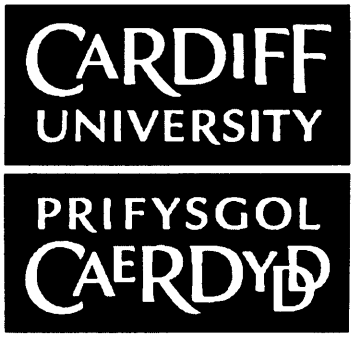
Signal Processing Techniques for the Interpretation of Microarray Measurements

---

Thesis submitted to the University of Cardiff in candidature for the degree of Doctor of Philosophy.

Thomas Bowles



Centre of Digital Signal Processing
Cardiff University
2006

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Abstract

Microarray technology allows the measurement of gene transcription on a genome wide scale. Signal processing approaches to the analysis of data from microarray time course experiments are the focus of this thesis.

Firstly, spectral estimation methods are explored as a method for the detection of cell-cyclic elements within microarray data. High resolution data-dependent filterbank methods are proposed as an improvement to the traditional periodogram approach. A spectral estimator is then designed specifically to deal with the errors in the sampling times inherent in microarray experiments, which is based on the robust Capon beamformer. A beamforming inspired approach is shown to yield a more robust, and higher resolution, estimate of the magnitude spectrum of the whole data set than the previous spectral estimation approaches.

Blind source separation is examined as a method for recovering sources which represent fundamental cellular processes. The linear mixing model is compared to its transpose form, and a dual form, in terms of their finite sample performance with real microarray data. Second order methods are proposed to recover sources which are spatio-temporally uncorrelated and may be more suitable with microarray data.

Both the spectral and blind source separation techniques are shown to yield useful feature extraction measures for microarray data clustering. The spectral feature extraction allows the clustering of cell-cyclic genes into a single functional group. Finally, sparse source separation is introduced as a possible blind separation technique with microarray data.

Acknowledgement

Firstly, thanks to my family for constant love and support during my studies, and to Kate, for always being there.

I am indebted to my supervisor, Prof. Jonathon Chambers for his continuous guidance and encouragement through the PhD process.

Special thanks to my second supervisor, Dr Andreas Jakobsson for his knowledge and enthusiasm in discussing challenging problems.

Thanks to Emily for continuous support, and saint-like patience in putting up with me during the writing up process! Thanks also to Mike for sharing the parallel experience of a PhD student and to the Cardiff University Mountaineering Club for good times.

I am grateful to the EPSRC and Cardiff University for funding my PhD study.

Statement of Originality

Chapters 2, 3 and 4 in this thesis comprise original work to the author's best knowledge, except where otherwise stated and referenced. In particular, the following are novel.

Chapter 2: Spectral estimation and beamforming for cell cycle detection

1. The application of filterbank estimation methods to the spectral analysis of microarray data.

2. The derivation of a spectral estimator for use with mis-sampled microarray data, based on the robust Capon beamformer.

3. The application of a beamforming inspired method to gain a high resolution estimate of the frequency content of microarray data and the use of this method with non-uniformly sampled microarray data.

Chapter 3: Independent component analysis for microarray data

1. The analysis of the finite data aspect of Independent Component Analysis of microarray data and its implications for separation performance.

2. The analysis of the validity of the dual assumption in the linear mixing model and an assessment of its performance.

3. The application of second order methods to microarray data in order to enforce spatiotemporal uncorrelatedness in the recovered sources and an explanation of why this is more feasible than methods utilising higher order statistics.

4. An analysis of the error in the linear mixing model of microarray data in terms of the spectrum of its residuals.

Chapter 4: Clustering of microarray data

1. The ability to cluster cell-cyclic genes into a single functional group, which is enabled by the use of a spectral feature extraction step.

2. The use of independent component analysis as a feature extraction step for clustering in the context of microarray data.

3. The application of sparse component analysis to microarray data and demonstration that it has the potential for high separation performance with the number of time points typically obtained in microarray experiments.

Publications

The original contribution of this thesis is partially supported by the following publications.

A. Kapoor, T. Bowles and J. Chambers, "A novel combined ICA and clustering technique for the classification of gene expression data", *IEEE International Conference on Acoustics Speech and Signal Processing, Philadelphia, USA*, vol. 5, 2005.

T. Bowles and J. Chambers, "The transpose model form in independent component analysis and its applications to microarray data", *Proceedings of the Institute of Mathematics and its Applications: International Conference on Mathematics and Signal Processing, Cirencester, UK*, 2004.

T. Bowles, A. Jakobsson and J. Chambers, "Detection of cell-cyclic elements in mis-sampled gene expression data using a robust Capon estimator", *IEEE International Conference on Acoustics Speech and Signal Processing, Montreal, Canada*, vol. 5, 2004.

T. Bowles, A. Jakobsson and J. Chambers, "Advanced spectral estimation for the identification of cell-cycle regulated genes", *IEEE EMBS UK and RI postgraduate conference in biomedical engineering and medical physics, Aston, UK*, 2003.

List of Acronyms

ANOVA	Analysis of Variance
ASC	Amplitude Spectrum Capon
BSS	Blind Source Separation
DOA	Direction Of Arrival
DFT	Discrete Fourier Transform
cDNA	complementary DNA
DNA	DeoxyRibonucleic Acid
ICA	Independent Component Analysis
IID	Independent Identically Distributed
FastICA	Iterative algorithm for ICA
FFT	Fast Fourier Transform
JADE	Joint Approximate Diagonalisation of Eigenmatrices
MI	Mutual Information
mRNA	messenger RNA
MSE	Mean Square Error

MVDR	Minimum Variance Distortionless Response
PCA	Principle Component Analysis
pdf	Probability Density Function
PI	Performance Index
PLS	Partial Least Squares
PSC	Power Spectrum Capon
QT	Quality Threshold clustering
RCB	Robust Capon Beamformer
RNA	Ribonucleic Acid
SCA	Sparse Component Analysis
SOBI	Second Order Blind Identification

List of Symbols

$\odot$	Schur-Hadamard (elementwise) product
$(\cdot)^H$	Hermitian transpose operator
$(\cdot)^T$	Transpose operator
$\mathbf{a}_L$	L-tap Fourier vector
$\mathbf{A}$	Mixing matrix
$\tilde{\mathbf{A}}$	Mixing matrix for the transpose form
$\mathbf{B}$	Estimated unmixing matrix
$\tilde{\mathbf{B}}$	Estimated unmixing matrix for the transpose form
$\mathbf{C}$	Matrix of cluster centroids
$E[\cdot]$	Expectation operator
$gcd[\cdot]$	Greatest common divisor
$\mathbf{h}_\omega$	L-tap bandpass filter, centred on ω
K	Number of sources
L	Filter length
m	Number of sources

N	Number of time samples
ω	Frequency
$\mathbf{R}$	Covariance matrix
$\mathbf{S}$	Source matrix
$\tilde{\mathbf{S}}$	Source matrix for the transpose form
$\mathbf{U}$	Rotation matrix
$\mathbf{W}$	Whitening matrix
$\mathbf{X}$	Data matrix
$\mathbf{Y}$	Estimated source matrix

List of Tables

List of Figures

Contents

Chapter 1

INTRODUCTION

1.1 Fundamental biology

Inside every cell of almost[1] every organism is DNA (deoxyribonucleic acid). In eucaryotes (more complex organisms) DNA is tightly packaged into chromosomes and contained within the cell nucleus. Procaryotes (single celled organisms, mainly bacteria) have no cell nucleus and their DNA is contained within the cell cytoplasm. This DNA encodes the genetic information of the organism. The genetic information of a whole organism is known as a genome and provides a blueprint for the function of the organism.

1.1.1 The DNA molecule

The DNA molecule is composed of two polynucleotide chains, fixed together by hydrogen bonds in the famous double helix structure [5], shown in Figure 1.1. Each polynucleotide chain is composed of many nucleotides. Each nucleotide is composed of a five carbon sugar and a phosphate group which are common to each nucleotide, and a base which may be either adenine, cytosine, guanine or thymine. These bases are denoted (A,C,G,T) and their symbols are also used to identify

[1] Some retroviruses actually store their genetic information as RNA (ribonucleic acid).

the corresponding nucleotide. These four symbols are the fundamental base-4 alphabet for our representation of genetic information. Each base pairs only with one other base; A always pairs with T, and G always pairs with C. Because of this, the nucleotide sequence in one polynucleotide chain completely determines the sequence in the other, they are thereby said to be complementary. The polarity of the DNA chain (and hence nucleotide sequence) is indicated by defining one end of the DNA molecule as the 5' end and the other as the 3' end. By convention, nucleotide sequences are usually given in the order 5'-3'.

1.1.2 DNA replication

When cells divide the DNA they contain must be replicated accurately if excessive mutation is not to occur. As suggested in [5] the double helix structure of DNA is ideal for replication. The complementary nature of the nucleotide sequences allows one polynucleotide chain to identify uniquely its complementary partner. Each strand can therefore be used as a template for the synthesis of a new strand. The task of DNA replication is performed by a cluster of proteins, known as a replication machine. Initiator proteins prise the two strands apart by breaking the hydrogen bonds. The positions at which this occurs are known as replication origins and are indicated by a particular sequence of nucleotides. A whole genome has many of these replication origins, greatly speeding up the process of replication. This concurrent replication strategy results in many replication forks along the DNA strand. The central component of the replication machine is called DNA polymerase which synthesises the new DNA strand using one of the parent strands as a template. The replication machine also includes a proof

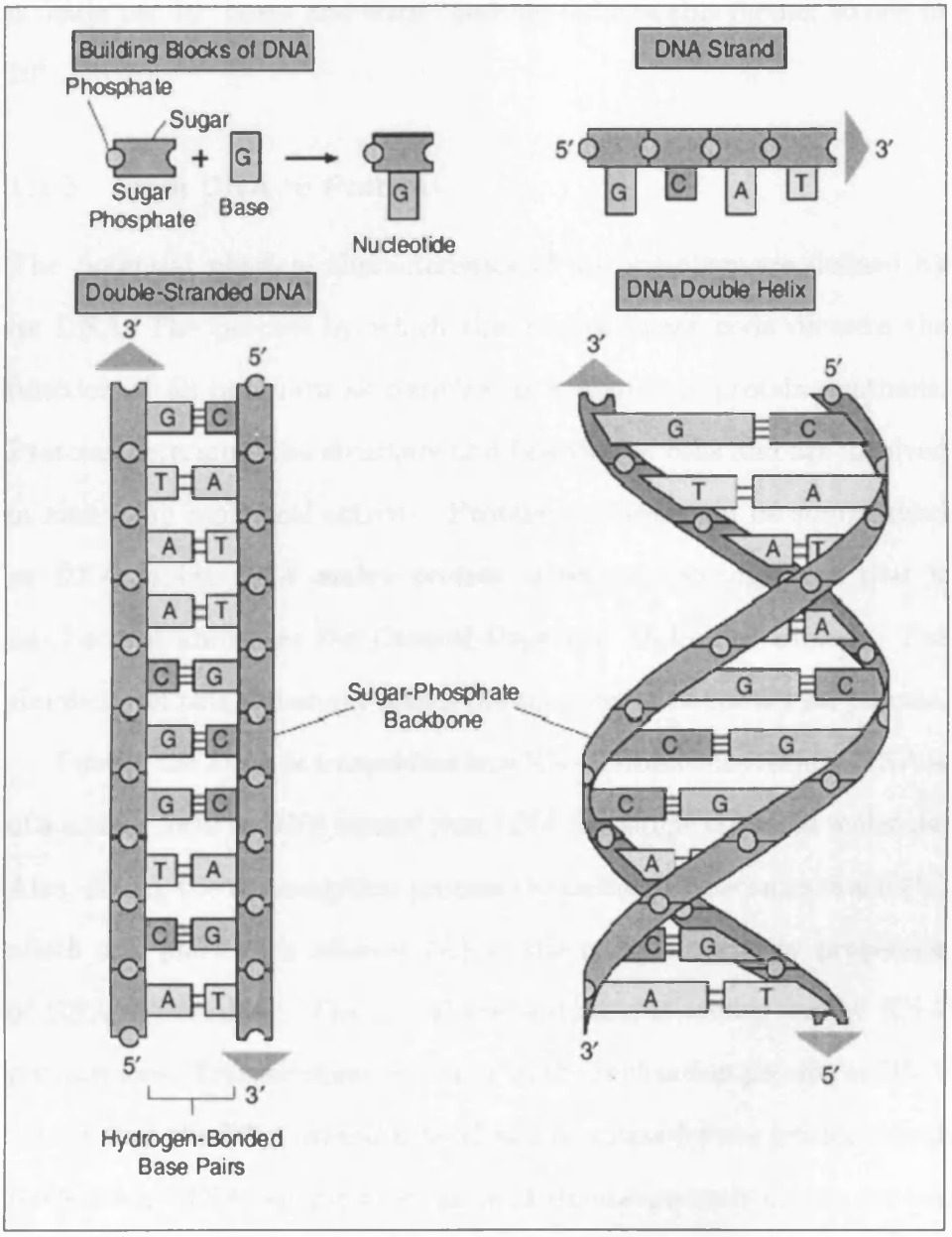


Figure 1.1. DNA composition and structure (reproduced from [3]). The composition of the DNA strands from base nucleotides is shown, along with the pairing of complementary nucleotides to create the double helix structure.

reading system for checking whether the newly formed DNA is correct. The accuracy of replication is such that, on average, only one mistake is made per 10^7 bases and error checking reduces this further to one in 10^9 .

1.1.3 From DNA to Protein

The potential physical characteristics of an organism are defined by its DNA. The process by which this base-4 linear code dictates the function of an organism as complex as a human is protein synthesis. Proteins determine the structure and function of cells and are involved in almost all biological activity. Protein synthesis can be summarised as *DNA makes RNA makes protein*, a premise so ingrained that it has become known as the *Central Dogma* of Molecular Biology. The simplicity of this statement masks the complexity of the actual process.

Firstly, the DNA is *transcribed* into RNA (ribonucleic acid). RNA is of a similar form to DNA except that RNA is a single stranded molecule. Also, during the transcription process thyamine (T) becomes uracil (U) which still pairs with adenine (A) so the complementarity properties of DNA are retained. The actual transcription is carried out by RNA polymerases. Transcription is similar to the replication process of DNA except that the DNA strand is used as a template for the production of RNA. Many RNA polymerases can work simulataneously on one stretch of DNA so the transcription process can be rapid. Unlike DNA replication there is no proof reading mechanism in transcription so the error rate is higher at about one error per 10^4 base pairs. This relatively high error rate is tolerable as RNA is created only temporarily as a step towards protein synthesis. DNA contains special sequences of nucleotides

called promoter regions which signal to the RNA polymerase where to start and stop transcription. Upon finding a promoter site, the RNA polymerase begins transcription and continues along the DNA until a stop site is reached. In complex organisms, such as humans, the genes can be interspersed by long non-coding regions of up to 10^5 bases. The freshly transcribed RNA is called the primary transcript to distinguish it from RNA in other stages of processing.

Eucaryotic genes have their coding regions (exons) interupted by long non-coding stretches of DNA (introns). The primary transcript includes these introns. In fact, in humans, the preponderance of introns means that only 5 percent of RNA in the primary transcript directly codes for protein. The introns are removed in a process called RNA splicing and the resulting exons joined together to give a continuous coding RNA, known as messenger RNA (mRNA). The RNA segments cut from the primary transcript can actually be reconstructed in varying permutations to allow the creation of different proteins from the same gene. Hence, one gene can produce any one of a domain of similar proteins. The mRNA resulting from the splicing procedure is ready for translation for production of a specific protein[2].

The translation procedure maps the four letter code of the mRNA into the linear sequence of amino acids which define a protein. The rules for this translation are known as the *genetic code*, shown in Figure 1.2. The mRNA is read in groups of three. One group of three nucleotides is called a codon and codes directly for a specific amino acid. There are 20 common amino acids used to build proteins and $4^3 = 64$ possible codons. All of the possible codons are in use and

[2]Some procaryotic mRNA is actually able to code for multiple proteins.

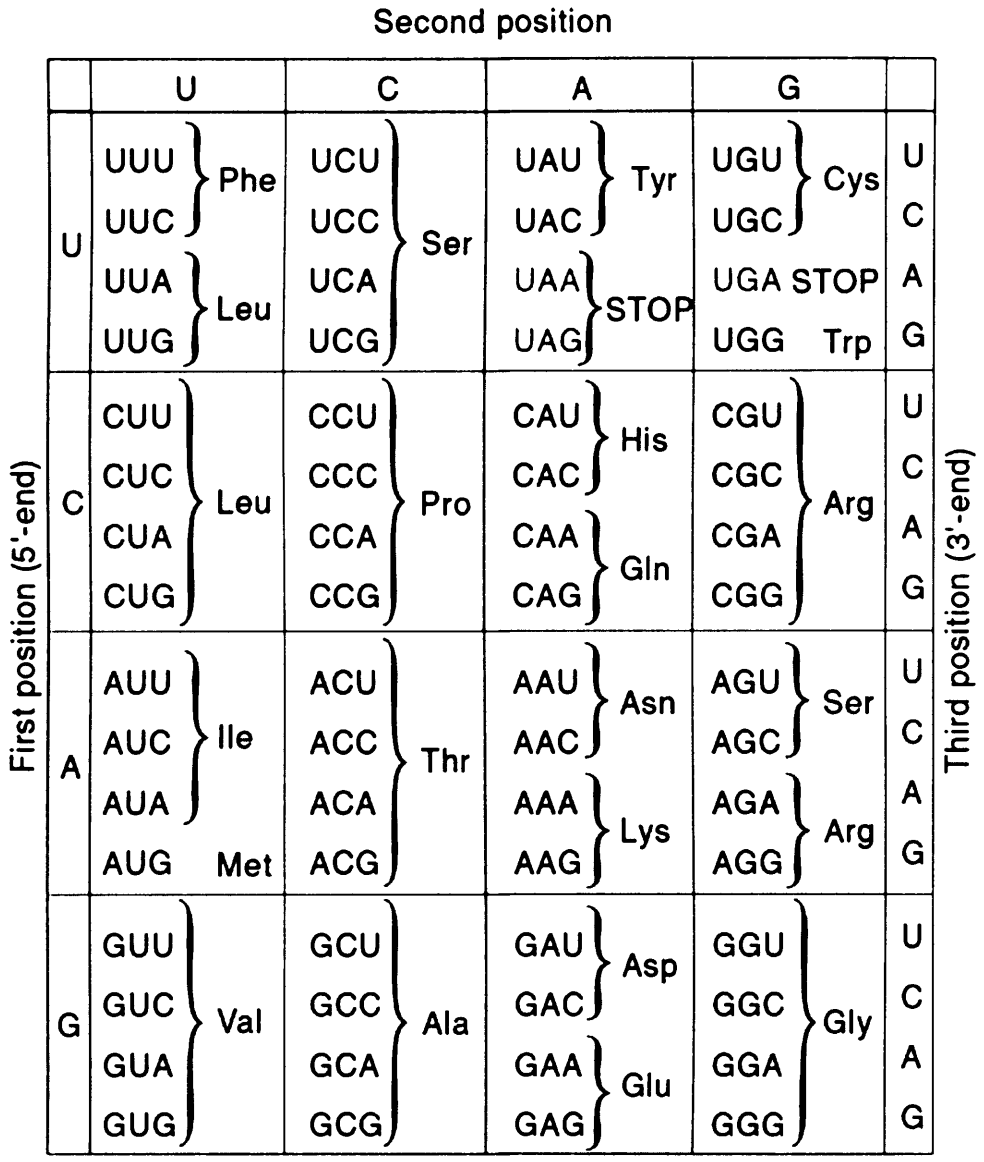


Figure 1.2. The genetic code, showing the mapping between different codons and their respective amino acids. For example, the first codon in the top left box of the table is UUU, and this maps to the amino acid Phe (Phenylalanine). The next codon down is UUC, which also maps to Phe.

so some amino acids must be specified by more than one codon. In fact, some amino acids are represented by up to six codons and others by only one. This many to one mapping has been shown to be near optimal in the sense that the final protein formed is robust to errors in the mRNA [6]. One codon (AUG) is known as the start codon and denotes the starting point for protein construction. Three codons (UAA, UAG, UGA) are recognised as stop codes to halt protein translation. The resultant amino acid chain then folds into the complex three dimensional structure of the protein. In most cases this folding is spontaneous, though some proteins are guided though the folding process by molecules called chaperones. Currently, the prediction of three dimensional protein shape from the one dimensional amino acid chain is achieved by exhaustive search though the possible permutations to find the permutation with the lowest energy required to hold it. This is computationally very intensive and a reliable analytical method for determining protein shape (which defines its function) from a given amino acid sequence is something of a holy grail within the structural genomics community.

1.1.4 Gene regulation

Identical DNA is contained within every cell in the human body and yet the diversity in cell characteristics is huge. This is possible through the regulation of gene expression. When a gene is *expressed*, the protein that the gene codes for is actually produced by the cell. Gene regulation is the control of gene expression to produce the proteins needed by the body in the correct quantities at the correct time. Gene regulation is a complex network of processes which is not yet fully understood.

However, it is known that regulation can occur at four stages during protein synthesis:

1. Transcription, control of when a particular gene is transcribed from DNA into RNA.

2. RNA processing, control of the splicing process from the primary transcript RNA to mRNA.

3. Translation, control of when a particular gene is translated from RNA into protein.

4. Protein control, activating and deactivating proteins.

Wasted intermediary products are minimised by performing regulation early in the protein synthesis process. Because of this, RNA transcription regulation is the primary control mechanism for most genes and this is the area we will focus on.

The promoter region of a gene always contains an initiation site, which is where the transcription of the DNA into RNA by the RNA polymerase begins, and a general promoter region immediately upstream from the initiation site. This promoter region contains sites that the RNA polymerase requires to bind to the DNA. These are required by every gene for transcription to occur and so cannot be a regulatory mechanism. However, there are other sequences which can be as short as five base pairs which are scattered further upstream from the promoter region that are present in almost all genes but in differing configurations. These are the regulatory sequences. Some of these respond to a single biochemical signal and have a binary effect, effectively switching a gene on or off. These are common in bacteria. More complex

organisms tend to have longer, multiple sequences which act in a combinatorial fashion to dictate the rate of transcription. These regulatory sequences, or motifs, are recognised by one or more regulatory proteins that act as the agent to control expression. These regulatory sequences and their corresponding proteins can either act as activators to encourage transcription or repressors to discourage transcription. In addition to these regulatory proteins, eucaryotes require the presence of a group of proteins known as *general transcription factors* which are thought to play a role in positioning the RNA polymerase and pulling apart the two DNA strands. Analysis of the regulatory process in eucaryotes is complicated by the fact that regulatory motifs can occur thousands of base pairs upstream of the promoter region. This is feasible because the DNA loops over itself, allowing relatively close contact between both the promoter region and the regulatory sequences. Figure 1.3 shows this looping, and other aspects of eucaryotic gene transcription activation. The gene regulation process is very complex and detailed identification of the gene regulation network is beyond current technology. However, measurement of gene mRNA levels in response to simple physical or biochemical stimuli is possible using DNA microarrays.

1.2 DNA Microarray technology

Microarrays provide a systematic, high-throughput method of measuring relative mRNA levels of thousands of genes concurrently. The possible uses of microarrays in genomics are diverse because the generality of the microarray hardware imposes few constraints on experiment design.

The objective is to measure mRNA levels under given experimental

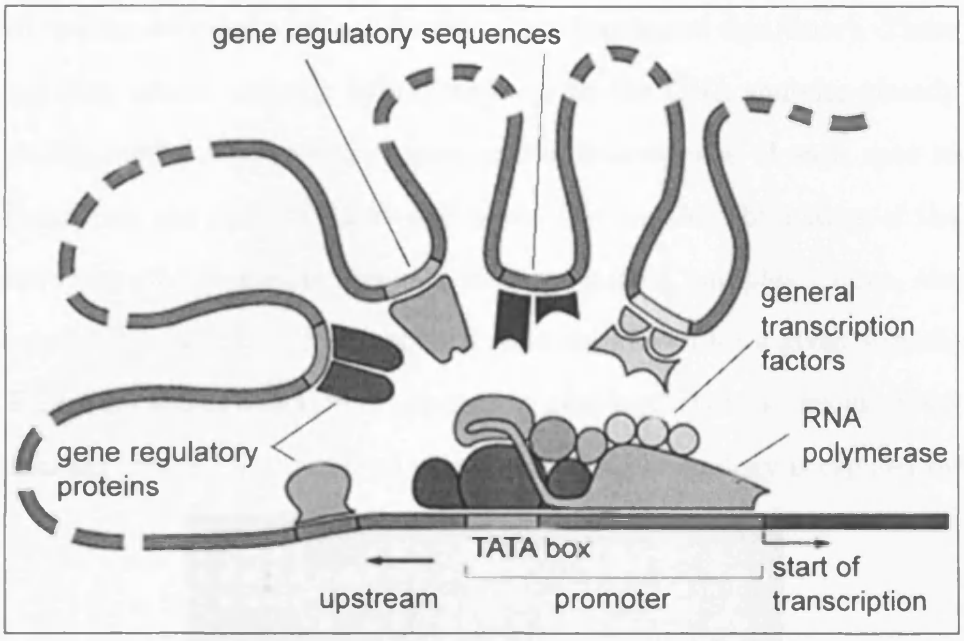


Figure 1.3. Eucaryotic gene activation (reproduced from [3]). The looping formulation of the DNA is shown, which allows close contact between the regulatory sequences and the promoter sites and hence allows the regulatory sequences to control transcription.

conditions. These experimental conditions could be the application of heat, an external chemical or an internal cell messenger agent. Samples can be arranged in order to give the response of one gene to multiple stimuli or the response of all the genes in a genome to a single stimuli, or a hybrid of the two. This flexibility means that the experiment design stage is crucial if useful results are to be obtained. Usually, DNA representing a single gene is assigned to each spot. In the case of cDNA (complementary DNA) arrays[3], for each sample spot on the array, mRNA is sampled from the cell populations under two different experimental conditions. One condition could be the active application of some stimuli and the other a measure of the cells in a reference state. This then undergoes *reverse transcription* into cDNA. Each cDNA sam-

[3]also known as two channel arrays.

ple is then labelled with a different colour flourescent dye (fluor). These are then mixed, causing hybridization with the DNA samples already on the spot. A laser then measures the flourescence of each spot to determine the ratio of fluors and hence the relative abundance of the sequence of the specific gene in the two mRNA samples. Thus, the relative transcription levels of each gene are known for a given stimuli. Figure 1.4 shows a section of microarray grid typical of the layout of two channel arrays. Another type of microarray technonology is typified by

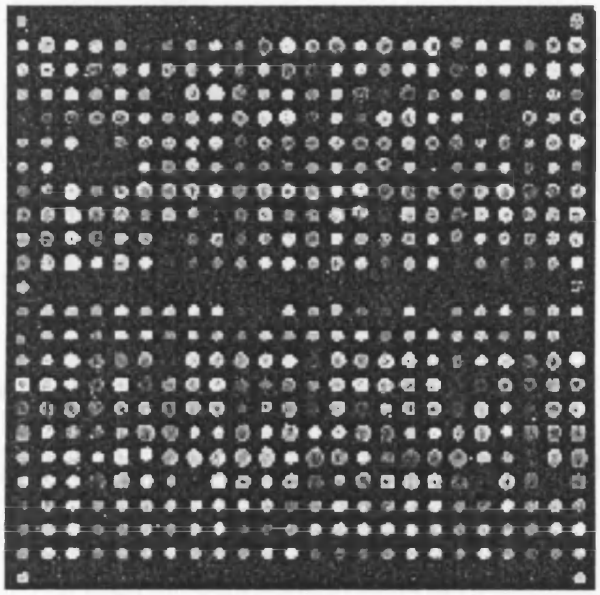


Figure 1.4. Enlarged subsection of microarray image of a two channel array. One channel is commonly assigned to red and the other green, and the two channels superimposed on a single image.

the Affymetrix (Santa Clara, USA) approach, whereby oligo probes are synthesized *in situ* on the array by photolithography. Another *in situ* approach was pioneered by Agilent (Palo Alto, USA) which uses inkjet printing technology to build oligonucleotide probes directly onto the array. Whether this *in situ* approach is used, or the two channel technology, the aim is identical; to measure gene transcription activity on a large scale.

