, one can design a subset rule which protects oneself against false

Table 2.2: Quantiles of standard deviations σ_j, $j = 1, \ldots, p$ based on data set of a 12 colon-cancer patients using truncation $M = \log(5) = 1.6, M = \log(7.5) = 2$ and $M = \log(10) = 2.3$, respectively.

M	0.5	0.7	0.9	max
1.6	0.48	0.55	0.67	1.38
2	0.51	0.60	0.74	1.55
2.3	0.52	0.62	0.78	1.74

positives due to high empirical correlations. To start with one can just use a higher rate of initial screening so that it becomes much harder to make it into the clustering routine. For example, one only selects highly differentially expressed genes (so that one end up clustering e.g. only 100 genes) or one only selects genes with relatively small standard deviation. Moreover, we can thin out the clusters by requiring strong correlations with the centers (medoids) of the clusters: in this way, a gene which happens to have a strong correlation with various genes while in truth it is not correlated at all might still not make the subset since it needs to be strongly related with the actual center of the cluster. Suppose that one is interested in determining which genes among $m \leq p$ selected genes cluster with a *given* gene. In this case one just needs to estimate m pairwise correlations of genes with the given gene. The bootstrap can be used to compare various subset rules w.r.t to their performance in finding the correct subset of genes and the number of false positives.

In this section we study the distribution of the maximal difference (over all p genes) between true and observed averages, true and observed standard deviations and true and observed correlations under various sample sizes. In order to do this at an appropriate noise level we use a noise level observed in an actual data set of 12 colon cancer patients. In order to create a worst case scenario and to protect outselves against false positives we will determine these distributions in the context that all genes are unrelated: if many genes are correlated than the true dimension of the problem can be much lower than p. We also study the performance of empirical correlations when the true correlation is high in order to determine how well we can do in discovering all highly correlated genes.

The observed noise level in a data set depends on the truncation level M one uses: the larger M the larger the noise level. We first compute the 0.5,0.7,0.9-quantile and maximum of the p standard deviations of the by M truncated log-ratios of a colon-cancer data set with 12 patients. Table 2.2 reports these quantiles for $M = \log(5), \log(7), \log(10)$.

2.6.1 Sample sizes for estimation of the population mean.

Consider a $p = 10000$ dimensional vector of log-relative gene-expressions where each component is truncated by M. Let σ_j be the standard deviation of component j and let $\sigma = \max_j \sigma_j$. The following function computes the sample size

$$n^* = \frac{2\sigma^2 + 2M\epsilon/3}{\epsilon^2} \{\log(p) + \log(2/\delta)\}$$

such that with probability $1 - \delta$ the maximal difference $\max_j \mid \mu_{nj} - \mu_j \mid$ over p genes is smaller than ϵ.

Table 2.3: Consider a $p = 10,000$ dimensional vector of log-relative gene-expressions where each component is truncated by M. Let σ_j be the standard deviation of component j and let $\sigma = \max_j \sigma_j$. The following table gives the sample sizes such that with probability 0.95 the maximal difference $\max_j \mid \mu_{nj} - \mu_j \mid$ over $p = 10000$ genes is smaller than ϵ. We provide this sample size for $M = \log(5), \log(7.5), \log(10,$ corresponding 0.5,0.7,0.9,1-quantiles of the p-standard deviations as observed in the colon-cancer data set and $\epsilon \in \{0.1, 0.2, \ldots, 1\}$.

M,Sigma	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
1.6,0.48	732	217	112	72	51	39	32	26	23	20
1.6,0.55	918	264	133	83	59	45	36	29	25	22
1.6,0.67	1296	358	175	107	74	55	43	35	30	25
1.6,1.38	5051	1297	592	341	224	159	120	94	76	63
2,0.51	843	254	132	85	61	47	38	32	27	24
2,0.60	1101	318	161	101	72	54	44	36	31	26
2,0.74	1585	439	214	131	91	68	53	44	37	31
2,1.55	6370	1636	746	430	282	201	151	118	96	79
2.3,0.52	895	273	143	93	67	52	42	36	31	27
2.3,0.62	1189	347	176	111	79	61	48	40	34	30
2.3,0.78	1767	491	240	148	102	77	60	49	41	35
2.3,1.74	8009	2052	934	538	352	250	188	147	118	98

Table 2.4: Below we report the median, 0.7-quantile and 0.9 quantile of the distribution of the maximal difference $\max_{j\in\{1,\ldots,p\}} \mid \mu_{nj} - \mu_j \mid$ between $p = 10,000$ sample means and true means. Here μ_{nj} is the sample mean of logratios based on n observations with $N(mu_j, sigma = 0.55)$ distribution, $j = 1, \ldots, 10000$.

n	0.5-q	0.7-q	0.9-q
n=15	0.56	0.59	0.63
n=30	0.40	0.41	0.43
n=60	0.28	0.29	0.31
n=100	0.22	0.22	0.24
n=150	0.18	0.19	0.20

Since n^* only depends on p through $\log(p)$ the effect of p on the required sample size n^* is very minimal. For example, suppose the noise level is $\sigma = 0.55$, $M = \log(5)$ and one wants to be sure that the sample means of p genes are within a distance $\epsilon = 0.5$ of the true means, then the required sample size for $p = 1$ is 17 and the required sample size for $p = 100000$ is 69.

The following table 2.3 provides now these sample sizes for $M = \log(5), \log(7.5), \log(10)$ and the values of σ as computed in the table above.

This sample size formula is a conservative formula since it holds for any data generating distribution function. We will now carry out a simulation to determine the distribution of $\max_{j\in\{1,\ldots,p\}} \mid \mu_{nj} - \mu_j \mid$ for $p = 10,000$ at a noise level corresponding with the 0.7-quantile of the σ_{nj}'s we observed in a coloncancer patient data set at truncation level $M = \log(5)$. In all these simulations we assume that all p genes are uncorrelated which corresponds with a worst case scenario. For example, if $n = 15$, then with probability 0.9, each of the 10,000 observed averages are within a distance 0.63 of the truth.

