Bayesian Clustering of Curves and the Search of the Partition Space

by

Silvia Liverani

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

Department of Statistics

September 2009

THE UNIVERSITY OF
WARWICK

Contents

List of Tables

List of Figures

Acknowledgments

I would like to express my gratitude to all those who gave me the possibility to complete this thesis.

First and foremost, I would like to thank Professor Jim Q. Smith, the best supervisor one could wish for. Without his expertise, active involvement, sound advice, and encouragement, this work would not have been possible. Thank you for pushing me, for getting 100% involved and for always having my best interests in mind. I appreciate you both as a supervisor and as a person.

The University of Warwick and the Department of Statistics have provided great support throughout my postgraduate studies. Special appreciation goes to Mrs Paula Matthews for always being very helpful. I am grateful to EPSRC, through CRiSM (Centre for Research in Statistical Methodology), for providing financial support. Thanks also go to the Centre for Scientific Computing.

I would like to thank my collaborators. In particular, Paul Anderson for helping to get me started on my PhD and Andrew Millar for interesting discussions and encouragement.

On a more personal note, I would like to thank all the friends and colleagues that made many aspects of my life better. Special appreciation goes to Chris Cantwell for proof-reading but more importantly because his programming tips saved me years of

work. Also, thanks to Mylène Bédard for priceless advice and Maria Costa for being a great friend from beginning to end.

Infine, il ringraziamento più sentito alla mia famiglia, per il suo continuo supporto.

Declarations

I declare that this thesis contains my own work, except where stated otherwise. It has not been submitted in this form or similar elsewhere for the award of any other higher degree. The methods and results contained in Chapters 4, 5, and 6 have been incorporated into five publications with other authors. Chapter 7 is my own work.

The work on the geometry of Bayes Factors was started by Jim Q. Smith. I have contributed to a complete rewrite of a paper on this topic by streamlining the explanations, proofs and notation, correcting several technical errors and coding up the new algorithm based on the proposed new settings and so demonstrating their efficacy numerically. The paper, titled *Separation Measures and the Geometry of Bayes Factor Selection for Classification*, appeared in the Series B of the Journal of the Royal Statistical Society in 2008 (Smith et al., 2008b) and it is included in Chapter 4 of this thesis. Moreover, the new algorithm was used for an applied paper with biologists: *Light-regulated transcriptional networks in Ostreococcus provides insight into the biology and physiology of the marine eukaryotic picophytoplancton*, submitted to PNAS (Monnier et al., 2009). My contribution to this paper is the statistical analysis with the Bayesian Fourier Clustering.

Chapter 5 is based on a paper, *Efficient Utility-based Clustering over High Dimensional Partition Spaces*, accepted for publication by the Journal of Bayesian Analysis

(Liverani et al., 2009a). The third and fourth authors contributed the datasets for this paper, while the second author's main contribution was on the adaptation of the code by Heard et al. (2006) to this new set of problems. My main contributions to this paper are in the theoretical sections and Section 5.5.2.

The work on encoding MAX-SAT solvers for clustering problems, presented in Chapter 6, resulted in a conference proceedings paper: *Searching a multivariate partition space using weighted MAX-SAT*, included in the Proceedings of the 6th International Meeting of Computational Intelligence Methods for Bioinformatics and Biostatistics (Liverani et al., 2009b) This paper is mainly my own work, with James Cussens' contribution on the technical aspects of the encoding of the MAX-SAT as a clustering problem.

Abstract

This thesis is concerned with the study of a Bayesian clustering algorithm, proposed by Heard et al. (2006), used successfully for microarray experiments over time. It focuses not only on the development of new ways of setting hyperparameters so that inferences both reflect the scientific needs and contribute to the inferential stability of the search, but also on the design of new fast algorithms for the search over the partition space. First we use the explicit forms of the associated Bayes factors to demonstrate that such methods can be unstable under common settings of the associated hyperparameters. We then prove that the regions of instability can be removed by setting the hyperparameters in an unconventional way. Moreover, we demonstrate that MAP (maximum a posteriori) search is satisfied when a utility function is defined according to the scientific interest of the clusters. We then focus on the search over the partition space. In model-based clustering a comprehensive search for the highest scoring partition is usually impossible, due to the huge number of partitions of even a moderately sized dataset. We propose two methods for the partition search. One method encodes the clustering as a weighted MAX-SAT problem, while the other views clusterings as elements of the lattice of partitions. Finally, this thesis includes the full analysis of two microarray experiments for identifying circadian genes.

Abbreviations

MAP:	Maximum A Posteriori
AHC:	Agglomerative Hierarchical Clustering

Chapter 1

Introduction

Clustering, the process of dividing sets of observations into a smaller number of groups, is widely used in exploratory analysis. The advances of genome-scale sequencing in recent years, such as new assay techniques like DNA microarrays, allow the simultaneous recording of tens of thousands of variables and, together with the rapid growth of computer power, they have enabled Bayesian statisticians to implement computer-intensive inferential methods for clustering.

In particular, microarray experiments that measure the expression of tens of thousands of genes are now widespread and their shear size presents a challenge to any probability distribution guided clustering algorithm. More recently a large number of experiments have been performed that collect short longitudinal time courses - or time profiles - of microarrays. These profiles have been very useful in aiding the discovery of new genes in the various regulatory pathways in the studied organisms.

Because of its transparency, one particularly successful methodology within this domain has been the use of MAP model selection (Heard et al., 2006) on partitions of different clusters. The usual assumption is that genes in the same cluster share the same expression profile - albeit with some residual measurement error - and are otherwise expressed independently of genes outside that cluster. Each gene within a cluster is modelled using a regression model with respect to a basis, sensitively chosen to reflect the underlying science concerning the class of shapes. This basis function also respects the order and dependence over time of each cluster profile. Because regression models of this kind admit various conjugate Bayesian analyses, the logarithm of the marginal likelihood $S(\mathcal{C})$ of each possible partition $\mathcal{C}$ of the genes into clusters has an *explicit* closed form expression. Since the MAP Bayes Factor partition $\mathcal{C}^*$ is simply the partition $\mathcal{C}$ with the highest value of $S(\mathcal{C})$, we have an easily calculable score function over partitions where the score has been customised to the scientifically plausible basis functions and chosen prior hyperparameters.

In fact, under appropriate prior assumptions, the score $S(\mathcal{C})$ is not only expressible in closed form but can also be expressed as a linear function of component scores on clusters. This means that the vast partition space can be searched quickly using local moves, making the fast identification of close to optimal high scoring partitions feasible in a short time.

These methods have been used successfully in a wide range of applications, such as social sciences, medical sciences, financial markets and commerce (Denison et al., 2002; Zhou et al., 2006; Ray and Mallick, 2006; Lau and Green, 2007). Their speed, their ability to respect time ordering of expression and their faithfulness to some of the science behind the experiments have made the outputs of these analyses particularly useful.

This thesis focuses on recent contributions to this development by not only designing new fast search algorithms of the vast heterogeneous space, but also by developing new ways of setting the model hyperparameters so that inferences both reflect the scientific needs and contribute to the inferential stability of the search.

The structure of the thesis is as follows. The first two chapters introduce the terminology and give background information on both biology and statistics with a focus on clustering of time-course microarray experiments. These are followed by two chapters that focus on MAP model selection, its instability and its implementation when used in conjunction with a utility function. Then we present two chapters on two alternative approaches to the search of the partition space, and we conclude with a chapter on further research ideas and the appendices. Abstracts for the chapters follow.

Chapter 2 is a short introduction to the relevant biology and genetics, with a focus on microarray experiments. This is necessary to understand the motivation and relevance of the work presented in this thesis. Chapter 3 includes a review of clustering algorithms, with a particular focus on Bayesian clustering algorithms for curves and time-course observations.

Conjugacy assumptions are often used in Bayesian selection over a partition because they allow the otherwise unfeasibly large model space to be searched very quickly. The implications of such models can be analysed algebraically. In Chapter 4 we use the explicit forms of the associated Bayes factors to demonstrate that such methods can be unstable under common settings of the associated hyperparameters. We then prove that the regions of instability can be removed by setting the hyperparameters in an unconventional way. Under this family of assignments we prove that model selection is determined by an implicit separation measure: a function of the hyperparameters and the sufficient statistics of clusters in a given partition. We show that this family of separation measures has desirable properties. The proposed methodology is illustrated through the selection of clusters of longitudinal gene expression profiles. Moreover, in Chapter 4 we implement the results obtained and perform the analysis of a real dataset of microarray experiments on the tiny algae *Ostreococcus tauri*.

Even when Bayes factors have a closed form, in model-based clustering a comprehensive search for the highest scoring (MAP) partition is usually impossible, due to the huge number of partitions of even a moderately sized dataset. However, when each cluster in a partition has a signature and it is known that some signatures are of scientific interest whilst others are not, it is possible, within a Bayesian framework, to develop search algorithms which are guided by these cluster signatures. Such algorithms can be expected to find better partitions more quickly. In Chapter 5 we develop a framework within which these ideas can be formalised and we then illustrate the efficacy of the proposed guided search on a microarray time-course data set.

In Chapter 6 we present a different approach to the search of the partition space. The main contribution of this chapter is to encode the formal Bayes factor search on partitions as a weighted MAX-SAT problem and use well-known solvers for that problem to search for good partitions. We demonstrate how, with the appropriate priors over the partition space, this method can be used to fully search the space of partitions in small problems and how it can be used to enhance the performance of more familiar algorithms in large problems. We illustrate our method on clustering of time-course microarray experiments.

Chapter 7 offers a different approach to the search over the partition space by viewing clusterings as elements of the lattice of partitions. Great benefit can often be derived by searching the partition space in a different, more structured way. Here we define properties of the moves on the lattice and propose an algorithm that is consistent with them, explicitly using the lattice structure of the partitions in conjunction with linear properties of the score function of the partitions.

Finally, there is much further work that could be done in this area. The concluding Chapter 8, in particular, presents a research plan for the development of a method that moves away from partitions and towards the encoding of the biological regulatory network underlying these processes directly into the statistical model.

The appendices include the graphical output of the two data analyses presented in Chapters 5 and 6.

Chapter 2

Biological Processes

The work in this thesis is motivated by several discussions with biologists from the Millar Research Group[1] and the Circadian Clock and Cell Division Cycle Group of the University of Paris 06[2]. They provided us with datasets, obtained from microarray experiments that they performed, to identify the genes involved with the circadian rhythms of several organisms.

In this chapter we briefly introduce our terminology and motivation, since we will refer to these datasets throughout this thesis to illustrate the statistical methods that we propose. This chapter is based on introductions to biology for statisticians, such as Emmert-Streib and Dehmer (2008), Amaratunga and Cabrera (2003) and Deshmukh and Purohit (2007), for the sections on DNA and microarrays, and also on introductory books on circadian rhythms, such as Refinetti (2006) and Sehgal (2004). However, the focus of this thesis is Statistics.

[1] Research Group lead by Professor Andrew Millar, Centre for Systems Biology and School of Biological Sciences, University of Edinburgh.

[2] Research Group lead by François-Yves Bouget, Observatoire Océanologique

2.1 DNA Microarrays

All living organisms contain *DNA (DeoxyriboNucleic Acid)*, a molecule that encodes all the information required for the development and functioning of an organism. In order to understand how such a simple molecule can give rise to the amazing biological diversity of life, scientists find and decipher the information encoded in DNA. Microarrays provide a view into the biology of DNA, and thus a rich way to examine living systems.

DNA is a physical molecule that is able to encode information in a linear structure. Cells express information from different parts of this structure in a context-dependent fashion. DNA encodes genes, and regulatory elements control whether genes are on or off. In a loose sense, DNA could be described as existing in some number of states. Microarrays are a tool used to read the states of DNA.

In recent years, microarray analysis has become a key experimental tool, enabling the analysis of genome-wide patterns of gene expression. Individual experiments generate thousands of data points or observations, turning experiments into hypothesis-generating endeavours. Moreover, experiments generate more information than the experimenter could possibly interpret, transforming what used to be wet science into information science.

2.1.1 Biology of Microarray Experiments

Microarrays measure events in the genome. An event may be the transcription of a gene, the binding of a protein to a segment of the DNA, the presence or absence of a mutation, a change in the copy number of a locus, a change in the methylation state of the DNA, or any of a number of states or activities that are associated with DNA or RNA molecules. The purpose of a microarray is to measure expression of multiple genes simultaneously in response to some biological perturbation.