As previously noted, control of transcription level is the primary gene regulation method. It should be noted that, as stated in the central dogma of molecular biology, the final product of gene expression is a protein, which is not measurable using microarrays. Although technology in this area is developing rapidly (see [7] for an overview of proteomic technologies), it is not yet possible to measure protein on a genome-wide scale and so the measurement of mRNA with microarrays is currently the best estimate of gene expression on a genome-wide scale. Nevertheless, the term 'gene expression' is often used in relation to microarray experiments and this should be assumed to refer to transcription levels.

For a further introduction to microarrays, see [8–10]. A review of different microarray technologies can be found in [11].

1.2.1 Data quality issues in microarray experiments

Microarray data are the product of a long experimental pipeline. A typical simplified data collection process would be:

- Plan experiment.
- Grow, or otherwise obtain, cells under the relevant environmental conditions.
- Sample cell culture and hybridise on a microarray.
- Place microarray in scanner and read in fluorescence values to obtain microarray images.
- Use image processing software on the microarray images to assess the spot locations and assign a value to the expression level depending on the spot brightness.

The process of data collection is subject to noise at every stage. Some common sources of noise include:

- Cell culture impurity or cross contamination.
- Human imprecision in the sampling and hybridisation procedure.
- Microarray scanning imprecision.
- Imperfect image processing of microarray images.

These major sources of noise result in data that are considerably more noisy than that usually used in many signal processing applications. The noise characteristics of microarray data are not widely understood although some initial work has been done to model the noise [12]. In [13], replicate experiments (Affymetrix based) were performed to isolate noise sources. They found that noise originating from the sample preparation stage was relatively small and could be modelled by a small constant component. Hybridization noise was found to be more significant, with noise dependent on the expression level and showing some Poisson-like characteristics. This study concentrates on a few of the more easily quantified sources of noise but serves to demonstrate that the characteristics of microarray noise are not trivial and any signal processing technique must reflect this.

Given the high degree of uncertainty in microarray data, microarray experiments should be performed in parallel so that some measure of the variance in the output can be obtained. However, large scale microarray experiments are expensive and so parallelism is inevitably constrained by cost. As the cost of microarray experiments decreases, parallelism should become more common and provide more reliable data than are currently available.

Even if noiseless measurements were possible, the analysis of microarray data would still present a considerable challenge. Each set of data provides a snapshot of the process of gene regulation at limited instants in time, for specific environmental conditions. Gene regulation is controlled by complex biological processes which are understood only in parts. In contrast to many signal processing applications, *generative models for the data are unavailable*, except in the most basic qualitative forms.

1.2.2 Microarray data preprocessing

Microarray data, as generated by image processing software must be preprocessed before it can be reliably used. Data from cDNA microarrays are usually given as a log ratio between the mRNA levels of the two cell populations. A log ratio is used to ensure numerical symmetry between upregulated and downregulated genes. A base two logarithm is most usually used and hence values of -1 and $+1$ represent two fold down and up regulated genes, respectively.

More extensive normalisation is often performed to compensate for differences between microarrays and even across the surface of each microarray. For a review of microarray normalisation see [14].

1.2.3 Time course experiments

A single microarray allows the measurement of the transcription levels of P genes at an instant in time. This can be used to give an instantaneous snapshot of the transcriptional state of a set of genes. In order to measure the dynamic behaviour of gene transcription it is necessary to sample cell populations over time and, for each time point, measure

the transcription levels of all P genes with a microarray. Such an experiment is called a time course experiment and provides insight into transcription levels over time. The results of such a time course experiment can be expressed as a matrix of expression levels $\mathbf{X} \in \mathbb{R}^{P \times N}$, given N microarrays, each representing a point in time, and P genes per microarray. Each row of $\mathbf{X}$ hence represents the transcription levels of one particular gene over time. This kind of time course data is naturally of particular interest from a signal processing point of view, and will be the focus of this thesis.

Missing values are common in microarray time course data, primarily because of poor spot quality in the microarray images. Various methods have been suggested to interpolate missing values [15–17]. Throughout this thesis genes with more than a quarter of the values missing in a time course are discarded and the remainder of the missing values are calculated with a cubic spline interpolant. A cubic spline interpolant allows smooth and robust interpolation, without resorting to assumptions of a particular generative model [18].

1.3 Signal processing in genomics

Many signal processing techniques have been applied to the analysis of microarray data, including clustering [4,19,19,20], spectral analysis [21], independent component analysis [22–24], and network modelling [25]. Microarray data analysis represents a significant challenge to signal processing techniques[4]. Amongst the issues for the development of

[4]This has motivated a philosophy which draws upon four cornerstone concepts in signal processing; i.e. parametric and non-parametric methods, together with supervised and unsupervised learning. The expectation is that the current best solutions for microarray data analysis will result from a fusion of such ideas and this is indeed the heart of the methodology in Chapter 4.

successful algorithms are:

- Robustness to significant uncertainties in generative models.
- Robustness to variable data quality, missing data, non-uniform sampling and short data lengths.
- The incorporation of limited probabalistic domain knowledge into algorithms.
- The fusion of diverse data.
- Analysis of massively multi-variable datasets.
- Efficient implementations of algorithms to work with vast datasets.

1.4 Thesis outline

This thesis is split into three primary contribution chapters. In chapter 2, filterbank spectral estimation is introduced as a method for detecting cell cyclic elements within gene expression data. A method for dealing explicitly with temporal mis-sampling, and other noise sources is then developed from the robust Capon estimator. Methods derived from beamforming are then discussed, including how to cope with the non-uniform sampling often found in microarray data.

Chapter 3 introduces blind source separation techniques for microarray analysis. Different ways of blindly extracting sources which represent underlying cellular processes are discussed, including transpose forms and second order methods. The different methods are analysed in terms of limited sample size performance and model error.

Chapter 4 analyses how the two previous spectral estimation and blind source separation approaches can be used to enhance the clus-

tering of microarray data. Sparsity is then introduced as a possible criterion for source separation and parallels are drawn with clustering approaches. Finally, conclusions on the thesis work are drawn and promising topics for future study highlighted.

Chapter 2

SPECTRAL ESTIMATION AND BEAMFORMING FOR CELL CYCLE DETECTION

2.1 The cell cycle

A cell reproduces by duplicating its genetic material and then dividing in two [3]. Cells have a finite life and so this division and growth is necessarily regular to maintain a cell population. This natural process of cell division and growth is called the cell cycle. The control of the cell cycle is part of the gene regulatory process and so the cell cycle can manifest itself as a cyclic element in the transcription activity of some genes. Genes exhibiting this cyclic activity could be either regulators of the cell cycle, genes whose transcription is directly affected by the cell cycle, or genes whose transcription is affected by other genes connected to the cell cycle.

An important early application of microarray time course experiments has been to identify genes with cyclic elements. In order for cell cyclic elements to be detected in time courses the cell culture must be synchronised to the same point in the cell cycle, i.e. every cell in the

cell culture should start the time course at the same stage in the cell cycle. Without this synchronisation, the cells within each time point sample would be in varying stages of the cell cycle, and the microarray measurement would record an average value of transcription activity of cells spread over different stages in the cell cycle.

The validity of this type of study has been debated, primarily through the work of Professor Stephen Cooper [1] (University of Michigan USA). In particular, the ability of whole culture treatment methods to synchronise cells is questioned. It is also doubted whether the cyclic components observed in the microarray data are actually due to the cell cycle or are a reaction of the cells to the shock of the attempted synchronisation. These issues are reviewed in [26] and detailed more fully in the references therein. This doubt over the results obtained should not discourage research into the detection of cell cyclic components as other experimental methods exist and more reliable data will become available. In fact, a later study addressed the issue of synchronisation and provided evidence of good population synchrony [27].

2.2 Cell cycle studies in the literature

In one study, which became something of a benchmark for subsequent research, Spellman *et al.* [1] identified 800 genes which could be cell cycle regulated from the *Saccharomyces cerevisiae* (budding yeast) genome. The Fourier transform was used to obtain the frequency content of the genes' expression time series and thereby rank them according to the magnitude of the Fourier transform at the estimated cell cycle frequency. It should be noted that the decision which genes were actually

[1] http://www-personal.umich.edu/~cooper/

cell cycle regulated was rather arbitary because of the lack of any obvious cut-off point in the ranking of possible cell cyclic genes. The methods and analysis used in this study, along with that of [28], were widely cited and clearly influential in subsequent work. In particular, the use of the Fourier transform in [1] to identify periodic elements in profiles was used in much of the later work.

Later studies used human cell lines to attempt to identify cell cyclic genes in the human genome [29,30]. A further study on the fission yeast *Schizosaccharomyces pombe*, revealed a similar proportion of genes involved in the cell cycle to that of earlier studies on the budding yeast *Saccharomyces cerevisiae*. This study was also notable for using independent time series replicates, which allow a measure of data quality. Two recent studies also found similar numbers of cell cycle regulated genes in fission yeast [27,31]. The later work of A. Oliva *et. al.* [27] is notable as it uses a relatively high number of time points; up to 52 are used in one experiment.

The most common technique used by biologists analysing time course microarray data is clustering. However, clustering is of limited value in the identification of cell cyclic elements because the times of peak expression (or phases) vary widely between genes. Therefore, genes which exhibit perfect cell cyclic behaviour are not clustered together unless their phases happen to be coincident. In practice, the phases are sufficiently widely distributed that the boundaries between clusters is rather arbitrary.

In addition to the primary data producing studies, work has been done into the analysis methods of gene cell cycle data. Proposed methods of cell cycle detection generally fall within two categories: Fourier

based methods and model based methods. In [21], the periodogram was used to obtain the spectra of the gene expression series and Fisher's g statistic used to identify a cut off point for genes deemed cell cycle regulated. In [32], an improved Fourier based technique is suggested to cope with errors caused by phase variation in short data lengths.

A model based approach was used in [33–35]. A model cell cycle is defined and then an intelligent search algorithm ranks the genes' distance from this reference. In model based (and clustering) approaches, the problem of variable cell cycle phases must be addressed. In Fourier based analysis, the amplitude and phase information are effectively decoupled.

2.3 Spectral estimation for cell cycle detection

2.3.1 Filterbank spectral estimation methods

The estimation of the spectrum of microarray data presents specific challenges. In particular, the data sequence is typically short (just 18 samples were taken in the *alpha* experiment from [1]) and negligible prior information is known about the generative model or noise characteristics. The lack of model knowledge of the generative process precludes the use of a parametric estimator. The filter-bank class of non-parametric spectral estimators has received significant attention lately [36]. The filter bank approach is based on the filtering of the data with a bandpass filter, centred at a given frequency at which the spectrum is to be estimated. The power in the filter output is then divided by its bandwidth to obtain an estimate of the spectrum at that frequency.

Consider a data sequence $\{y(t)\}_{t=0}^{N-1}$ of length N. The power in the filter output is given by:

$$\phi(\omega) = E\left[|\mathbf{h}_\omega^H \mathbf{y}_L(t)|^2\right] \tag{2.3.1}$$

$$= \mathbf{h}_\omega^H E\left[\mathbf{y}_L(t)\,\mathbf{y}_L^H(t)\right]\mathbf{h}_\omega \tag{2.3.2}$$

$$= \mathbf{h}_\omega^H \mathbf{R}_y \mathbf{h}_\omega \tag{2.3.3}$$

where $E[\cdot]$ is the expectation operator, $\mathbf{h}_\omega$ is the coefficient vector of an L-tap finite impulse response bandpass filter, centered at frequency ω, $\mathbf{R}_y$ is the data covariance matrix, defined as $\mathbf{R}_y = E\left[\mathbf{y}_L(t)\,\mathbf{y}_L(t)^H\right]$ and

$$\mathbf{y}_L(t) = \begin{bmatrix} y(t) & \dots & y(t+L-1) \end{bmatrix}^T \tag{2.3.4}$$

for $t = 0, \dots, M-1$, where $M = N - L + 1$, $(\cdot)^T$ denotes the transpose operator and $(\cdot)^H$ the Hermitian transpose.

The classical periodogram can be couched in filterbank terms, and is equivalent to applying the filter [37]

$$\mathbf{h}_\omega = \frac{\mathbf{a}_L}{\sqrt{L}} \tag{2.3.5}$$

where the filter length is equal to the number of samples, $L = N$ and $\mathbf{a}_L$ is defined as the Fourier vector

$$\mathbf{a}_L = \begin{bmatrix} 1 & e^{i\omega} & \dots & e^{i\omega(L-1)} \end{bmatrix}^T \tag{2.3.6}$$

and $i = \sqrt{-1}$. Note that this method is a non-adaptive, or data-independent, method in the sense that the design of $\mathbf{h}_\omega$ does not depend on the data sequence $\{y(t)\}_{t=0}^{N-1}$. This interpretation of the pe-

riodogram can be easily shown to be equal to the more traditional expression [37]

$$\phi_{PER}(\omega) = \mathbf{h}_{\omega}^{H}\mathbf{R}_{y}\mathbf{h}_{\omega} \tag{2.3.7}$$

$$= \frac{1}{L^2}\mathbf{a}_{N}^{H}\mathbf{y}_{N}(t)\,\mathbf{y}_{N}(t)^{H}\,\mathbf{a}_{N} \tag{2.3.8}$$

$$= \frac{1}{L^2}\left|\mathbf{a}_{N}^{H}\mathbf{y}_{N}(t)\right|^{2} \tag{2.3.9}$$

where $\mathbf{a}_{N}^{H}\mathbf{y}_{N}(t)$ is nothing but a vector expression for the value of the Discrete Fourier Transform (DFT) at ω. The periodogram can therefore be efficiently computed over a uniform frequency grid using the Fast Fourier Transform (FFT). This is effectively the method used by most of the literature concerned with cell cycle detection.

The performance of the periodogram, in terms of resolution and variance, can be improved through the use of data adaptive methods to design the bandpass filters. The classical Capon, or Power Spectrum Capon (PSC), estimator, is one such method [38, 39]. The PSC minimises the power in the filter output whilst enforcing unit gain at the frequency of interest, giving the minimisation problem:

$$\mathbf{h}_{\omega} = \min_{\mathbf{h}_{\omega}} \mathbf{h}_{\omega}^{H}\mathbf{R}\mathbf{h}_{\omega} \text{ subject to } \mathbf{h}_{\omega}^{H}\mathbf{a}_{L} = 1 \tag{2.3.10}$$

The solution is found through application of a Lagrangian formulation as [37]

$$\mathbf{h}_{\omega} = \frac{\mathbf{R}^{-1}\mathbf{a}_{L}(\omega)}{\mathbf{a}_{L}^{H}(\omega)\,\mathbf{R}^{-1}\mathbf{a}_{L}(\omega)} \tag{2.3.11}$$

substituting (2.3.11) into (2.3.3) yields the power spectrum Capon estimate of $\phi(\omega)$

$$\hat{\phi}_{PSC}(\omega) = \frac{1}{\mathbf{a}_{L}^{H}(\omega)\,\mathbf{R}^{-1}\mathbf{a}_{L}(\omega)} \tag{2.3.12}$$

In [40], we proposed using the amplitude spectrum Capon (ASC) estimator to identify cell-cyclic elements within gene expression data. This estimator improves on the accuracy of the PSC, making it particularly suited to the short data sequences of gene expression time series. The estimator is obtained by estimating the amplitude spectrum of data, which is modelled as

$$y(t) = \alpha_\omega e^{i\omega t} + e(t), \quad t = 0 \ \ldots \ N-1 \tag{2.3.13}$$

where α_ω is the complex amplitude of the generic sinusoid at ω and $e(t)$ is coloured noise representing the remainder of the signal. In vector form,

$$\mathbf{y}_L(t) = \left[\begin{array}{ccc} y(t) & \ldots & y(t+L-1) \end{array} \right]^T \tag{2.3.14}$$

$$= \alpha_\omega \left[\begin{array}{ccc} e^{i\omega t} & \ldots & e^{i\omega(t+L-1)} \end{array} \right]^T + \mathbf{e}_L(t) \tag{2.3.15}$$

$$= \alpha_\omega \mathbf{a}_L(\omega) e^{i\omega t} + \mathbf{e}_L(t) \tag{2.3.16}$$

for $t = 0, \ldots, N-1$, where $\mathbf{e}_L(t)$ is a vector formed similarly to $\mathbf{y}_L(t)$. We wish to minimise the effect of the noise term, whilst enforcing unit gain for the frequency of interest, giving the constrained optimisation

$$\mathbf{h}_\omega = \min_{\mathbf{h}_\omega} \mathbf{h}_\omega^H \mathbf{Q}_\omega \mathbf{h}_\omega \text{ subject to } \mathbf{h}_\omega^H \mathbf{a}_L = 1 \tag{2.3.17}$$

where $\mathbf{Q}_\omega = E\left[\mathbf{e}_L(t)\, \mathbf{e}_L^H(t)\right]$ is the covariance matrix of the noise term.

However, we note that

$$\mathbf{R} = E\left[\mathbf{y}_L(t)\,\mathbf{y}_L^H(t)\right] \tag{2.3.18}$$

$$= |\alpha_\omega|^2 \mathbf{a}_L(\omega)\,\mathbf{a}_L^H(\omega) + \mathbf{Q}_\omega \tag{2.3.19}$$

$$\mathbf{Q}_\omega = \mathbf{R} - |\alpha_\omega|^2 \mathbf{a}_L(\omega)\,\mathbf{a}_L^H(\omega) \tag{2.3.20}$$

where we have assumed that the noise is uncorrelated with the signal. Therefore,

$$\mathbf{h}_\omega^H \mathbf{Q}_\omega \mathbf{h}_\omega = \mathbf{h}_\omega^H \left[\mathbf{R} - |\alpha_\omega|^2 \mathbf{a}_L(\omega)\,\mathbf{a}_L^H(\omega)\right] \mathbf{h}_\omega \tag{2.3.21}$$

$$= \mathbf{h}_\omega^H \mathbf{R} \mathbf{h}_\omega - |\alpha_\omega|^2 \tag{2.3.22}$$

From (2.3.22), we note that

$$\min_{\mathbf{h}_\omega} \mathbf{h}_\omega^H \mathbf{Q} \mathbf{h}_\omega = \min_{\mathbf{h}_\omega} \mathbf{h}_\omega^H \mathbf{R} \mathbf{h}_\omega \tag{2.3.23}$$

and hence, the minimisation (2.3.17) is equivalent to the minimisation in the classical Capon design, given in equation (2.4.11). The filter is thus given by (2.3.11), the filtered signal being

$$\mathbf{h}_\omega^H \mathbf{y}_L(t) = \alpha_\omega \mathbf{h}_\omega^H \mathbf{a}_L(\omega)\, e^{i\omega t} + \mathbf{h}_\omega^H \mathbf{e}_L(t) \tag{2.3.24}$$

$$= \alpha_\omega e^{i\omega t} + \mathbf{h}_\omega^H \mathbf{e}_L(t) \tag{2.3.25}$$

The least-squares estimate of α_ω is

$$\hat{\alpha}_\omega = \frac{1}{M} \sum_{t=0}^{M-1} \mathbf{h}_\omega^H \mathbf{y}_L(t)\, e^{-i\omega t} \triangleq \mathbf{h}_\omega^H \mathbf{Y}_\omega \tag{2.3.26}$$

The amplitude spectrum Capon estimate is obtained by the substitu-

tion of the filter result (2.3.11) into equation (2.3.26), yielding

$$\hat{\phi}_{ASC}(\omega) = |\hat{\alpha}_\omega|^2 \tag{2.3.27}$$

$$= |\mathbf{h}_\omega^H \mathbf{Y}_\omega|^2 \tag{2.3.28}$$

$$= \left| \frac{\mathbf{a}_L^H(\omega)\mathbf{R}^{-1}\mathbf{Y}_\omega}{\mathbf{a}_L^H(\omega)\mathbf{R}^{-1}\mathbf{a}_L(\omega)} \right|^2 \tag{2.3.29}$$

2.3.2 Detection of cell cyclic components using spectral estimators

The method for the detection of cell cyclic elements is as follows:

1. Estimate the magnitude spectrum of the p^{th} gene as $\hat{\phi}_p(\omega)$, for $p = 1, \ldots, P$.

2. Estimate the ensemble average[2] of the amplitude spectra:

$$\bar{\hat{\phi}}_p = \frac{1}{P}\sum_{p=1}^{P} \hat{\phi}_p(\omega) \tag{2.3.30}$$

   A dominant peak in $\bar{\hat{\phi}}_p$ occurs if a significant number of genes have cell cyclic components. This frequency location of this peak provides an estimate of the cell cycle frequency ω_{cc}.

3. Rank the P genes according to their spectral amplitude at the estimated cell cycle frequency, i.e. $\hat{\phi}_p(\omega_{cc})$.

The use of the ensemble average for cell cycle frequency detection was first proposed explicitly in [21]. We test the performance of the spectral estimators on microarray data from the *alpha* experiment

[2]Note that this is not an ensemble in the strict statistical sense, unless each gene can be viewed as one realisation of a single underlying process.

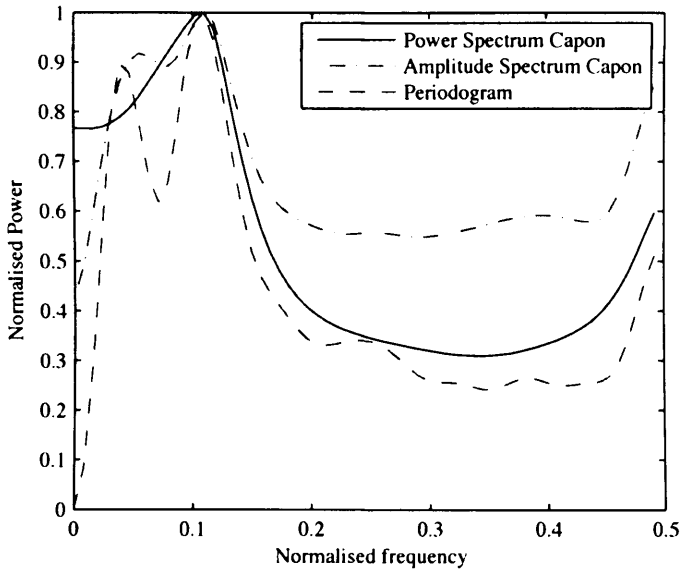


Figure 2.1. Ensemble average power spectrum estimates from the *alpha* data, with the Periodogram, Amplitude Spectrum Capon and Power Spectrum Capon estimators. The filter length for the PSC and ASC methods was 7. Note the distinct peak in the cell cycle location.

from [1]. *In the context of the work in this Chapter the simulation studies are based on real microarray data as no theoretically justified quantative models are available for the production of synthetic datasets.* The data length is $N = 18$, with $P = 6075$ usable gene profiles.

Figure 2.1 shows the resulting ensemble average spectrum for the periodogram, PSC and ASC methods. The filter length for the PSC and ASC methods is 7. The figure displays the distinct peak expected from a dataset with a significant cell cyclic component. The three methods all place the peak in a similar location but the estimate of the remainder of the spectrum is rather more varied.