Table 2.5: Below we report the median, 0.7-quantile and 0.9 quantile of the distribution of the maximal diffference $\max_{j\in\{1,\ldots,p\}} \mid \sigma_{nj} - \sigma_j \mid$ between $p = 10,000$ sample standard deviations and true standard deviations. Here σ_{nj} is the sample standard deviations of logratios based on n observations with $N(mu_j, sigma = 0.55)$ distribution, $j = 1, \ldots, 10000$.

n	0.5-q	0.7-q	0.9-q
n=15	0.40	0.43	0.46
n=30	0.28	0.3	0.31
n=60	0.2	0.21	0.22
n=100	0.15	0.16	0.18
n=150	0.13	0.13	0.14

Table 2.6: Below we report the median, 0.7-quantile and 0.9 quantile of the distribution of the maximal diffference $\max_{i<j\in\{1,\ldots,p\}} \mid \rho_{n,ij} - \rho_{ij} \mid$ of the sample correlations and true correlations which are equal to zero. Here $\rho_{n,ij}$ is the sample correlation of uncorrelated logratios for genes i and j with standard deviation $\sigma = 0.55$ and $p = 1000$.

n	0.5-q	0.7-q	0.9-q
n=15	0.92 0.93 0.94		
n=30	0.76 0.77 0.77		
n=60	0.58 0.59 0.6		
n=100	0.46	0.47	0.48
n=150	0.38	0.39	0.41

2.6.2 Sample size for estimation of the standard deviations.

Various interesting subset rules rely on estimates of the gene-specific standard deviations σ_j. Therefore it is also of interest to know with high certainty that the true standard deviation is within a reasonable distance from the observed standard deviation.

2.6.3 Sample size for estimation of the correlations.

If a pair of genes have an observed correlation larger than a certain number, then that might be an important finding in the process of drug development. For example, one might find that an unknown gene has a large correlation with a gene which is well known (from the literature) to be an important cause of cancer. In that case, one might decide to carry out experiments controlling this unknown gene. In addition, if one observes clusters of genes in the data then that will be interpreted as that genes are working together. Our cluster routines are purely functions of the observed correlations and can thus only be trusted if we do a good job in estimating the true correlation matrix of the genes we decided to cluster. Note that we typically only cluster a subset of all p genes: for example, we might just cluster all 3-fold differentially expressed genes. Therefore we are particularly interested to know what sample size we need to estimate a 300 by 300 (if we cluster 300 genes) or at most 1000 by 1000 correlation matrix. Table 2.6 reports the performance in estimation of a 1000 by 1000 diagonal correlation matrix (i.e. all correlations are zero). Table 2.7 reports the performance in estimation of a 300 by 300 diagonal correlation matrix (i.e. all correlations are zero).

Suppose now that all the true correlations one is interested in are high. Now, one might

Table 2.7: Below we report the median, 0.7-quantile and 0.9 quantile of the distribution of the maximal diffference $\max_{i<j\in\{1,\ldots,p\}} \mid \rho_{n,ij} - \rho_{ij} \mid$ of the sample correlations and true correlations which are equal to zero. Here $\rho_{n,ij}$ is the sample correlation of uncorrelated logratios for genes i and j with standard deviation $\sigma = 0.55$ and $p = 300$.

n	0.5-q	0.7-q	0.9-q
n=15	0.87	0.89	0.91
n=30	0.69	0.71	0.73
n=60	0.52	0.54	0.57
n=100	0.42	0.42	0.43
n=150	0.33	0.34	0.34

Table 2.8: Below we report the median, 0.7-quantile and 0.9 quantile of the distribution of the maximal difference $\max_{j\in\{1,\ldots,p\}} \mid \rho_{n,j} - \rho_j = 0.8 \mid$ of the $p = 100,000$ sample correlations and true correlations. Here $\rho_{n,j}$ is the sample correlation based on n observations of $X \sim N(0, 0.55)$,Y with $Y = 0.445X + N(0, 0.2)$ so that the true correlation between X and Y is 0.77.

n	0.5-q	0.7-q	0.9-q
n=15	1.1	1.1	1.2
n=30	0.59	0.61	0.64
n=60	0.35	0.36	0.38
n=100	0.24	0.25	0.27
n=150	0.18	0.18	0.19

want to be able to discover them all as highly correlated pairs of genes. Table 2.8 provides the performance in estimating 100,000 highly correlated pairs simultaneously at various sample sizes.

Suppose now that all the true correlations one is interested in are high. Now, one might want to be able to discover them all as highly correlated pairs of genes. Table 2.8 provides the performance in estimating 100,000 highly correlated pairs simultaneously at various sample sizes.

Suppose now that one gene j^* is given and one just wants to estimate the the p correlations ρ_{jj^*} with this gene j^*, $j = 1, \ldots, p$. Tables 2.9 and 2.10 provides the performance in estimating p zero-correlations for various sample sizes and $p = 1000, 300$. Table 2.11 provides the performance in estimation of $p = 300$ high correlations for various sample sizes.