DNA is made up of four chemical building blocks called bases: adenine, cytosine, guanosine and thymidine (abbreviated as A, C, G or T). These building blocks are referred to as nucleotides. DNA consists of two long polymers of simple units called nucleotides, with backbones made of sugars and phosphate groups. Because these four bases can form sequences, it is possible to use them to encode information based on their patterns of occurrence. The amount of DNA, and therefore the length of the sequences, varies from organism to organism.

DNA often consists of two strands, antiparallel to each other. The two strands are hydrogen bonded together by interactions between the bases, forming a helical structure. Each type of base on one strand forms a bond with just one type of base on the other strand (A with T and C with G). Thus, if one knows the sequence of one strand, by definition, one knows the sequence of the opposite strand. This property has profound consequences on the study of biology and it is used by cells to replicate themselves. Therefore, the strands can essentially be melted apart and separated thus opening the way for a copying mechanism to read each single strand and re-create the second complementary strand for each half of the pair, resulting in a new double-stranded molecule for each cell. This is also the mechanism by which cells express genes.

This property of complementarity is also what is used for measuring gene expression on microarrays. Just as energy can melt strands apart and separate them into single molecules, the process is reversible so that single strands that are complementary to each other can come together and re-anneal to form a double-stranded complex.

This process is called hybridisation and is the basis for many experiments in molecular biology. The molecules can come from completely different sources, but if they match, they will hybridise.

Microarray methods label the complex mixture that is in a solution and utilise a

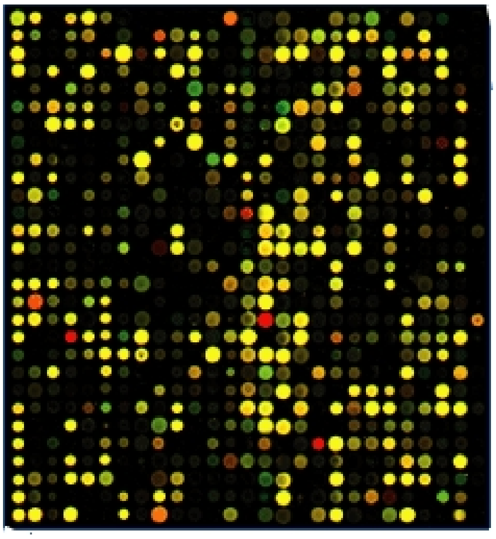


Figure 2.1: Example of a probe spotted oligo microarray. Each spot represents a gene.

two-dimensional surface of known molecules or *probes* (fragments of DNA) in discrete locations, as a readout. Complementarity between target molecules in the complex mixture and probes arrayed on the solid surface will result in annealing and hybridisation, thus capturing the labelled molecules on the surface. See Fig. 2.1 for a photograph of a microarray chip. See Emmert-Streib and Dehmer (2008), Amaratunga and Cabrera (2003) and Deshmukh and Purohit (2007) for an extensive overview of microarray experiments and a comprehensive literature review on the topic.

Types of Arrays

There are mainly three kinds of microarray technologies: spotted microarrays, Affymetrix GeneChips composed of relatively short oligonucleotides synthesised on a chip surface, and other *in situ* synthesis platforms such as arrays made by Agilent and NimbleGen. In this thesis we will analyse data from Affymetrix experiments only.

Affymetrix GeneChips are single sample microarrays. These arrays measure the relative abundance of every gene in a single sample. In this way, one can examine whether one gene is expressed at a higher or lower level than some other gene in the same sample. If samples are to be compared, a separate chip must be performed for each sample, and the data adjusted by scaling or normalisation before comparison.

Normalisation and Scaling

Affymetrix GeneChips use hybridisation intensities of single samples as a readout of gene expression. Since many factors unrelated to gene expression can affect the hybridisation properties of a gene, each gene is represented not by one probe but by a population of probes. Summarising a readout of several probes into a single value for gene expression adds a layer of complexity to data analysis because there are several ways probe sets can be polled and opinions differ on which method is best.

There are two basic steps involved in data analysis. The first is summarising the probe set representing a gene into a readout of expression for that gene. If the gene expression of two samples is to be compared, the results from each chip must be normalised in a second step to account for any differences in labelling and scanning of the samples. There are numerous approaches to carry out these steps. Regardless of the analysis method, microarrays are sensitive and it is important to be a good experimentalist. Many studies in the literature present the importance of pre-processing and show how this can influence the results in terms of a differential expression. See e.g. Sebastiani et al. (2003), Bolstad et al. (2003), Cope et al. (2004), Blangiardo and Richardson (2008) and Seidel (2008) for recently proposed methods and reviews of microarray normalisation and scaling methods.

Note that in this thesis we do not focus on the important issue of how the microarray experiments were carried out by the biologists with which we collaborated.

However, Dr. Julia Brettschneider (Department of Statistics, University of Warwick) agreed to compute several quality assessment measures (Brettschneider et al., 2008) on the Arabidopsis thaliana microarray dataset that we analyse in Chapters 4, 5 and 6. Microarray quality is assessed by comparing suitable numerical summaries across microarrays, so that outliers and trends can be visualized and poor-quality arrays or variable-quality sets of arrays can be identified. The tools proposed by Brettschneider et al. (2008) highlight different aspects in the wide spectrum of potential quality problems. The result of the quality assessment measures above on our Arabidopsis thaliana datasets indicated that the experiments appear to be free of temporal trends and patterns, batch effects, and quality biases related to sample properties or to experimental conditions. This is possibly due to the careful experiments conducted by our collaborators in the Millar lab. We did not investigate this aspect further as this is beyond the scope of this thesis.

Uses of Microarrays

Microarrays can serve many purposes and novel applications continue to emerge. A common application of microarrays has been the measurement of gene expression, from characterising cells and processes to clinical applications such as tumour classification. In this thesis we will focus on experiments that aim to identify the genes subject to circadian rhythms of several organisms.

2.2 Circadian Rhythms

The *circadian clock* is defined as an internal body clock possessed by living organisms including plants. It explains why an organism's behaviour alters according to the time of the day. The term *circadian* comes from the Latin words *circa* (around) and *diem* (day) meaning literally ‘approximately one day’. Circadian rhythms are the most important rhythms in chronobiology. Chronobiology is a field of science that examines periodic (cyclic) phenomena in living organisms and their adaptation to solar and lunar related rhythms.

Circadian rhythms are studied in many disciplines. They interact with medical and other research fields such as jet-lag, sleep disorders, endocrinology, geriatrics, sports medicine, space medicine, asthma, epilepsy, oncology, osteoarthritis, hypertension and many more.

Circadian rhythms are endogenously generated, and can be entrained by external cues, such as daylight and temperature. These rhythms allow organisms to anticipate and prepare for precise and regular environmental changes. The mechanism of the circadian clocks have been difficult to determine, but molecular and genetic studies indicate that the 24-hour period arises from a system of interconnected feedback loops that control the transcription of a small number of *clock genes*. However, note that circadian rhythms can be entrained to slightly shorter and longer periods than the Earth’s 24 hours, but in this thesis we assume that they remain as 24-hour cycles.

Circadian rhythms are outwardly very similar in all species but the genes that make up the clock mechanisms are quite different (comparing animals, plants, fungi and cyanobacteria). See Refinetti (2006) and Sehgal (2004) for an extensive overview of circadian rhythms and a comprehensive literature review on the topic.

The Millar Research Group at the University of Edinburgh and the Circadian Clock and Cell Division Cycle Group of the University of Paris 06 both study the circadian clock in different organisms. The Millar Research Group carries out experiments on a plant called *Arabidopsis thaliana*, whilst the work of the Circadian Clock and Cell Division Cycle Group focuses on a green algae called *Ostreococcus tauri*.

We can confidently say that over 16% of *Arabidopsis thaliana* genes (Edwards et al., 2006) and probably over 35% (Covington et al., 2008) are circadian-regulated in constant conditions, but over 80% during light/dark or warm/cold cycles (Michael et al., 2008). The rhythmic function of these genes controls many processes, including leaf and petal movements, the opening and closing of stomatal pores, the discharge of floral fragrances and many metabolic activities, especially those associated with photosynthesis. The circadian clock also influences seasonal cycles that depend on day-length, including the regulation of flowering. This photoperiodic system appears to depend on the circadian clock to measure the duration of the day or night, thus monitoring the passage of the seasons.

2.2.1 Arabidopsis thaliana

The *Arabidopsis thaliana*, commonly known as wall cress or mouse-ear cress, is a plant that is often used in biological research. It is widely used as model organism for a variety of reasons, such as its rapid life cycle (about 6 weeks from germination to mature seed), good germination rate and the easiness with which it may be cultivated in restricted spaces. There is an international research community of academic, government and industrial laboratories studying this plant and its characteristics. Arabidopsis has a short gene sequence, about 25,000 genes, which has been known in full since year 2000.

In particular, Arabidopsis thaliana has been widely studied in the chronology community because it exhibits visible circadian rhythms in leaf movement and less obvious rhythms in the expression of many genes. One may conclude that the plant simply responds to the environment, but it has been shown that the plant continues to exhibit the circadian rhythms even if taken out of its environment, as discussed in Robertson McClung et al. (2002). Biologists have therefore been able to deduce that

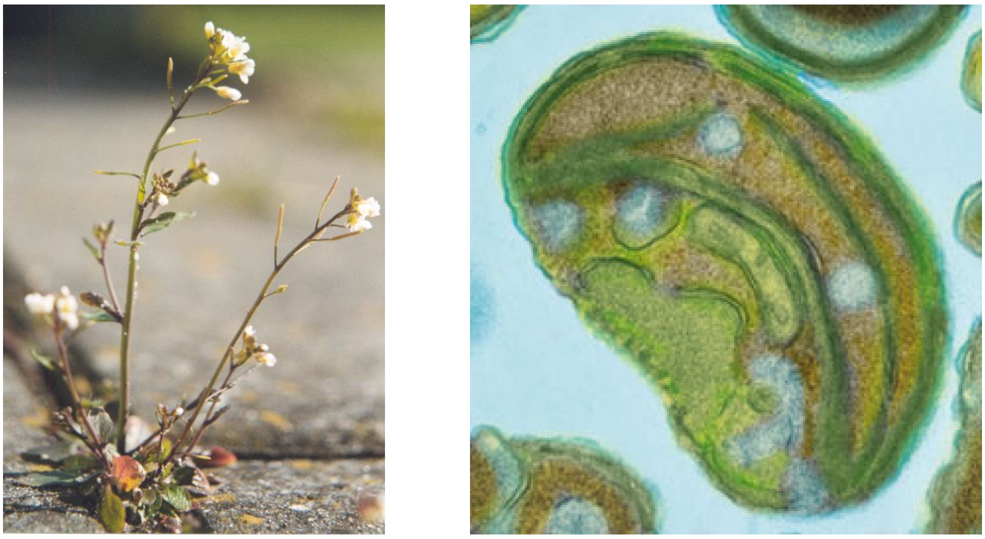


Figure 2.2: Arabidopsis thaliana (left) and Ostreococcus micrograph (right), courtesy Wenche Eikrem and Jahn Throndsen, University of Oslo.

the plant must have some kind of internal clock system, called the central oscillator. Environmental factors serve to synchronise the internal clock so that it stays on track with exogenous conditions. See Weigel and Glazebrook (2002) and references therein for more literature on Arabidopsis thaliana.

2.2.2 Ostreococcus tauri

Ostreococcus tauri is a genus of unicellular coccoid green alga belonging to the class Prasinophyceae, discovered in 1994 in the Thau lagoon. In addition to its very small size (1 micrometer), Ostreococcus has a compact genome of 12.5 Mbp, where Mbp stands for million base pairs[3]. Ostreococcus has a reduced set of genes involved in the regulation of both the circadian clock and the cell cycle. Furthermore, exponentially growing populations of algal cells are in a homogeneous physiological state compared

[3] The size of an individual gene or an organism's entire genome is often measured in base pairs (bp) because DNA is usually double-stranded. Hence, the number of total base pairs is equal to the number of nucleotides in one of the strands. 1 Mbp = 1,000,000 bp. For reference, Arabidopsis has a genome of 125 Mbp.

to multicellular organisms, which contain different tissues. Therefore, Ostreococcus appears to be a promising model through which to apprehend the complex interactions between the circadian clock and the cell division cycle at the molecular level. See Bowler and Allen (2007) and references therein for more literature on Ostreococcus tauri.

2.3 Our Data

In this thesis we analyse microarray experiments carried out by the Millar Research Group at the University of Edinburgh on Arabidopsis thaliana and the Circadian Clock and Cell Division Cycle Group of the University of Paris 06 on Ostreococcus tauri to identify the genes potentially regulated by the circadian clock in these organisms.