One of the issues with the PSC and the ASC estimators is the need to specify a filter length L. The filter length effectively governs a bias/variance tradeoff [41]. A longer filter length enables a higher resolution estimate at the expense of higher variance. Figure 2.2 shows

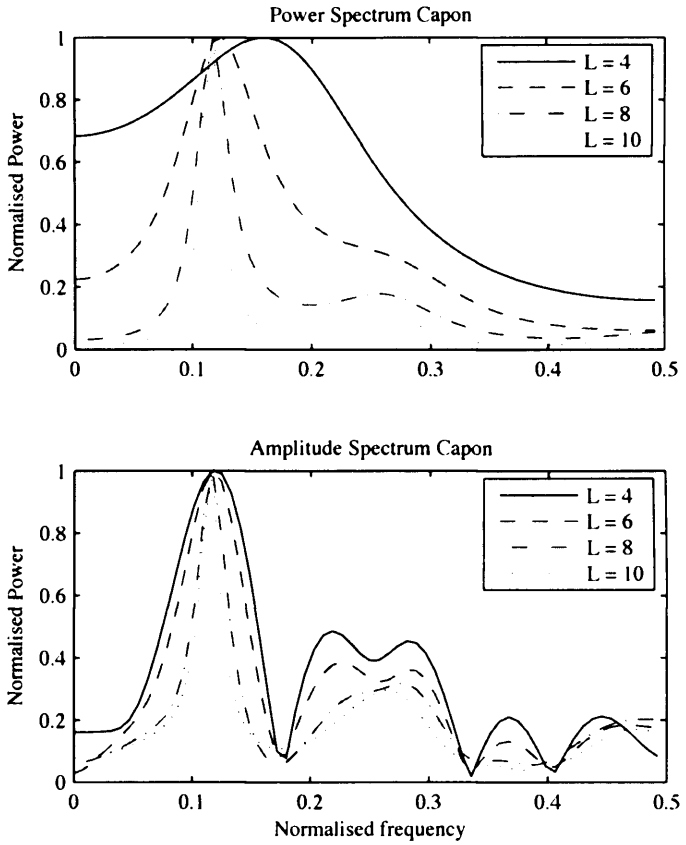


Figure 2.2. Power spectrum estimates of an example gene (YGR065C) in the *alpha* data using the PSC and ASC estimators for a range of filter lengths. A spectrum with sharper peaks is indicative of a higher resolution estimate. The increase in resolution with filter length is clear.

the PSC and ASC estimates for a range of filter lengths on a single example cell cyclic gene. The increase in resolution with filter length is clear, with the ASC giving a higher resolution estimate than the PSC for a given filter length. Given a long filter length, both methods are capable of giving a very high resolution estimate [42]. However, with data lengths such as $N = 18$ the variance is likely to be significant. A high filter length will place a precise peak, but the short data length and high noise could mean that this peak is misplaced. If this situation is replicated over the full set of gene profiles then the resulting 'jitter' in the frequency location could have an unpredictable effect on the

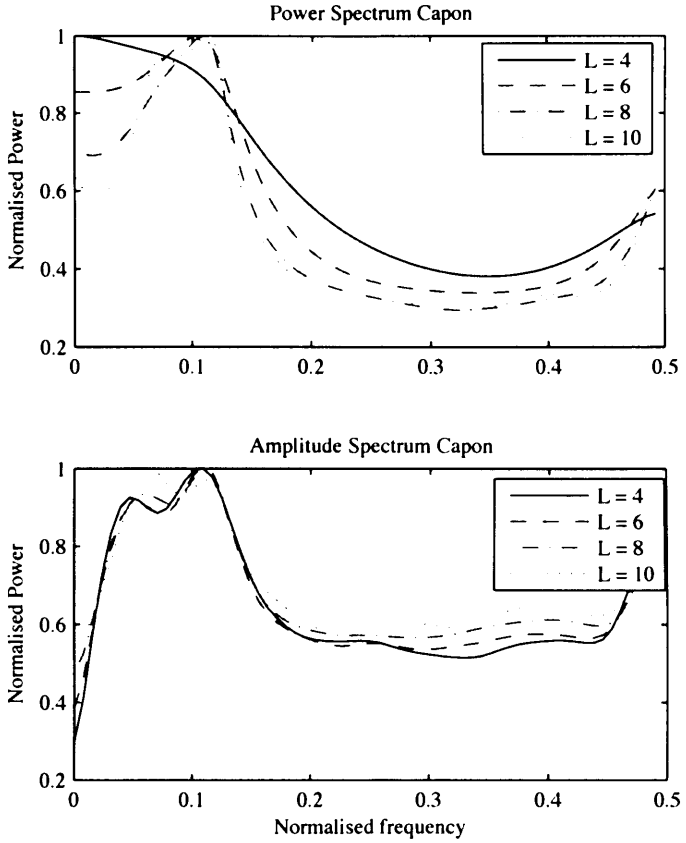


Figure 2.3. Ensemble average power spectrum estimates for the *alpha* data using the PSC and ASC estimators for a range of filter lengths. The effect of filter length on the ensemble average is varied.

ensemble average. Figure 2.3 shows the effect on the ensemble average of the variation of filter length. The ASC clearly shows less variation in ensemble average with filter length but for a long filter length $L = 10$ the peak is placed in a rather different location. Clearly, a poor choice of filter length could result in an erroneous estimate of the frequency location.

2.3.3 Robust Capon approach

One of the problems inherent in microarray data is timing errors. The cell populations in microarray time course experiments are sampled by hand and so the timing is imprecise. For example, in [1] the nominal

sampling period in the *alpha* factor experiments was 7 minutes but errors were estimated at up to 20 seconds [43]. We design a spectral estimator to be robust to mis-sampling [44], based on the robust Capon beamformer.

The robust Capon beamformer (RCB) presented in [45, 46] is able to determine the power in a signal of interest given imprecise knowledge of the array steering vector. As the beamforming problem is directly analogous to spectral estimation, the steering vector uncertainty is equivalent to uncertainty in the Fourier vector in the case of spectral estimation. Here, we show that errors in temporal sampling can be represented as an uncertainty disc around the Fourier vector. Let the ideally sampled data be represented as

$$y(t) = \alpha_\omega e^{i\omega t} + n(t), \tag{2.3.31}$$

for $t = 0, \ldots, N-1$, where α_ω is the (complex) amplitude of a generic sinusoidal component at frequency ω, where $\omega \in [0, 2\pi)$, and $n(t)$ is an additive zero mean coloured noise process containing component power at frequencies other than ω (see, e.g., [36]). Introducing sampling errors, we rewrite (2.3.31) as

$$y(t) = \alpha_\omega e^{i\omega(t+\Delta_t)} + n(t), \tag{2.3.32}$$

where Δ_t is a random variable representing the sampling error at time t. Here, we make the natural assumption that $\{\Delta_t\}_{t=0}^{N-1}$ are independent identically distributed (IID) variables, with $\Delta_t \sim N(0, \sigma_\Delta^2)$, where σ_Δ^2

models the level of uncertainty in the sampling process. Let

$$\begin{aligned}\mathbf{y}_L(t) &= \begin{bmatrix} y(t) & y(t+1) & \dots & y(t+L-1) \end{bmatrix}^T \\ &= \alpha_\omega \widetilde{\mathbf{a}}_L e^{i\omega t} + \mathbf{e}_L(t), \end{aligned} \tag{2.3.33}$$

for $t = 0, \dots, N-L$,

$$\begin{aligned}\widetilde{\mathbf{a}}_L &= \mathbf{a}_L \odot \mathbf{a}_\Delta \\ \mathbf{a}_L &= \begin{bmatrix} 1 & e^{i\omega} & \dots & e^{i\omega(L-1)} \end{bmatrix}^T \\ \mathbf{a}_\Delta &= \begin{bmatrix} e^{i\omega\Delta_t} & e^{i\omega\Delta_{t+1}} & \dots & e^{i\omega\Delta_{t+L-1}} \end{bmatrix}^T \end{aligned}$$

with $\odot$ denoting the Schur-Hadamard (elementwise) product. To form the uncertainty region created by the sampling uncertainty, we proceed to evaluate the expected value and the covariance matrix of $\widetilde{\mathbf{a}}_L$. The expectation of $\widetilde{\mathbf{a}}_L$ is

$$\bar{\widetilde{\mathbf{a}}}_L = E\left[\widetilde{\mathbf{a}}_L\right] \tag{2.3.34}$$

$$= \mathbf{a}_L \odot E\left[\mathbf{a}_\Delta\right] \tag{2.3.35}$$

$$= \mathbf{a}_L E\left[e^{i\omega\Delta_t}\right] \tag{2.3.36}$$

where we exploited the assumption that $\{\Delta_t\}_{t=0}^{N-1}$ are IID. Noting that $E\left[e^{i\omega\Delta_t}\right]$ is the characteristic function of a zero-mean Gaussian random variable yields

$$\bar{\widetilde{\mathbf{a}}}_L = e^{\frac{-\omega^2\sigma_\Delta^2}{2}} \mathbf{a}_L \tag{2.3.37}$$

Similarly,

$$\begin{aligned} \mathbf{C}_{\tilde{\mathbf{a}}} &= E\left[(\widetilde{\mathbf{a}}_L - \bar{\widetilde{\mathbf{a}}}_L)(\widetilde{\mathbf{a}}_L - \bar{\widetilde{\mathbf{a}}}_L)^H\right] \\ &= \left(1 - e^{-\omega^2\sigma_\Delta^2}\right)\mathbf{I}_L \end{aligned} \tag{2.3.38}$$

This covariance model for the sampling uncertainties could be easily enhanced with additional prior knowledge from laboratory experiments. Based on the above derivation, we assume that $\widetilde{\mathbf{a}}_L$ belongs to the uncertainty ellipsoid

$$\left(\widetilde{\mathbf{a}}_L - \bar{\widetilde{\mathbf{a}}}_L\right)^H \mathbf{C}_{\tilde{\mathbf{a}}}^{-1}\left(\widetilde{\mathbf{a}}_L - \bar{\widetilde{\mathbf{a}}}_L\right) \leq 1 \tag{2.3.39}$$

where $\mathbf{C}_{\tilde{\mathbf{a}}}$ is given by (2.3.38). Using (2.3.38), the hyperspherical uncertainty region is given by

$$\left\| \widetilde{\mathbf{a}}_L - e^{\frac{-\omega^2\sigma_\Delta^2}{2}}\mathbf{a}_L \right\|_2 \leq \epsilon \tag{2.3.40}$$

where $\epsilon = \beta\left(1 - e^{-\omega^2\sigma_\Delta^2}\right)$, and $\|\cdot\|_2$ denotes the Euclidean norm. Note that the radius of the hypersphere is a function of ω and σ_Δ. The reliance on σ_Δ is, of course, expected, but the presence of ω is also intuitive as the estimation of the spectral content at low frequency should be less affected by sampling errors than at higher frequencies. The extra scalar parameter β allows the uncertainty disc to be extended to give a more conservative estimate, which is useful for allowing extra unstructured uncertainty due to short data lengths and unknown noise characteristics. The robust Capon estimator [45, 46] is then obtained

using the solution to the constrained optimisation

$$\min_{\widetilde{\mathbf{a}}_L} \widetilde{\mathbf{a}}_L^H \mathbf{R}^{-1} \widetilde{\mathbf{a}}_L \quad \text{subject to} \quad \left\| \widetilde{\mathbf{a}}_L - \bar{\widetilde{\mathbf{a}}}_L \right\|_2 \leq \epsilon \tag{2.3.41}$$

where $\mathbf{R}$ is the (estimated) covariance matrix of the measured data. To eliminate the trivial solution $\widetilde{\mathbf{a}}_L = \mathbf{0}$, it is assumed that $\|\bar{\widetilde{\mathbf{a}}}_L\|_2^2 > \epsilon$. In this case, the solution will lie on the boundary of the constraint, simplifying the problem to a minimization with equality constraint

$$\min_{\widetilde{\mathbf{a}}_L} \widetilde{\mathbf{a}}_L^H \mathbf{R}^{-1} \widetilde{\mathbf{a}}_L \quad \text{subject to} \quad \left\| \widetilde{\mathbf{a}}_L - \bar{\widetilde{\mathbf{a}}}_L \right\|_2 = \epsilon \tag{2.3.42}$$

The solution to (2.3.42) is obtained using a Lagrange multiplier [45]

$$f = \widetilde{\mathbf{a}}_L^H \mathbf{R}^{-1} \widetilde{\mathbf{a}}_L + \lambda \left(\left\| \widetilde{\mathbf{a}}_L - \bar{\widetilde{\mathbf{a}}}_L \right\|_2^2 - \epsilon \right) \tag{2.3.43}$$

The optimal solution $\hat{\widetilde{\mathbf{a}}}_L$ is found by differentiation of (2.3.43) with respect to $\widetilde{\mathbf{a}}_L$, yielding the solution:

$$\hat{\widetilde{\mathbf{a}}}_L = \bar{\widetilde{\mathbf{a}}}_L - (\mathbf{I} + \lambda \mathbf{R})^{-1} \bar{\widetilde{\mathbf{a}}}_L \tag{2.3.44}$$

The Lagrange multiplier λ is obtained by the solution of the constraint equation:

$$g(\lambda) \triangleq \left\| (\mathbf{I} + \lambda \mathbf{R})^{-1} \bar{\widetilde{\mathbf{a}}}_L \right\|_2^2 = \epsilon \tag{2.3.45}$$

A unique solution to (2.3.45) is obtained through gradient descent (see [45] for details and the formulation of upper and lower bounds). With the Lagrange multiplier determined, $\hat{\widetilde{\mathbf{a}}}_L$ is given by (2.3.44). The robust Capon spectral estimate is given by using $\hat{\widetilde{\mathbf{a}}}_L$ in place of $\mathbf{a}_L$ in the classical power spectrum Capon estimator, i.e. the estimated power

spectral density is obtained as

$$\phi(\omega) = \frac{1}{\hat{\tilde{\mathbf{a}}}_L^H \mathbf{R}^{-1} \hat{\tilde{\mathbf{a}}}_L} \tag{2.3.46}$$

The robust Capon estimator was tested on the *alpha* factor microarray data from [1]. The cell cycle frequency was estimated using the ensemble average and the data pre-processed in the same manner as for the PSC and ASC methods, outlined in the previous section. The estimates given in Figure 2.4 show typical examples of genes which in [1] were judged to be cell-cycle regulated. In all cases, the robust Capon estimator places a definite peak at the location of the estimated cell cycle frequency. The classical Capon tends to place a very sharp peak in the vicinity of the cell cycle frequency but the amplitude value at the cell cycle frequency can be relatively low. It is likely that the sharp peak is misplaced because of the significant uncertainty in the data. The periodogram has a broader peak, but this too is often misplaced and, as expected, suffers from spurious peaks due to the large sidelobes. The periodogram and classical Capon estimators both show more variation than the robust Capon estimator in the spectrum outside the region containing the estimated cell cycle.

2.4 Beamforming methods for cell cycle detection

In the previous section, we considered the use of data adaptive filter-bank spectral estimators. It is worth noting that the resulting set of L-tap filters, for each frequency, will differ for each gene - we are thus effectively designing P filters for *each and every* frequency. We now proceed to instead form only a single filter for each frequency, i.e. for

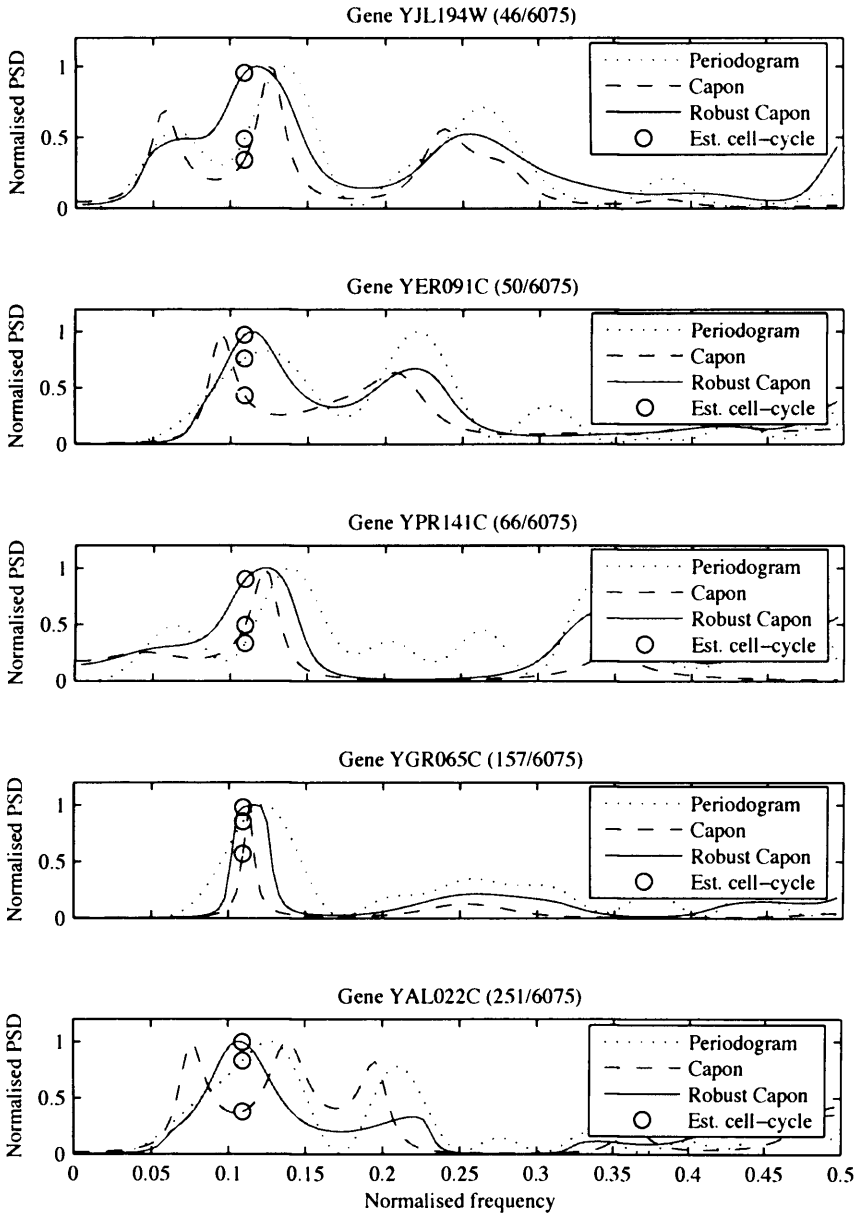


Figure 2.4. Spectrum estimates of selected genes by robust Capon and classical Capon and periodogram methods. The estimated cell cycle frequency is circled. Both axes are normalised.

each particular frequency we suggest applying the same filter to all genes.

This conceptual difference gives several benefits: first, we can exploit all the P genes to construct the frequency dependent filters, and secondly, each filter can be extended to N taps in length. Previously, the filters were restricted to be $L < \frac{N}{2}$ long to ensure that the required covariance matrix be non-singular. However, if all the P genes are used to form the filters, these can be extended to N taps without loss of rank in the covariance matrix, yielding significantly higher resolution in the resulting spectral estimates.

Let $\mathbf{h}_\omega$ denote the N-tap data adaptive filter designed to minimise the power of the filter output, while passing the frequency of interest, ω undistorted, i.e.

$$\mathbf{h}_\omega = \arg\min_{\mathbf{h}_\omega} \quad \mathbf{h}_\omega^H \mathbf{R} \mathbf{h}_\omega \quad \text{subject to} \quad \mathbf{h}_\omega^H \mathbf{a}_\omega = 1 \tag{2.4.1}$$

where $\mathbf{R}$ is the covariance matrix of the considered gene. As $\mathbf{R}$ is unknown, we form an estimate by averaging all the P genes, i.e. $\mathbf{R} = \frac{1}{P} \sum_k \mathbf{x}_k^T \mathbf{x}_k$, where $\mathbf{x}_k$ denotes a row vector containing the samples of one gene, implicitly assuming that each gene has the same statistical properties. This is clearly not the case, and we will comment further on this assumption below. Examining (2.4.1) we note that this formulation is identical to the MVDR beamformer; clearly designing a single filter for each frequency, for all the genes results in a problem formulation identical to the traditional beamformer. Beamforming is the spatial equivalent of the spectral estimation problem. In this case, the data are obtained from a number of sensors typically located in a linear,

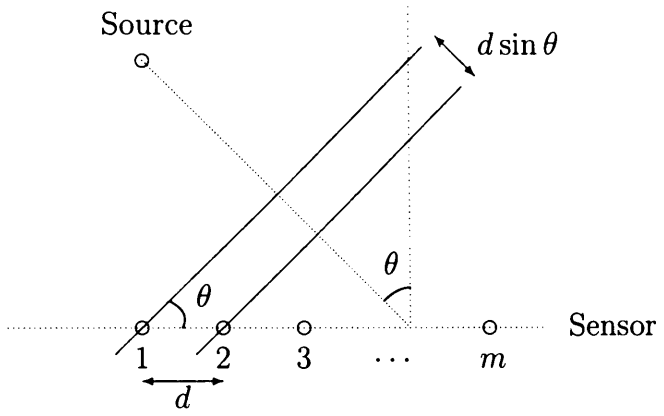


Figure 2.5. Uniform linear array diagram with the assumption that the source is in the far-field, so that the wavefront which traverses the array can be assumed planar.

preferably uniform, array as shown in Figure 2.5. Hence, the problem of interest is to determine the direction of arrival (DOA) of sources, typically located far away from the array. In this case, the estimation of the power as a function of DOA, which can also be expressed as spatial frequency, is directly analogous to the estimation of power in the frequency location, ω, in the case of spectral estimation. Thus, by using beamforming approaches designed to estimate the power distribution as a function of the spatial frequency, we will obtain an estimate of the distribution of power over the entire data set, providing an estimate equivalent to the previously discussed ensemble average. In our problem domain, we use the beamforming framework by treating each gene data vector as an N-dimensional sample impinging on an array of N sensors. Note that, whereas in the ensemble average approach we effectively had N scalar samples, we now have P vectorial samples. In our application, $P \gg N$ and so we can expect significant benefits from this approach.

We note that in the beamforming formulation, the impinging sources do not need to be stationary, i.e. each of the P N-tap vectorial data samples are not required to have the same statistical properties; the resulting spatial spectral estimate will only indicate from which DOA power is impinging on the array - not when it does so. Indeed, the time dimension usually present in beamforming is instead replaced by a set of gene data - the ordering of which is entirely arbitary. Thus, in our case, we do not require the false assumption that all the P genes share the same statistical properties. Our method will hence measure the frequency content of the entire dataset, not from which genes it originates. Hereafter, we denote the suggested approach a temporal beamformer to stress the similarities to the spatial beamformer in the array case.

To reflect the different problems, the following beamforming derivations will use θ, consistent with the beamforming approach, whilst plots will be in frequency, reflecting the trivial conceptual transformation to our original problem domain. The array covariance matrix can be expressed:

$$\mathbf{R_X} = E\left[\mathbf{X}^T\mathbf{X}\right] \tag{2.4.2}$$

where $\mathbf{X}$ denotes the full $P \times N$ data matrix. The strength of this formulation is clear; an estimate of the covariance matrix will be highly robust because each of the P genes is now viewed as a sample. The typically very high values of P ensure an excellent quality covariance matrix estimate. The power in the direction of arrival (and hence frequency in the original problem) onto the array can now be determined using a beamforming approach.

2.4.1 The standard beamformer

The power in a given direction of arrival θ can be estimated by filtering the incoming data with a spatial filter, $\mathbf{h}_\theta$. The power in the filter output for a given direction θ can be expressed as

$$\phi(\theta) = \mathbf{h}_\theta^H \mathbf{R}_\mathbf{X} \mathbf{h}_\theta \tag{2.4.3}$$

The filter can be designed in various ways; the filter used in the standard beamformer is the array steering vector, $\mathbf{a}_\theta$, normalised by the number of array sensors, i.e.

$$\mathbf{h}_\theta = \frac{\mathbf{a}_\theta}{\sqrt{N}} \tag{2.4.4}$$

For a uniformly spaced array, the array steering vector is given by

$$\mathbf{a}_\theta = \left[1 \; e^{2\pi\theta i} \; \ldots \; e^{2\pi\theta i(N-1)}\right]^T \tag{2.4.5}$$

Figure 2.6 shows the estimate of the power spectrum of the *alpha* data using the standard beamformer and the ensemble average periodogram. It is worth stressing that the estimates for the ensemble average periodogram and the standard beamformer are identical. The ensemble average periodogram estimate is given by

$$\phi(\omega) = \frac{1}{P} \sum_{n=1}^{P} \frac{\mathbf{a}_\omega^H}{N} \mathbf{R}_{\mathbf{X}n} \frac{\mathbf{a}_\omega}{N} \tag{2.4.6}$$

$$= \frac{\mathbf{a}_\omega^H}{N} \frac{1}{P} \sum_{n=1}^{P} \mathbf{R}_{\mathbf{X}n} \frac{\mathbf{a}_\omega^H}{N} \tag{2.4.7}$$

$$= \frac{\mathbf{a}_\omega^H}{N} \frac{1}{P} \mathbf{X}^T \mathbf{X} \frac{\mathbf{a}_\omega}{N} \tag{2.4.8}$$

$$= \frac{\mathbf{a}_\omega^H}{N} E\left[\mathbf{X}^T \mathbf{X}\right] \frac{\mathbf{a}_\omega}{N} \tag{2.4.9}$$

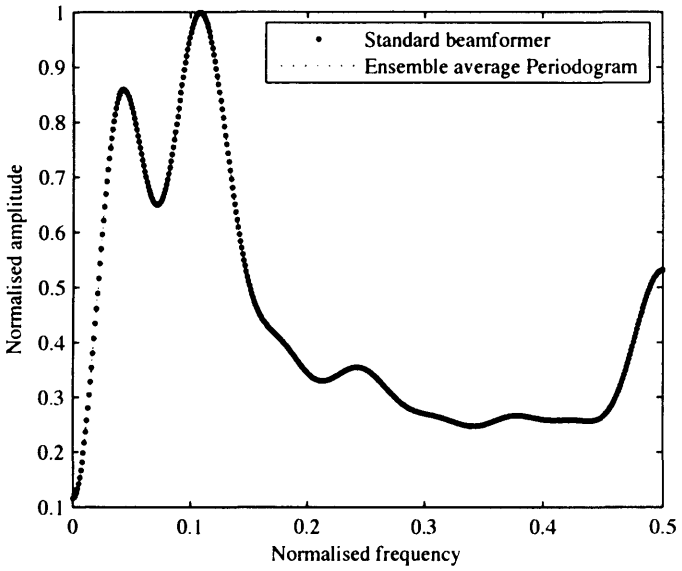


Figure 2.6. Power spectrum of *alpha* data using the standard beamformer and the ensemble average periodogram. The two estimates are coincident.

which is equivalent to the standard beamformer solution:

$$\phi(\theta) = \frac{\mathbf{a}_\theta^H}{N} E\left[\mathbf{X}^T\mathbf{X}\right] \frac{\mathbf{a}_\theta}{N} \tag{2.4.10}$$

As mentioned previously, the strength in the beamforming formulation lies in the accuracy of the covariance estimate. However, as the standard beamformer filter design is invariant to the data this robustness is not fully exploited, yielding the same spectral estimates as the ensemble average discussed in Section 2.3.2. A data adaptive method should be able to use the accurate covariance estimate afforded by the beamforming approach to yield a filter that is, in some sense, optimal. One such method is now examined.