2.7 Simulation to assess clustering performance

The goal of this simulation study is to explore the performance of $\widehat{\mathbf{S}}_n$ and the bootstrap in the context of a known data-generating distribution. We are particularly interested in assessing the difficulty of applying cluster labels in the presence of genes that belong to no cluster and how that is affected by sample size. The simulation shows that it is beneficial to screen unrelated genes prior to applying a clustering algorithm. We also see that unrelated genes tend to depress conventional measures of the clustering strength. Lastly, it is apparent that post-screens affected by isolated extreme values, such as the smallest entries in a column of a correlation matrix, will require large sample sizes to achieve good performance of $\widehat{\mathbf{S}}$ and alternative screens should be

Table 2.9: Below we report the median, 0.7-quantile and 0.9 quantile of the distribution of the maximal difference $\max_{j\in\{1,\ldots,p\}} \mid \rho_{n,jj^*} - \rho_{jj^*} \mid$ of the $p = 1000$ sample observed correlations and true correlations with a fixed gene j^*. Here ρ_{n,jj^*} is the sample correlation based on n observations of $X_{j*} \sim N(0, 0.55)$, $Y_j \sim N(0, 0.55)$.

n	0.5-q	0.7-q	0.9-q
n=15	0.78	0.8	0.85
n=30	0.58	0.61	0.67
n=60	0.43	0.45	0.49
n=100	0.33	0.35	0.37
n=150	0.27	0.29	0.31

Table 2.10: Below we report the median, 0.7-quantile and 0.9 quantile of the distribution of the maximal difference $\max_{j\in\{1,\ldots,p\}} \mid \rho_{n,jj^*} - \rho_{jj^*} \mid$ of the $p = 300$ sample observed correlations and true correlations with a fixed gene j^*. Here ρ_{n,jj^*} is the sample correlation based on n observations of $X_{j*} \sim N(0, 0.55)$, $Y_j \sim N(0, 0.55)$.

n	0.5-q	0.7-q	0.9-q
n=15	0.72	0.74	0.79
n=30	0.55	0.57	0.61
n=60	0.38	0.41	0.44
n=100	0.3	0.32	0.36
n=150	0.25	0.26	0.28

Table 2.11: Below we report the median, 0.7-quantile and 0.9 quantile of the distribution of the maximal difference $\max_{j\in\{1,\ldots,p\}} \mid \rho_{n,jj^*} - \rho_{jj^*} \mid$ of the $p = 300$ sample observed correlations and true correlations with a fixed gene j^*. Here ρ_{n,jj^*} is the sample correlation based on n observations of $X_{j*} \sim N(0, 0.55)$, $Y_j = 0.445X_{j*} + Z_j$, $Z_j \sim N(0, 0.55)$, so that the true correlations are 0.77.

n	0.5-q	0.7-q	0.9-q
n=15	0.34	0.45	0.65
n=30	0.22	0.27	0.36
n=60	0.15	0.18	0.21
n=100	0.12	0.14	0.17
n=150	0.092	0.11	0.13

considered.

2.7.1 Data-generating distribution

We create a data-generating distribution by assuming a multivariate normal model and selecting the parameters $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The first priority is to impose a cluster structure; in particular, we want $K = 3$ clusters of genes. There are three clusters – cluster A, cluster B, and cluster C – and each contains 100 genes. Each cluster has a core set of genes that are more highly correlated with one another and a more weakly correlated set of peripheral genes. The genes in a given cluster have no correlation to genes in the other clusters. The clustered genes are embedded in a set of 300 other genes that have absolutely no correlation with other genes at all. The correlation matrix of the full set of 600 genes $\boldsymbol{\rho}$ is block diagonal. With this set-up we are trying to simulate what seems to be an important data structure: a fraction of the genes being studied are involved in the phenomenon of interest and even break down into several well-defined clusters, but there are many "noisy" genes on the array, which are not involved and whose presence makes it difficult to find the relevant clusters.

The mean expression levels are also set with the cluster structure in mind. The noisy genes have means near zero, with some individual genes exhibiting a mild amount of differential expression. Cluster A contains genes that are over-expressed, many quite strongly. Most genes in Cluster B are differently expressed, with slightly more being under-expressed than over-expressed. Cluster C contains genes with a wide range of expressions. Gene-specific standard deviations have different distributions for each cluster and for the noisey genes. Throughout this section, we use yellow for cluster A, violet for cluster B, and blue for cluster C.

2.7.2 Subset rule

The subset rule is applied to the true mean and covariance (not simulated data), so that we can examine properties of the target subset $\mathcal{S}$. The rule we use is typical of those applied in many microarray data analyses: first, screen for differently expressed genes and then apply cluster analysis. We will exclude genes with an absolute mean $|\mu_j| < \log_2 1.5 = 0.58$, which corresponds to 1.5-fold differential expression. Of the 600 genes, 318 are retained and 282 are excluded based on this screen.

The remaining 318 genes are provided to a cluster analysis routine, with the dissimilarity between two genes defined as 1 minus the absolute value of the correlation. For a fixed number of clusters K, a partitioning method finds the best grouping and, by exploring different values of K, we can assess the evidence for different K values (see Kaufman and Rousseeuw, 1990, chap. 1, sect. 3). It is also valuable to have a way to assign meaning to a particular cluster label k; most scientific papers that employ cluster analysis to analyze microarray data discuss the unifying theme of the genes found in each cluster. In the context of one data set, any clustering algorithm will likely yield at least one partition that can be interpreted. However, when one views a data set and its clustering as just one realization of a stochastic phenomenon, it is desirable to have a way to enforce a coherent meaning for cluster label k. The cluster centers that are important in most partitioning methods play this role very well. By fixing cluster centers, one can ensure it is sensible to compare genes with label k from one realization of the experiment to the next. Lastly, we prefer an algorithm called "partitioning around medoids" (PAM) (Kaufman and Rousseeuw, 1990, chap. 2) to k-means because we like being able to use any distance metric and we prefer that cluster centers be one of the underlying objects (in this

Table 2.12: Average Silhouettes for $K = 2, 3, 4$ in simulation study.