Our collaborators provided us with several datasets of microarray experiments performed at different time points to detect those genes that are potentially clock controlled. One of the main issues with these datasets is their size which can be of up to several tens of thousands of observations. Each observation is a short time series, usually of between 10 and 20 time points collected every few hours. Experimental conditions vary: experiments may be carried out in normal environmental conditions or in extreme conditions. For example, some experiments on Arabidopsis run on constant light for days.

The aim of the statistical analysis motivated by microarray experiments and presented in this thesis is to identify genes with similar expression profiles and potentially regulated by the circadian clock.

Chapter 3

Bayesian Clustering of Time-course Data

Classification can be described as the process of assigning a set of observations to subsets so that observations in the same cluster are similar in some sense to one another, and dissimilar to observations in other clusters. Classification is a useful tool for several exploratory applications such as pattern-analysis, grouping, decision-making, machine-learning, data mining, document retrieval, image segmentation, bioinformatics and pattern classification. Two broad categories of classification problems can be distinguished.

Discriminant analysis The groups are determined beforehand and the objective is to determine a method to discriminate among the groups. Individuals have a known group membership and other variables measured. In pattern recognition literature this type of classification problem is referred to as *supervised pattern recognition* or *learning with a teacher* whilst in statistical terminology it is referred to as *discriminant analysis* or *assignment* (Gordon, 1987).

Cluster analysis The groups (clusters) are not predetermined and in fact the object is to determine how best to cluster these objects into groups. Individuals (in our case, genes) have some variables measured (in our case, the expression levels at various times). However, the groups are not known *a priori* but have to be discovered. The conclusion of cluster analysis may well be that any summarisation of the data would be misleading. This type of classification problem is also referred to as *clustering*, *unsupervised pattern recognition* or *learning without a teacher*, *automatic classification*, *numerical taxonomy*, *botryology*, *typological analysis*.

Terminology varies and in the literature classification is used to describe the whole subject or it may have either of the restricted uses given above.

In the vast literature on classification, clustering and discriminant analysis, see Gnanadesikan (1989), Gordon (1987) and Sebastiani et al. (2003) for reviews of classification methods.

In this thesis we will focus on cluster analysis, as the most appropriate collection of statistical methods for grouping genes with similar profiles and answering the biological questions introduced in the previous chapter. We will refer to it interchangeably as either cluster analysis or clustering. In this chapter we introduce the statistical concepts and methods for clustering of observations, heuristic and model-based in Section 3.1. In Section 3.2 we introduce some of the methods used for clustering of time series and in Section 3.3 we focus on Bayesian clustering algorithms.

3.1 Data Clustering

Clustering is an exploratory method that aims to reduce the complexity of large datasets by grouping together observations with similar features. It requires the organisation of a collection of patterns into clusters based on similarity. In most of the literature on clustering the term implies a *partition* of a set of data points.

In mathematics, a partition $\mathcal{C}$ of a set $D = \{y_1, \ldots, y_n\}$ of cardinality n is a division of D into non-overlapping non-empty subsets that cover all of D. The units or observations y_i that belong to D are r-dimensional vectors. In our context of the biological application outlined in Chapter 2, the observations y_i correspond to the log expressions of gene i over r time points at which measurements are taken. Note, however, that in this section y_i is an observation, not necessarily a time series.

The output of a clustering algorithm is a number of *clusters* which are non-empty subsets of the set of observations D. Let c_k for $k = 1, \ldots, N$ represent a subset of D and the kth element of partition $\mathcal{C}$. The following holds:

1) $c_k \in \mathcal{C}$ for $k = 1, \ldots, N$

2) $\bigcup_{k=1}^{N} c_k = D$

3) $c_k \cap c_l = \emptyset$ if $k \neq l$

However, it is worth mentioning another area of research, explored particularly in biological taxonomy, which studies non-partitioning methods where clusters can overlap. For example, see Hubert (1974), Peay (1975) and the Fuzzy C-means method developed by Dunn (1973) and improved by Bezdek (1981). Much less attention is devoted to such methods. Clustering methods that partition have substantial practical advantages in analysing large datasets because they generally produce greater information reduction and for this reason will be the focus of this thesis. Regarding the terminology, usually a *grouping* (or *fuzzy clustering*) of a set of data points is a collection of its subsets, such that each point lies in at least one subset. A *clustering*, as defined above, is a grouping which partitions the set of data points.

As mentioned above, several research communities have different terminologies and assumptions for the components of the clustering process in their field of application. Therefore, a complete review would be a monumental task given the sheer mass of literature in this area. Also, the accessibility of the review would also be questionable given the need to reconcile very different vocabularies and assumptions regarding clustering in the various communities.

The goal of this chapter is therefore only to survey the core concepts and techniques in the large subset of cluster analysis with its roots in statistics and decision theory, with a focus on clustering of multidimensional observations with dependent data points.

Clustering methods range from those that are largely heuristic to more formal procedures based on statistical methods. In the following sections we will review some of the most common clustering algorithms, such as heuristic clustering and model-based clustering.

3.1.1 Heuristic Clustering

The term *heuristic* refers to those clustering algorithms that do not require the specification of a probabilistic model. These are based on mathematical distances between observations but do not account for any random error. One of the main advantages of such methods is their simplicity and interpretability which has made them very popular in several fields of application. The literature on such methods is extensive and we will limit ourselves to mention a few of them below. See Hartigan (1975) for an extensive review of such clustering algorithms.

One of the most popular clustering algorithms is the *k-means* algorithm (MacQueen, 1967). A number k is chosen and k cluster centre locations are randomly selected. Each data point discovers which centre it is closest too and then each centre calculates the centroid of the points it owns. This is repeated until a certain convergence criterion is satisfied. Some authors refer to algorithms such as k-means as *relocation methods* because of their nature of relocating observations to different clusters in the convergence step. The k-means algorithm is strongly influenced by the choice of k that the scientist has to choose beforehand.

There are many alternative methods that use the same concept of minimising the distance between observations in the same cluster and maximising the distance between observations in different clusters. The construction of a relevant measure of pairwise dissimilarity (or similarity, in some cases) is often the first step in these studies. Single link, complete link, group average link, weighted average link, sum of squares, incremental sum of squares, centroid, median and flexible are just a few of the similarity measures that belong to this category: see Gordon (1987) for a list of references. In contrast to the k-means algorithm, the number of clusters is not fixed and it is selected once the algorithm has run. This is a strong advantage of these methods as clustering is often used as an exploratory technique to reduce the dimensionality of the data and choosing the number of clusters a priori is not always possible or sensible, especially because often in practice it strongly affects the result of the clustering.

The algorithms above which are based on similarity measures must be used in conjunction with a procedure that selects the portions of the partition space to explore, as there is no convergence step as, for instance, in the k-means clustering. Such a search is often carried out by hierarchical algorithms.

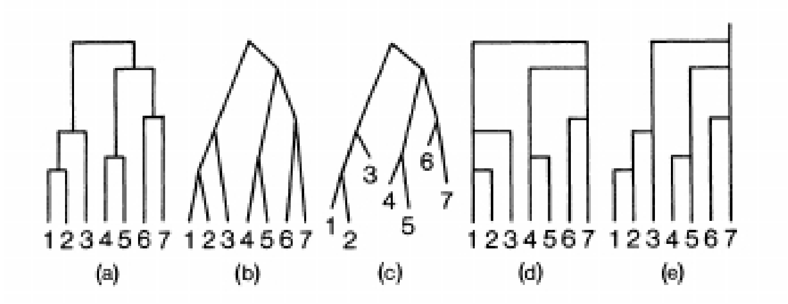


Figure 3.1: Five different formats for representing dendrograms (Gordon, 1987).

Hierarchical Algorithms

Hierarchical algorithms were initially used to search the space of partitions for heuristic clustering, but they are now widely used in conjunction with probabilistic clustering too. See Gordon (1987) for a review of hierarchical algorithms, with a focus on heuristic clustering.

The *Agglomerative Hierarchical* Algorithm (AHC) is the most popular one in this area. The outline of this algorithm follows: Initially there are n singleton classes, each one corresponding to one of the observations available. At each stage in the algorithm the most similar pair of classes is amalgamated. Different clustering strategies are distinguished by the manner in which the similarity of two classes of objects is defined. Agglomerative algorithms are often represented by *dendrograms*. These are trees which specify hierarchically nested sets of subsets, each subset corresponding to a class of similar objects, and with the additional property that a height is associated with each of the internal nodes. See Fig. 3.1 for an example. We use an algorithm of this type in Chapters 4 and 5.

An alternative is the *divisive* algorithm. Initially there is one class containing all n objects. At each stage an existing class is divided into two. However, these algorithms are computationally very demanding and their use is not as widely spread as AHC. Other alternatives are the *constructive* and the *direct optimisation* algorithms (Gordon, 1987).

Agglomerative algorithms do not require the number of clusters to be fixed *a priori*. Therefore there are several techniques that have been proposed to address this issue, such as cutting the dendrogram obtained at a prespecified level of similarity, cutting where the gap between two successive combination similarities is largest, prespecifying the number of clusters N and selecting the cutting point that produces N clusters, or using a measure that balances distortion and model complexity.

3.1.2 Model-based Clustering

As mentioned above, heuristic procedures do not account for the probability distribution of the clusters. In this thesis we focus on model-based clustering methods which assume that the data are generated by a mixture of underlying probability distributions in which each component represents a different group or cluster.

The three main components of model-based clustering are the probabilistic model, the algorithm for the search of the partition space and the choice of the number of clusters.

Probability model Given observations $D = (y_1, \ldots, y_n)$, let $f_k(y_i|\theta_k)$ be the density of an observation y_i from the kth component, where θ_k are the corresponding parameters, and let N be the numbers of components in the mixture. Note that y_i is an observation and it can be multidimensional, that is, a vector. There are two ways to model our population. The *classification likelihood* approach maximises

$$L_{CL}(\theta_1, \ldots, \theta_N; \gamma_1, \ldots, \gamma_n|D) = \prod_{i=1}^{n} f_{\gamma_i}(y_i|\theta_{\gamma_i}) \tag{3.1}$$

where γ_i are discrete values labelling the classification: $\gamma_i = k$ if y_i belongs to the kth component. The *mixture likelihood* approach maximises

$$L_{ML}(\theta_1, \ldots, \theta_N; \tau_1, \ldots, \tau_N | D) = \prod_{i=1}^{n} \sum_{k=1}^{N} \tau_k f_k(x_i|\theta_k) \qquad (3.2)$$

where τ_k is the probability that an observation belongs to the kth component ($\tau_k \geq 0$; $\sum_{k=1}^{N} \tau_k = 1$).

The case where $f_k(y_i|\theta_k)$ is multivariate Normal has been solved analytically by Scott and Symons (1971) when the covariance matrices are kept constant. Banfield and Raftery (1993) extend the result covariance matrices of different size or orientation or shape. They also propose a local parameterisation for non-Gaussian clustering.

In this thesis we focus on a Bayesian model-based approach, an extension of the classification method approach above. In our framework, instead of maximising the likelihood, we search for the partition that maximises a score, as we discuss in Section 3.3.

Search of the partition space The classification likelihood can be used as the basis for agglomerative hierarchical clustering, but iterative relocation methods are widely used.

Iterative relocation methods for clustering via mixture models are possible through EM (Expectation–Maximisation) and related techniques, as in Fraley and Raftery (1998). The EM algorithm iterates between an E-step in which parameters of the distributions are estimated and an M-step in which the data likelihood is maximised with respect to the parameters until the convergence criteria are satisfied.

Choice of the number of clusters Hierarchical methods explore a subset of the partition space and there are a number of criteria to determine the optimal number of clusters. In addition to the usual methods for determining the number of clusters for hierarchical algorithms, the cardinality of the partition can be ascertained by choosing the model which maximises the classification likelihood among the models explored by the hierarchical search algorithm.

Alternatively, when EM is used to find the maximum mixture likelihood, an approximation to the Bayesian Information Criterion (BIC) (Schwarz, 1978) can be used to determine the number of clusters. When a problem with a discrete set of competing models is proposed, the term Bayes factor is used for the ratio of the marginal likelihood under one model, say $\mathcal{C}_i$, to the marginal likelihood under another model, say $\mathcal{C}_j$, that is

$$BF(\mathcal{C}_i, \mathcal{C}_j) = \frac{p(D|\mathcal{C}_i)}{p(D|\mathcal{C}_j)}, \tag{3.3}$$

The BIC provides a close approximation to the Bayes factor when the prior over the parameters is the unit information prior, that is, a multivariate normal prior with mean at the maximum likelihood estimate and variance equal to the expected information matrix for one observation (Kass and Wasserman, 1995; Stanford and Raftery, 2002). Twice the log marginal likelihood of the model is approximated as follows.

$$2 \log p(D|\mathcal{C}) + \text{constant} \approx 2 \log L_{\mathcal{C}}(D, \hat{\theta}) - m_{\mathcal{C}} \log(n). \tag{3.4}$$

where $L_{\mathcal{C}}(D, \hat{\theta})$ is the maximised mixture for the model $\mathcal{C}$ and $m_{\mathcal{C}}$ is the number of independent parameters to be estimated in the model. However, it is well known that BIC is a poor approximation to the logarithm of the Bayes Factor in high dimensional problems (Stone, 1979).