2.4.2 The Capon beamformer

The Capon beamformer[3] seeks to minimise the power in the filter output whilst passing power from the direction of interest undistorted. The optimisation is hence [37]

$$\mathbf{h}_\theta = \arg\min_{\mathbf{h}_\theta} \quad \mathbf{h}_\theta^H \mathbf{R} \mathbf{h}_\theta \quad \text{subject to} \quad \mathbf{h}_\theta^H \mathbf{a}_\theta = 1 \tag{2.4.11}$$

The minimisation problem in (2.4.11) has the well known solution [37]

$$\mathbf{h}_\theta = \frac{\mathbf{R}^{-1}\mathbf{a}_\theta}{\mathbf{a}_\theta^H \mathbf{R}^{-1} \mathbf{a}_\theta} \tag{2.4.12}$$

[3]Also known as the Minimum Variance Distortionless Response (MVDR) beamformer.

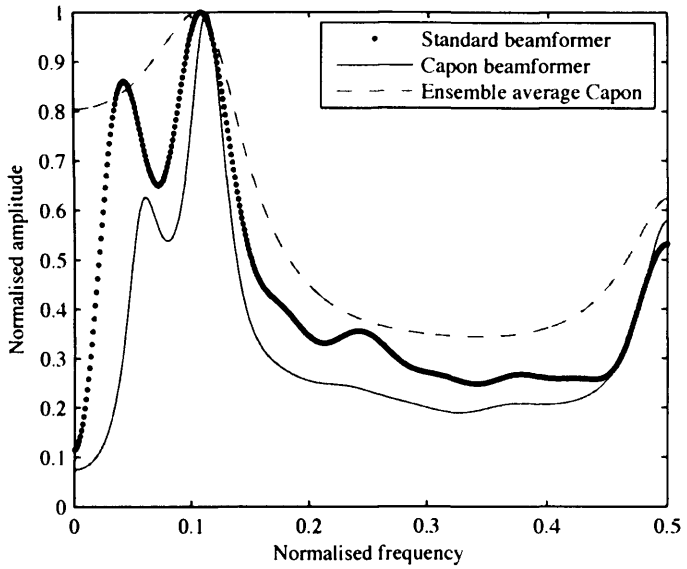


Figure 2.7. Power spectrum of *alpha* data using the standard beamformer, the Capon beamformer and the ensemble average Capon estimate. The ability of the Capon beamformer to take advantage of the robust covariance matrix estimate afforded by the beamforming formulation allows the use of a longer filter length and results in a higher resolution estimate.

which, if inserted into (2.4.3), yields the Capon estimate of the spatial spectrum

$$\phi(\theta) = \frac{1}{\mathbf{a}_\theta^H \mathbf{R}^{-1} \mathbf{a}_\theta} \tag{2.4.13}$$

It is interesting to note that the beamforming approach allows the use of a filter length equal to the full data length, yielding significantly higher resolution in the resulting spectral estimate[4]. Figure 2.7 shows the normalised estimates obtained from the Capon beamformer, along with the estimates from the standard beamformer, and the ensemble average Capon. As is clear from the figure, the Capon beamformer yields a higher resolution spectral estimate compared to the standard

[4]Note that, for the filterbank approaches discussed in Section 2.3.1, L is limited to $L < N/2$ to ensure that the used covariance matrix estimate is non-singular. Here, we note that the covariance matrix estimate will be full rank even for $L = N$. However, this is not always the case - see also the discussion in Section 2.4.3.

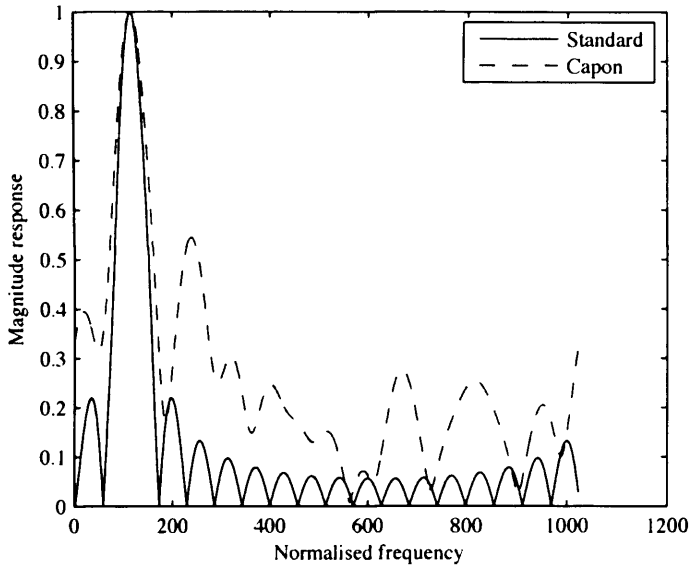


Figure 2.8. Magnitude of filter responses for the Capon and standard beamformers at the estimated cell cycle frequency.

beamformer and its own ensemble average equivalent. In contrast to the ensemble average method, the use of a maximal filter length with an accurate covariance estimate allows a high resolution estimate of the power spectrum to be obtained. To illustrate the data dependence of the Capon filter, Figure 2.8 shows the magnitude response of the filter $\mathbf{h}_\theta$ at the estimated cell cycle frequency. The Capon filter is shown to adapt to the data by placing sharp nulls in regions of significant interference. Notice that the Capon filter gain can be higher than the beamformer filter in (2.4.4). It is important to stress that it will only be so for frequencies containing little or no power, and the increased gain thus will not have any significant adverse effect on the resulting spectral estimates. Furthermore, the Capon filter will place deep nulls at the locations of power different from the frequency of interest, ensuring that these frequencies do not significantly affect the estimate.

2.4.3 Removing zero frequency values

Gene expression profiles over time are not constrained to be zero mean, and the row means of the data matrix $\mathbf{X}$ are generally non zero. The mean value of the case study data sets from [1] are given in Table 2.1. Naturally, this mean value is reflected as a zero frequency component

X	Mean absolute values of row means
alpha	0.0097
cdc15	0.3587
cdc28	0.1900
elu	0.0025

Table 2.1. Mean absolute values of row means for the case study data sets from [1]

in the power spectrum. The zero frequency component is of little interest and may easily dominate the power spectrum. A trivial solution is to subtract the row mean from each row of $\mathbf{X}$. Figure 2.9 shows the estimated power spectrum of the *cdc28* data using the standard beamformer, with and without first subtracting the row means from $\mathbf{X}$. The significant zero frequency components in the *cdc28* data are clear from the power spectrum, resulting in an erroneous cell-cycle frequency estimate. Clearly, subtraction of the row mean from the data is necessary for accurate assessment of the power spectrum in data sets with significant row means, especially where the frequency of interest lies relatively close to zero. Removing the row mean would seem to be a simple solution to the problem of zero frequency components dominating the spectrum. However, for the Capon beamformer doing so will present a problem. Subtraction of the row means results in the loss of a linearly independent component, and a corresponding drop in rank of the covariance matrix $\mathbf{R}$. As seen in (2.4.13), the Capon solution

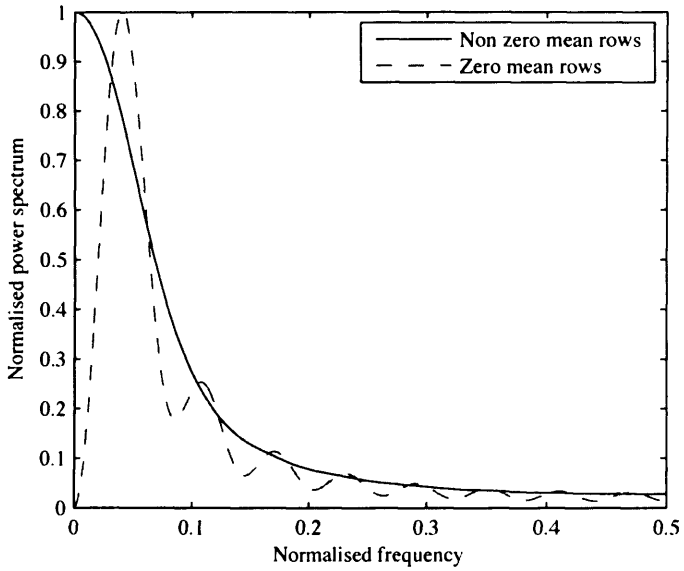


Figure 2.9. Power spectrum estimate from the standard beamformer with zero row mean *cdc28* data and non zero row mean *cdc28* data.

depends on $\mathbf{R}$ being invertible, and hence full rank. Possible solutions to this problem will now be examined.

Generalised optimum approach and diagonal loading

As discussed in Section 2.4.2, the Capon beamformer is given by equation (2.4.13), which requires that $\mathbf{R}$ is full rank. However, a more general solution, that does not assume full rank $\mathbf{R}$ exists [47]. The general Capon solution is given by

$$h_\theta = \frac{\left(\mathbf{R} + \mathbf{a}_\theta \mathbf{a}_\theta^{\mathbf{H}}\right)^\dagger \mathbf{a}_\theta}{\mathbf{a}_\theta^{\mathbf{H}} \left(\mathbf{R} + \mathbf{a}_\theta \mathbf{a}_\theta^{\mathbf{H}}\right)^\dagger \mathbf{a}_\theta} \tag{2.4.14}$$

where $(\cdot)^\dagger$ denotes the Moore-Penrose pseudo-inverse, i.e. $\mathbf{A}^\dagger = \left(\mathbf{A}^H \mathbf{A}\right)^{-1} \mathbf{A}^H$. This solution works but is sensitive to the numerical tolerances used in the calculation of the pseudo inverse. However, it can be shown that this solution is equivalent to simply diagonally loading $\mathbf{R}$ by a factor of λ, yielding the solution [47]

$$h_\theta = \frac{\left(\mathbf{R} + \lambda \mathbf{I}\right)^{-1} \mathbf{a}_\theta}{\mathbf{a}_\theta^{\mathbf{H}} \left(\mathbf{R} + \lambda \mathbf{I}\right)^{-1} \mathbf{a}_\theta} \tag{2.4.15}$$

The loading factor λ is chosen to ensure that $(\mathbf{R} + \lambda \mathbf{I})$ is sufficiently well conditioned to allow for numerically stable matrix inversion. However, in practice, the solution is fairly insensitive to the choice of λ. Figure 2.10 shows the performance of the Capon beamformer with diagonal loading of $\lambda = 0.05$ on the *alpha* data with zero mean rows. This has clearly achieved our objective in combining the high resolution Capon estimate with the elimination of the zero frequency component. Note that the Capon beamformer given in (2.4.13) can not be computed for this case without diagonally loading due to the singular

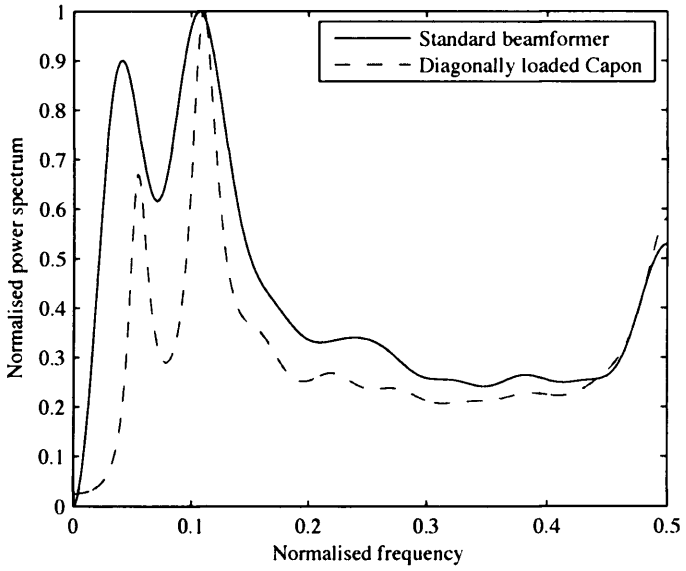


Figure 2.10. Power spectrum estimate from the standard beamformer and the diagonally loaded Capon beamformer with zero row mean *alpha* data.

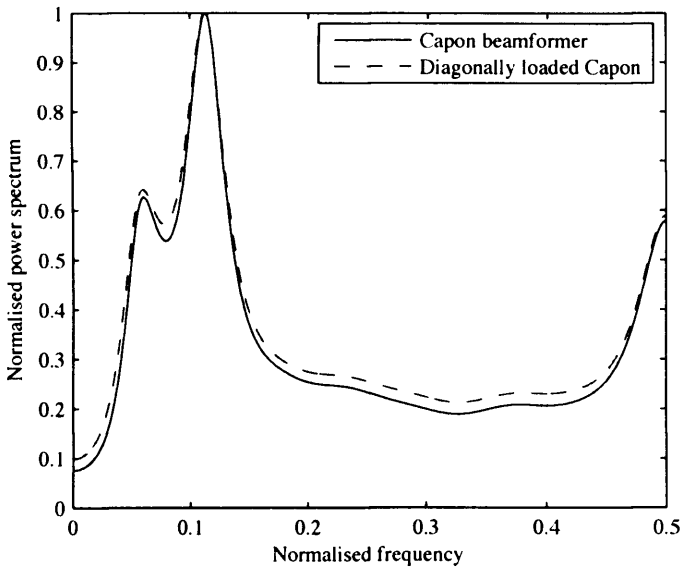


Figure 2.11. Power spectrum estimate from the Capon beamformer and the diagonally loaded Capon beamformer with non zero row mean *alpha* data.

covariance matrix.

However, it is natural to assume that the diagonal loading causes some loss of resolution in the Capon beamformer. Figure 2.11 shows the normalised power spectrum estimate of the Capon beamformer and the diagonally loaded Capon, with $\lambda = 0.05$, using the *alpha* data which has non zero mean rows. Clearly, the diagonal loading has not significantly degraded the power spectrum. We conclude that the diagonally loaded Capon is able to cope with the rank deficient case caused by enforcing zero mean rows with negligible loss of resolution.

Forward-backward covariance matrix estimate

The standard covariance estimate can be improved in the case where $\mathbf{R}$ is centro-Hermitian [48], which means that

$$\mathbf{R} = \mathbf{J}\mathbf{R}^H\mathbf{J} \tag{2.4.16}$$

where the exchange matrix

$$\mathbf{J} = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ 0 & \ddots & 1 & 0 \\ \vdots & \cdots & \vdots & \vdots \\ 1 & \cdots & 0 & 0 \end{bmatrix} \tag{2.4.17}$$

$\mathbf{R}$ will be centro-Hermitian for the case of uniformly sampled real data. A suitable estimate for the case where $\mathbf{R}$ is centro-Hermitian (and real) is the forward-backward estimate:

$$\mathbf{R}_{\text{FB}} = \frac{1}{2}\left(\mathbf{R} + \mathbf{J}\mathbf{R}^T\mathbf{J}\right) \tag{2.4.18}$$

The forward-backward estimate has been shown to have half the asymptotic bias of the standard estimate in cases where $\mathbf{R}$ is centro-Hermitian [48]. In addition it is full rank, even in cases where $\mathbf{X}$ has enforced zero mean rows. Hence, for the case of uniform sampling, the forward-backward estimate of $\mathbf{R}$ can be used which is invertible and avoids the need for diagonal loading.

2.4.4 Non-uniform sampling

The beamforming approach is able to estimate the spectrum given non-uniformly sampled data. This is equivalent to the non-linearly spaced array case, and requires only the adjustment of the steering vector. Writing the steering vector as

$$\mathbf{a}_\theta = \begin{bmatrix} 1 & e^{i2\pi\theta} & \ldots & e^{i2\pi\theta(N-1)} \end{bmatrix} \tag{2.4.19}$$

$$= e^{i2\pi\theta\mathbf{\Lambda}} \tag{2.4.20}$$

where $\mathbf{\Lambda}$ is the time index vector, given by $\begin{bmatrix} 0 & 1 & \dots & N-1 \end{bmatrix}$ in the uniform sampling case. More generally, given a time vector:

$$\mathbf{t} = \begin{bmatrix} t_1 & t_2 & \dots & t_N \end{bmatrix} \tag{2.4.21}$$

which can be non-uniformly sampled, the sampling periods are given by

$$\mathbf{\Delta}_t = \begin{bmatrix} (t_2 - t_1) & (t_3 - t_2) & \dots & (t_N - t_{N-1}) \end{bmatrix} \tag{2.4.22}$$

The effective sampling time, δ_t, is defined as

$$\delta_t = gcd\left[\mathbf{\Delta}_t\right] \tag{2.4.23}$$

where $gcd\left[\cdot\right]$ denotes the greatest common divisor. The corresponding time index vector $\mathbf{\Lambda}$ can be defined recursively as

$$\Lambda_n = \begin{cases} 0 & \text{for } n = 1 \\ \Lambda_{n-1} + \frac{\Delta_{n-1}}{\delta} & \text{for } n = 2 \text{ to } N \end{cases} \tag{2.4.24}$$

To demonstrate the method, Figure 2.12 shows the normalised spectrum of the *cdc15* data, which is non-uniformly sampled.

2.5 Assessment of the cell-cyclic content of individual genes

The beamforming method gives an accurate assessment of the frequency content of a microarray dataset. A dominant peak indicates the presence of an underlying periodic component. In our test data sets, this component is the cell cycle. The location of the dominant peak hence provides an estimate of the cell cycle frequency. We now present a

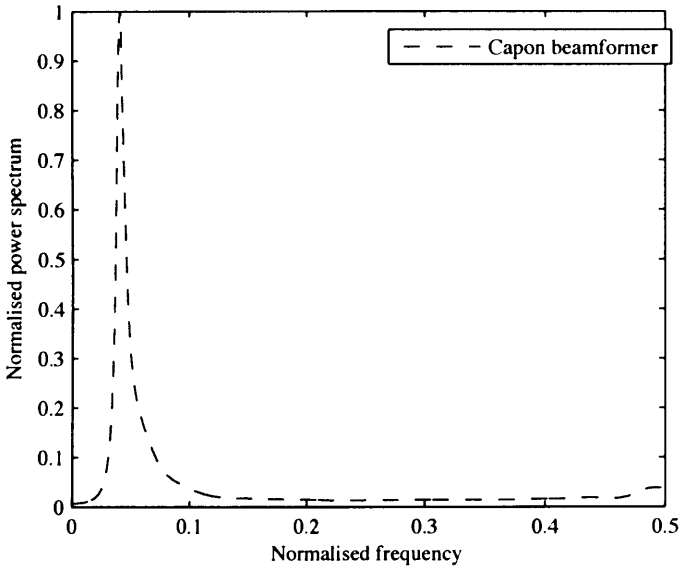


Figure 2.12. Power spectrum estimate from diagonally loaded Capon beamformer with non-uniformly sampled *cdc15* data. This dataset clearly contains a very dominant cell-cyclic component.

coherent method for the assessment of the extent to which this component is present in each gene. Our beamforming method has provided us with a set of filters $\mathbf{h}_\omega$. The time profile for each gene can now be filtered by $\mathbf{h}_\omega$ at the cell cycle frequency ω_{cc}. The power in the filter output now yields an estimate of the power in an individual gene at the cell cycle frequency. The estimate of the power in the expression profile of the p'th gene at the cell cycle frequency is hence given by

$$\sigma_p = \mathbf{h}_{\omega_{cc}}^H \frac{\mathbf{x}_p \mathbf{x}_p^H}{N} \mathbf{h}_{\omega_{cc}} \tag{2.5.1}$$

These values can be ranked to give an estimate of the relative power present in each gene at the estimated cell cycle location. This measure clearly depends heavily on the variance of individual gene expression profiles. A gene profile with a high variance will rank higher than one with a low variance, though the sinusoidal component may not be as distinct. Normalising the rows of $\mathbf{X}$ to unity variance solves this problem but many low amplitude gene profiles are biologically not significant. A good approach is to compare the power in the filter output with that of several random permutations of the gene expression profile [30], i.e. for each gene the time points are randomly permuted and the power in the filter output calculated. If the obtained power for each random permutation is consistently lower than the power of the true permutation then a sinusoidal component is deemed present with high confidence. The proportion of times that the true permutation yields the highest power is a measure of the confidence that the sinusoidal component is present. Hence, the power in the k'th random permutation of the expression profile of the p'th gene at the estimated cell cycle

frequency is given by

$$\sigma_{p,k} = \mathbf{h}_{\omega_{cc}}^{H} \frac{P_k\left[\mathbf{x}_p\right] P_k\left[\mathbf{x}_p^{H}\right]}{N} \mathbf{h}_{\omega_{cc}} \tag{2.5.2}$$

where $P_k[\cdot]$ represents the random permutation of the k-th trial. Given K random permutations or trials, our measure of the extent to which the sinusoidal component is present in the p'th gene is denoted by Ω_p and given by

$$\text{for} \quad k = 1 \quad \text{to} \quad K \tag{2.5.3}$$

$$\Omega_p = \Omega_p + \frac{1}{K} \quad \text{iff} \quad \sigma_p > \sigma_{p,k} \tag{2.5.4}$$

For example, a value of $\Omega_p = 1$ means that the power in the true permutation is greater than all other tested permutations, giving confidence that the sinusoidal component is present (given a high enough number of trials). A value of $\Omega_p = 0.5$ means that the power in the true permutation was greater than only half of other tested permutations, this would be the expected value for random data. Figure 2.13 shows Ω for all genes in the *alpha* data, sorted in descending order for both the standard and diagonally loaded Capon beamformer. The increased selectivity of the Capon method is evident. The crucial region is in the range $\Omega_p \in [0.9, 1]$, which is where cell cyclic genes would be expected to lie. Figure 2.14 shows the scores in this region. This region highlights the increased selectivity of the Capon method. For example, if a score of $\Omega_p = 1$ were demanded to give maximum confidence then the standard beamformer (itself equivalent to the Periodogram approach used in the literature) gives 256 genes whereas the Capon method gives only 56 genes. Similarly for $\Omega_p = 0.95$ the standard approach gives

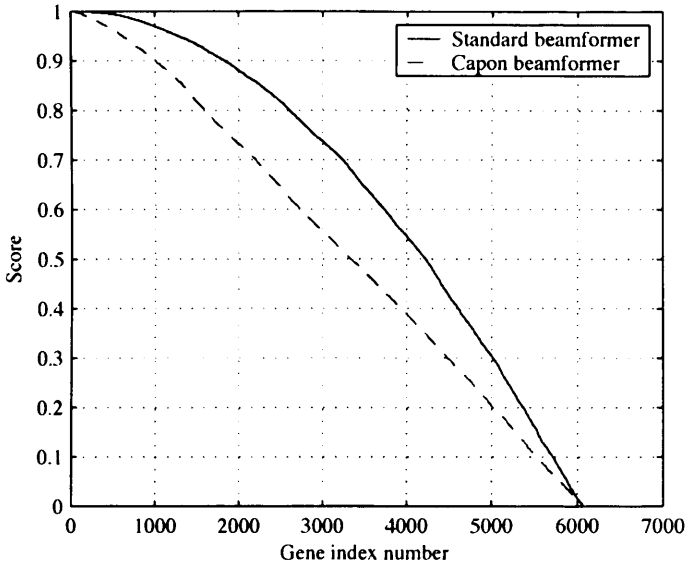
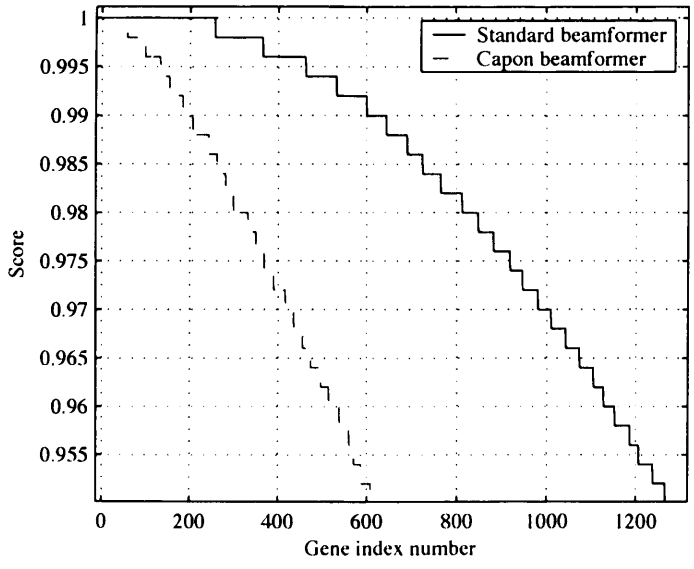


Figure 2.13. Ω for all genes in the *alpha* data, sorted in descending order for both the standard and diagonally loaded Capon beamformer.



htbp

Figure 2.14. $\Omega_p \in [0.9, 1]$ for all genes in the *alpha* data, sorted in descending order for both the standard and diagonally loaded Capon beamformer.

1265 genes and the Capon approach 330 genes. These results corroborate the findings of the analysis in the literature which claim that the number of cell cyclic genes is frequently overestimated [26, 49].

2.6 Conclusions

The estimation of spectral information from sample sizes as short as those typically found in microarray time course experiments is clearly a challenging problem. Using current technology, sample sizes are too small to be able to conclude definitively which genes are actually cell-cycle regulated. With longer time series, use of filterbank spectral estimators with long filter lengths should enable accurate assessment of which gene are cell-cyclic. Nevertheless, worthwhile advances have been obtained in terms of the performance analysis of filterbank estimation methods and the subsequent development in the robustness to temporal mis-sampling. The use of a beamforming approach has been shown to yield a high resolution estimate of the spectral content of microarray data and to be suitable for use with non-uniformly sampled expression data.