Which genes?	K	Overall Avg. Silh.	Cluster 1	Cluster 2	Cluster 3	Cluster 4
318	2	0.09	0.06	0.17		
genes	3	0.13	0.12	0.17	0.11	
(67 noise)	4	0.09	0.12	0.02	0.04	0.11
251	2	0.07	0.03	0.17		
genes	3	0.09	0.04	0.17	0.10	
(no noise)	4	0.05	0.04	0.06	0.02	0.10

case, a gene) instead of an average of objects, a quantity that is difficult to interpret and less robust to outliers.

Given any partition, Kaufman and Rousseeuw (1990) define for each object a quantity called the silhouette, which reflects how well-matched an object is in its cluster versus the next closest cluster. Silhouettes take values in the interval $[-1, 1]$, with 1 corresponding to a perfect match. Silhouettes are a valuable tool for assessing what is basically the goodness-of-fit for a clustering. For a given data set and clustering method, silhouettes can be compared for different numbers of clusters in order to choose the optimal number. There were 251 genes in clusters A, B, and C that passed the differential expression screen (318 - 251 = 67 noise genes pass the screen, but have silhouettes of zero). Silhouettes were examined for $K = 2, 3$, and 4. When $K = 2$, we see that cluster B is fully recovered, while clusters A and C are lumped together. When $K = 3$, which we know to be the correct value of K, and we see that PAM recovers the underlying clusters. When $K = 4$, clusters A and C are fully recovered and Cluster B is split into two. The lack of evidence for $K = 4$ is apparent in the erratic, even negative, silhouettes for genes in cluster B. The core versus periphery structure of the underlying clusters is also reflected in the silhouettes. Table 2.12 presents average silhouettes for these clusterings, with and without the 67 noise genes that pass the differential expression screen. We see that the overall average silhouette is highest at the correct value of K, which is 3, regardless of the presence of the noise genes. But the noise genes have a dampening effect on the silhouettes in general. This points out the benefit of eliminating all unrelated genes prior to attempting any type of cluster analysis.

After clustering the genes, we chose to apply one last screen in another attempt to eliminate uninteresting genes. The goal is to remove genes that are not particularly well-matched to their cluster. We used two different approaches, one based on pairwise dissimilarities and one based on silhouettes. The dissimilarity screen DYS works in this manner: the cluster center (or "medoid") is automatically included. Any gene with a dissimilarity of less than 0.655 with the cluster center is included. Any gene with a dissimilarity of less than 0.655 with any previously included gene is also included. This last step is repeated until no changes occur. The silhouette screen SILH includes all genes with a silhouette greater than 0.08. Both of the screens result in target subsets $\mathcal{S}$ containing 150 genes. Table 2.13 presents target subset $\mathcal{S}$ membership by true cluster membership for both screens.

To summarize the subset rule, the genes were first screened for differential expression by requiring that $|\mu_j| > 0.58$. The remaining 318 genes are clustered by PAM, with the cluster number $K = 3$. The cluster centers are noted and will be fixed in future analyses. In light of the clustering, genes are screened again based either on dissimilarities or silhouettes to yield a

Table 2.13: Target subset membership by true cluster membership.

	Target Subset $\mathcal{S}_j$ =					
True	0	1	2	3	> 0	All
DYS screen						
Noise	300				0	300
Cluster A	50	50			50	100
Cluster B	27		73		73	100
Cluster C	73			27	27	100
	450	50	73	27	150	600
SILH screen						
Noise	300				0	300
Cluster A	71	29			29	100
Cluster B	12		88		88	100
Cluster C	67			33	33	100
	450	29	88	33	150	600

final subset containing 150 genes and their cluster labels.

2.7.3 Sampling distribution of $\widehat{\mathbf{S}}_n$

We generated 100 samples of size $n = 25, 50$, and 150 from the chosen data-generating distribution $N((\boldsymbol{\mu}, \boldsymbol{\Sigma}))$ and applied the two subset rules described above. Based on these samples, we can estimate the reappearance probabilities p_j and p_j^k. In Figure **??**, we examine the effect of sample size in the DYS screen. The results are somewhat counter-intuitive but illustrate an important phenomenon. At the series of sample sizes considered here ($n = 25, 50, 150$), overall p_j tend to decrease for all genes. Average p_j within different values of $\mathcal{S}$ are presented in the lower left panel. But it is important to examine the cluster-specific reappearance probabilities. The top panel presents this information for 4 typical genes, one for each value of $\mathcal{S}$, and the lower right panel presents averages within values of $\mathcal{S}$. We see that, while overall p_j may be declining, the correct cluster-specific p_j^k are climbing steadily. One expects that, had we added a larger sample size such as $n = 300$, even the overall p_j would begin to increase as n does.

The results of this simulation demonstrate that the mean requires much less data to estimate than the covariance structure. For all sample sizes, the expected number of genes passing the differential expression screen is very close to the true number of 318. It is approximately 325, 323, and 321 for $n = 25, 50$, and 150, respectively. From other simulations not reported here, in which the subset rule consists solely of the differential expression screen, we know that both sensitivity and positive predictive value at this stage are extremely high (between 0.95 and 0.99) and, therefore, the *correct* genes are almost always passing this initial screen at all sample sizes. The problem occurs in the clustering and DYS screen – that is, the steps of the subset rule that depend on the covariance. At $n = 25$, many genes are misclassified into the incorrect cluster, but frequently pass the dissimilarity screen due to sampling variability in the covariance. Since a gene can pass this screen by exhibiting even one extremely small pairwise distance, it is almost always passed for small samples. Therefore, the probability of appearing in the $\widehat{\mathbf{S}}_n$ has significant contributions from all three cluster-specific probabilities p_j^1, p_j^2, and p_j^3. This can be seen in the first stacked column for each of the 4 genes highlighted in the top panel of Figure **??**. As the

Table 2.14: Cluster-wide quality measures for the DYS rule in the simulation study.

	n = 25	n = 50	n = 150
E{Sens}	0.98	0.97	0.77
E{PPV}	0.45	0.50	0.84
E{PEFP}	0.00	0.00	0.00
PAFP	0.48	0.09	0.00

sample size increases to 50 and 150, this misclassification decreases and p_j becomes dominated by the correct cluster-specific probability. This can be seen in the second and third stacked columns. These simulation results suggest a modification of the DYS screen in which a gene must have a sufficiently small dissimilarity specifically with the cluster center.