The model-based clustering algorithm described above is widely used and it has been implemented in R in the MCLUST package (Fraley and Raftery, 2002). However, this method has a number of limitations: the rate of convergence can be slow and it is not practical for models with a large number of components and where no dependence is allowed between variables. Therefore, as we will illustrate in later sections, it does not account for time dependence.

3.2 Clustering of Time-course Data

In the last decade the technology of biological experiments has advanced and equipment capable of producing large outputs has become widely spread throughout the biology community. However, sometimes biologists themselves do not have the tools to analyse such results as this is beyond their field of expertise. Many biological labs around the world teamed up with statisticians, mathematicians and bioinformaticians in the last few years.

Microarray experiments are an excellent example of such experiments, with outputs of millions of data points. In particular, in the last ten years these experiments have become increasingly wide spread and many labs have used them for different purposes. As mentioned in the previous chapter, in this thesis we focus on the analysis of time-course microarray experiments. Clustering is the main statistical tool used on such experiments to reduce the dimensionality and to allow the biologists to identify the main features of the data.

Clustering is one of the classical problems that statisticians have worked on for over 50 years, e.g. Cox (1957) and Fisher (1958). Many methods have been developed over the years but only at the end of the 1990's, with the availability of time-course microarray experiments, a new need for statistical techniques for clustering of time series became apparent. Initially statisticians adapted the available clustering algorithms to this new set of problems, but only at the beginning of this century models that incorporated the time component of such datasets became available. Therefore, most of the literature on clustering of time-course data is presented with examples on microarray experiments. This does not limit the areas of possible application of such techniques in the future, it only acknowledges the origin of the research interest in the area.

A wide range of clustering algorithms have been proposed to analyse time-course gene expression data. We review here some of the main approaches and in the following subsection we focus on the Bayesian methods.

One of the first attempts in this field was by Eisen et al. (1998) where the objective was to identify similarity in gene expression patterns. The correlation coefficient is used as a gene similarity metric to identify *co-expressing* genes and then a single-linkage cluster analysis is performed. This belongs to the family of hierarchical algorithms.

Tamayo et al. (1999) proposes a different approach by adapting Self-Organising Maps (SOMs) for clustering gene expression patterns. Initially Tamayo et al. (1999) preprocess the data, eliminating the genes that did not change significantly across samples. Then, to identify clusters, they use SOMs, an algorithm very similar to k-means: the number of clusters is fixed, an initial grid of cluster centroids is defined and iterations are performed to find the optimal position in the space for those centroids. The authors wrote the code to implement their algorithm (GENECLUSTER), making this paper a highly cited one.

In the same year the authors of another paper (Ben-Dor et al., 1999) provided their own algorithm implemented in a package called CAST (Cluster Affinity Search Technique), making it another highly cited paper. They proposed a graphic theoretic (based on theory of graphical models) approach which made no assumptions on the number of clusters sought. In the case of gene expression patterns they assume that some underlying cluster structure exists for a graph that represents correlation between patterns of different genes. However, this pattern is obscured by the complexity of biology and corrupted by experimental errors. Ben-Dor et al. (1999) propose an algorithm to retrieve such structure, which they assume would take the form of a clique graph.

However, model-based algorithms have been proven to outperform heuristic methods generally. Yeung et al. (2001) use model-based clustering in conjunction with the EM algorithm and BIC for the choice of the number of clusters. They compare the package MCLUST for model-based clustering, implemented by Fraley and Raftery (1998), to CAST by Ben-Dor et al. (1999) and they find that model-based approaches produce slightly higher quality clusters.

All the methods introduced above, however, ignore the chronological order of the time-course gene expression, i.e. arbitrarily permuting the time points of the observations does not affect the results of the clustering. Since gene expression levels evolve over time, time can be an important factor for the gene expression levels. This is especially true of the type of data modelled in this thesis. Biomedical informatics investigators have demonstrated the risks incurred by disregarding the dependency among observations in the analysis of time series (Ramoni et al., 2002). Therefore, time-invariance (changing the order of the observed time points changes the results) is a desirable property of a clustering algorithm for time series.

A first attempt towards a time-invariant model-based clustering is by Li et al. (2002) with a mixed-effects model using B-splines, treating gene expression levels as a continuous function of time. Then the EM algorithm is used in the framework of a mixture model. Luan and Li (2003) compare the mixed effects model with B-splines, with the latter having a better performance in the simulation studies. In the same year, independently, Bar-Joseph et al. (2002) published a similar approach with cubic splines using the EM algorithm, but fixing the number of clusters. Ma et al. (2006) propose a similar approach with cubic splines and a variation of the EM algorithm, using BIC for choosing the number of clusters.

Song et al. (2007) introduce the use of functional principal component analysis for identifying the most prominent features of the data and retaining the time dependence of the data, but then turn back to MCLUST for the clustering of such features without an elegant idea for incorporating functional analysis in the clustering algorithm. The use of functional analysis for clustering is explored more by Chiou and Li (2007). This paper contains many interesting ideas but it has a disadvantage that does not allow its use in practice: it is computationally intensive and the authors include examples on less than 100 genes only. Moreover, Chiou and Li (2007) assume that the number of clusters is known a priori and their results depend strongly on the initial clustering of the functional scores.

3.2.1 Bayesian Clustering of Time-course Data

Bayesian hierarchical modeling refers to a generic strategy for model building in which unobserved quantities are organized into a small number of discrete levels with logically distinct and scientifically interpretable functions and probabilistic relationships between them. These capture inherent features of the data. The hierarchy of levels makes it particularly suitable for modeling gene expression data, which arises from a number of processes and is affected by many sources of variability. In the Bayesian framework there are many approaches to modeling these different sources of variability using fixed effects, random effects and distributional assumptions.

We present in this section the main Bayesian contributions to clustering of time course data. However, it should be also mentioned the great contribution of reversible jump MCMC (Green, 1995) to the field of Bayesian clustering because it can be applied to mixture models (Richardson and Green, 1997) to allow the number of clusters to vary. The implementation of this approach can be fully Bayesian since all parameters of interest can be treated as random variables and their posterior distribution can be approximated with reversible jump MCMC.

The first attempt to a model-based Bayesian time-invariant approach for time-course data is relatively recent, by Ramoni et al. (2002) with a pseudo-Bayesian method: an agglomerative clustering is performed with an heuristic search and a Bayesian approach with improper priors is used to determine the number of clusters and score each partition, due to the computational effort otherwise necessary for a fully Bayesian model. Four years later the authors reviewed their algorithm (Wang et al., 2006) and proposed a polynomial basis function with proper priors, a more appropriate model for short time series. However, the heuristic search using a Euclidean measure is not convincing when used to explore partitions evaluated by a different method.

A significant hurdle in the identification of periodically expressed genes by microarray experiments arises from the substantial amount of noise in the observations. Only when the sampled cells are in good synchrony can time course readings reflect cell cycle course transcriptions. Obtaining a pure synchronise dpopulation is non-trivial. For example, Lu et al. (2004) presents a model for resynchronising time series expression data by assuming that expression profiles follow a specific pattern (sinusoids) and employing an empirical Bayes method to detect periodically expressed genes. Resynchronisation is an important aspect of microarray experiments but this is beyond the scope of this thesis.

A full Markov chain Monte Carlo (MCMC) approach is used in an early work by Wakefield et al. (2003), and then refined by Zhou et al. (2006). The use of a basis function representation with random effects is promising but the method is very computationally intensive, with the marginal likelihood not available analytically under their model. The size of microarray experiments makes this approach infeasible because the run time to obtain reasonably accurate approximations to the marginal likelihood for a full hierarchy would be excessive. Zhou et al. (2006) use a filtering technique to reduce the high dimensionality of the data before running the clustering algorithm in order to overcome this problem, whilst clustering itself should be the tool used for this reduction.

There are few papers on Bayesian clustering of time-course data. Ray and Mallick (2006) proposed a nonparametric Bayesian wavelet model for clustering functional data, relying on a Dirichlet process prior for the distribution of the wavelet coefficients. The model is promising for those applications for which the use of wavelets is appropriate, even though it is computationally intensive and in the paper Ray and Mallick (2006) only include examples with up to six hundred genes. Quintana and Iglesias (2003), Vogl et al. (2005) and Lau and Green (2007) are excellent papers on Bayesian clustering, but they are not directly interested in the clustering of time-course data. Note also that our clustering problem is different from the one approached, for example, by Muller et al. (2008) who include a regression on covariates. We do not have any covariate information available.

Finally, it is the paper by Heard et al. (2006) that, in our opinion, stands out in the field of clustering of time-course data. The model they propose is based on the ideas developed in Denison et al. (2002). Heard et al. (2006) propose a fully Bayesian approach with a conjugate family and a hierarchical search, without approximating techniques such as MCMC. This allows the clustering of many thousands of genes without pre-filtering the data. Moreover, it is not necessary to use approximating measures such as BIC, as the exact marginal probabilities are available. For the specific purpose of time-course data, this model has the advantage of being time invariant, and it is also very flexible.

For example, instead of observations over time, this could be adapted for observations treated with different doses or exposed to different treatments. Finally, as for any Bayesian analysis, the summary statistics of the posterior distribution of the regression coefficients have a clear interpretation. We therefore review the method proposed by Heard et al. (2006) in the next section.

3.3 Bayesian Hierarchical MAP Clustering

Our data is contained in a matrix $D = (y_1, \dots, y_n)$. We will refer to the data about gene i, as y_i, the row i in the data matrix D. Therefore, D is a $n \times r$ matrix, where r is the number of time points that we observe for each gene and n is the number of genes. Following the notation by Gupta and Nagar (2000), for a matrix $X(m \times n)$, $vec(X)$ is the $mn \times 1$ vector defined as

$$vec(X) = \left(\mathbf{x}'_1, \dots, \mathbf{x}'_m\right)' \tag{3.5}$$

where $\mathbf{x}_i, i = 1, \dots, n$ is the i^{th} column of X. Therefore, we define $y = vec(D)$, a column vector of length $n \times r$. So,

$$y = \left(\underbrace{y_{11}, \dots, y_{1r}}_{y_1}, \dots, \underbrace{y_{n1}, \dots, y_{nr}}_{y_n}\right)'. \tag{3.6}$$

Note that for simplicity in this thesis we will refer to r as the number of time points observed for each gene. Therefore we will be working with time series data. However, this can be generalised to any experiment with multidimensional data, not necessarily a microarray experiment. For example, the model above can be used for observations treated with different doses, exposed to different treatments or followed up over time.

Note that the model presented here applies directly to time series with only one observation for each gene at each time point (no replicates) but it can be extended to include replicates by adding a layer to the hierarchical model. However, the data available to us did not include replicates. See Angelini et al. (2007) for a more in depth discussion on this topic.

3.3.1 Bayesian Regression

Heard et al. (2006) propose the use of a regression framework for the data, that for gene i at time t is given by

$$y_{it} = X_i(t)\beta + \varepsilon_{it} \tag{3.7}$$

where β is a p-vector of basis coefficients, $X_i(t) = (X_{i1}(t), \ldots, X_{ip}(t))$ is in general a p-vector of specified basis functions of t and ε_{it} is an error that we model as Gaussian and independent.