Spectral estimation is suitable for identifying cyclic profiles, but many processes underlying gene expression are not necessarily cyclic. In order to detect these, a more general approach is needed. Blind Source Separation (BSS) allows the blind estimation of components, according to a variety of statistical criteria. In the following chapter we examine BSS as a technique for extracting fundamental components from gene expression data.

Chapter 3

INDEPENDENT COMPONENT ANALYSIS FOR MICROARRAY DATA

3.1 Review of Independent Component Analysis for microarray data

Independent Component Analysis (ICA) refers to the task of recovering statistically independent sources from a set of mixtures. ICA has been successfully applied to many problem domains, see [50] for an overview of biomedical applications. ICA has been applied to microarray data in a number of publications [22–24, 51–55]. The ICA approach to microarray data analysis is attractive because of the technique's ability to extract statistically independent sources blindly. Here, 'blindly' refers to the ability to estimate sources and mixing parameters using solely the output and a few key assumptions. This is useful in the study of microarray data because both the input and the mixing process are certainly unknown. The generation of statistically independent sources is also appealing as these could intuitively represent fundamental cellular processes.

The first application of ICA to the microarray analysis problem is in [22,56]. ICA is used to classify genes into biologically relevant groups using no prior knowledge by generating independent components and measuring the proximity to each gene through correlation. It appears to use a dual formulation, shown in section 3.3.3, though this is not made explicit in the explanation. ICA was shown to outperform Principle Components Analysis (PCA) in comparison to handpicked benchmark profiles.

In [24], the FastICA algorithm [57] is used to find 'modes' of gene expression, apparently using a dual formulation. Constrained ICA is used in [58] to generate independent components given some prior knowledge of the genes' relationship to each other. Whilst this may give more accurate results, it rather negates one of the primary strengths of ICA; the lack of need for additional prior knowledge, and reduces it to a supervised method.

Nonlinear ICA is introduced in [51], as a possible improvement over the traditional linear mixing model, and is shown to give some benefits with smaller scale microarray experiments. The transpose model form is used, see section 3.3.2 for the significance of this.

In [52], the transpose model form is used to perform ICA in order to group genes relevant to the development of cancer. This study is slightly different in that the data are not time series data. Further work on the analysis of cancer data using ICA is done in [53].

ICA is compared to Analysis of Variance (ANOVA), Partial Least Squares (PLS) and PCA in [55] and found to be the best technique for grouping genes which belong to the same biological family.

The study in [59] is important as it explicity compares the temporal

model form (section 3.3.1) with the tranpose model form (section 3.3.2). It also introduces a hybrid form, for trading off between independence in the temporal and transpose model forms.

3.2 Introduction to ICA

3.2.1 Statistical principles

The fundamental aim of ICA is to recover a set of statistically independent sources, $\mathbf{s} \in \mathbb{R}^m$, given data $\mathbf{x} \in \mathbb{R}^P$ generated by some function of the sources $\mathbf{x} = f(\mathbf{s})$. Sources are said to be independent if their joint Probability Density Function (pdf) can be factorised into the product of the marginal pdfs. Hence, for m independent sources, the joint pdf can be written

$$q(\mathbf{s}) = \prod_{i=1}^{m} q_i(\mathbf{s}_i) \tag{3.2.1}$$

where $q(\mathbf{s})$ is the joint pdf and $q_i(\mathbf{s}_i)$ is the pdf of the i-th source. A measure of closeness between two pdfs, $f(\mathbf{s})$ and $g(\mathbf{s})$ is given by the Kullback divergence

$$\mathcal{K}(f|g) \triangleq \int_{\mathbf{s}} f(\mathbf{s}) \log\left(\frac{f(\mathbf{s})}{g(\mathbf{s})}\right) d\mathbf{s} \tag{3.2.2}$$

The Kullback divergence is an example of a contrast function, which are used as the objective functions of ICA. They must, in some sense, quantify the independence between sources and reach a minimum (or maximum) when source separation is achieved. It can be shown that the Kullback divergence is the contrast function associated with the

maximum likelihood estimate of the sources [60]

$$\phi^{\mathrm{ML}}[\mathbf{y}] = \mathcal{K}[\mathbf{y}|\mathbf{s}] \tag{3.2.3}$$

where $\phi^{\mathrm{ML}}[\mathbf{y}]$ is the maximum likelihood contrast function of the distribution of the estimated sources $\mathbf{y} \in \mathbb{R}^m$. In practical cases, direct use of the Kullback divergence is precluded by the lack of knowledge of the source density functions. ICA algorithms must explicitly or, more often, implicitly, estimate the source density functions. Practical contrast functions operate on finite sample data to give an estimate, in some sense, of the degree of independence between sources, with no prior knowledge, or limited prior knowledge of the source distributions. The ICA problem is intractable for arbitary functions $\mathbf{x} = f(\mathbf{s})$, it is, however, solvable for specific cases. We examine the linear mixing model.

3.2.2 Linear mixing model formulation

The data are represented by a matrix $\mathbf{X} \in \mathbb{R}^{P \times N}$, with P sensors (genes) and N samples (time points). The rows of $\mathbf{X}$ are henceforth assumed zero mean. The generative model for the data is linear mixing

$$\mathbf{X} = \mathbf{AS} \tag{3.2.4}$$

where $\mathbf{A} \in \mathbb{R}^{P \times m}$ is an unknown mixing matrix, assumed to have full column rank. $\mathbf{S} \in \mathbb{R}^{m \times N}$ is the unknown matrix of sources, whose rows represent m statistically independent sources, no more than one of which is Gaussian distributed. Problems can be defined as underdetermined ($m > P$), complete ($m = P$), or overdetermined ($P > m$). In

general, underdetermined problems require special techniques or extra assumptions for a unique solution. In the case of microarray data, P is so large that problems are generally overdetermined, though they could be regarded as complete in the case of transposed data with very few time points.

Given these assumptions, and a known data matrix $\mathbf{X}$, the decomposition can be solved up to two indeterminacies:

1. Permutation - The ordering of the sources is arbitrary and not guaranteed to be preserved, given that, for any row swap in $\mathbf{S}$, $\mathbf{X}$ can be restored with the equivalent column swap in $\mathbf{A}$.

2. Scaling - The amplitudes of the sources are indeterminate, given that any change in amplitude in the rows in $\mathbf{S}$ is trivially corrected by the inverse change in amplitude in the corresponding columns of $\mathbf{A}$. Note that this amplitude ambiguity includes possible changes of sign. In recognition that the scaling of the sources is entirely arbitary, *the y axes of all source plots in this chapter are unscaled.*

The solution can be written

$$\mathbf{Y} = \mathbf{BX} \tag{3.2.5}$$

where $\mathbf{Y} \in \mathbb{R}^{m \times N}$ is an estimate of $\mathbf{S}$, up to the permutation and scaling ambiguities and $\mathbf{B} \in \mathbb{R}^{m \times P}$ is the $m \times P$ estimated unmixing matrix.

For a perfect estimate, the unmixing matrix is given by

$$\mathbf{B} = \mathbf{A}^{\dagger} \tag{3.2.6}$$

$$= \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T \tag{3.2.7}$$

where $\mathbf{A}^{\dagger}$ denotes the Moore-Penrose pseudo inverse of $\mathbf{A}$ [61]. The ambiguities in the ICA model prevent such an estimate. In practice, the best that can be achieved is

$$\mathbf{B} = \mathbf{P}\mathbf{A}^{\dagger} \tag{3.2.8}$$

where $\mathbf{P}$, in the errorless case, is a non-mixing (or permutation) matrix, i.e. a matrix with exactly one non-zero entry in each row and each column. This suggests that a natural Performance Index (PI) for testing ICA algorithms (assuming that the true mixing matrix is known) is the measurement of the extent to which $\mathbf{P} = \mathbf{BA}$ is a true non-mixing matrix [62]. One such non-negative PI is

$$\mathrm{PI}\left(\mathbf{P}\right) = \sum_{j=1}^{m}\left(\sum_{i=1}^{m}\frac{|p_{ij}|^2}{\max_i |p_{ij}|^2} - 1\right) \tag{3.2.9}$$

The $\mathrm{PI}\left(\mathbf{P}\right) \rightarrow 0$ as $\mathbf{P}$ approaches a pure non-mixing matrix.

3.2.3 The JADE algorithm

The JADE (Joint Approximate Diagonalisation of Eigenmatrices) algorithm is an algebraic approach to ICA. The algorithm was first presented in [63], but several other publications are useful for a fuller understanding of the algorithm [60, 64–66]. In this version of the algorithm, the data are assumed real - as is always the case with microarray

data. The algorithm then adopts a two stage approach to first whiten the zero mean data and then calculate a rotation matrix to minimise certain higher order correlations in the data. The estimated unmixing matrix is hence decomposed as

$$\mathbf{B} = \mathbf{UW} \tag{3.2.10}$$

where $\mathbf{W}$ is the spatial whitening matrix, and $\mathbf{U}$ a rotation matrix. Independence implies uncorrelatedness[1] so all ICA algorithms must whiten the data. In JADE, pre-whitening is an explicit step. The whitened data are given by

$$\mathbf{Z} = \mathbf{WX} \tag{3.2.11}$$

The condition for white data is

$$\mathbf{R}^{\mathbf{Z}} = \mathbf{W}\mathbf{R}^{\mathbf{X}}\mathbf{W}^T \tag{3.2.12}$$

$$= \mathbf{W}E\left[\mathbf{X}\mathbf{X}^T\right]\mathbf{W}^T \tag{3.2.13}$$

$$= \mathbf{I} \tag{3.2.14}$$

The whitening matrix $\mathbf{W}$ can be obtained using the eigenvalue decomposition of the covariance matrix of $\mathbf{X}$

$$\mathbf{R}^{\mathbf{X}} = E\left[\mathbf{X}\mathbf{X}^T\right] = \mathbf{E}\mathbf{D}\mathbf{E}^T \tag{3.2.15}$$

where $\mathbf{E}$ is an orthogonal matrix of eigenvectors and $\mathbf{D}$ is a matrix with the eigenvalues on the leading diagonal and zeros elsewhere, i.e.

[1] Though the converse is not true, except in the case of Gaussian random variables.

diag($\lambda_1 \ldots \lambda_N$). The whitener is then given by

$$\mathbf{W} = \mathbf{D}^{-\frac{1}{2}}\mathbf{E} \tag{3.2.16}$$

If the number of sources is less than the number of sensors, then the $(P - m)$ least significant eigenvalues and corresponding eigenvectors are discarded to yield an $m \times P$ whitening matrix. The dimensionality of the whitened data $\mathbf{Z}$ is thus reduced to $m \times N$, the dimensionality of the required matrix of sources. The problem is then reduced to that of finding $\mathbf{U}$. The matrix $\mathbf{U}$ is a rotation matrix as it relates two spatially white matrices $\mathbf{S}$ and $\mathbf{WX}$ through the relation $\mathbf{S} = \mathbf{UWX}$ [63]. Following pre-whitening, the number of free parameters is reduced from Pm to $m(m-1)/2$.

The rotation matrix $\mathbf{U}$ can be obtained using the joint diagonalisation of fourth order cumulant matrices. The fourth order cumulants for zero mean real random variables x_i, x_j, x_k, x_l are

$$\begin{aligned} Cum\,(x_i, x_j, x_k, x_l) = {} & E\,[x_i x_j x_k x_l] \\ & - E\,[x_i x_j]\,E\,[x_k x_l] \\ & - E\,[x_i x_k]\,E\,[x_j x_l] \\ & - E\,[x_i x_l]\,E\,[x_j x_k] \end{aligned} \tag{3.2.17}$$

Cumulants involving two or more random variables are known as cross cumulants whilst cumulants of one variable are known as auto cumulants. The fourth order autocumulant of a real, zero mean random

variable x is known as the kurtosis and is defined as

$$\begin{aligned} k(x) &= Cum(x,x,x,x) \\ &= E\left[x^4\right] - 3E\left[x^2\right]^2 \end{aligned} \tag{3.2.18}$$

Kurtosis is used as a measure of Gaussianity. Gaussian distributed variables have a kurtosis of zero and are known as mesokurtic. Super-Gaussian, or leptokurtic, distributions are characterised by sharp peaks with quickly decaying tails and have positive kurtosis. Sub-Gaussian, or platykurtic, distributions, are flatter with heavy tails and have negative kurtosis.

The true nature of cumulants is tensorial, however fourth order cumulants of a $P \times N$ matrix of data $\mathbf{X}$ can be defined in a matrix notation that is more amenable to algebraic manipulation. For any $\mathbf{M}$, the ij-th entry of the $P \times P$ fourth order cumulant matrix $\mathbf{Q}^{\mathbf{X}}(\mathbf{M})$ can be defined by [66]

$$\left[\mathbf{Q}(\mathbf{M})\right]_{ij} \equiv \sum_{k,l=1}^{m} \mathrm{Cum}\left(\mathbf{X}_i, \mathbf{X}_j, \mathbf{X}_k, \mathbf{X}_l\right) \mathbf{M}_{kl} \tag{3.2.19}$$

Hence, for zero mean data, the matrix $\mathbf{Q}^{\mathbf{X}}(\mathbf{M})$ can be calculated as

$$\mathbf{Q}^{\mathbf{X}}(\mathbf{M}) = E\left[\left(\mathbf{X}^T\mathbf{M}\mathbf{X}\right)\mathbf{X}\mathbf{X}^T\right] - \mathbf{R}^{\mathbf{X}}\mathrm{tr}\left(\mathbf{M}\mathbf{R}^{\mathbf{X}}\right) - \mathbf{R}^{\mathbf{X}}\mathbf{M}\mathbf{R}^{\mathbf{X}} - \mathbf{R}^{\mathbf{X}}\mathbf{M}^T\mathbf{R}^{\mathbf{X}} \tag{3.2.20}$$

where $\mathrm{tr}(\cdot)$ denotes the trace of a matrix. Hence, the cumulants of our matrix of spatially white data $\mathbf{Z} \in \mathbb{R}^{m \times N}$, for any $\mathbf{M} \in \mathbb{R}^{m \times m}$, are given by the $m \times m$ matrix

$$\mathbf{Q}^{\mathbf{Z}}(\mathbf{M}) = E\left[\left(\mathbf{Z}^T\mathbf{M}\mathbf{Z}\right)\mathbf{Z}\mathbf{Z}^T\right] - tr(\mathbf{M}) - \mathbf{M} - \mathbf{M}^T \tag{3.2.21}$$

$\mathbf{Q}^{\mathbf{Z}}(\mathbf{M})$ therefore represents a slice through the fourth order cumulant tensor of $\mathbf{Z}$, the co-ordinates of which are defined by $\mathbf{M}$. There exists a set $\mathcal{M} = \{ \mathbf{M}_1 \ \ldots \ \mathbf{M}_J \}$, of J $m \times m$ matrices for which $\mathbf{Q}^{\mathbf{Z}}(\mathbf{M}) \forall \mathbf{M} \in \mathcal{M}$ encapsulates all of the fourth order information in $\mathbf{Z}$. This maximal set is obtained whenever $\mathcal{M}$ constitutes a basis for the linear space of $m \times m$ matrices [66]. An intuitive basis for the space $\mathcal{M}$ is given by $\mathbf{e}_p \mathbf{e}_q^T$ where $\mathbf{e}_p$ is an m dimensional column vector with 1 in the p-th entry and zero elsewhere. This ensures that only one element of any $\mathbf{M}$ in $\mathcal{M}$ is non zero and so the entries of $\mathbf{Q}^{\mathbf{Z}}(\mathbf{M})$ are simply the cumulants of $\mathbf{Z}$[2]. Figure 3.1 shows the structure of the resultant cumulant set using this basis.

The JADE criterion for minimising the fourth order correlation is given by

$$\phi^{\text{JADE}}(\mathbf{Z}) = \sum_{ijkl \neq iikl} \left(\mathbf{Q}^{\mathbf{Z}}_{ijkl}\right)^2 \tag{3.2.22}$$

This is equivalent to minimising the off-diagonal entries in $\mathbf{Q}^{\mathbf{Z}}(\mathbf{M})$. Note that this criterion does not explicitly cover all cross cumulants in $\mathbf{Q}^{\mathbf{Z}}(\mathbf{M})$, only those where $i \neq j$. Figure 3.1 shows which cumulants are explicitly minimised in the criterion. Note, however, that the cumulant set is non redundant in the sense that individual cumulant values appear in multiple locations in the set and so all cross cumulants are minimised. The contrast function can hence be written

$$\phi^{\text{JADE}}(\mathbf{Z}) = \sum_{\mathbf{M}_i \in \mathcal{M}} \text{Off}\left(\mathbf{U}^T \mathbf{Q}^{\mathbf{Z}}(\mathbf{M}_i)\, \mathbf{U}\right) \tag{3.2.23}$$

where Off $(\cdot)$ is defined as the sum squared of off-diagonal elements in

[2]The original version of the JADE algorithm used eigenmatrices to achieve a more compact basis [63]. However, this basis is only accurate when the model can be guaranteed to hold exactly.

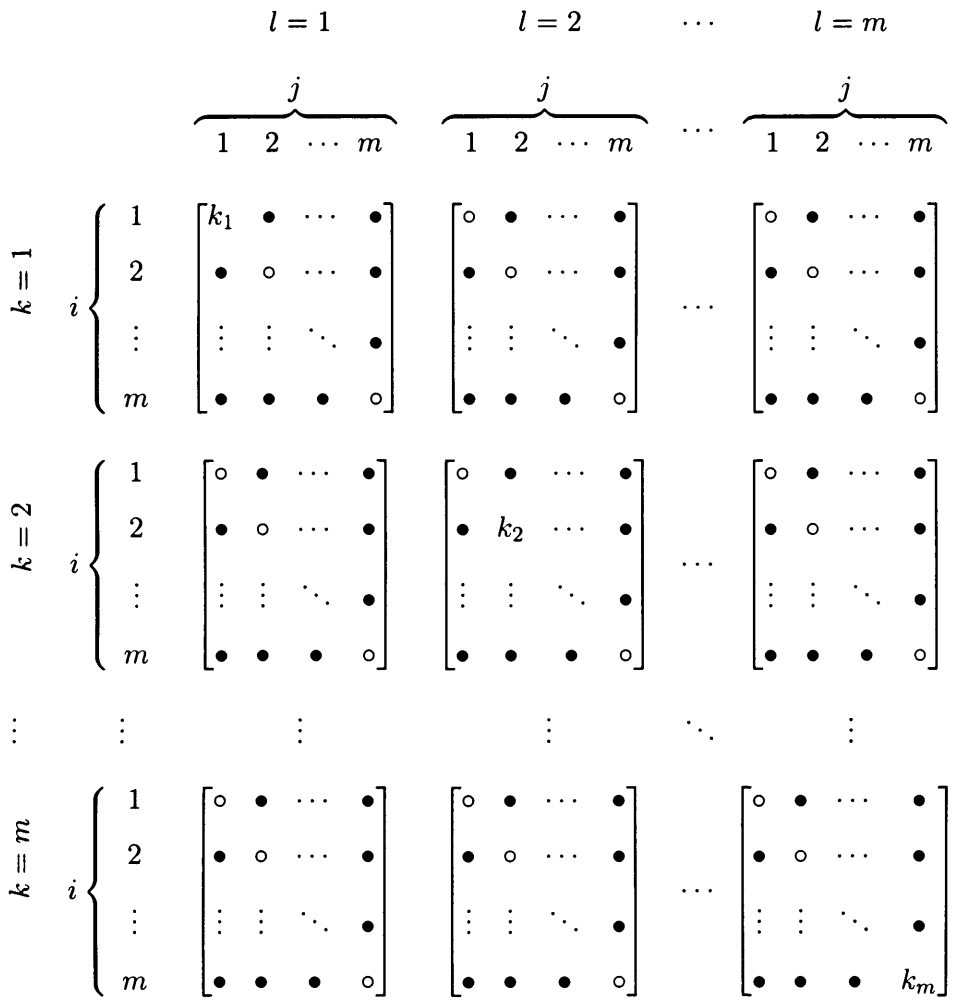


Figure 3.1. Diagram showing the structure of $\mathbf{Q}^{\mathbf{z}}(\mathbf{M})$. k_i are the autocumulants, i.e. the kurtosis values of the i-th source. • represents the cross-cumulants for which $i \neq j$ and hence are explicitly minimised in the JADE contrast function. ○ represents crosscumulants for which $i = j$ and hence are not explicitly minimised in the JADE criterion, but are represented elsewhere in the cumulant set.

a matrix, i.e.

$$\text{Off}\,(\mathbf{H}) \equiv \sum_{i \neq j} (h_{ij})^2 \tag{3.2.24}$$

The diagonalisation of a matrix can be achieved using a Jacobi algorithm, see e.g. [61], which uses successive rotations to minimise the off-diagonal elements. This technique can be extended to the joint diagonalisation of a matrix set [65]. The technique is numerically very robust and converges quickly. Note, however, that contrast function (3.2.23) is minimised but will be driven to zero only in the theoretical case of infinite sample statistically stationary data, and hence exact cumulant values.

3.3 Independent component analysis of microarray data

3.3.1 Generating independent time series

Given the $P \times N$ data matrix $\mathbf{X}$, with P genes and N samples, the standard ICA decomposition can be applied to find m independent sources.

$$\begin{array}{ccccc} \mathbf{X} & = & \mathbf{A} & & \mathbf{S} \\ (P \times N) & & (P \times m) & & (m \times N) \end{array} \tag{3.3.1}$$

The matrix of sources $\mathbf{S}$ represents m sources, each comprising N samples. The sources are then hoped to represent either fundamental processes underlying the genes' expression, or other independent time profiles common to multiple genes. The mixing matrix $\mathbf{A}$ gives a measure of the extent to which each source is present in each gene. The decomposition can also be viewed as a clustering algorithm with a sample independence distance measure, subject to a linear mixing model. The rows of $\mathbf{S}$ would then represent the cluster centroids, and $\mathbf{A}$ represents

the level of membership each gene has to a given cluster. The number of sources is not known *a priori* and is usually obtained heuristically on the basis of the biological plausability of the estimated sources.

The PI measure in equation (3.2.9) only applies when the true mixing matrix, or equivalently, the true sources are known. This is clearly not the case for real microarray data and so the usual performance index cannot be used. The independence of the resulting sources can, however, be estimated. One measure is an estimate of the Mutual Information (MI) using second and fourth order cumulants [66].

$$\phi^{\mathrm{MI}}(\mathbf{Y}) \approx \frac{1}{4}\sum_{ij\neq ii}\left(\mathbf{R}^{\mathbf{Y}}_{ij}\right)^2 + \frac{1}{48}\sum_{ijkl\neq iiii}\left(\mathbf{Q}^{\mathbf{Y}}_{ijkl}\right)^2 \qquad (3.3.2)$$

The first term is the sum of squared off diagonal terms in the covariance matrix, the second is the sum of squared cross cumulants in the fourth order cumulant set. The weighting between the terms stems from the origin of the approximation in an Edgeworth expansion of the pdf [67]. In ICA algorithms that use explicit prewhitening, such as JADE, the first term is necessarily zero as the off-diagonal terms of the covariance matrix are driven to zero by the prewhitening step. In general, the quantification of independence from finite data is notoriously challenging. Particular issues with $\phi^{\mathrm{MI}}(\mathbf{Y})$ are

1. The Edgeworth expansion from which the approximation is derived is only valid for near-normal distributions [67]. As the sources estimated by ICA are as non-normal as possible, the weighting between terms in the expression for the evaluation of estimated sources is innaccurate as a true measure of independence. It is still accurate in the sense that it is driven to zero for

independent sources.

2. The second term is a fourth order quantity squared, rendering it highly sensitive to outliers. In fact, given a leptokurtic source with frequent high amplitude outliers, the term is almost completely dominated by outliers and becomes disproportionately large. Even with no obvious outliers the measure is still far more sensitive to values in the tails of a distribution than those around the mean.

In addition to these caveats, the measure is not invariant to m. As it is a pure addition of cumulants then it rises as m grows. This means it cannot be used to compare results from different numbers of sources. In order to make a measure which is invariant to m, we propose using a pairwise measure of Mutual Information $\phi^{\mathrm{PMI}}(\mathbf{Y})$. This measure effectively calculates the mean value of the Mutual Information from all pairwise combinations of sources.

$$\phi^{\mathrm{PMI}}(\mathbf{Y}) = \frac{1}{m^2} \sum_{i \neq j} \phi^{\mathrm{MI}}\left(\begin{bmatrix} \mathbf{y}_i \\ \mathbf{y}_j \end{bmatrix} \right) \tag{3.3.3}$$

The data from the *alpha* dataset includes some high magnitude outliers. The reliance of the JADE contrast function and the ϕ^{PMI} measure on fourth order quantities squared means that they are particularly susceptible to these outliers. In order for these not to dominate the results, absolute values of the *alpha* data were limited to 4. The total proportion of data affected is less than 0.04%. Figure 3.2 shows the estimated sources which result from applying JADE to the *alpha* data from [1]. Three sources were specified. Figure 3.3 shows the equivalent sources from a principle components analysis for comparison. Note

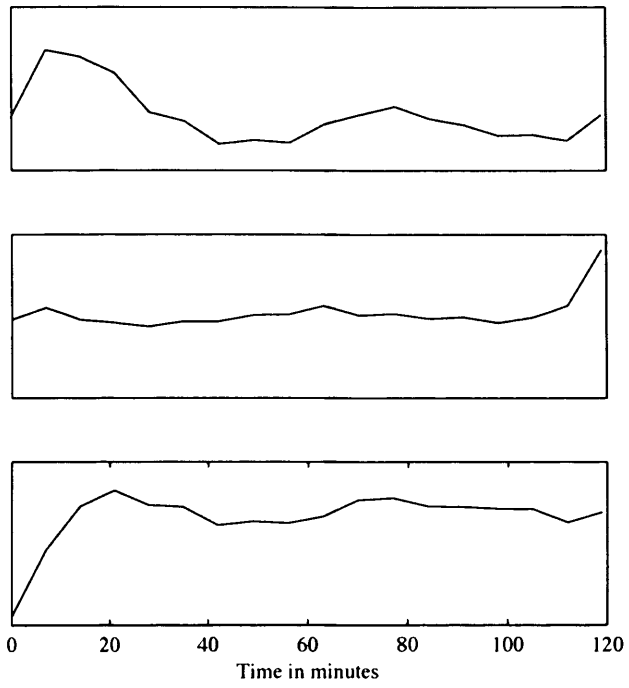


Figure 3.2. Estimates of sources for $m = 3$ using the JADE algorithm on the *alpha* data. $\phi^{\mathrm{PMI}}(\mathbf{Y}) = 0.061$.

that the principle components can be obtained simply by applying the whitening matrix obtained in the first stage of the JADE algorithm, i.e. $\mathbf{Y} = \mathbf{WX}$. For both the ICA and PCA approaches, the second order component of $\phi^{\mathrm{PMI}}(\mathbf{Y})$ is zero. This is, of course, expected as the initial PCA step in JADE is designed to drive the covariance matrix diagonal. The fact that the second order correlation remains zero in the estimated JADE sources serves to demonstrate that the matrix derived in the second step of the JADE algorithm $\mathbf{U}$ is indeed orthogonal and so preserves spatial whiteness.