The behavior described above is also apparent in subset-wide measures of quality, reported in Table 2.14. As expected, the sensitivity decreases at these sample sizes, but the positive predictive value increases. Once again, we conjecture that the sensitivity would increase for $n > 150$. Extremely false positives were defined as genes with absolute mean expression less than $\log_2 1.1 \approx 0.14$. The expected proportion of extremely false positives (E{PEFP}) is essentially zero for all n and the probability of any extremely false positives (PAFP) decreases as n grows.

The situation is quite different for the subset rule SILH that screens based on the silhouettes. Summary information on p_j and p_j^k is depicted graphically in Figure **??**. It is immediately apparent that the reappearance probabilities are much lower in general than those seen with the DYS rule. This is due to the fact that, compared to the silhouettes produced by the true block diagonal correlation matrix, the silhouettes in observed data are lower. The average silhouette in the target subset $\mathcal{S}$ is 0.09. The expected average silhouette in the sample subset $\widehat{\mathbf{S}}$ is 0.02. The non-zero empirical correlation that arises between even unrelated genes has the effect of making the clustering appear to be less strong. Therefore, when applying the silhouette cutoff to a clustering based on a finite amount of data, we are left with a smaller set of genes. The average size of $\widehat{\mathbf{S}}_n$ is approximately 32, 27, and 43 for $n = 25, 50$, and 150, respectively. Both the expected subset size and the p_j and p_j^k seem to grow very slowly as the sample size increases. We have also noted here and in other analyses that the values of the silhouettes are very dependent on the dimension of the data set (number of genes), so that universal cutoff values as described in Kaufman and Rousseeuw (1990) are not appropriate in the gene expression context. One screen that may be more useful than absolute cutoffs based on silhouettes is to always retain a fixed number of top-ranked genes based on silhouettes or estimated cluster-specific probabilities. If one wishes to test the significance of a silhouette, we propose using a simulation from an appropriate null distribution (i.e.: one with no clustering).

Table 2.15 presents subset-wide measures of quality for the SILH rule. Both sensitivity and positive predictive value increase with n and the probability of any false positive is extremely small even at $n = 25$ and quickly falls to zero.

We are also interested in the actual gene-specific probabilities p_j and p_j^k. For genes in $\mathcal{S}$ for the DYS rule, although the overall p_j decrease, the correct cluster-specific probabilities p_j^k increase with n. In fact, at $n = 150$, essentially no genes appear in $\widehat{\mathbf{S}}$ carrying the incorrect cluster label. This observation supports the above discussion of the DYS rule. Consistent with the above findings regarding the stringency of the silhouette-based screen, we see relatively low p_j, which grow very slowly with n, for the SILH rule. The misclassification of genes is practically

Table 2.15: Cluster-wide quality measures for the SILH rule in the simulation study.

	n = 25	n = 50	n = 150
E{Sens}	0.18	0.18	0.28
E{PPV}	0.86	0.98	0.99
E{PEFP}	0.00	0.00	0.00
PAFP	0.04	0.00	0.00

impossible with this rule.

2.7.4 Bootstrap results

For each of the 100 size n samples generated from the data-generating distribution $N((\boldsymbol{\mu}, \boldsymbol{\Sigma}))$, we carried out the parametric bootstrap as described in van der Laan and Bryan (2001). Since the simulated data is multivariate normal distributed here, the use of this distribution in the bootstrap is appropriate. The empirical distribution of the bootstrap subsets allows us to estimate interesting features of the sampling distribution of $\widehat{\mathbf{S}}$. The probability of gene j appearing in $\widehat{\mathbf{S}}_n$, i.e. p_j, is estimated by the proportion of bootstrap subsets in which gene j appears. An analogous approach leads to estimates of p_j^k. Figures **??** and **??** plot true reappearance probabilities against average bootstrap probabilities for the DYS and SILH rules, respectively.

In finite samples, the expected bootstrap probabilities are biased estimators of the true probabilities. For certain simple rules, this bias is relatively straightforward to quantify and is discussed in **?**). For complicated rules such as DYS and SILH, the only relevant result is that, as $n \to \infty$, the expected bootstrap probabilities will approach 1 for genes in $\mathcal{S}$ and 0 for all other genes. Graphically, this means that as $n \to \infty$, we will eventually see points only at (0,0) and (1,1). But for finite n, plots such as **??** and **??** are the best way to understand the relationship between the expected bootstrap and true reappearance probabilities.

2.7.5 Distribution of the sample mean

For a fixed n and δ, the formula stated below in equation 2.4.2 can be solved for the ϵ such that the probability of even one component of the p-dimensional sample mean $\widehat{\boldsymbol{\mu}}_n$ varying by more than ϵ from the corresponding component of the true mean $\boldsymbol{\mu}$ is less than $0 < \delta < 1$. The sample size is quite conservative, since it does not exploit the correlation among the genes. That is, when one computes values of $\epsilon > 0$ as described below, the actual probability of $\max_j |\widehat{\mu}_j - \mu_j| > \epsilon$ is much less then δ. Table 2.16 illustrates this and also shows that one can use a value of σ that is much smaller than the actual maximum of the gene-specific log ratio standard deviations and still see favorable results. In all instances, $n = 25$ and $M = 5$.