Moreover, Heard et al. (2006) assume that each gene profile is exchangeable within the cluster that it belongs to and consider the Normal Inverse-Gamma conjugate Bayesian linear regression model, which takes the form

$$y^{(k)} = X^{(k)}\beta^{(k)} + \varepsilon^{(k)} \tag{3.8}$$

for cluster k, where $\beta^{(k)} = (\beta_1^{(k)}, \ldots, \beta_p^{(k)})$ is the vector of parameters with $p \leq r$, $X^{(k)}$ is the design matrix of size $n_k r \times p$, $\varepsilon^{(k)} \sim N(0, \sigma_k^2 I_{rn_k})$ where n_k is the number of genes in cluster k and I_{rn_k} is the identity matrix of size $rn_k \times rn_k$. See Section 3.3.2 for discussion on the choice of the design matrix $X^{(k)}$. Under this notation, a partition $\mathcal{C}$ of the genes divides them into N clusters of cardinality n_1, ..., n_N and it holds that

$$\sum_{i=1}^{N} n_i = n.$$

The well known Normal Inverse-Gamma (NIG) conjugate family is extensively reviewed by O'Hagan and Forster (2004). We will use the notation by Denison et al. (2002) and we can define the priors

$$p(\beta^{(k)}|\sigma_k^2) = N(m, \sigma_k^2 V) \quad \text{and} \quad p(\sigma_k^2) = \text{IGamma}(a, b) \tag{3.9}$$

where $V \in \mathbb{R}^{p\times p}$ is the prior covariance matrix, $m \in \mathbb{R}^p$ is the prior mean and $a \in \mathbb{R}$ and $b \in \mathbb{R}$ are hyperparameters or prior parameters. Since the authors assume that $\varepsilon^{(k)} \sim (0, \sigma_k^2 I_{rn_k})$, the likelihood takes the form

$$p(y^{(k)}|X^{(k)}, \beta^{(k)}, \sigma_k^2) = N(X^{(k)}\beta^{(k)}, \sigma_k^2 I) \tag{3.10}$$

and this leads us to the posterior distribution, that is,

$$p(\beta^{(k)}|y^{(k)}, \sigma_k^2) = N(m_k^*, \sigma_k^2 V_k^*) \quad \text{and} \quad p(\sigma_k^2|y^{(k)}) = \text{IGamma}\left(a_k^*, b_k^*\right) \tag{3.11}$$

where

$$\begin{aligned}
m_k^* &= (V^{-1} + X^{(k)\prime}X^{(k)})^{-1}(V^{-1}m + X^{(k)\prime}y^{(k)}), && (3.12)\\
V_k^* &= (V^{-1} + X^{(k)\prime}X^{(k)})^{-1}, && (3.13)\\
a_k^* &= a + n_k r/2, && (3.14)\\
b_k^* &= b + \gamma_k/2, && (3.15)\\
\gamma_k &= y'y + m'V^{-1}m - (m_k^*)'(V_k^*)^{-1}m_k^*. && (3.16)
\end{aligned}$$

In regression modelling it is usual to consider a centred parameterisation for $\beta^{(k)}$ so that $m = \mathbf{0} \in \mathbb{R}^p$.

The critical quantity in this clustering procedure is the marginal likelihood, or prior predictive distribution, for each cluster k and is given by

$$\begin{aligned}
p(y^{(k)}|\mathcal{C}) &= \int\int p(y^{(k)}|\beta^{(k)}, \sigma_k^2)p(\beta^{(k)}|\sigma_k^2)p(\sigma_k^2)d\beta^{(k)}d\sigma_k^2 && (3.17)\\
&= \left(\frac{1}{\pi}\right)^{n_k r/2} \frac{b^a}{(b_k^*)^{a_k^*}} \frac{|V_k^*|^{1/2}}{|V|^{1/2}} \frac{\Gamma\left(a_k^*\right)}{\Gamma\left(a\right)}, && (3.18)
\end{aligned}$$

as shown in Poirier (1995) (pp. 542–543).

3.3.2 Design Matrix

The design matrix, or basis function, $X^{(k)}$ is a key feature of the regression model above, as it approximates the true relationship between the explanatory variable and the response. The choice of design matrix is related to the data and the features of interest. For example, Heard et al. (2006) choose a family of basis functions called the *truncated power spline basis* while Ray and Mallick (2006) choose a wavelet basis function.

The matrix consists of rows of linear or non-linear functions of the time ordinates at which the gene expression measurements are taken. In our context a Fourier basis function seems appropriate in the context of our applications and following discussion with the biologists regarding the features that they are interested in. Moreover, when a full basis is chosen, as explained below, a Fourier basis can capture asymmetric behaviour too.

When p is an even number the basis function takes the form

$$\begin{aligned} X_i(t) &= (\, 1, \\ &\quad \cos\left(2\pi t/\mathcal{T}\right), \sin\left(2\pi t/\mathcal{T}\right), \\ &\quad \cos\left(2\pi t(2)/\mathcal{T}\right), \sin\left(2\pi t(2)/\mathcal{T}\right), \\ &\quad \cos\left(2\pi t(3)/\mathcal{T}\right), \sin\left(2\pi t(3)/\mathcal{T}\right), \\ &\quad \ldots, \\ &\quad \cos\left(2\pi t(p/2)/\mathcal{T}\right)) \end{aligned}$$

where $\mathcal{T}$ is the total time taken by the microarray experiments. In our context $\mathcal{T}$ is the number of hours between the first and last microarray experiment. When p is an odd number the last sine term in the equation above is added. The highest frequency we can fit to the data is the Nyquist frequency, given by π, while the lowest frequency we can reasonably fit completes one cycle in the whole length of the time series and it is given by $2\pi/\mathcal{T}$.

An analysis along these lines is sometimes called a Fourier analysis or a harmonic analysis. The Fourier series representation has p parameters to describe r observations and so it can be made to fit the data exactly. Note also that when $r = p$ the error term is still applicable as it refers to the random error within a cluster. Note that when a full Fourier basis is chosen $(r = p)$ the basis function can describe non sinusoidal behaviour. This is shown in our results chapters and the appendices. However, for computational reasons it is not always possible to include a full representation. In those cases $(r >> 15)$ the harmonics need to be selected according to the features of interest in the data. See, for example, Section 4.7.

The overall effect of the Fourier analysis of the data is to partition the variability of the series into components at frequencies $2\pi/\mathcal{T}$, $4\pi/\mathcal{T}$, $\ldots$, π. The component at frequency $2\pi j/\mathcal{T}$ is frequently called the jth harmonic. When p is even, for $X_i(t)\beta$, it is often useful to write the jth harmonic in the equivalent form

$$\beta_j \cos(\omega_j t) + \beta_{j+1} \sin(\omega_j t) = R_j cos(\omega_j t + \phi_j) \tag{3.19}$$

where j is even, $\omega_j = 2\pi t(j/2)/\mathcal{T}$,

$$R_j = \sqrt{\beta_j^2 + \beta_{j+1}^2}$$

is the amplitude of the pth harmonic and

$$\phi_j = \tan^{-1}(-\beta_{j+1}/\beta_j)$$

is the phase of the pth harmonic. See Chatfield (2003) for more discussion on Fourier analysis.

3.3.3 Prior Modelling and Setting of the Hyperparameters

Prior Covariance

Treating the hyperparameters of the prior covariance matrix V as fully unknown and following some multivariate distribution is ideal in a fully Bayesian approach but it would carry a great computational burden because conjugacy would be lost. Therefore, for speed, Heard et al. (2006) assume independence of the components of β so that $V = \text{diag}(v)$, where $v \in \mathbb{R}$. Then they calculate scores over a grid of values of the single parameter v choosing that value which maximises the marginal likelihood of the resulting clustering.

Partition Prior

We also require a prior model for the partition $\mathcal{C}$. In the absence of real prior information about the items, it is common practice to assign positive prior probability to every possible partition. Lau and Green (2007) review several partition priors in the literature. In this thesis we focus on two partition priors, the Multinomial-Dirichlet distribution used by Heard et al. (2006) and the Crowley prior, proposed by Crowley (1997), that has additional properties. Recall the notation: a partition $\mathcal{C}$ of the N genes divides them into N clusters of cardinality $\{n_1, \ldots, n_N\}$ with $n = \sum n_i$.

Heard et al. (2006) specify a uniform distribution on the number of clusters over the set $\{1, \ldots, n\}$

$$p(N) = \frac{1}{n}, \tag{3.20}$$

and a prior on the cluster sizes which is defined by the Multinomial-Dirichlet conjugate family. The Multinomial likelihood is given by

$$p(n_1, \ldots, n_N | N, \theta_1, \ldots, \theta_N) \propto \prod_{i=1}^{N} \theta_i^{n_i} \tag{3.21}$$

where $\theta = (\theta_1, \ldots, \theta_N)$ are the prior cluster cardinalities which have a Dirichlet distribution, given by

$$p(\theta_1, \ldots, \theta_N | N, \alpha_1, \ldots, \alpha_N) = \frac{\Gamma(\sum_{i=1}^N \alpha_i)}{\prod_{i=1}^N \Gamma(\alpha_i)} \prod_{i=1}^N \theta_i^{\alpha_i}$$

where $\alpha = (\alpha_1, \ldots, \alpha_N)$ are the prior parameters. See Castelo (2002) for more details on the Multinomial-Dirichlet conjugate family. Therefore,

$$p(\mathcal{C}|\alpha_1, \ldots, \alpha_N) = \frac{1}{n} \frac{\Gamma\left(\sum_{i=1}^N \alpha_i\right)}{\prod_{i=1}^N \Gamma(\alpha_i)} \frac{\prod_{i=1}^N \Gamma(n_i + \alpha_i)}{\Gamma\left(n + \sum_{i=1}^N \alpha_i\right)} \tag{3.22}$$

becomes

$$\begin{aligned} p(\mathcal{C}) &= p(n_1, \ldots, n_N | N) p(N) && (3.23) \\ &= \frac{(N-1)! n_1! \ldots n_N!}{n(n+N-1)!} && (3.24) \end{aligned}$$

when $\alpha_1 = \ldots = \alpha_N = 1$.

An alternative partition prior is proposed by Crowley (1997). We will refer to it as the Crowley prior. It takes the form

$$p(\mathcal{C}) = \frac{\Gamma(\lambda)\lambda^N}{\Gamma(n+\lambda)} \prod_{i=1}^N \Gamma(n_i) \tag{3.25}$$

where $\lambda > 0$ is the parameter of the partition prior, N is the number of clusters and n is the total number of observations, with n_i the number of observations in cluster c_i. Note that a particular example of a Crowley prior is the Multinomial-Dirichlet prior with uniform hyperparameters, where λ is set so that $\lambda \in (1/n, 1/2)$.

Both of the above partition priors have the desirable property of *exchangeability*, as discussed in Booth et al. (2008). First of all, $p(\mathcal{C})$ depends only on N and $\{n_1, \ldots, n_N\}$, so two partitions that share the same values of N and $\{n_1, \ldots, n_N\}$, and only differ by a permutation of the objects in the clusters, will have the same probability.

For example, there are 5 possible partitions for $n = 3$,

$$\mathcal{C}_1 : 123, \quad \mathcal{C}_2 : 12|3, \quad \mathcal{C}_3 : 1|23, \quad \mathcal{C}_4 : 13|2, \quad \mathcal{C}_5 : 1|2|3, \tag{3.26}$$

and $p(C_2) = p(C_3) = p(C_4)$ when the property of exchangeability holds. This property is a minimal requirement in our context given the arbitrariness of the assignment of the labels $1, \dots, n$ to the data.

Another property of interest is *consistency*, discussed in McCullagh and Yang (2006). The consistency property these authors demand requires that the prior remains exchangeable over the remaining genes if a gene is deleted. If this property were to fail, then, for example,

$$p(\mathcal{C}_1) \neq p(\mathcal{C}_2) + p(\mathcal{C}_3) \tag{3.27}$$

with

$$\mathcal{C}_1 : 123, \quad \mathcal{C}_2 : 1234, \quad \mathcal{C}_3 : 123|4. \tag{3.28}$$

This property holds for the Crowley prior but it does not hold for the Multinomial-Dirichlet partition prior. However, whilst this property is a compelling one when units of a partition lie in a potentially infinite family, it is not such an attractive condition to demand in our context, where the number of units (genes) is fixed to a finite number.

3.3.4 Hierarchical Clustering

Heard et al. (2006) use an agglomerative hierarchical algorithm, that is, the algorithm starts with all the genes in different clusters and, at each step, merges the two clusters with greatest *inter-cluster closeness*. The inter-cluster closeness is a Bayes factor which calculates the increase in marginal posterior likelihood that would be gained by merging two clusters, say k and l,

$$c_{kl} = c_{lk} = \frac{p(\mathcal{C}_1)p(y|\mathcal{C}_1)}{p(\mathcal{C}_2)p(y|\mathcal{C}_2)} \tag{3.29}$$

where $\mathcal{C}_1$ is the partition with the two clusters c_k and c_l merged and $\mathcal{C}_2$ is the partition from the previous step of the algorithm, with the two clusters c_k and c_l apart. Therefore,

$$c_{kl} = c_{lk} = \frac{p(\mathcal{C}_1) \prod_{k=1}^{N'} p(y^{(k)})}{p(\mathcal{C}_2) \prod_{k=1}^{N} p(y^{(k)})} \tag{3.30}$$

where $N' = N - 1$ because clusters k and l have been merged and $y^{(kl)}$ is the marginal likelihood for the cluster obtained by merging cluster k and l. It can be simplified and it takes the form

$$c_{kl} = c_{lk} = \frac{(n + N - 1)(n_k + n_l)!p\left(y^{(kl)}\right)}{(N-1)n_k!n_l!p\left(y^{(k)}\right) p\left(y^{(l)}\right)} \tag{3.31}$$

when comparing two partitions which are identical outside of clusters k and l.