The value of $\phi^{\mathrm{PMI}}(\mathbf{Y})$ is lower for the sources estimated by JADE than by PCA. This is because the extra step in JADE reduces the fourth order order cross-correlation and so decreases the second term of $\phi^{\mathrm{PMI}}(\mathbf{Y})$, making the sources more independent. The fact that the drop in $\phi^{\mathrm{PMI}}(\mathbf{Y})$ between the PCA and JADE algorithms is not large hints

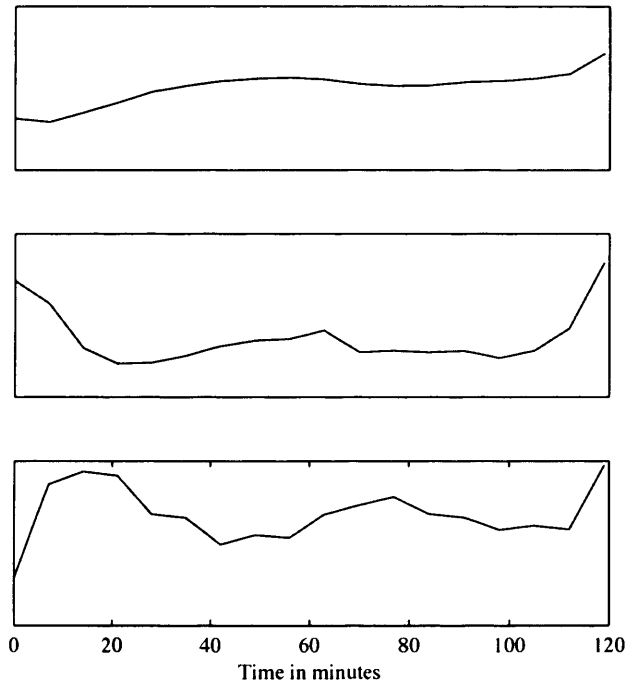


Figure 3.3. Estimates of sources for $m = 3$ from PCA of the *alpha* data. $\phi^{\mathrm{PMI}}(\mathbf{Y}) = 0.093$.

m	PCA	JADE	FastICA
3	0.093	0.061	0.060
4	0.114	0.099	0.103
5	0.138	0.093	0.218
6	0.103	0.090	N/A*

Table 3.1. Table of $\phi^{\text{PMI}}(\mathbf{Y})$ for the *alpha* data over a range of m for PCA, JADE and FastICA algorithms. *The FastICA algorithm failed to converge in the case where $m = 6$, minor convergence failures were also reported for other values of m.

at a key point in the ICA of microarray data: That the estimation of higher order statistics from limited data is challenging. The *alpha* data contains only $N = 18$ timepoints, two orders of magnitude lower than the number of samples typically used to demonstrate the performance of ICA algorithms. Despite this, the decrease in the value of $\phi^{\text{PMI}}(\mathbf{Y})$ following the second JADE step indicates that *some* benefit is being derived from the higher order statistics.

Table 3.1 shows the values of $\phi^{\text{PMI}}(\mathbf{Y})$ for the *alpha* data as the number of sources m varies. Values for another ICA algorithm, FastICA [57] are also given to show that the results are generally applicable. The values of $\phi^{\text{PMI}}(\mathbf{Y})$ in Table 3.1 are generally lower for the JADE algorithm than the PCA algorithm, reflecting the former's aim of generating independent, rather than merely uncorrelated, sources. The FastICA values are similar to JADE, in the cases where the algorithm successfully converged. The reduction in $\phi^{\text{PMI}}(\mathbf{Y})$ between the PCA and JADE algorithms is greatest at $m = 5$, here the higher order statistics seem to have the most effect and so $m = 5$ could be an interesting set of sources to examine.

The estimated components for $m = 5$ are shown for the JADE and PCA approaches in Figures 3.4 and 3.5 respectively. The fourth

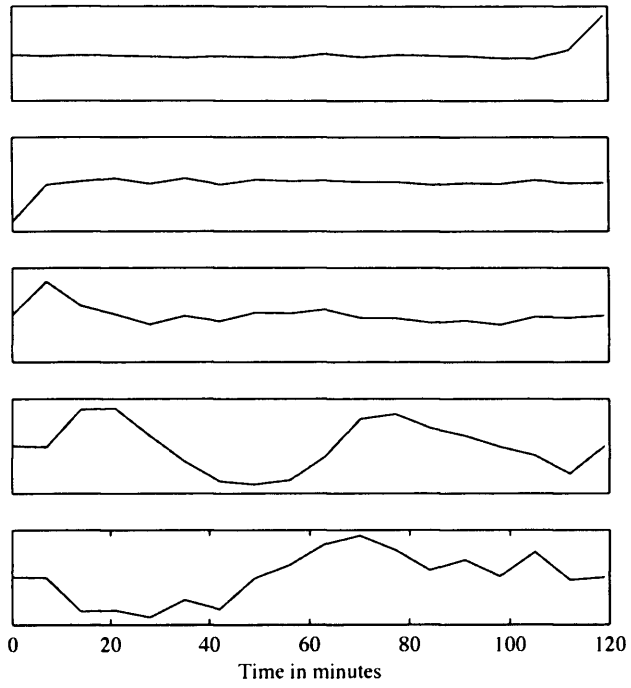


Figure 3.4. Estimates of $m = 5$ sources from JADE on the *alpha* data. Note the distinct cyclic profile of the fourth source.

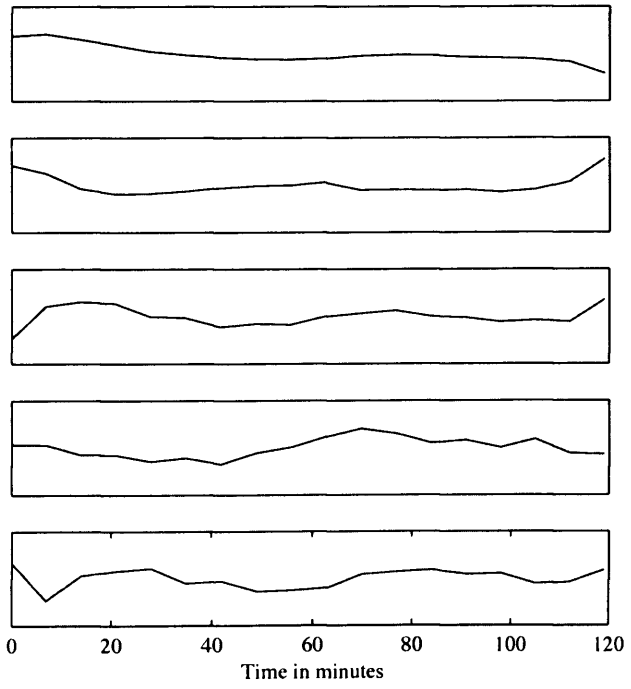


Figure 3.5. Estimates of $m = 5$ sources from PCA on the *alpha* data.

m	PCA	JADE	FastICA
3	0.014	0.013	0.013
4	0.015	0.008	0.009
5	0.085	0.026	0.026
6	0.164	0.033	N/A*

Table 3.2. Table of $\phi^{\text{PMI}}(\mathbf{Y})$ for the *Plasmodium* data from [2] over a range of m for PCA, JADE and FastICA algorithms. The values are significantly lower than those in Table 3.1, reflecting the increased data length. In addition, the drop in $\phi^{\text{PMI}}(\mathbf{Y})$ between the PCA and ICA algorithms is more significant. *The FastICA algorithm failed to converge in the case where $m = 6$.

source from JADE shows a distinctive cell cyclic component. The same distinct cyclic source does not feature in the PCA sources and so the higher order statistics stage of the JADE algorithm does appear to help here to reveal significant components in the *alpha* data. It appears that some benefit is being derived from the higher order statistics, though the small drop in the $\phi^{\text{PMI}}(\mathbf{Y})$ values between the PCA and ICA algorithms in Table 3.1 is indicative of the limitations of higher order statistics in the face of small sample sizes.

In order to assess the effect of a slightly increased sample size on the values of $\phi^{\text{PMI}}(\mathbf{Y})$, we examine data from [2], which profiles the gene transcription of the malaria causing parasite *Plasmodium falciparum.* This study used $N = 48$ time points, rather than $N = 18$ for the *alpha* data in [1]. Table 3.2 shows the values of $\phi^{\text{PMI}}(\mathbf{Y})$ for a range of m. The values of $\phi^{\text{PMI}}(\mathbf{Y})$ are significantly lower than in Table 3.1 because of the increase in sample size, from 18 to 48. In addition, the drop in the values of $\phi^{\text{PMI}}(\mathbf{Y})$ between the PCA and ICA algorithms is more significant, reflecting the utility of higher order statistics with increasing sample size. The study in [2] is close to the maximum number of time points used in microarray experiments but further gains are to

be expected when future technology, and cost reductions, enable more time measurements. To illustrate this, we give an example of how the estimation of $\phi^{\text{PMI}}(\mathbf{Y})$ varies with sample size using synthetic data.

In order to demonstrate how the estimation of sample statistics varies with sample size, Figure 3.6 shows the estimation of $\phi^{\text{PMI}}(\mathbf{Y})$ for synthetic sources, one uniform, one Laplacian and one Gaussian, against a range of sample sizes. The mean value of $\phi^{\text{PMI}}(\mathbf{Y})$ drops to zero only as the sample size approaches 10^4. The mean error and variance is shown separately for the second and fourth order terms of $\phi^{\text{PMI}}(\mathbf{Y})$. The fourth order term decreases far more slowly with sample size and the variance is especially high at small sample sizes. Comparing the sample sizes typical in microarray data with the estimation error in synthetic sources at small sample sizes, we can see why the estimation of sources from microarray data is such a challenging scenario.

ICA is a useful technique in the analysis of microarray data, and shows demonstrable improvements over PCA, in terms of mutual information measurements and biological plausibility, but the estimation of higher order statistics clearly suffers from the lack of samples in the time dimension. To try and overcome this difficulty, we now examine operating on the transpose of the data.

3.3.2 Operating on the transpose of the data

Much of the literature on ICA of microarray data operates on the transpose of the data [51–54,59]. The strength of this approach is clear; with the data matrix transposed, each of the P genes is viewed as a sample. P is typically in the thousands, so estimates of higher order statistics should be significantly more accurate then in the previous section. The

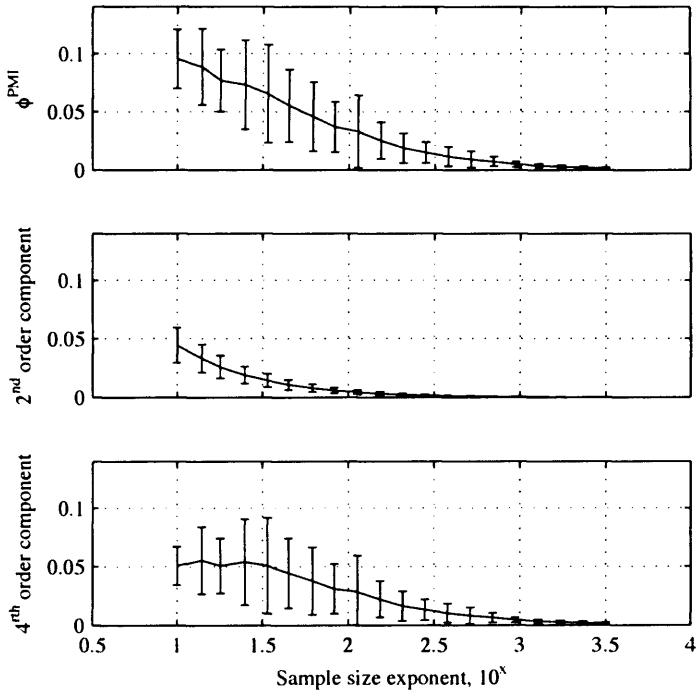


Figure 3.6. Plot shows the mean value of $\phi^{\mathrm{PMI}}(\mathbf{Y})$ for three sources; uniform, Laplacian and Gaussian, for a range of sample sizes from 10 to 10000. Also shown are the individual contributions from the second and fourth order terms of $\phi^{\mathrm{PMI}}(\mathbf{Y})$. Means and standard deviations were obtained over 1000 Monte Carlo runs. The mean value of $\phi^{\mathrm{PMI}}(\mathbf{Y})$ is seen to drop rather slowly as sample size increases. The error bars denote one standard deviation away from the mean. The standard deviation too, drops as the sample size is increased. Clearly, the fourth order component is contributing most significantly to both the mean and variance at the lower sample sizes.

linear mixing model is hence

$$\underset{(N \times P)}{\tilde{\mathbf{X}}} = \underset{(N \times m)}{\tilde{\mathbf{A}}} \underset{(m \times P)}{\tilde{\mathbf{S}}} \tag{3.3.4}$$

where $\tilde{\cdot}$ has been used to distinguish $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{S}}$ from the mixing matrix and sources in the previous formulation. Note that $\tilde{\mathbf{X}} = \mathbf{X}^T$. Each of the m sources is now P samples long, which should result in a more accurate estimation of the higher order statistics and hence a more accurate decomposition, however this formulation is rather less intuititive as a generative model, though it can still be viewed as a model based clustering routine.

The demixing matrix $\tilde{\mathbf{B}} \in \mathbb{R}^{m \times N}$ can now be estimated as

$$\tilde{\mathbf{B}} = \text{JADE}\left(\tilde{\mathbf{X}}, m\right) \tag{3.3.5}$$

yielding the estimated sources $\tilde{\mathbf{Y}} \in \mathbb{R}^{m \times P}$ as

$$\tilde{\mathbf{Y}} = \tilde{\mathbf{B}}\tilde{\mathbf{X}} \tag{3.3.6}$$

Table 3.3 shows the values of $\phi^{\text{PMI}}\left(\tilde{\mathbf{Y}}\right)$, where $\tilde{\mathbf{Y}}$ is an estimate of $\tilde{\mathbf{S}}$ from the transpose model form. The values are not directly comparable to those of the standard model form in Table 3.1 because the distributions are rather more leptokurtic, yielding universally higher values of $\phi^{\text{PMI}}\left(\tilde{\mathbf{Y}}\right)$. However, the drop in $\phi^{\text{PMI}}\left(\tilde{\mathbf{Y}}\right)$ between the PCA and ICA algorithms, in Table 3.3, shows that the vastly increased data length allows the higher order statistics based ICA algorithms to substantially outperform PCA.

m	PCA	JADE	FastICA
3	1.758	0.738	0.765
4	2.499	0.538	0.680
5	2.427	0.486	0.639
6	1.697	0.416	0.469

Table 3.3. Table of $\phi^{\text{PMI}}\left(\tilde{\mathbf{Y}}\right)$ for the *alpha* data over a range of m for PCA, JADE and FastICA algorithms, using the transpose model form. The significant drop in $\phi^{\text{PMI}}\left(\tilde{\mathbf{Y}}\right)$ between the PCA and ICA methods reflects the more accurate estimation of higher order statistics from the increased data lengths afforded by the transposed data.

m	PCA	JADE	FastICA
3	0.085	0.061	0.059
4	0.091	0.105	0.130
5	0.110	0.200	0.220
6	0.082	0.180	0.161

Table 3.4. Table of $\phi^{\text{PMI}}(\mathbf{Y})$ for the *alpha* data over a range of m for PCA, JADE and FastICA algorithms, using the dual model form $\mathbf{Y} = \tilde{\mathbf{B}}^{\dagger}$.

3.3.3 Duality in the transpose form

The transpose form provides independent sources $\tilde{\mathbf{S}}$ of length P. Intuitively, we would like sources that are length N and so can be considered independent components underlying the set of length N gene time series profiles. Such profiles do exist in the transpose model form; in the columns of $\tilde{\mathbf{A}}$, but in general there is no guarantee that these will be independent. Models where both the rows of $\tilde{\mathbf{S}}$ and the columns of $\tilde{\mathbf{A}}$ are independent are said to be dual and a sensible strategy to recover the sources, given $P \gg N$, would be to estimate the separation matrix $\tilde{\mathbf{B}}$ using JADE from the transpose data matrix $\tilde{\mathbf{X}}$ and then to estimate the length N sources from the columns of $\tilde{\mathbf{A}} = \tilde{\mathbf{B}}^{\dagger}$, i.e. $\mathbf{Y} = \left(\mathbf{B}^{\dagger}\right)^{T}$. This is effectively the technique used in [22,24,56], though it is not explicitly stated that the returned components will be independent only when the dual assumption holds true.

Figure 3.7 shows $m = 5$ sources using this dual form from the *alpha* data. The sources are similar to the temporal model form in Figure 3.4, lending some support to the dual assumption.

Table 3.4 shows the values of $\phi^{\text{PMI}}(\mathbf{Y})$ for a range of m for the PCA, JADE and FastICA algorithms. The values for the PCA algorithm are somewhat lower than those from the ICA algorithms. In fact,

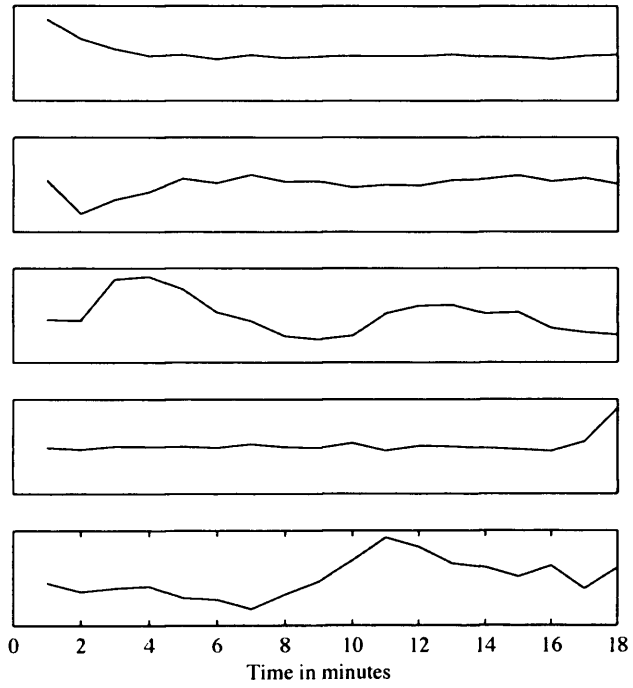


Figure 3.7. Estimates of $m = 5$ sources $\mathbf{Y} = \tilde{\mathbf{B}}^{\dagger}$ from the *alpha* data. The source profiles are similar to the temporal model form, in Figure 3.4, and so the dual assumption may have some merit.

the sources estimated as $\mathbf{Y} = \tilde{\mathbf{B}}^{\dagger}$ by the ICA algorithms are spatially correlated, leading to a non zero first term of $\phi^{\text{PMI}}(\mathbf{Y})$. The rotation matrix applied in the second stage of the ICA algorithms destroys the second order decorrelation. Given that the algorithm is able to use P samples, and the resulting values of $\phi^{\text{PMI}}(\mathbf{Y})$ are not significantly lower than those using the traditional model form with N samples, as in Table 3.1, we conclude that the dual formulation is not necessarily totally accurate in terms of microarray data. However, the sources in Figure 3.7 indicate that the method can be used to return sources that are close to those from the temporal ICA model.

3.3.4 Second order methods

We now consider a second order method to return sources that are not necessarily independent, but spatio-temporally uncorrelated. The justification for this is twofold:

1. The estimation of second order statistics from limited data is a more realistic prospect than the estimation of higher order statistics. As shown by synthetic data in Figure 3.6, fourth order statistics require significantly more data to estimate acurately than second order statistics. The small difference between $\phi^{\text{PMI}}(\mathbf{Y})$ values for the PCA and ICA in Table 3.1 attests to the difficulty in estimating higher order statistics from real, limited sample size microarray data. Solutions involving a dual formulation seem not to give good results, probably reflecting a lack of duality in real microarray data.

2. The justification for seeking to extract independent components from microarray data is unclear. Undoubtedly, there should be

certain fundamental processes underlying the gene expression which are generated by independent processes. However, these are likely to be dynamic time processes and not simply values drawn from independent probability density functions. In the context of underlying time processes, the ability to extract components which are spatio-temporally uncorrelated may actually be more attractive than those which are independent.

The SOBI (Second Order Blind Identification) algorithm is able to estimate sources which are spatio-temporally uncorrelated [69]. The algorithm is similar to the JADE algorithm in principle, but rather than diagonalising fourth order cumulant matrices, it diagonalises time lagged covariance matrices. The diagonalisation of time lagged covariance matrices enforces spatio-temporal decorrelation at the given lags. For a set of K time lags $\{\tau_1 \quad \tau_2 \quad \dots \quad \tau_K\}$, the set of matrices to be diagonalised is therefore

$$\mathcal{R} = \left\{ \mathbf{R}^{\mathbf{Z}}(\tau_1) \quad \mathbf{R}^{\mathbf{Z}}(\tau_2) \quad \dots \quad \mathbf{R}^{\mathbf{Z}}(\tau_K) \right\} \tag{3.3.7}$$

where $\mathbf{R}^{\mathbf{Z}}(\tau)$ is defined as

$$\mathbf{R}^{\mathbf{Z}}(\tau) = \frac{1}{N-\tau} \sum_{t=1}^{N-\tau} \mathbf{z}(t)\, \mathbf{z}^T(t+\tau) \tag{3.3.8}$$

Note that $\mathbf{R}^{\mathbf{Z}}(0)$ is effectively already diagonalised in the prewhitening stage of the algorithm. This is equivalent to jointly diagonalising the set $\{\mathbf{R}^{\mathbf{Z}}(0) \quad \dots \quad \mathbf{R}^{\mathbf{Z}}(K)\}$, whilst assigning an infinite weight to the diagonalisation of the $\mathbf{R}^{\mathbf{Z}}(0)$ covariance matrix.

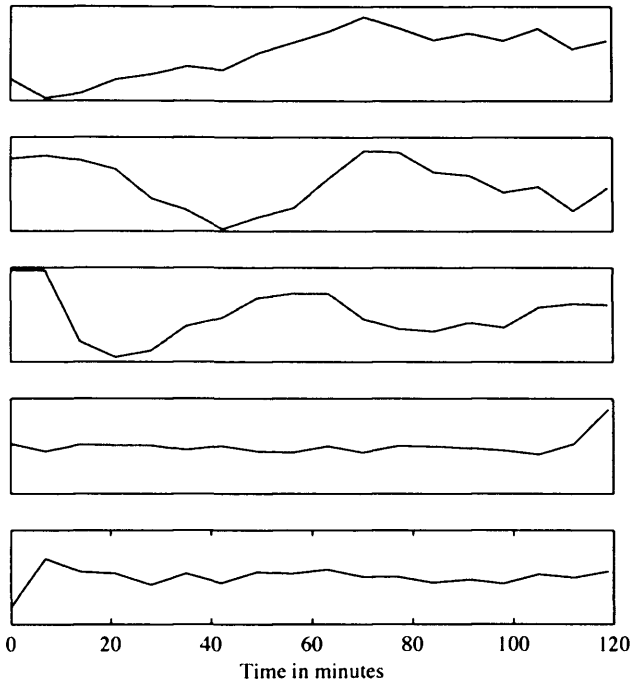


Figure 3.8. Estimates of $m = 5$ sources from SOBI on the *alpha* data, with $K = 7$ lags. A distinctive cell cyclic source is shown in the second source.

Figure 3.8 shows the estimated sources recovered by the SOBI algorithm for $m = 5$ sources. The sources are somewhat similar to those returned by JADE in Figure 3.4, a cell cyclic source is present in both.