An alternative, less conservative approach to determining the sample size needed for a certain precision is to perform simulations utilizing the correlation structure in the data. By the central limit theorem, we have that the sample mean $\hat{\mu}$ is asymptotically distributed $N(\hat{\mu}_{\mathbf{n}}, \hat{\boldsymbol{\Sigma}}_{\mathbf{n}})$. By simulating from this distribution, we can determine the sample sizes needed for different levels of precision. Non-parametric simulations could also be employed.

Table 2.16: Demonstration that the sample size formula is conservative.

Nominal δ	ϵ	Actual δ	σ
0.05	2.64	0.000	$\max_j \sigma_j = 2.06$
0.50	2.27	0.000	$\max_j \sigma_j = 2.06$
0.20	1.52	0.005	75-th quantile of $\sigma_j = 0.89$
0.40	1.14	0.030	25-th quantile of $\sigma_j = 0.37$
0.80	1.01	0.055	0.25

2.7.6 Conclusions

These simulations illustrate some important issues encountered in cluster analysis of gene expression data. In particular, we see that sampling variability of the covariance structure and the presence of unrelated genes can have a strong impact on partitioning algorithms and measures of cluster strength and stability. We have found that pre- and post-screening of the genes helps to avoid some of these problems. The simulations show that screens based on differential expression are accurate even for small sample sizes, whereas screens based on the covariance are harder to estimate accurately. One drawback of screening the genes, however, is that important or interesting genes can be excluded along with the "noisy" genes we wish to remove.

In response to this issue, we have developed an algorithm called Hierarchical Ordered Partitioning And Collapsing Hybrid (HOPACH), which incorporates both partitioning and agglomerative steps in order to identify clustering patterns in the data even in the presence of many unrelated genes. We have conducted simulations which illustrate that this methodology does better than simple partitioning or agglomerative methods at identifying small clusters in the presence of many noisy genes (van der Laan and Pollard (2001)). In Section 2.5 we outline the HOPACH method and apply it to a cell line data set with two subpopulations. We also demonstrate methods for selecting differently expressed genes using a null distribution.

2.8 Simulations to compare different bootstrap methods.

We report here on a simulation study carried out in Pollard and van der Laan (2001). The nonparametric bootstrap has the advantage of being computationally much easier than the parametric bootstrap. In addition, the nonparametric bootstrap avoids distributional assumptions about the parameter of interest, whereas the estimation of the distribution of $\sqrt{n}(\Sigma_n - \Sigma)$ using the parametric bootstrap is only consistent under the model assumption. There is reason to believe, however, that the parametric bootstrap might perform better in the gene-expression context, where the number of observations n is typically very small relative to the dimension p (number of genes). The performance of the bootstrap is measured by how well the distribution of $\theta_n^{\#}$ approximates the distribution of θ_n. It is clear that this performance is mainly dependent on how close $\mathbf{P}_n$ is to P. Our initial feeling was that in this setting, the empirical distribution P_n (i.e. nonparametric bootstrap) might be an inappropriate estimate of P. Another fact of interest is that the nonparametric bootstrap is known to be inconsistent in various low-dimensional examples, while the parametric bootstrap is consistent under minimal additional assumptions given that the parametric model is correct Giné and Zinn (1990).

With these ideas in mind, we conducted a simulation study to assess the asymptotic validity of the nonparametric, convex, and parametric bootstraps for estimating the distribution of a

gene clustering parameter. We used $p = 3000$ genes and $n = 40$ samples. These choices reflect typical dimensions of the data matrix X (possibly after prescreening) as seen in commercial and academic settings. In order to investigate the effect of asymptotics on our results, we repeated the simulations using $n = 250$ samples.

2.8.1 Simulation: Multivariate Normal Data (with diagonal covariance)

This simulation investigates gene clustering. The true data generating distribution was chosen to be a multivariate normal with diagonal covariance matrix so that the genes were uncorrelated. For simplicity, a fourth of the genes was generated from each of four distributions: $N(0.5, 0.25)$, $N(-0.5, 0.5)$, $N(1, 1)$, $N(-1, 0.75)$. The summary measures of interest were selected to be the 0.9 quantile of the maximum absolute difference in the mean vector, median vector, and correlation matrix. These measures give a good indication of how far a distribution is from the truth.

In order to define the "true" values of the summary measures, a large number N draws from the true distribution were compared to the known mean, median and correlation. Results were compared for $N = 100, 1000, 10000$ and showed little dependence on N so that $N = 100$ was deemed sufficient. Next, a single draw from the true distribution was identified as the "observed" data and the three types of bootstrap were performed with convex repeated for $d = 0.1, 0.3, 0.5$. In each case, $B = 100$ bootstrap samples were generated from which the 0.9 quantiles were calculated. In order to investigate the variability of these measures, we repeated each simulation twenty times with $n = 40$, obtaining twenty sets of 0.9 quantiles. From these, we calculated a mean and standard deviation. The coefficient of variation was on the order of 2.5% for the mean, 3.0% for the median and 1.25% for the correlation. These values were sufficiently small that we chose to use the results from just one simulation of $B = 100$ bootstrap samples in each case.

Table 2.17 shows the results of Simulation 1. We found that the bootstrap is good at $n = 250$ and a little conservative at $n = 40$. At both sample sizes, the bootstrap performed poorly for the median, which is a known result. It is interesting to note that in contrast to our hypothesis, the nonparametric bootstrap actually performed well relative to the convex and parametric bootstrap. We had expected the convex bootstrap, a smoothed version of the nonparametric, to perform consistently better than the nonparametric, but instead found that the convex was more biased for the mean than the nonparametric, performing best when d was smallest ($d = 0$ is equivalent to nonparametric).

This simulation suggests that the nonparametric and parametric bootstraps can be used to assess the variability of summary measures of gene clustering (see also van der Laan and Bryan (2001)). Since estimated variability in the means is quite accurate and estimated variability in the correlation is accurate at $n = 250$ and conservative at $n = 40$, then we should be able to assess the variability of subset rules of the form $S(\mu, \Sigma)$ accurately (or at least conservatively) for reasonable sample sizes.