The algorithm proceeds as follows.

1. Start with $n = N$ clusters, each containing one gene. Calculate the marginal posterior unnormalised probability kernel

$$\pi_N = \frac{(N-1)!}{N(2N-1)!} \prod_{k=1}^{N} p(y^{(k)}).$$

2. Calculate the inter-cluster closeness for all possible combinations of pairs of clusters k and l. This corresponds to $N(N-1)/2$ calculations at each stage.

3. Identify the clusters k and l that maximise the inter-cluster closeness and merge them. Set $N = N - 1$ and calculate the revised kernel $\pi_N = c_{kl}\pi_{N+1}$.

4. Repeat steps 2-3 until $N = 1$. Looking back over the clusterings visited, find the partition that maximised the marginal posterior unnormalised probability kernel. This is the optimal clustering.

Bayes factor methods have a direct meaning within the Bayesian framework. For example, given the distributional assumptions defined within and between clusters, a MAP partition is the most probable explanation of the data. This gives it a direct, easily understood, validity albeit with the caveat that the modelling assumptions might be inappropriate. For example, this allows us to adjust the clustering analysis to the scientific purpose of the experiment, e.g. focus on decision analysis as in Chapter 5.

3.3.5 Computational Efficiency

The high dimensionality of our data is crucial to examine the implications on the feasibility of the implementation. Being able to use a conjugate model, instead of a time-consuming MCMC, is challenging for such datasets, but it enables us to obtain an analytic expression for the marginal likelihood. This and other clever choices of covariance matrices and basis functions leads to simplifications in the calculations.

In our data the time points at which given expression profiles are observed are identical and this implies

$$X^{(k)'} = (B', \ldots, B') \tag{3.32}$$

where B is the $r \times p$ design matrix. Although assuming that all profiles are observed at the same set of time points leads to some of the computational savings in this section, this assumption can be relaxed without losing any of the model properties, unlike other clustering algorithms.

Therefore, we can rewrite the model from equation (3.8) as

$$y_i^{(k)} = B\beta^{(k)} + \varepsilon_i^{(k)} \tag{3.33}$$

for each gene which belongs to cluster k, where $k = 1, \ldots, N$ and $i = 1, \ldots, n_k$. Moreover, this implies

$$X^{(k)'}X^{(k)} = \big(B'B, \ldots, B'B\big) = n_k B'B \tag{3.34}$$

and

$$X^{(k)\prime}y^{(k)} = \sum_{i=1}^{n_k} B'y^{(k)} \tag{3.35}$$

simplifying the marginal likelihood in equation (3.17).

These procedures, that exploit various types of symmetry in the problem, and a few more, whose details can be found in Heard et al. (2006), lead to considerable savings in computational time.

3.4 Evaluation of Clustering Methods

In some classification problems, such as discriminant analysis, there is a *true* classification against which to compare the results. In this scenario a measure of the *goodness* of such a method is a simple count of the points misclassified, or a normalisation of it into a percentage error. However, when clustering a real dataset like we do in our applications, there is no absolute scheme with which to measure clusterings, only a natural extension of this idea involving the comparison of two arbitrary clusterings.

There are two general ways to compare clustering algorithms. The first is to consider how easy they are to use, and since this is a computer-oriented problem, it involves time and storage requirements. The second is to evaluate how well they perform when used and in the literature this concerns the development of a measure of similarity between clusterings. Note that a clustering can be evaluated by an *internal* criterion (e.g. distortion, likelihood) that is usually algorithm dependent, or by an *external* criterion, that measures the distance between two clusterings based solely on the comparison of the clusters obtained.

A widely used similarity measure between two partitions is the Rand index (Rand, 1971). The Rand index is defined as the number of pairs of objects that are either in the same group in both partitions or in different groups in both partitions, divided by the total number of pairs of objects. Thus, given n observations, $y_1, \ldots, y_n$, and two clusterings of them $\mathcal{C} = \{c_1, \ldots, c_{N_1}\}$ and $\mathcal{C}' = \{c'_1, \ldots, c'_{N_2}\}$, the Rand index $R(.,.)$ is defined as

$$R(\mathcal{C}, \mathcal{C}') = \sum_{i<j}^{n} \gamma_{ij} \Big/ \binom{n}{2} \tag{3.36}$$

where

$$\gamma_{ij} = \begin{cases} 1 & \text{if there exists } k \text{ and } k' \text{ such that} \\ & \text{both } y_i \text{ and } y_j \text{ are in both } c_k \text{ and } c_{k'} \\ 1 & \text{if there exists } k \text{ and } k' \text{ such that} \\ & y_i \text{ is in both } c_k \text{ and } c_{k'} \text{ while } y_j \text{ is in neither } c_k \text{ and } c_{k'} \\ 0 & \text{otherwise.} \end{cases} \tag{3.37}$$

Other popular measures are the Jaccard index (Ben-Hur et al., 2002), the Fowlkes-Mallows index (Fowlkes and Mallows, 1983), the Hubert and Arabie index (Hubert and Arabie, 1985) as well as statistically adjusted versions of some of the above (Hubert and Arabie, 1985). Gordon (1987) includes a review of evaluation and validation methods for clustering and also reviews methods for testing the absence (or presence) of class structure in data to determine the strength of evidence for any clustering.

An interesting paper by Meilă (2005) gives an axiomatic characterisation of some criteria for comparing clusterings and distances between partitions. Meilă (2005) views clusterings as elements of the lattice of partitions and defines distances in terms of its edges, introducing a general framework for similarity measures. One of the results of this paper is that many of the widely used indices (such as the Jaccard index, the Fowlkes-Mallows index, the Hubert and Arabie index and all of the adjusted indices) are non-local. An index is defined as non-local when a change inside a single cluster counts differently depending on how the data is clustered. These measures rate worse on the scale of understandability because of their non-intuitive interpretation in this respect.

However, one of the main issues with the use of any of the similarity measures mentioned above is that there is no gold standard with which a clustering can be compared. Rand (1971) proposes four procedures for the evaluation of fundamental aspects of clustering methods. The first is the retrieval of an obvious structure as a test of the ability of the algorithm to identify the generating clustering. The second is the sensitivity of a method to perturbation of the data, obtained by adding an error. The similarity measure is then used to compare the original clustering and the clustering of perturbed data. The third procedure tests the algorithm's sensitivity to missing individuals by clustering a subset of the observations available. The fourth and most popular procedure uses the similarity measure to test whether two clustering algorithms produce the same results when applied to the same data.

Recently, the vast field of the evaluation, comparison and reproducibility of clustering has also been studied by bioinformaticians, especially in relation to the performance of such methods in the classification of microarray experiments. See, for example, McShane et al. (2002) and Thalamuthu et al. (2006).

The evaluation of a clustering is an essential aspect of the validation of our work in this field. However, the existing methods reviewed above are sometimes restrictive in our context. For example, our field of application and the features of our data do not indicate the necessity to undergo tests for the absence or presence of a data structure (Gordon, 1987; McShane et al., 2002). Moreover, external methods for clustering evaluation produce coarse comparisons and they are not always appropriate in the absence of a gold standard to compare with, as in our case, and when the features of the data included in the model are very different.

Therefore, even though many authors use the Rand index to compare their clustering to other methods, in our context it is becoming more frequent to validate results with their biological interpretation or to resort to visual comparison. For example, Heard et al. (2006) compare to other standard methods by examining visually the expression profile clusters.

Throughout this thesis we use different methods to compare clusterings according to the context and the objective of each example included. In particular, we validate our results with their biological interpretation and resorting to visual intepretation, but we also attempt to retrieve simulated data to compare the performance of our algorithm to other methods (see Chapter 4) and we introduce outliers to test the algorithm's sensitivity (Chapters 5 and 6).

In the context of Bayesian clustering introduced in this chapter, we now focus on MAP model selection and its instability in certain conditions.

Chapter 4

Geometry of Bayes Factors

When a model space is vast, it is often expedient to select a Bayesian model using conjugate priors; see for example Barry and Hartigan (1992) and Heard et al. (2006). The Bayes factors then have a simple algebraic form and so the comparison of two models is almost instantaneous. This makes search algorithms for models with high posterior probability in this huge partition space orders of magnitude faster than their numerical non-conjugate analogues.

In this chapter we demonstrate that the explicit nature of this type of selection algorithm has another advantage. The properties and characteristics of the algorithm can be studied algebraically. In our particular case, its underlying geometry is linked with the well-studied behaviour of products of t-distributions (see for example O'Hagan and Le (1994), Chipman et al. (2001) and references therein). However, these authors focus on the geometry of t-distribution posteriors rather than on the geometry of Bayes factors. This enables us to explain not only how and why conjugate Bayesian model selection can break down under default settings of hyperparameters, but also to show that most of these apparent anomalies are removed if the hyperparameters are calibrated to plausible pre-posterior predictions, within a particular subfamily of these conjugate models.

In the next section we briefly review the geometry of the types of products of t-densities which form the marginal likelihoods of this class. In Section 4.2 we demonstrate how this geometry impinges on model selection based over partitions with particular emphasis on the methodology proposed in Heard et al. (2006). We illustrate how and why standard settings of hyperparameters can produce poor selection characteristics in Section 4.3. In Section 4.4 we derive explicit characterisations ensuring that Bayes factor selection prefers partitions that combine clusters when they are close with respect to a certain separation measure. In Section 4.5 we illustrate these new settings in certain idealised contexts and in Section 4.6 we examine properties of this implicit separation measure. This enable us to make a direct link between Bayes Factor selection and more conventional separation based clustering methods; see Chipman et al. (2001), Gordon (1999) and Hastie et al. (2001). We demonstrate that a partition, $\mathcal{C}_1$, is preferred to another, $\mathcal{C}_2$, (which is identical to $\mathcal{C}_1$ except that two particular clusters in $\mathcal{C}_1$ are combined into one cluster in $\mathcal{C}_2$) if, and only if, the sufficient statistics of the two clusters in $\mathcal{C}_1$ are different enough from one another in a certain, very natural, sense. Finally, in Section 4.7 we present the results of the clustering using the setting of the hyperparameters that we propose in this chapter on a real dataset.

In a careful study of model selection over large spaces of linear models, Chipman et al. (2001) argue that hyperparameters should be set to make prior assumptions minimally influential. However, when selecting across a space of partition models, we argue that such a strategy is futile and all settings of hyperparameters have a different and strong effect on model selection over this domain. We demonstrate that it is simple to elicit values of hyperparameters within the class of proportional models so that they calibrate to pre-posterior predictions associated with the model space in a given context. It is also possible to demonstrate both analytically and numerically that these settings are robust to moderate misspecification.

4.1 A Simple Likelihood Ratio

4.1.1 Conjugate Bayesian Estimation of Profiles

Using the same notation introduced in Section 3.3, the Bayes factor associated with this model can then be calculated from its marginal likelihood, $L(y)$. Note that we omit k here for simplification. Thus

$$L(y) = \left(\frac{1}{\pi}\right)^{nr/2} \frac{b^a}{(b^*)^{a^*}} \frac{|V^*|^{1/2}}{|V|^{1/2}} \frac{\Gamma(a^*)}{\Gamma(a)},$$

which can be written as

$$2\log L(y) = 2l(y) = K(V,a,b,n) - 2a^* \log(b + \gamma/2),$$

where

$$K(V,a,b,n) = 2\log\left(\left(\frac{1}{\pi}\right)^{nr/2} b^a \frac{|V^*|^{1/2}}{|V|^{1/2}} \frac{\Gamma(a^*)}{\Gamma(a)}\right).$$

Because $X'X$ is full rank, the maximum likelihood estimate $\widehat{\boldsymbol{\beta}}$ of the mean vector $\boldsymbol{\beta}$ is uniquely defined and is given by

$$\widehat{\boldsymbol{\beta}} = (X'X)^{-1}X'y = n^{-1}(B'B)^{-1}B'D\mathbf{1}$$

with $\mathbf{1}$ an r-vector of ones. This is obtained by maximising the likelihood, noting that $y \sim N(X\boldsymbol{\beta}, \sigma^2 I)$. Also,

$$\gamma = y'y + m'V^{-1}m - (m^*)'(V^*)^{-1}m^* = rn\widehat{\sigma}^2 + \widehat{\boldsymbol{\beta}}' \left(V + (X'X)^{-1}\right)^{-1} \widehat{\boldsymbol{\beta}},$$

where $\widehat{\sigma}^2 = (y - X\widehat{\boldsymbol{\beta}})'(y - X\widehat{\boldsymbol{\beta}})/rn$ is the maximum likelihood estimate of σ^2. This result is pointed out by O'Hagan and Forster (2004), pp. 310, and it is obtained by noting that

$$\left(V^{-1} + (X'X)\right)^{-1} V^{-1} = (X'X)^{-1} \left(V + (X'X)^{-1}\right)^{-1}$$

and

$$\begin{aligned}\left(V^{-1} + (X'X)\right)^{-1} &= (X'X)^{-1} - (X'X)^{-1} \left(V + (X'X)^{-1}\right)^{-1} (X'X)^{-1} \\ &= V - V \left(V + (X'X)^{-1}\right)^{-1} V,\end{aligned}$$

as pointed out by Zhang (1999), pp. 93.