3.3.5 Model error

The linear mixing model $\mathbf{X} = \mathbf{AS}$ is unlikely to be a true generative model for the microarray data as the gene regulatory process is a network of complex biological functions which is likely to involve nonlinearity in some form. The estimated mixing matrix is given by

$$\hat{\mathbf{A}} = \mathbf{B}^{\dagger} \tag{3.3.9}$$

The matrix of residuals, Υ , is given by the difference between the actual data matrix and the data matrix given by the estimated model,

m	MSE $(\mathbf{X} - \mathbf{B}^{\dagger}\mathbf{B}\mathbf{X})$	MSE $\left(\tilde{\mathbf{X}} - \tilde{\mathbf{B}}^{\dagger}\tilde{\mathbf{B}}\tilde{\mathbf{X}}\right)$
3	0.051	0.049
4	0.041	0.039
5	0.033	0.031
6	0.027	0.026

Table 3.5. Table of mean square error values for the transpose and non-transpose model form.

i.e.

$$\Upsilon = \mathbf{X} - \hat{\mathbf{A}}\mathbf{Y} = \begin{cases} \mathbf{X} - \mathbf{B}^{\dagger}\mathbf{B}\mathbf{X} & \text{for ICA,} \\ \mathbf{X} - \mathbf{W}^{\dagger}\mathbf{W}\mathbf{X} & \text{for PCA.} \end{cases} \tag{3.3.10}$$

In fact, the estimated data, and hence residuals, for the PCA and prewhitened ICA approaches are identical, because $\mathbf{B}^{\dagger}\mathbf{B} = \mathbf{W}^{\dagger}\mathbf{W}$, this is readily verified by the following equivalences

$$\begin{aligned} \mathbf{B}^{\dagger}\mathbf{B} &= (\mathbf{U}\mathbf{W})^{\dagger}(\mathbf{U}\mathbf{W}) && (3.3.11) \\ &= \mathbf{W}^{T}\left(\mathbf{U}^{T}\mathbf{U}\mathbf{W}\mathbf{W}^{T}\right)^{-1}\mathbf{U}^{T}\mathbf{U}\mathbf{W}, \quad \text{as } \mathbf{U}\mathbf{W} \text{ is rank } m && (3.3.12) \\ &= \mathbf{W}^{T}\left(\mathbf{W}\mathbf{W}^{T}\right)^{-1}\mathbf{W}, \quad \text{as } \mathbf{U}^{T}\mathbf{U} = \mathbf{I} && (3.3.13) \\ &= \mathbf{W}^{\dagger}\mathbf{W} && (3.3.14) \end{aligned}$$

The Mean Square Error (MSE) in the estimated model is given by:

$$\text{MSE}(\Upsilon) = \frac{1}{PN}\sum_{ij}\Upsilon_{ij}^{2} \tag{3.3.15}$$

Table 3.5 shows the values of MSE (Υ) for the transpose and non-transpose case. The difference in the MSE values for the non-transpose and transpose case is negligible. The MSE for the dual formulation is identical to that of the transpose case, because it is simply a reformulation of the transpose model.

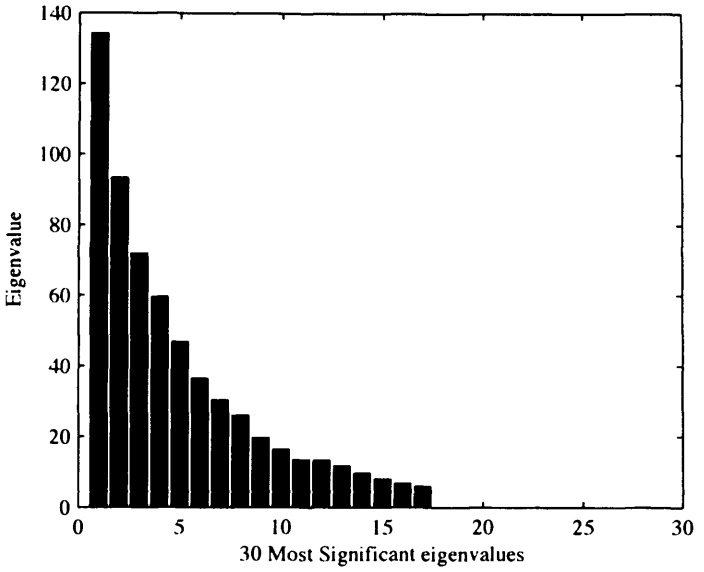


Figure 3.9. First 30 eigenvalues of $\mathbf{R}^{\mathbf{X}}$ for the *alpha* data.

The MSE decreases as the number of sources increases. The reason for this is clear, in PCA (and hence the first step of JADE) the reduction to m sources is achieved through discarding the $(P - m)$ least significant eigenvectors and eigenvalues and projecting the data onto the m remaining eigenvectors. Unless the eigenvalues which are discarded are all zero, some information is necessarily lost. Figure 3.9 shows the 30 most significant eigenvalues of $\mathbf{R}^{\mathbf{X}}$ for the *alpha* data. Note that there are only 18 non zero eigenvalues because $\mathbf{X}$ is 6075×18 and so the rank of $\mathbf{X}$ cannot exceed 18.

There is no sudden drop in the eigenvalues and so the proportion of variance explained will tend slowly towards unity as $m \rightarrow \text{rank}\left(\mathbf{R}^{\mathbf{X}}\right)$. The choice of m is hence a tradeoff between the proportion of variance explained by the model, and corresponding MSE, and the validity of the returned sources. A further examination of the residual Υ should help to determine whether any important information is being lost, or if the discarded information is largely noise.

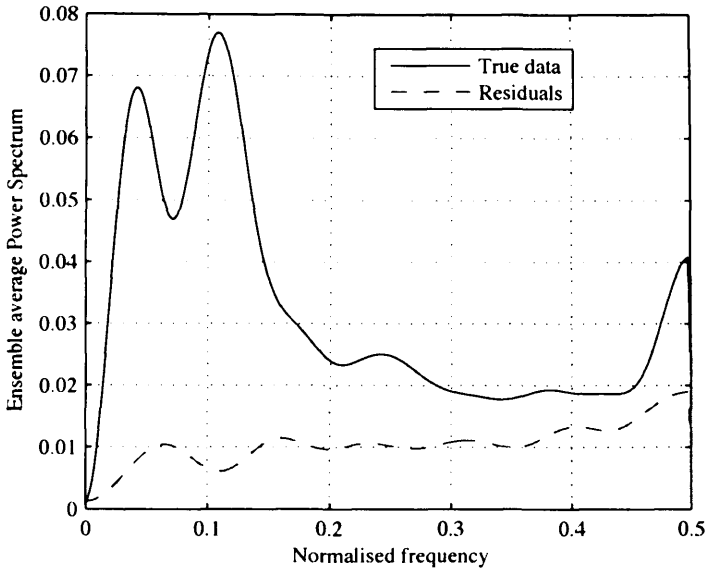


Figure 3.10. Ensemble average power spectra of the actual data $\mathbf{X}$, and the matrix of residuals $\boldsymbol{\Upsilon}$ for $m = 5$ on the *alpha* data. The power spectrum of the residual data is dominated by the high frequency noise region.

Figure 3.10 shows the ensemble average power spectrum of the residual data $\boldsymbol{\Upsilon}$, together with that of the actual data for $m = 5$ on the *alpha* data. The power spectrum in the residual is dominated by the high frequency noise region, indicating that the information lost by the discarded eigenvectors is largely noise.

3.4 Conclusions

The standard ICA model does seem to provide demonstrable improvements over PCA even at low sample sizes, however sample size is clearly the limiting factor in performance - as tests with synthetic data indicate. The dual formulation bypasses this limitation by transposing the data matrix but assumes independence in the columns of the mixing matrix, in addition, spatial uncorrelatedness is not enforced in the recovered sources. We highlight the effectiveness of second order methods

to recover sources which are spatio-temporally uncorrelated. This approach is more practical with the short sample sizes available and the recovery of spatio-temporally uncorrelated sources is as at least as attractive as those which are independent. The model error has been shown to be dependent on the magnitude of the eigenvalues discarded and, providing a sensible number of components is chosen, the discarded information is primarily high frequency noise.

The linear mixing model is not likely to be a true generative model for gene expression, however it does provide a first step towards a multiple input, multiple output model for gene expression. More advanced non-linear models can be used, but these generally require more data to estimate the parameters than the linear mixing model. As technology develops to provide larger sample sizes, more advanced models of gene expression may become feasible to estimate using blind, or semi-blind, techniques. In addition, as sample sizes become larger, the use of convolutive models may become practicable. These would allow the modelling of time lags in the generative model which are likely to be a factor in some parts of the gene regulatory process.

Given the current state of the art in microarray genomics, and corresponding low sample sizes, development of the generative model is challenging. In the next chapter we examine clustering, which is non-parametric in the sense that no underlying model is assumed. We show that clustering results can be improved through the use of spectral and BSS feature extraction steps. We also examine sparsity as a criterion for separation, which uses the same linear mixing model but requires rather less data to achieve good performance.

Chapter 4

CLUSTERING OF MICROARRAY DATA

4.1 Clustering in microarray data analysis

Clustering is of prime importance in the analysis of microarray data (see e.g. [19, 70]), and is a central feature of most microarray data analysis software packages. It allows the unsupervised grouping of thousands of gene profiles into a few clusters of similar profiles. The centroids of these clusters can then be examined and the time profiles explained by real biological processes, see for example [20]. In this way, thousands of gene profiles are decomposed into a few primary functional groups. This dimensional reduction allows biologists to concentrate on the key cellular processes apparent in a dataset. It also allows the discovery of the function of genes whose effects where previously unknown by comparing these genes with other, well characterised genes, in the same cluster. In this chapter, we examine how both the spectral estimation and the ICA work in the previous two chapters can be used as feature extraction steps to enhance clustering. We then introduce sparse component analysis as a method for source separation, show its advantages over the BSS methods studied thus far and demonstrate that its

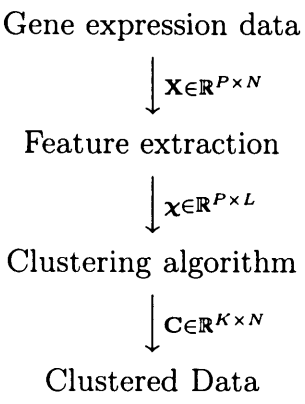


Figure 4.1. Summary of the clustering process. The feature extraction step acts on the data to produce an L dimensional vector of features χ. The feature extraction attempts to represent the data in the most separable form, either by discarding data which does not aid to discrimination between clusters or by a transformation of the data into a more separable form. $\mathbf{C}$ is the matrix of cluster centroids.

transpose form is equivalent to a standard cluster analysis of the data.

Given a P genes by N samples (usually time points) data matrix, $\mathbf{X}$, the P genes can be assigned to K clusters. Each cluster is represented by a cluster *centroid* which represents the average profile within the cluster. The membership level of each gene to each cluster is then given by the distance from each gene to the nearest cluster centroid. The clustering process is summarised in Figure 4.1. Many clustering algorithms have been designed, each of which typically has many variants, see [71] for a further overview of classification and clustering.

The most well known clustering algorithm is called the K-means algorithm and would cluster $\mathbf{X}$ (assuming no prior feature extraction step) into K clusters as follows:

1. Generate K initial cluster centroids. These can be provided on the basis of prior knowledge or randomly generated.

2. Assign each gene to the cluster with the closest centroid, as mea-

sured by a given distance measure.

3. Recompute the cluster centroid as the ensemble average of all the genes assigned to that cluster.

4. Go back to step 2 and repeat until the centroids move less than a set tolerance threshold.

The K-means algorithm can be represented by

$$\mathbf{C} = \text{kmeans}\,(\mathbf{X}, K) \tag{4.1.1}$$

where $\mathbf{C} \in \mathbb{R}^{K \times N}$ is a matrix of cluster centroids, each row of which represents a cluster centroid. The K-means method has been extensively studied in the literature and fast implementations exist [72].

The K-means algorithm requires the prior specification of the number of clusters K. This can be taken from prior biological knowledge, or tuned heuristically on the basis of the biological plausibility of the cluster centroids. Other clustering methods, e.g. hierarchical clustering [19], or Quality Threshold (QT) clustering [73], may be used if prior specification of K is not desirable. The K-means algorithm is deterministic for a given set of initial cluster centroids. However, for randomly generated initial centroids, it is advisable to run the algorithm multiple times to check the stability of the computed clusters, with respect to the initial centroids. The distance measure is critical to any clustering routine. It provides the actual criteron by which the closeness of gene expression profiles are measured and should be chosen to reflect the application. We now show that the clustering of cell-cyclic data can be significantly enhanced by the use of magnitude spectrum feature extraction.

4.2 Frequency domain feature extraction for microarray data

The clustering procedure outlined in the previous section operates directly on the gene expression profiles. This is problematic in the case of cell cyclic data because of the phase differences in the cyclic profiles. Two genes with perfectly cyclic profiles of identical frequency but significantly differing phases would likely be placed in different clusters. This is clearly not desirable if one cluster is meant to represent a functional group of cell cyclic genes. In practice, the spread of phases in the cyclic elements means that large number of clusters must be used to try and imperfectly approximate many different phases. The novel solution we propose to this problem is a Fourier based feature extraction step.

In this case the feature extraction function is simply the magnitude squared Fourier transform of the zero mean gene profile. In this way, the phase component of the profile is effectively discarded, allowing genes which could be cell cycle regulated to be clustered into the same functional group. Hence, the feature vector $\boldsymbol{\chi} \in \mathbb{R}^{1\times L}$, corresponding to a given gene profile $\mathbf{x} \in \mathbb{R}^{1\times N}$ can be defined elementwise as

$$\chi(k) = \left| \sum_{n=1}^{L} x_L(n)\, e^{\frac{-j2\pi(k-1)(n-1)}{L}} \right|^2 \quad \text{for} \quad 1 < k < L \tag{4.2.1}$$

where $\mathbf{x}_L(n)$ denotes the n-th index of $\mathbf{x}$, zero padded to length L. We note that any of the spectral estimators in Chapter 2 could be used as a feature extraction step, and use of the high resolution filterbank methods could provide benefits with longer length data.

Figure 4.2 shows the frequency domain centroids of clusters computed using the K-means algorithm with $K = 4$ and magnitude spectrum feature extraction on the *alpha* data. The centroids all appear to

represent biologically, or experimentally, valid functional groups: cell cycle related genes, high frequency noise, a low frequency component and a broadband coloured noise component. The clustering method is able to delineate the cell cyclic component from the lower frequency component which is evident in the twin peaks of the ensemble average periodogram in Figure 2.1. The lower frequency component is representative of a slowly changing dynamic process, most likely a rising or falling profile, in response to the shock of the chemical agent used to synchronise the cell culture. The high frequency noise is likely to be measurement noise, whilst the broadband component is more likely to be an amalgamation of other noise, or low amplitude biological effects, occuring throughout the experimental process. Though this example uses a K-means algorithm with a Euclidean distance measure, the Fourier feature extraction is equally applicable to other clustering algorithms and a sensibly chosen distance measure. In particular the use of QT clustering [73] with spectral feature extraction would allow the spread of the clusters to be controlled, enforcing smaller clusters of higher quality. The ability to group cell cyclic genes in a single functional group through a clustering routine is novel.

4.3 ICA feature extraction

In this section we show that ICA can also be used as a useful feature technique in gene clustering, as demonstrated in [74]. ICA is used to transform the gene profiles into a space of lower dimensionality whilst preserving the salient features. We show that this feature extraction step enhances the separability of the data and can produce clusters which are closer to those estimated using specific domain knowledge

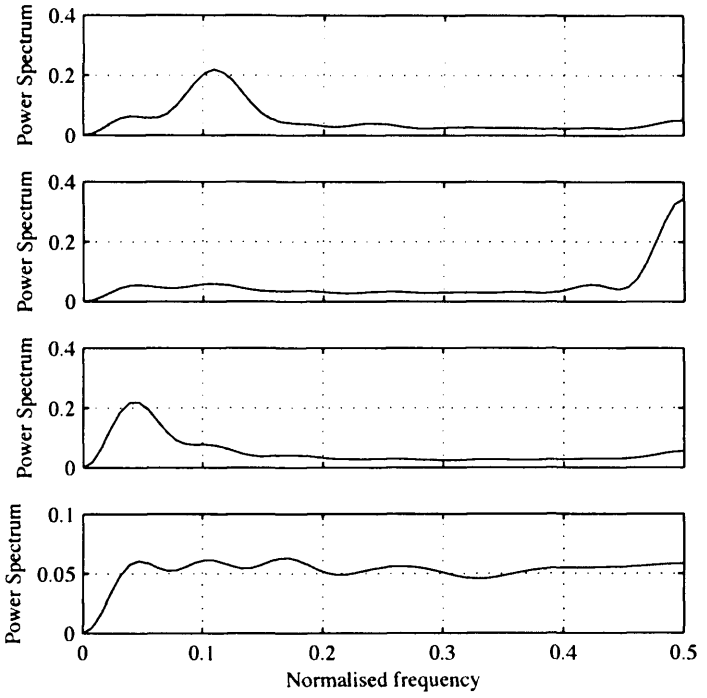


Figure 4.2. The plots show the frequency domain cluster centroids from the *alpha* data using K-means clustering with a Fourier preprocessing step for $K = 4$, $L = 128$, and a Euclidean distance measure. The centroids appear to be very interesting biologically and seem to represent distinct functional groupings: A distinct cell cyclic group, a high frequency noise group, a group with a distinct low frequency component and a group which seems to be a broadband coloured noise component. The ability to identify a cell cyclic functional group using a clustering routine is enabled by the Fourier preprocessing step.

than a standard clustering process.

We decompose the data matrix $\mathbf{X}$ using the standard linear mixing formulation as in Equation (3.2.2)

$$\mathbf{X} = \mathbf{AS} \tag{4.3.1}$$

We note that the j-th row of $\mathbf{A}$ represents the set of m weighting coefficients which map the space of statistically independent components back to the j-th gene. The j-th row of $\mathbf{A}$ is hence a compact representation of the j-th row of $\mathbf{X}$, with respect to the basis of statistically independent components. The j-th row of $\mathbf{A}$ now represents the feature vector for the j-th gene. The clustering is then performed as

$$\Theta = \text{kmeans}\,(\mathbf{A}, K) \tag{4.3.2}$$

and the cluster centroids in the transformed domain Θ are transformed back into the original domain by

$$\mathbf{C} = \Theta\mathbf{Y} \tag{4.3.3}$$

The same approach can be taken using the transpose formulation, given in Section 3.3.2. In this case, the approach is

$$\Theta = \text{kmeans}\left(\tilde{\mathbf{Y}}^T, K\right) \tag{4.3.4}$$

$$\mathbf{C} = \Theta\tilde{\mathbf{A}}^T \tag{4.3.5}$$

The dual approach may be useful where the number of timepoints is too low for the standard formulation to estimate higher order statistics

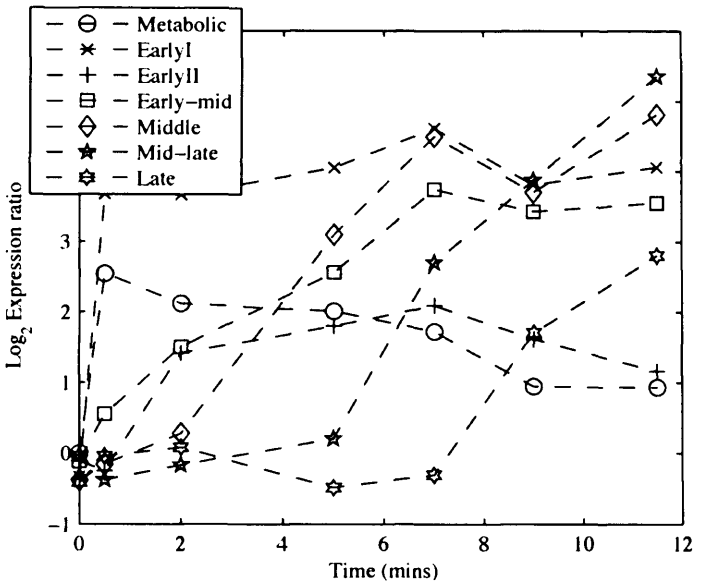


Figure 4.3. Benchmark profiles of salient underlying cellular processes, generated from sets of genes representative of those processes, which were selected using domain knowledge (from [4]).

effectively.

In order to test the performance of the ICA feature extraction step, data from [4] were used. This study monitored the gene expression of $P = 6118$ budding yeast genes over $N = 7$ timepoints. Despite having a relatively low number of time points, this study is particularly suitable for performance assessment of the clustering feature extraction because benchmark profiles are provided. These benchmark profiles were generated by averaging small sets of genes which were known to be representative of certain cellular processes. These benchmark profiles hence represent target profiles against which we can test our clustering procedures.

The benchmark patterns are termed Metabolic, EarlyI, EarlyII, Early-mid, Middle, Mid-late and Late and are shown in Figure 4.3. The matrix of benchmark profiles is denoted $\mathbf{C}_B \in \mathbb{R}^{K \times N}$, where each row of $\mathbf{C}_B$ represents a benchmark profile.

Method	Mean PI	PI Variance
Direct clustering	1.283	0.262
Dual ICA feature extraction	1.247	0.181
ICA feature extraction	1.278	0.276
PCA feature extraction	1.323	0.244

Table 4.1. Performance Index values for a range of clustering schemes, over 200 Monte Carlo trials. Kmeans clustering was used, with different initial centroids in each trial. $m = 3$ sources were used in the ICA and PCA approaches.

The closeness between the profiles which are blindly estimated through clustering routines and the benchmark profiles is difficult to ascertain by eye. The quality of the blind estimation can be measured by the same PI, given in Equation (3.2.9), as is used to assess ICA performance. The performance is measured as

$$\zeta = \mathrm{PI}\left(\mathbf{C}_B^{\dagger\,T}\mathbf{C}^T\right) \tag{4.3.6}$$

Both $\mathbf{C}_B$ and $\mathbf{C}$ are transposed to make the PI invariant to row swaps, which is necessary as the orders of both the benchmark profiles, and the cluster centroids, are arbitary.

The PIs of the feature extraction schemes are shown in Table 4.1. The dual ICA formulation achieved the best performance, with the lowest mean PI and lowest variance. This is to be expected with such few timepoints ($N = 7$), as the number of timepoints increases, it is likely the performance of the standard ICA solution would overtake that of dual formulation ICA. The benefit of the ICA based feature extraction comes from the use of higher order statistics to extract the salient features from the data, whilst reducing the noise that hampers direct clustering of the data.

4.4 Sparse component analysis

In Chapter 3 we examined blind source separation (BSS) techniques where blind recovery of sources was performed under the assumption of statistical independence or spatiotemporal uncorrelatedness. We discovered that the short data length of gene expression data was typically the limiting factor in the performance of the blind source separation algorithms. Here, we examine sparsity as a criteron for separation, which is the basis for a family of algorithms known as Sparse Component Analysis (SCA) [75]. The achievable separation performance of these algorithms is far less constrained by sample size than ICA algorithms, which are reliant on higher order statistics.

Given our data matrix $\mathbf{X}$ we assume a linear mixing model with m sources as in Section 3.3.1

$$\mathbf{X} = \mathbf{AS} \tag{4.4.1}$$

except that here we assume that $\mathbf{S}$ is not statistically independent but instead sparse, i.e. each column of $\mathbf{S}$ has at least one zero value. In practice, the performance of SCA algorithms improves as the number of non-zero entries in each column of $\mathbf{S}$ tends to one. Given such a model, the unknown mixing matrix $\mathbf{A}$[1] and corresponding source matrix $\mathbf{S}$ can be estimated by a two step procedure, up to the same permutation and scale ambiguities as the ICA case.

The columns of the mixing matrix $\mathbf{A}$ are estimated as the normalised

[1] Also called a signal dictionary in the SCA literature.

centroids of a clustering routine [76]. $\mathbf{A}$ can therefore be estimated as

$$\mathbf{C} = \text{kmeans}\,(\mathbf{X}, m) \tag{4.4.2}$$

$$\mathbf{A} = \mathbf{C}^T \tag{4.4.3}$$

We note that this estimate of $\mathbf{A}$ is suboptimal in the sense that the returned sources will not necessarily be the best global solution to (4.4.1) under a sparsity constraint[2], but in general will give a good solution [77].

The sources can now be estimated by the following minimisation [77, 78]

$$\min \sum_{i=1}^{m} \sum_{j=1}^{N} |s_{ij}| \quad \text{subject to} \quad \mathbf{AS} = \mathbf{X} \tag{4.4.4}$$

where $|\cdot|$ denotes the l^1 norm. This can be solved using the following dynamic programming solution with non-negative constraints

$$\begin{aligned} \min \sum_{i=1}^{m} (u_{ij} + v_{ij}) \quad \text{subject to} \quad & [\mathbf{A}, -\mathbf{A}] \left[\mathbf{u}_j^T, \mathbf{v}_j^T\right]^T = \mathbf{x}\,(j) \\ & \mathbf{u}_j \geq 0, \quad \mathbf{v}_j \geq 0, \end{aligned} \tag{4.4.5}$$

where $j = 1, \ldots, N$, $\mathbf{u}_j$ represents the j-th column of $\mathbf{U}$, and $\mathbf{S} = \mathbf{U} - \mathbf{V}$.

The SCA literature is focussed primarily on the underdetermined case (i.e. $m > P$), which makes the use of the l^1 norm method essential for a unique solution to the underdetermined equation $\mathbf{AS} = \mathbf{X}$ for a known $\mathbf{A}$ and $\mathbf{X}$. Our scenario is overdetermined (i.e. $m < P$) and so the unique solution can also be obtained using the l^2 norm solution $\mathbf{S} = \mathbf{A}^{\dagger}\mathbf{X}$.

In order to illustrate the performance of the SCA algorithm on short

[2]The estimation of the globally optimum mixing matrix remains an open problem.

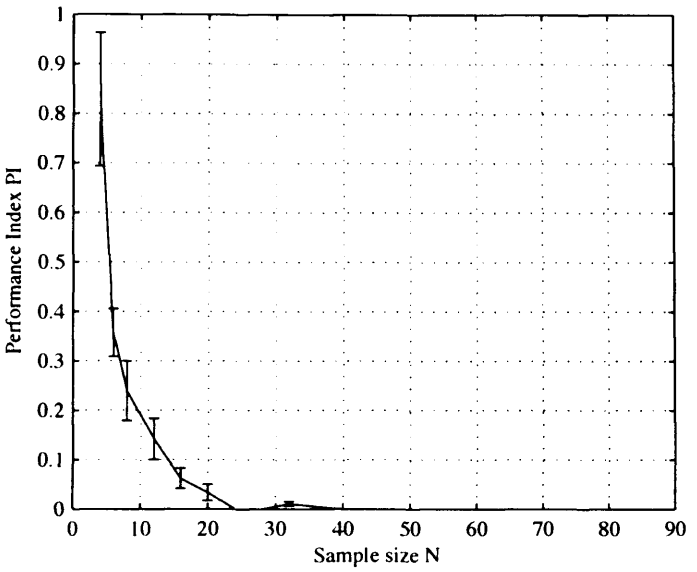


Figure 4.4. The Mean PI of the two stage SCA algorithm with synthetic data against sample size. $m = 5$ sources and $P = 100$ sensors were used with a randomly generated mixing matrix and perfectly sparse sources. 20 Monte Carlo trials were used to obtain the mean values, with the error bars denoting the variance. The PI falls to zero very quickly against sample size, in contrast to the ICA approaches in chapter 3, which typically require thousands of samples for the PI to approach zero.

data lengths, Figure 4.4 shows the PI of synthetic data over a range of short sample sizes. It is clear from the plot that the sparse component analysis problem requires only small sample sizes for excellent performance, in contrast to ICA.