	0.9 quantile of maximum absolute difference		
Parameter:	Mean	Median	Correlation
n=40			
True distribution	0.60	0.74	0.75
Nonparametric	0.60	0.98	0.89
Convex d=0.1	0.59	0.97	0.89
Convex d=0.3	0.54	0.86	0.88
Convex d=0.5	0.50	0.78	0.87
Parametric	0.63	0.93	0.84
n=250			
True distribution	0.25	0.30	0.35
Nonparametric	0.26	0.38	0.36
Convex d=0.1	0.23	0.34	0.36
Convex d=0.3	0.21	0.30	0.36
Convex d=0.5	0.20	0.27	0.36
Parametric	0.24	0.36	0.35

Table 2.17: Results of Simulation 1 for gene clustering. $B = 100$ i.i.d. bootstrap samples were used in each simulation. Every bootstrap sample included $n = 40$ or $n = 250$ observations of a 3000-dimensional gene expression vector. The 0.9 quantile of the maximum absolute difference in each summary measure is reported.

2.9 Data Analysis in Human Acute Leukemia

Golub et al. (1999) analyze gene expression data in human acute leukemias to demonstrate a proposed method for discovering cancer classes (within a broader cancer diagnosis such as leukemia) and for predicting the class membership of a new tumor. The primary data consists of profiles for 38 leukemia patients, 27 of which have acute lymphoblastic leukemia (ALL) and 11 of which have acute myeloid leukemia (AML). For each patient there is a gene expression profile obtained from hybridization of bone marrow RNA to Affymetrix oligonucleotide microarrays. With oligonucleotide arrays, a specific probe (DNA fragment) is deposited on each spot on the array in a fixed quantity. With the cDNA arrays described earlier, there is much less control over the amount of probe placed on the array and that is the main reason for hybridizing two samples at once. By competitive hybridization, we can measure *relative* expression and avoid relying on the absolute intensity measured from one sample alone. This technical distinction means that one can actually interpret the raw intensities from an Affymetrix chip and compare them from one patient to another.

We use our methodology to search for the subset of genes that are the best classifiers for diagnosis. It is clinically important and, apparently, difficult to distinguish the two tumor classes. Obviously, we want to look for genes which are differentially expressed in ALL patients versus AML patients. Since there is no natural pairing of measurements, we have chosen to form a reference AML expression for each gene by taking the geometric mean of the intensities across all 11 subjects. We use this as the denominator and form a ratio for each gene for all 27 ALL patients. Therefore $n = 27$ and, after data pre-processing recommended by Golub et al. (1999), we have $p = 5925$ genes.

Table 2.18: Data analysis bootstrap results on subset quality.

	Avg Bootstrap Estimate	
	$K = 2$	$K = 3$
Sensitivity	0.88	0.88
Positive Predictive Value	0.85	0.84
Prop. of Ext. False Pos.	0.00	0.00
Any Ext. False Pos.	0.00	0.00
0.90 quantile of max abs dev. (mean)	1.20	1.25
0.90 quantile of max abs dev. (std dev'n)	3.34	3.31
0.90 quantile of max abs dev. (corr)	1.00	1.00
0.90 quantile of max abs dev. (covar)	11.15	10.93

We retained genes with at least 3-fold differential expression, which translates into an absolute log-ratio mean of at least 1.585. Of the original 5925 genes, 147 passed this pre-screen. We then ran PAM for several cluster numbers. The distance D_{ij} between genes i and j was defined as one minus the modified correlation proposed by Eisen et al. (1998). This quantity is obtained when one uses the normal formula for correlation $\rho_{ij} = \sigma_{ij}/\sigma_i\sigma_j$ but replaces the means μ_i and μ_j with a user-specified reference value (in this case zero) in the usual calculation of covariance and standard deviation (e.g. $\sigma'_{ij} = E(Y_iY_j)$, $\sigma'_j = \sqrt{E(Y_i^2)}$, and $\rho'_{ij} = \sigma'_{ij}/\sigma'_i\sigma'_j$). As recommended by Kaufman and Rousseeuw we chose the number of clusters K by inspecting the average silhouette widths for various values of K. For $K = 2, 3$, and 4, the average silhouette widths were $0.87, 0.74,$, and 0.24 respectively (for $K = 5, \ldots, 9$, average silhouettes widths were consistently below 0.24). We decided to run the bootstrap for both $K = 2$ and $K = 3$ and omitted the post-screen in both cases. Genes with absolute mean less than 0.07, which corresponds to 1.05-fold differential expression or less, were deemed particularly unsuitable as classifiers and 1415 genes met this criterion in the observed data. We carried out 100 bootstrap iterations and, once the medoids for the observed data were found, the cluster centers were fixed at these medoids throughout the bootstrap.

Table 2.18 provides basic quality measures for the bootstrap subsets. We see that sensitivity and positive predictive value are high (around 85%) for both bootstraps and we see no extremely false positives. Figure **??** presents the single-gene proportions from the bootstrap both the $K = 2$ and $K = 3$ cases; the length of each horizontal bar represents the number of bootstrap iterations in which a particular gene appears in the bootstrap subset. Since the cluster centers were fixed, we can also report the stability of cluster labels and the relative frequency of each label is depicted by the shading within the horizontal bars. We see that, when increasing the cluster number from 2 to 3, in fact we split one existing cluster into two and leave one cluster untouched. In both cases, the genes in the estimated subset reappear extremely often and almost always carry the same label as in the estimated subset. Overall, the stability of these clusters is quite strong. This is confirmed by the cluster-specific quality measures presented in table 2.19.

Table 2.19: Data analysis bootstrap results on cluster stability.