4.1.2 Comparing Two Regression Profiles

Define the observation vector $y = (y^{(1)}, y^{(2)})'$, where $y^{(j)} = (y_1^{(j)}, \ldots, y_{n_j}^{(j)})'$. The components $\{y_i^{(j)} : 1 \leq i \leq n_j, j = 1, 2\}$ are profiles of a fixed length $r \geq p$ with

$$y_i^{(j)} = B\boldsymbol{\beta}_j + \varepsilon_i^{(j)}$$

where $\varepsilon_i^{(j)} \sim N(\mathbf{0}, \sigma_j^2 I)$, $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2)'$ and $\coprod_{i,j} \varepsilon_i^{(j)} | \boldsymbol{\beta}$ with $\coprod$ representing independence between random variables. Thus, the profile vectors containing the longitudinal data on each unit, $y_i^{(j)}$, each follow the same linear model with a design matrix B of rank p. $y^{(j)}$ is a vector of length rn_j, $j = 1, 2$.

Let model M_s assume that the vectors $y^{(1)}$, $y^{(2)}$ are independent and also assume that $(\boldsymbol{\beta}_1, \sigma_1^2) \coprod (\boldsymbol{\beta}_2, \sigma_2^2)$, where $(\boldsymbol{\beta}_j, \sigma_j^2)$ is assumed to have the prior density $NIG(\mathbf{0}, V_j, a_j, b_j)$. Then, with the obvious extension of the notation given above, its log marginal likelihood $l_s(y)$ is given by

$$2l_s(y) = \sum_{j=1,2} K(V_j, a_j, b_j, n_j) - 2 \sum_{j=1,2} a_j^* \log \left(b_j + \gamma_s^{(j)}/2\right),$$

where

$$\gamma_s^{(j)} = rn_j\widehat{\sigma}_j^2 + \widehat{\boldsymbol{\beta}}_j' \left(V_j + n_j^{-1}(B'B)^{-1}\right)^{-1}\widehat{\boldsymbol{\beta}}_j$$

and

$$\widehat{\boldsymbol{\beta}}_j = n_j^{-1}(B'B)^{-1}B'D_j\mathbf{1}.$$

Now, compare the model M_s with a model M_t that assumes the vectors $y^{(1)}$, $y^{(2)}$ share the same parameter values. Under M_t, $\boldsymbol{\beta}_1 = \boldsymbol{\beta}_2$ and $\sigma_1^2 = \sigma_2^2$ where $(\boldsymbol{\beta}_1, \sigma_1^2)$ has the prior density $NIG(\mathbf{0}, \overline{V}, \overline{a}, \overline{b})$. So the log marginal likelihood $l_t(y)$ of this model satisfies

$$2l_t(y) = K(\overline{V}, \overline{a}, \overline{b}, n_{12}) - 2\overline{a}^* \log(\overline{b} + \gamma_t/2),$$

where $n_{12} = n_1 + n_2$ and

$$\gamma_t = rn_{12}\widehat{\sigma}^2 + \widehat{\boldsymbol{\beta}}' \left(\overline{V} + n_{12}^{-1}(B'B)^{-1}\right)^{-1}\widehat{\boldsymbol{\beta}}.$$

$\widehat{\sigma}^2$ is the standard maximum likelihood estimate of the variance of the combined sample,

$$\widehat{\boldsymbol{\beta}} = n_{12}^{-1}\sum_{j=1}^{2} n_j\widehat{\boldsymbol{\beta}}_j$$

and

$$rn_{12}\widehat{\sigma}^2 = \sum_{j=1,2} n_j r\widehat{\sigma}_j^2 + \frac{n_1 n_2}{n_{12}}(\widehat{\boldsymbol{\beta}}_1 - \widehat{\boldsymbol{\beta}}_2)'B'B(\widehat{\boldsymbol{\beta}}_1 - \widehat{\boldsymbol{\beta}}_2).$$

We note that both models have a marginal likelihood which is a function of their hyperparameters and the four familiar statistics $\{\widehat{\boldsymbol{\beta}}_j, \widehat{\sigma}_j^2 : j = 1, 2\}$.

4.1.3 Bayesian MAP Model Selection

One popular method is Bayesian Maximum A Posteriori or MAP model selection: see e.g. Bernardo and Smith (1994). This simply chooses the model with the highest posterior probability. If the prior log odds for model M_t against model M_s are κ, then the distinct or separate vector model M_s is preferred to the combined vector model M_t when the posterior log odds are greater than κ. This occurs when $l_s(y) - l_t(y) > \kappa$ or, equivalently,

$$\Phi = \log(\overline{u} + \widehat{\boldsymbol{\beta}}_1' C_{11}\widehat{\boldsymbol{\beta}}_1 - 2\widehat{\boldsymbol{\beta}}_1' C_{12}\widehat{\boldsymbol{\beta}}_2 + \widehat{\boldsymbol{\beta}}_2' C_{22}\widehat{\boldsymbol{\beta}}_2) - \sum_{j=1,2} \rho_j \log(u_j + \widehat{\boldsymbol{\beta}}_j' A_j \widehat{\boldsymbol{\beta}}_j) > \kappa',$$

where

$$\begin{array}{rclrcl} A & = & (\overline{V} + \frac{1}{n_{12}}(B'B)^{-1})^{-1}, & A_j & = & (V_j + \frac{1}{n_j}(B'B)^{-1})^{-1}, \\ C_{11} & = & \frac{n_1}{n_{12}}\left(\frac{n_1}{n_{12}}A + n_2 B'B\right), & \overline{u} & = & 2\overline{b} + r(n_1\widehat{\sigma}_1^2 + n_2\widehat{\sigma}_2^2), \\ C_{22} & = & \frac{n_2}{n_{12}}\left(\frac{n_2}{n_{12}}A + n_1 B'B\right), & u_j & = & 2b_j + rn_j\widehat{\sigma}_j^2, \\ C_{12} & = & \frac{n_1 n_2}{n_{12}}\left(\frac{1}{n_{12}}A + B'B\right), & \rho_j & = & a_j^*/\overline{a}^*. \end{array}$$

Note that the threshold,

$$\kappa' = \left[2\kappa - \sum_{j=1,2} K(V_j, a_j, b_j, n_j) + K(\overline{V}, \overline{a}, \overline{b}, n_{12}) + 2(\overline{a} - a_1 - a_2)\log 2\right] / 2\overline{a}^*,$$

depends on the data only through (n_1, n_2) and the specified prior log odds κ between the two models. In principle, the prior parameter κ and hence κ' can take any value, so the behaviour of this selection algorithm is formally explained simply through the geometry of the contours of the function Φ. For the remainder of the chapter we will use the condensed notation $K(n)$ to denote $K(V, a, b, n)$.

The function Φ can be further simplified by introducing some new notation. We set $\mathbf{w}_j$ so that

$$\mathbf{w}_j' A_j \mathbf{w}_j = \mathbf{w}_j'(V_j + n_j^{-1}(B'B)^{-1})^{-1}\mathbf{w}_j = 1.$$

Further, we define $\overline{z}_j = ||Q_j\widehat{\boldsymbol{\beta}}_j||$, where Q_j is any matrix satisfying $Q_j'Q_j = A_j$ and let

$\lambda_1 = \mathbf{w}_1' C_{11} \mathbf{w}_1,\ \lambda_{12} = \mathbf{w}_1' C_{12} \mathbf{w}_2,\ \lambda_2 = \mathbf{w}_2' C_{22} \mathbf{w}_2$. Therefore, $C_{jj} = A_j \lambda_j$ and

$$\begin{aligned}
\widehat{\boldsymbol{\beta}}_j' C_{jj} \widehat{\boldsymbol{\beta}}_j &= \lambda_j \widehat{\boldsymbol{\beta}}_j' A_j \widehat{\boldsymbol{\beta}}_j = \lambda_j \widehat{\boldsymbol{\beta}}_j' Q_j' Q_j \widehat{\boldsymbol{\beta}}_j = \lambda_j ||Q_j \widehat{\boldsymbol{\beta}}_j||^2 = \lambda_j \overline{z}_j^2, \\
\widehat{\boldsymbol{\beta}}_1' C_{12} \widehat{\boldsymbol{\beta}}_2 &= \lambda_{12}^0 \widehat{\boldsymbol{\beta}}_1' Q_1' Q_2 \widehat{\boldsymbol{\beta}}_2 = \lambda_{12} \overline{z}_1 \overline{z}_2, \\
\widehat{\boldsymbol{\beta}}_j' A_j \widehat{\boldsymbol{\beta}}_j &= \widehat{\boldsymbol{\beta}}_j' Q_j' Q_j \widehat{\boldsymbol{\beta}}_j = ||Q_j \widehat{\boldsymbol{\beta}}_j||^2 = \overline{z}_j^2,
\end{aligned}$$

for values of λ_{12}^0 and λ_{12} given in the next section. We then prefer M_s to M_t if, and only if,

$$\Phi = \log(\overline{u} + \lambda_1 \overline{z}_1^2 - 2\lambda_{12} \overline{z}_1 \overline{z}_2 + \lambda_2 \overline{z}_2^2) - \sum_{j=1,2} \rho_j \log(u_j + \overline{z}_j^2) > \kappa'.$$

Note that $(\overline{z}_1, \overline{z}_2)$ are the distances of the two profiles from zero, each scaled by a factor reflecting the deviation from zero we expected a priori under the separating model M_s. The statistics u_j depend on the data only through $\widehat{\sigma}_j^2$. The statistic $\overline{u}$ is a linear function of u_1 and u_2 and so is a linear function of the two corresponding sums of squares, and λ_j corresponds to the distance from zero expected for the profile $\widehat{\boldsymbol{\beta}}_j$ under M_t relative to that expected under M_s.

4.1.4 Using g-priors for Conjugate Clustering

Employing a general form of covariance matrix V demands that the space of prior hyperparameters is very large. For simplicity, transparency and to ensure invariance to linear transformations of bases various authors (Chipman et al., 2001; Fernández et al., 2001; Smith and Kohn, 1996; Zellner, 1971) have advocated the use of g-priors for prior covariance matrices.