Clearly, SCA is capable of good separation performance using the kind of data lengths typically generated in microarray time course experiments. The caveat, of course, is that the sources must be sparse for the technique to be successful. It is certainly plausible that some underlying biological sources could be sparse, though perfect sparsity in all sources is unlikely. Figure 4.5 shows $m = 5$ sources generated from the SCA algorithm on the *alpha* data from [1]. The rather jagged nature that the enforced sparsity gives the sources is unlikely to be

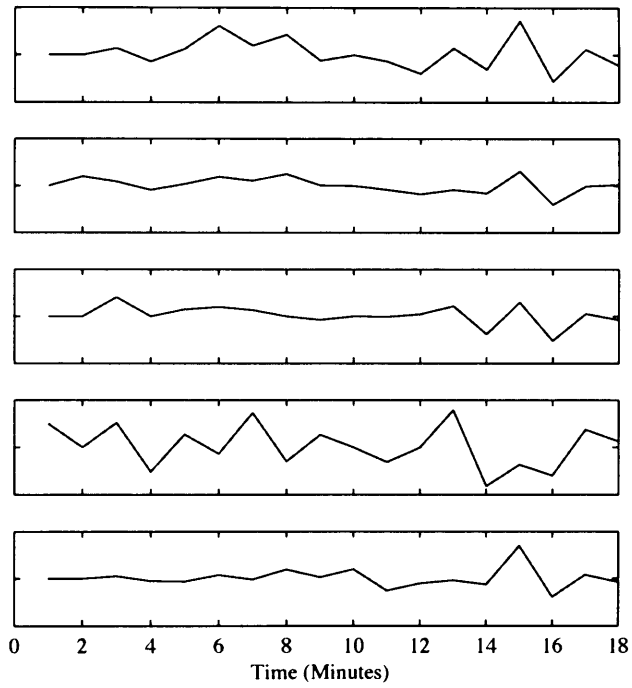


Figure 4.5. $m = 5$ sources generated from the *alpha* dataset of [1] using SCA. The components are certainly markedly different from those generated by ICA as in Figure 3.4. The enforcement of sparsity gives the sources a rather jagged appearance. This is unlikely to be accurate for all sources, though is rather plausible for noise components.

correct for all sources, though some could certainly be plausible. It is likely that the assumption of sparsity is not an accurate one for all the processes underlying gene expression. However, in order for SCA to recover more plausible sources, sparsity can be enforced by the application of suitable pre-processing. Wavelet approaches are common for this kind of pre-processing [75–77] and this is a promising topic for future research.

An alternative approach is to transpose the linear mixing model, as described in [75], to give

$$\mathbf{X}^T = \mathbf{S}^T\mathbf{A}^T \tag{4.4.6}$$

We note that, in this formulation, $\mathbf{S}^T$ is now recovered directly from the first stage of the SCA algorithm by simply clustering the data. The recovered sources are then nothing but the centroids of the computed clusters and so recovery of the sources can be achieved using a standard cluster analysis of the data. In this case, the columns of $\mathbf{A}^T$, i.e. the rows of $\mathbf{A}$ are assumed sparse. This is intuitive as a perfectly sparse $\mathbf{A}$ matrix is non-mixing and so each row of $\mathbf{X}$ simply represents one of the five sources, allowing the linear mixing to be decomposed by a simple clustering routine. The extent to which $\mathbf{A}$ is, in practice, sparse (and hence non-mixing) gives a measure of the quality of the clusters. Figure 4.6 shows $m = 5$ sources computed from the *alpha* data using this method.

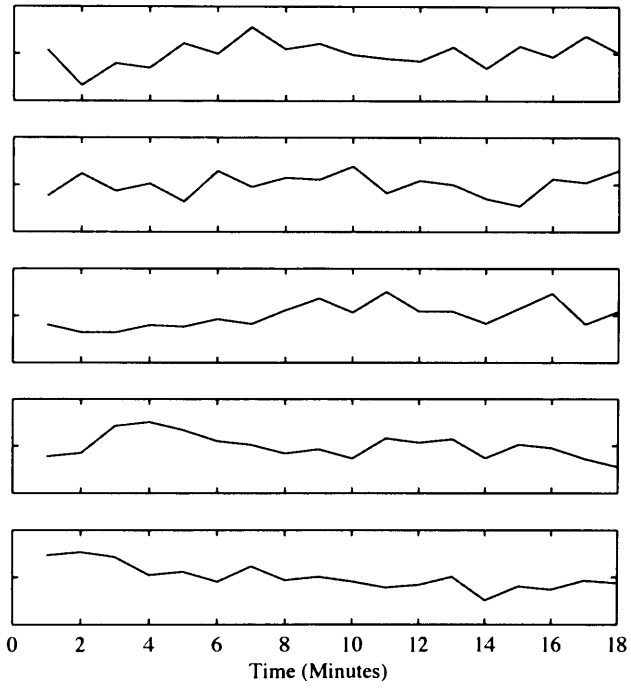


Figure 4.6. $m = 5$ sources generated by K-means clustering of the *alpha* data.

4.5 Conclusions

We have shown how both spectral estimation and BSS methods can be used as effective feature extraction techniques to improve gene expression clustering. In the case of the spectral feature extraction step, the novel ability to generate a single cluster of cell-cyclic genes is provided. The BSS based feature extraction allows the discarding of noise components and the transformation of the data into a space more amenable to clustering. Sparse component analysis is shown to yield good performance with the data lengths typical of microarray experiments and this may be a worthwhile avenue for future work.

Chapter 5

CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

The analysis of microarray data is in its infancy. Signal processing methods clearly have an important role to play in the analysis but classical, and cutting edge, signal processing techniques do not necessarily translate directly into successful methods for the analysis of microarray data. There are four primary reasons for this:

1. The data tend to be high dimensional in the sensor domain but with very limited time samples. Classical signal processing algorithms are typically designed with the opposite scenario in mind. The high number of sensors can lead to computational difficulties, whilst the limited time samples inhibit many estimation techniques.

2. The generative model for the data is unclear. The process of gene expression is partially understood on a qualitative level but numerical models can only be approximated for very small and well defined sections of the biological system. This limits the utility of many parametric signal processing techniques.

3. Microarray data contain significant noise with largely unknown characteristics. Missing data and outliers also hamper inference techniques.

4. The aim of most microarray studies is not well defined in a mathematical sense. Many studies tend to be exploratory experiments in which the aim is to gather knowledge about the relationships between genes, or their behaviour in certain environmental conditions, whereas signal processing algorithms tend to have a definite mathematical aim. The divergence between the knowledge sought by biologists and that which can be provided by tightly defined mathematical algorithms creates a challenge for the interpretation of results.

In addition to these technical considerations, the importance of human factors is not to be underestimated. The research fields of genomics and signal processing and statistics are traditionally distinct. The lack of mutual knowledge and understanding between the two disiplines is largely the reason why microarray data, often obtained at a cost of hundreds of thousands of pounds, are analysed using rudimentary statistical techniques. Often, the choice of which statistical analysis to use is governed solely by the options in the commercial microarray software package available rather than an assessment of its suitability to answer the question at hand. In order for the data analysis of microarray data to catch up with the ever advancing methods of data acquisition, greater co-operation is required between research communities.

Despite the challenges of the research area, substantial progress has been made in the analysis of signal processing approaches to microarray data analysis. Spectral estimation has been shown to be a sound

approach to the problem of cell-cyclic element detection and the robust Capon estimator for mis-sampled microarray data shows how spectral estimators can be designed to cope with microarray-specific data quality issues. The beamforming framework allows the estimation of the spectral content of a microarray dataset with a higher resolution than previously possible, and has been shown to work with non-uniformly sampled data. As more time samples become available, the higher resolution spectral estimation techniques will be able to assess the cyclic content of genes with a high degree of accuracy.

Blind source separation techniques introduce the concept of a mixing model to the problem and, despite the paucity of time samples, are shown to yield useful results. The standard ICA form is shown to yield benefits over PCA but the number of time points is the limiting factor. The dual form is shown to be not necessarily a perfectly accurate model but can nonetheless give good results, particularly where very short data lengths make the use of the standard form unfeasible. Second order methods are introduced, which may be a more realistic proposition than those reliant on higher order statistics - a fact which has yet to be recognised in the literature. The ability of SOBI to use second order statistics to extract components which are spatio-temporally uncorrelated is particularly attractive in gene expression analysis.

The use of both spectral and BSS techniques has been shown to benefit gene clustering. In particular, the use of a spectral feature extraction step allows the novel clustering of cell-cyclic genes into a single functional group. The use of BSS to transform the data into a space that is more amenable to clustering is shown to improve clustering results. Sparsity is introduced as a possible criterion for BSS of

microarray data and is shown to be effective with the number of time samples typically obtained in microarray studies.

5.2 Future work

There are many opportunities for future development of the work. With regard to the spectral estimation work for cell cycle detection, the availability of longer data lengths from microarray experiments is more critical than the further development of the algorithms. The short data lengths mean that it is impossible to ascertain whether the gene in question is truly cell-cyclic, or simply responding to the shock imposed by experimental conditions or is actually produced by chance. The availability of finely sampled data over multiple cell cycle periods would enable the use of a high resolution spectral estimator to place a sharp cut-off between those genes which are truly cell cyclic and those which are not.

The BSS separation work has numerous avenues for development. The validity of the mixing model for gene expression should be further examined and the development of more realistic models pursued. The use of convolutive and nonlinear models is likely to approximate the biological reality of gene expression more closely than a linear mixing model but more sophisticated models invariably require greater quantities of data for their accurate estimation and so these methods are not likely to be beneficial until longer time courses become available.

A more pressing source separation question, which should be feasible using currently available data lengths, is the application of preprocessing techniques to the sparse component analysis problem. We show that sparse component analysis is capable of excellent separation

performance with the limited data lengths typical of microarray experiments but the assumption of sparseness cannot be guaranteed to hold. Approaches such as the wavelet transform can provide a sparse basis for separation and this kind of method may yield good results with microarray data.

Clustering is so widely used in microarray studies that further development of clustering algorithms would be immediately beneficial to much current microarray based research. Research on feature extraction, similarity measures and the actual clustering algorithms themselves is all likely to be worthwhile, and immediately applicable to much recently available gene expression data. The design of a clustering procedure should be done with reference to aims and data of a specific experimental study. In fact, this approach is advised for signal processing approaches to microarray data in general. The challenging nature of the problem is such that application of known signal processing methods to microarray data can yield promising results, but these results are hard to assess. In order to advance the state of the art further, algorithmic approaches need to be designed with close co-operation between biologists and data analysts throughout the experimental process in order to answer specific biological questions. With this in mind, a good maxim for signal processing researchers interested in genomic applications would be an adaptation of an old quote.

> Ask not what genomics can do for you, ask what you can do for genomics.

With this kind of co-operation through the experimental process it is envisaged that signal processing will have a significant role to play

in understanding the fundamental function of organisms on a genomic scale.

Bibliography

[1] P. T. Spellman, G. Sherlock, M. Zhang, V. Lyer, K. Anders, M. Eisen, P. O. Brown, D. Botstein, and B. Futcher, "Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hydridization," *Molecular Biology of the Cell*, vol. 9, pp. 3273–3297, 1998.

[2] Z. Bozdech, M. Llinas, B. L. Pulliam, E. D. Wong, J. Zhu, and J. L. DeRisi, "The transcriptome of the intraerythrocytic developmental cycle of plasmodium falciparum," *Public Library of Science Biology*, vol. 1, no. 1, 2003.

[3] B. Alberts, D. Bray, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *Essential Cell Biology*. New York: Garland Publishing, 1998.

[4] S. Chu, J. DeRisi, M. Eisen, J. Mulholland, D. Botstein, P. Brown, and I. Herskowitz, "The transcriptional program of sporulation in budding yeast," *Science*, vol. 282, pp. 699–705, 1998.

[5] J. D. Watson and F. H. Crick, "A structure for deoxyribose nucleic acid," *Nature*, vol. 171, pp. 737–738, 1953.

[6] D. Gilis, S. Massar, N. J. Cerf, and M. Rooman, "Optimality of the genetic code with respect to protein stability and amino-acid frequencies," *Genome Biology*, vol. 2, no. 11, 2001.

[7] M. Tyers and M. Mann, "From genomics to proteomics," *Nature*, vol. 422, pp. 193–197, 2003.

[8] P. Brown and D. Botstein, "Exploring the new world of the genome with DNA microarrays," *Nature Genetics*, vol. 21, pp. 33–37, 1999.

[9] Y. F. Leung and D. Cavalieri, "Fundamentals of cDNA microarray data analysis," *Trends in Genetics*, vol. 19, no. 11, pp. 649–659, 2003.

[10] E. L. Liu, "Gene array technologies in biological investigations," *Proceedings of the IEEE*, vol. 93, no. 4, pp. 737–749, 2004.

[11] M. J. Heller, "DNA microarray technology: Devices, systems, and applications," *Annu. rev. Biomed. Eng.*, vol. 4, pp. 129–153, 2002.

[12] D. M. Rocke and B. Durbin, "A model for measurement error for gene expression arrays," *Journal of computational biology*, vol. 8, no. 6, pp. 557–569, 2001.

[13] Y. Tu, G. Stolovitzky, and U. Klein, "Quantitative noise analysis for gene expression microarray experiments," *PNAS*, vol. 99, no. 22, 2002.

[14] J. Quackenbush, "Microarray data normalization and transformation," *Nature Genetics*, vol. 32, pp. 496–501, 2002.

[15] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman, "Missing value estimation methods for DNA microarrays," *Bioinformatics*, vol. 17, no. 6, pp. 520–525, 2001.

[16] Z. Bar-Joseph, G. Gerber, D. Gifford, T. Jaakola, and I. Simon,

"A new approach to analyzing gene expression time series data," in *RECOMB*, pp. 39–48, 2002.

[17] D. V. Nguyen, N. Wang, and R. J. Carroll, "Evaluation of missing value estimation for microarray data," *Journal of Data Science*, vol. 2, pp. 347–370, 2004.

[18] C. D. Boor, *A Practical Guide to Splines*. Springer, 1994.

[19] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," in *Proc. Natl. Acad. Sci. USA*, vol. 95, pp. 14863–14868, 1998.

[20] A. P. Gasch and M. B. Eisen, "Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering," *Genome Biology*, vol. 3, no. 11, 2002.

[21] S. Wichert, K. Fokianos, and K. Strimmer, "Identifying periodically expressed transcripts in microarray time series data," *Bioinformatics*, vol. 20, no. 1, pp. 5–20, 2004.

[22] G. Hori, M. Inoue, S. Nishmura, and H. Nakahara, "Blind gene classification based on ICA of microarray data," in *ICA2001*, (Malmö, Sweden), 2001.

[23] X. Liao, N. Dasgupta, S. Lin, and L. Carin, "ICA and PLS modelling for functional analysis and drug sensitivity for DNA microarray signals," in *IEEE International conference on acoustics, speech and signal processing*, (Orlando, USA), 2002.

[24] W. Liebermeister, "Linear modes of gene expression determined by

independent component analysis," *Bioinformatics*, vol. 18, pp. 51–60, 2002.

[25] J. Hasty, D. McMillen, F. Isaacs, and J. J. Collins, "Computational studies of gene regulatory networks: In numero molecular biology," *Nature Reviews Genetics*, vol. 2, pp. 268–279, 2001.

[26] S. Cooper and K. Shedden, "Microarray analysis of gene expression during the cell cycle," *Cell and Chromosome*, vol. 2, no. 1, 2003.

[27] A. Oliva, A. Rosebrook, F. Ferrezuelo, S. Pyne, H. Chen, S. Skiena, B. Futcher, and J. Leatherwood, "The cell cycle-regulated genes of *schizosaccharomyces pombe*," *Public Library of Science Biology*, vol. 3, no. 7, pp. 1239–1260, 2005.

[28] R. Cho, M. Campbell, E. Winzeler, L. Steinmetz, A. Conway, L.Wodicka, T. Wolfsberg, A. Gabrielan, D. Landsman, D. Lockhart, and R. Davis, "A genome-wide transcriptional analysis of the mitotic cell cycle," *Molecular Biology of the Cell*, vol. 2, pp. 65–73, 1998.

[29] R. J. C. et. al., "Transcriptional regulation and function during the human cell cycle," *Nature Genetics*, vol. 27, no. 1, pp. 48–54, 2001.

[30] M. L. Whitfield, G. Sherlock, A. J. Saldanha, J. I. Murray, C. A. Ball, K. E. Alexander, J. C. M. Charles, M. Perou, M. M. Hurt, P. O. Brown, and D. Botstein, "Identification of genes periodically expressed in the human cell cycle and their expression in tumors," *Molecular Biology of the Cell*, vol. 13, pp. 1977–2000, 2002.

[31] X. Peng, R. K. Karuturi, L. D. Miller, K. Lin, Y. Jia, P. Kondu, L. Wang, L.-S. Wong, E. T. Liu, M. K. Balasubramanian, and J. Liu,

"Identification of cell cycle regulated genes in fission yeast," *Molecular Biology of the Cell*, vol. 16, pp. 1026–1042, 2005.

[32] R. K. Katuri and L. Jian-Hua, "Improved Fourier transform method for unsupervised cell-cycle regulated gene prediction," in *IEEE Computational Systems Bioinformatics conference*, (Stanford, USA), 2004.

[33] C. J. Langmead, A. K. Yan, C. R. McClung, and B. R. Donald, "Phase-independent rhythmic analysis of genome-wide expression patterns," in *RECOMB*, pp. 205–215, 2002.

[34] C. J. Langmead, C. R. McClung, and B. R. Donald, "A maximum entropy algorithm for rhythmic analysis of genome-wide expression patterns," in *Proc. IEEE Computer society bioinformatics conference*, pp. 237–245, 2002.

[35] C. J. Langmead, A. K. Yan, C. R. McClung, and B. R. Donald, "Phase-independent rhythmic analysis of genome-wide expression patterns," *Journal of Computation Biology*, vol. 10, no. 3, pp. 521–536, 2003.

[36] E. G. Larsson, J. Li, and P. Stoica, *High-Resolution Nonparametric Spectral Analysis: Theory and Applications.* In *High-resolution and robust signal processing*, Y. Hua, A. B. Gershman and Q. Cheng, Eds., New York, N.Y., USA: Marcel-Dekker, 2003.

[37] P. Stoica and R. Moses, *Spectral Analysis of Signals.* Upper Saddle River, N.J.: Prentice Hall, 2005.

[38] J. Capon, "High resolution frequency wave number spectrum analysis," *Proceedings of the IEEE*, vol. 57, pp. 1408–1418, 1969.

[39] R. Lacoss, "Data adaptive spectral analysis methods," *Geophysics*, vol. 36, pp. 134–148, 1971.

[40] T. Bowles, A. Jakobsson, and J. Chambers, "Advanced spectral estimation for the identification of cell-cycle regulated genes," in *IEEE EMBS UK and RI postgraduate conference in biomedical engineering and medical physics*, (Aston, UK), pp. 19–20, 2003.

[41] S. M. Kay, *Modern Spectral Estimation: Theory and Application.* Prentice Hall, 1988.

[42] A. Jakobsson, *Model-based and Matched-Filterbank Signal Analysis.* PhD thesis, Uppsala University, Sweden, 2000.

[43] G. Sherlock, "Personal communication." Email communication, 2003.

[44] T. Bowles, A. Jakobsson, and J. Chambers, "Detection of cell-cyclic elements in mis-sampled gene expression data using a robust capon estimator," in *International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, (Montreal, Canada), 2004.

[45] J. Li, P. Stoica, and Z. Wang, "On robust capon beamforming and diagonal loading," *IEEE Trans. Signal Processing*, vol. 51, no. 7, pp. 1702–1715, 2003.

[46] J. Li, P. Stoica, and Z. Wang, "Robust capon beamforming," *IEEE Signal Processing Letters*, vol. 10, no. 6, pp. 172–175, 2003.

[47] A. Gorokhov and P. Stoica, "Generalized quadratic minimization and blind multichannel deconvolution," *IEEE Trans. Signal Processing*, vol. 48, no. 1, pp. 201–213, 2000.

[48] M. Jansson and P. Stoica, "Forward-only and forward-backward sample covariances-a comparative study," *Signal Processing*, vol. 77, pp. 235–245, 1999.

[49] K. Shedden and S. Cooper, "Analysis of cell-cycle gene expression in saccharomyces cerevisiae using microarrays and multiple synchronization methods," *Nucleic Acids Research*, vol. 30, no. 3, pp. 2920–2929, 2002.

[50] J. Stone, "Independent component analysis: An introduction," *Trends in cognitive science*, vol. 6, no. 2, pp. 59–64, 2002.

[51] S. Lee and S. Batzoglou, "Application of independent component analysis to microarrays," *Genome biology*, vol. 4, no. R76, 2003.

[52] J. Berger, S. Hautaniemi, H. Edgren, O. Monni, S. Mitra, O. Yli-Harja, and J. Astola, "Identifying underlying factors in breast cancer using independent component analysis," in *NNSP*, (Toulouse, France), pp. 81–90, September 2003.

[53] S. Saidi, C. Holland, D. Kreil, D. MacKay, D. Charnock-Jones, C. Print, and S. Smith, "Independent component analysis of microarray data in the study of endometrial cancer," *Oncogene*, vol. 23, no. 39, pp. 6677–6683, 2004.

[54] J. Berger, S. Mitra, and H. Edgrin, "Studying DNA microarray data using independent component analysis," in *ISCCSP*, (Hammamet, Tunisia), pp. 747–750, March 2004.

[55] A. Carpentier, A. Riva, P. Tisseur, G. Didier, and A. Henault, "The operons, a criterion to compare the reliability of transcriptome

analysis tools: ICA is more reliable than ANOVA, PLS and PCA," *Computational Biology and Chemisty*, vol. 28, no. 3, 2004.

[56] G. Hori, M. Inoue, S. Nishmura, and H. Nakahara, "Blind gene classification an application of a signal separation method," *Genome informatics*, vol. 12, pp. 255–256, 2001.

[57] A. Hyvarinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Transactions on Neural Networks*, vol. 10, no. 3, pp. 626–634, 1999.

[58] X. Liao and L. Carin, "Constrained independent component analysis of DNA microarray signals," in *Workshop on Genomic Signal Processing and Statistics*, 2002.

[59] S. Kim and S. Choi, "Independent arrays or independent time courses for gene expression time series," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, (Kobe, Japan), 2005.

[60] J.-F. Cardoso, "Blind signal separation: statistical principles," *Proceedings of the IEEE*, vol. 90, pp. 2009–2026, October 1998.

[61] G. H. Golub and C. F. V. Loan, *Matrix computations.* Baltimore and London: The John Hopkins University Press, 1996.

[62] S. Amari, A. Cichocki, and H. Yang, "A new learning algorithm for blind signal separation," in *Advances in Neural Information Processing Systems*, vol. 8, pp. 757–763, 1996.

[63] J.-F. Cardoso and A. Souloumiac, "Blind beamforming for non Gaussian signals," *IEE Proceedings*, vol. 140, pp. 362–370, December 1993.

[64] J. F. Cardoso and P. Comon, "Independent component analysis, a survey of some algebraic methods," in *Proceedings of ISCAS'96*, vol. 2, (Atlanta, USA), pp. 93–96, 1996.

[65] J.-F. Cardoso and A. Souloumiac, "Jacobi angles for simultaneous diagonalization," *SIAM Journal on Matrix Analysis and Applications*, vol. 17, no. 1, pp. 161–164, 1996.

[66] J.-F. Cardoso, "High-order contrasts for independent component analysis," *Neural Computation*, vol. 11, pp. 157–192, January 1999.

[67] P. McCullagh, *Tensor methods in statistics.* London: Chapman and Hall, 1987.

[68] T. Bowles and J. Chambers, "The transpose model form in independent component analysis and its application to microarray data," in *Proceedings of the Institute of Mathematics and its Applications: International conference on mathematics in signal processing*, (Cirencester, UK), 2004.

[69] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines, "A blind source separation technique using second-order statistics," *IEEE Transactions on Signal Processing*, vol. 45, no. 2, 1997.

[70] J. Quackenbush, "Computation analysis of microarray data," *Nature Reviews Genetics*, vol. 2, pp. 418–427, 2001.

[71] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification.* Wiley, 2001.

[72] C. Elkan, "Using the triangle inequality to accelerate k-means," in

International Conference on Machine Learning (ICML), (Washington DC, USA), 2003.

[73] L. J. Heyer, S. Kruglyak, and S. Yooseph, "Exploring expression data: Identification and analysis of coexpressed genes," *Genome Research*, vol. 9, pp. 1106–1115, 1999.

[74] A. Kapoor, T. Bowles, and J. Chambers, "A novel combined ICA and clustering technique for the classification of gene expression data," in *IEEE Internation Conference on Acoustics Speech and Signal Processing (ICASSP 05)*, vol. 5, (Philadelphia, USA), 2005.

[75] P. Georgiev, F. Theis, A. Cichocki, and H. Bakardjian, "Sparse component analysis: a new tool for data mining," in *Data Mining in Biomedicine*, Springer, 2005.

[76] M. Zibulevsky, B. A. Pearlmutter, P. Bofill, and P. Kisilev, *Independent Components Analysis*, ch. Blind source separation by sparse decomposition in a signal dictionary, pp. 181–208. Cambridge University Press, 2001.

[77] Y. Li, A. Cickocki, and S. Amari, "Analysis of sparse representation and blind source separation," *Neural Computation*, vol. 16, pp. 1193–1234, 2004.

[78] Y. Li, A. Cickocki, and S. Amari, "Sparse component analysis for blind source separation with less sensors than sources," in *Proceedings of 4th International Symposium on Independent Component Analysis and Blind Signal Separation*, (Nara, Japan), pp. 89–94, 2003.