	$K = 2$ Bootstrap Avg.				$K = 3$ Bootstrap Avg.			
Cluster	Medoid	Size	Sensitivity	Pred. Value	Medoid	Size	Sensitivity	Pred. Value
1	1936	106.2	0.90	0.89	1936	106.3	0.89	0.89
2	5706	47.6	0.85	0.78	2227	13.9	0.83	0.76
3					3816	34.2	0.81	0.75
		153.8	0.88	0.85	1	154.3	0.88	0.84

Bibliography

J.M. Cherry, C. Ball, K. Dolinski, S. Dwight, M. Harris, J. C. Matese, G. Sherlock, G. Binkley, H. Jin, S. Weng, and D. Botstein. Saccharomyces genome database. *http://genome-www.stanford.edu/Saccharomyces/*, 2001.

R.J. Cho, 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. A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol.Cell*, 2:65–73, 2001.

Jean-Michel Claverie. Computational methods for the identifications of differential and coordinated gene expression. *Human Molecular Genetics*, 8(10):1821–1832, 1999.

J. DeRisi, L. Penland, P.O. Brown, M.L. Bittner, P.S. Meltzer, M. Ray, Y. Chen, Y.A. Su, and J.M. Trent. Use of a cdna microarray to analyse gene expression patterns in human cancer. *Nature Genetics*, 14:457–460, December 1996.

B. Efron and R. Tibshirani. The problem of regions. *Ann. Statist.*, 26(5):1687–1718, 1998.

B. Efron and R.J. Tibshirani. *An introduction to the bootstrap*. Chapman and Hall, New York, 1993.

M. Eisen, P.T. Spellman, P.O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci.*, 95:14863–14868, 1998.

J. Felsenstein. Confidence limits on phylogenies: an approach using the bootstrap. *Evolution*, 39(4):783–791, 1985.

The Chipping Forecast. The chipping forecast. *Nature Genetics*, 21(1, suppl.), 1999.

E. Giné and J. Zinn. Bootstrapping general empirical measures. *Ann. Probability*, 18:851–869, 1990.

T.R. Golub, 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. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286: 321–531, October 15 1999.

T. Heinemeyer, E. Wingender, I. Reuter, H. Hermjakob, A. E. Kel, O. V. Kel, E. V. Ignatieva, E. A. Ananko, O. A. Podkolodnaya, F. A. Kolpakov, Podkolodny N. L., and N. A. Kolchanov. Databases on transcriptional regulation: Transfac, trrd, and compel. *Nucleic Acids Res.*, 26: 364–370, 1998.

R. Herwig, A.J. Poustka, C. Mller, C. Bull, H. Lehrach, and J. O'Brien. Large-scale clustering of cdna-fingerprinting data. *Genome Research*, 9:1093–1105, 1999.

T.R Hughes, M.J Marton, A.R Jones, C.J Roberts, R. Stoughton, C.D. Armour, H.A. Bennett, E. Coffey, H. Dai, Y.D. He, M.J Kidd, A.M. King, M.R. Meyer, D. Slade, Lum P.Y., S.B. Stepaniants, D.D. Shoemaker, D. Gachotte, K. Chakraburtty, J. Simon, M. Bard, and S.H. Friend. Functional discovery via a compendium of expression profiles. *Cell*, 102:109–126, 2000.

L. Kaufman and P.J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, New York, 1990.

Eliot Marshall. Do-it-yourself gene watching. *Science*, 286:444–447, 1999.

C.M. Perou, S.S. Jeffrey, M. Van de Rijn, C.A. Rees, M.B. Eisen, D.T. Ross, A. Pergamenschikov, C.F. Williams, S.X. Zhu, J.C.F. Lee, O. Lashkari, D. Shalon, P.O. Brown, and D. Botstein. Distinctive gene expression patterns in human mammary epithelial cells and breast cancers. *Proc. Natl. Acad. Sci.*, 96:9212–9217, 1999.

C.M. Perou, T. Sørlie, M.B. Eisen, M. van de Rijn, S.S. Jeffrey, C.A. Rees, J.R. Pollack, D.T. Ross, H. Johnsen, L.A. Akslen, Ø. Fluge, A. Pergamenschikov, C. Williams, S.X. Zhu, P.E. Lønning, A.L. Børresen-Dale, P.O. Brown, and D. Botstein. Molecular portraits of human breast tumours. *Nature*, 406(6797):747–752, 2000.

K.S. Pollard and M.J. van der Laan. Statistical inference for two-way clustering of gene expression data. Technical Report 96, Group in Biostatistics, University of California, July 2001. Accepted by Mathematical Biosciences.

D.T. Ross, U. Scherf, M.B. Eisen, C.M. Perou, C. Rees, P. Spellman, V. Iyer, S.S. Jeffrey, M. Van de Rijn, M. Waltham, A. Pergamenschikov, J.C.F. Lee, D. Lashkari, D. Shalon, T.G. Myers, J.N. Weinstein, D. Botstein, and P.O. Brown. Systematic variation in gene expression patterns in human cancer cell lines. *Nature Genetics*, 24:227–235, 2000.

M.J. van der Laan and J.F. Bryan. Gene expression analysis with the parametric bootstrap. *Biostatistics*, 2:1–17, 2001.

M.J. van der Laan and K.S. Pollard. Hybrid clustering of gene expression data with visualization and the bootstrap. Technical Report 93, Group in Biostatistics, University of California, May 2001. Submitted to IEEE Intelligent Systems in Biology.

A. van der Vaart and J.A. Wellner. *Weak Convergence and Empirical Processes*. Springer, New York, 1996.

J. Zhu and M.Q. Zhang. Scpd: A promoter database of yeast saccharomyces cerevisiae. *Bioinformatics*, 15:607–611, 1999.