In the given context, these priors would set $\overline{V}^{-1} = \overline{g} B'B,\ V_1^{-1} = g_1 B'B$, $V_2^{-1} = g_2 B'B$ for specified constants $(\overline{g}, g_1, g_2)$ associated with the combined cluster $\overline{c}$ and the smaller clusters c_1 and c_2. Here, g is a measure of noise-to-signal so, in particular, the larger the value of g the greater the shrinkage of the expected posterior profile towards zero. Let $\mathbf{z}_j = (z_1^{(j)}, z_2^{(j)}, \ldots, z_p^{(j)})'$ with $j = 1, 2$ where

$$\mathbf{z}_j = \sqrt{\frac{n_j g_j}{g_j + n_j}} B\widehat{\boldsymbol{\beta}}_j \quad \text{such that} \quad \sqrt{\frac{n_j g_j}{g_j + n_j}} B = Q_j$$

and it holds that $A_j = Q_j' Q_j$ and $\overline{z}_j = ||\mathbf{z}_j||$. The vectors $\mathbf{z}_1$ and $\mathbf{z}_2$ are the posterior expected profiles of the two clusters, normalised by their posterior variance. It can be shown that the parameters of Φ then simplify. Since

$$\begin{aligned} B'B &= \left((B'B)^{-1} \right)^{-1} \\ &= \left(\frac{g_j n_j (g_j + n_j)}{g_j n_j (g_j + n_j)} (B'B)^{-1} \right)^{-1} \\ &= \frac{(g_j + n_j)}{g_j n_j} \left(\frac{1}{g_j} (B'B)^{-1} + \frac{1}{n_j} (B'B)^{-1} \right)^{-1} \\ &= \frac{(g_j + n_j)}{g_j n_j} A_j \end{aligned}$$

then $\mathbf{w}_j' B' B \mathbf{w}_j = (g_j + n_j)/(g_j n_j)$. It follows, similarly, that

$$A = \frac{(g_j + n_j)\overline{g} n_{12}}{(n_{12} + \overline{g}) g_j n_j} A_j.$$

Also, note that

$$C_{12} = \frac{n_1 n_2}{\overline{g} + n_{12}} B'B. \tag{4.1}$$

Since $\lambda_j = \mathbf{w}_j' C_{jj} \mathbf{w}_j$, then,

$$\lambda_1 = \frac{(\overline{g} + n_2)(g_1 + n_1)}{(\overline{g} + n_{12}) g_1} \quad \text{and} \quad \lambda_2 = \frac{(\overline{g} + n_1)(g_2 + n_2)}{(\overline{g} + n_{12}) g_2}.$$

Similarly,

$$\begin{aligned} \widehat{\boldsymbol{\beta}}_1' C_{12} \widehat{\boldsymbol{\beta}}_2 &= \frac{n_1 n_2}{\overline{g} + n_{12}} \widehat{\boldsymbol{\beta}}_1' B' B \widehat{\boldsymbol{\beta}}_2 \\ &= \frac{n_1 n_2}{\overline{g} + n_{12}} \sqrt{\frac{g_1 + n_1}{g_1 n_1}} \sqrt{\frac{g_2 + n_2}{g_2 n_2}} \widehat{\boldsymbol{\beta}}_1' Q_1' Q_2 \widehat{\boldsymbol{\beta}}_2 \\ &= \lambda_{12} \overline{z}_1 \overline{z}_2 \end{aligned}$$

and therefore

$$\lambda_{12} = \lambda_{12}^0 \cos(\theta[\mathbf{z}_1, \mathbf{z}_2]),$$

where

$$\lambda_{12}^0 = \sqrt{\frac{n_1 n_2 (g_1 + n_1)(g_2 + n_2)}{(\overline{g} + n_{12})^2 g_1 g_2}}$$

because, given two vectors a and b, it holds that their dot product is given by

$$a'b = ||a||\,||b|| \cos\theta(a, b).$$

The parameter $\theta[\mathbf{z}_1, \mathbf{z}_2]$ is the angle between vectors $(\mathbf{z}_1, \mathbf{z}_2)$ on a plane through zero containing the two rays $(\mathbf{0}, \mathbf{z}_1), (\mathbf{0}, \mathbf{z}_2)$. So, this is a measure of the difference in the scaled shapes of the two profiles.

A common choice of prior for model selection would be to set $g_1 = g_2 = \overline{g} = g$. This assumes that knowing the size, n, of a cluster would not affect the strength of our prior beliefs about the mean profile of a unit in that cluster. The prior information about each unit conditional on σ^2 is implicitly assumed to be based on exactly the same sources as other units in its cluster. We call this the *dependence* setting. Note that in this case

$$\lambda_1 = \lambda_2 = 1 + \frac{n_1 n_2}{g(g + n_1 + n2)}$$

and

$$\frac{n_1 n_2}{g + n_1 + n2} < n_j$$

for $j = 1, 2$. Therefore for $j = 1, 2$ it holds that

$$1 < \lambda_j < 1 + \frac{\min\{n_1, n_2\}}{g}.$$

An alternative protocol is sometimes applicable to, for example, gene expression data, where learning that a cluster of genes is large increases the chance that the cluster profile is close to zero: i.e. the cluster is not involved in regulation. A prior structure consistent with these beliefs – here called the *independence* model – assumes that the sources of information about the prior density of each single gene in a cluster are independent and of equal strength conditional on σ^2. This implies that $g_j = \breve{g} n_j$ and $\overline{g} = \breve{g} n_{12}$ so that

$$\lambda_1 = 1 + \frac{n_2}{\breve{g} n_{12}}, \quad \lambda_2 = 1 + \frac{n_1}{\breve{g} n_{12}}, \quad \lambda_{12}^0 = \frac{1}{\breve{g}} \sqrt{\frac{n_1 n_2}{n_{12}^2}}.$$

4.2 Using Bayes Factors to Select Between Many Partition Models

4.2.1 A Typical Example of Conjugate Bayesian Model Selection

MAP model selection is used routinely in many tree and cluster models. In order to show how the performance characteristics of such selection can be linked to the study of the function Φ, we next review Bayesian model selection as it applies to the clustering algorithm in Heard et al. (2006). There, thousands of longitudinal profiles of genes are collected into a partition $\mathcal{C} \in \mathscr{C}$ whose sets are the clusters $c \in \mathcal{C}$. $\mathscr{C}$ is the set of all partitions. Microarrays measure the level of expression (a real number) for all of its genes over a sequence of times. In our running example there are 13 time points (Edwards et al., 2006).

The vector of profiles of the logged gene expressions, y_c, within each cluster are assumed to be exchangeable. $y^{(c)} = B\boldsymbol{\beta}_c + \varepsilon_c$, $\varepsilon_c \sim N(\mathbf{0}, \sigma_c^2 I_{rn_c})$, $\coprod \varepsilon_c | \boldsymbol{\beta}$, $\boldsymbol{\beta} =$ vec$(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_N)$, where y_c is a vector of length rn_c, where r is the length of the profile, n_c is the number of gene profiles in cluster c and N the number of sets in the partition $\mathcal{C}$. Using analogous notation to that in Section 4.1, we have that

$$y_i^{(c)} = B\boldsymbol{\beta}_c + \varepsilon_i^{(c)}$$

for $1 \leq i \leq n_c$, where $\varepsilon_i^{(c)} \sim N(\mathbf{0}, \sigma_c^2 I_r)$ and $\coprod_{i,c} \varepsilon_i^{(c)} | \boldsymbol{\beta}$, $c \in \mathcal{C}$. The design matrix B

is customised to the context. Thus a spline basis is employed in Heard et al. (2006), a Fourier basis is used in Anderson et al. (2006) and Edwards et al. (2006) and a wavelet basis is used in Ray and Mallick (2006). The profile vectors $\boldsymbol{\beta}_c$ and variances σ_c^2 of the different clusters $c \in \mathcal{C}$ are all assumed to be mutually independent of each other and to follow the conjugate distributions given in Section 4.1. So, in particular, each cluster has an associated multivariate t-distribution with log marginal likelihood $l_c(y)$. Furthermore, because of the assumed independencies between clusters in a given partition, the log marginal likelihood $l_{\mathcal{C}}(y)$ of any partition $\mathcal{C}$ is simply the sum of the marginal likelihoods of its components:

$$l_{\mathcal{C}}(y) = \sum_{c \in \mathcal{C}} l_c(y).$$

The log marginal likelihood of any partition can therefore be written down explicitly. Under MAP selection an optimal partition $\mathcal{C}^* \in \mathscr{C}$ will be any partition such that, for all $\mathcal{C} \in \mathscr{C}$,

$$l_{\mathcal{C}^*}(y) + \log \pi(\mathcal{C}^*) \geq l_{\mathcal{C}}(y) + \log \pi(\mathcal{C}),$$

where $\pi(\mathcal{C})$ is our prior probability that partition $\mathcal{C}$ generated the data.

4.2.2 Exchangeability and Cohesions

To preserve certain exchangeability properties for partition models, the following four assumptions are commonly made (Barry and Hartigan, 1992; Quintana and Iglesias, 2003).

1. The prior parameters (V_c, a_c, b_c) of cluster $c \in \mathcal{C}$ depend on c but not $\mathcal{C}$.

2. The parameters (V_c, a_c, b_c) are a function of c only through n_c, the number of genes in c.

3. The probabilities $\{\pi(\mathcal{C}) : \mathcal{C} \in \mathscr{C}\}$ satisfy

$$\pi(\mathcal{C}) \propto \prod_{c \in \mathcal{C}} \pi_c$$

   where the proportionality constant is the sum of all these products of *cohesions*, π_c, over $\mathcal{C} \in \mathscr{C}$.

4. The probability π_c is allowed to depend on c only through its cardinality n_c.

We call prior beliefs for clustering *balanced* if they are consistent with these four assumptions. Previous studies (Anderson et al., 2006; Edwards et al., 2006; Heard et al., 2006) make a stronger assumption than (b) that (V_c, a_c, b_c) are not a function of n_c. The default choice of Heard et al. (2006) is balanced and sets cohesions so that $\pi_c = (n_c!)^{-1}$. The appropriate choice of parametric form of a family of balanced priors - which determines the prior distribution of cardinalities of the vector of clusters in a given partition - is clearly highly dependent on the science and purpose underlying the statistical analysis. The default setting mentioned above tends to favour partitions with clusters of similar sizes, whilst Dirichlet priors tend to do the reverse. However, although this prior obviously influences which partition is optimal, all the instabilities we address in this chapter apply whatever the choice of partition prior (see Section 4.5). It is therefore possible to separate modelling issues associated with the three hyperparameters of each cluster from appropriate choices of balanced priors: an important issue, but beyond the scope of this chapter. Henceforth, when no confusion shall arise we will write $(V_{c_j}, a_{c_j}, b_{c_j}, n_{c_j})$ as $(V_j, a_j, b_j, n_j),\ j = 1, 2.$

We note that partition priors have been criticised because consistency is not preserved if exchangeability of units is demanded after deletion (McCullagh and Yang, 2006). However, it is easily deduced that Crowley process priors, which do have this consistency property, lead to the same separation issues of hyperparameters from choice of partition prior parameter. See the discussion of the prior modelling also in Section 3.3.3.

4.2.3 Model Search

When the number of units partitioned is large (for example in Anderson et al. (2006) we clustered over $22,000$ genes), the partition space is huge. So, even being able to calculate the scores of single cluster partitions quickly is not enough to ensure that the scores of all the partitions in the vast partition space $\mathscr{C}$ can be evaluated. In practice it is therefore often necessary to use an appropriate search algorithm to perform this optimisation task on a sensible subset of such partitions.

One useful feature of using $l_{\mathcal{C}}(y)$ for selection is that the difference between the scores of two partitions identical outside a given set $\overline{c}$ will depend only on their relative scores over $\overline{c}$. We call partitions $\mathcal{C}^+$ and $\mathcal{C}^-$ *adjacent* if the two partitions differ only on a set $\overline{c} \in \mathcal{C}^+$ where $\overline{c} = c_1 \cup c_2$ with $c_1 \cap c_2 = \emptyset$, $c_1, c_2 \in \mathcal{C}^-$ so that $\{c_1, c_2\}$ partition $\overline{c}$. Then

$$l_{\mathcal{C}^-}(y) = l_{\mathcal{C}^+}(y) - \Omega[\mathcal{C}^-, \mathcal{C}^+] - \log \pi(\mathcal{C}^-) + \log \pi(\mathcal{C}^+),$$

where

$$\Omega[\mathcal{C}^-, \mathcal{C}^+] = l_{c_1}(y) + l_{c_2}(y) - l_{\overline{c}}(y)$$

and $\pi(\mathcal{C}^-)$, $\pi(\mathcal{C}^+)$ are the prior probabilities of $\mathcal{C}^-$ and $\mathcal{C}^+$ respectively. The comparison of adjacent partitions when using balanced priors is therefore especially straightforward and is utilised in many search algorithms used in this context. For example, the improvement presented by $\mathcal{C}^-$ (the model assuming the genes in $\overline{c}$ are in two different groups c_1 and c_2) over $\mathcal{C}^+$ (the model assuming all genes in c are exchangeable) is measured by $\Phi - \kappa'(n_1, n_2)$ where

$$\kappa' = \frac{2\{\log \pi_{c_1}(n_1) + \log \pi_{c_2}(n_2) - \log \pi_{\overline{c}} n_{12}\} + K(n_1) + K(n_2) - K(n_{12})}{2\overline{a}^*}.$$

Note that κ' is a function of the two partitions only via a symmetry of the cardinalities (n_1, n_2) of the two potentially combined clusters. $\mathcal{C}^-$ has a higher posterior probability than $\mathcal{C}^+$ if, and only if, $\Phi - \kappa'(n_1, n_2) > 0$. Thus, any search algorithm that moves only between adjacent partitions, either merging or splitting two clusters depending on whether the function Φ is large enough to instigate a split relative to a splitting penalty κ' (a function depending on cluster cardinalities within the relevant partitions but not on the data), is especially fast.

The most popular technique that uses adjacent moves to search a partition space is a greedy search algorithm called agglomerative hierarchical clustering (AHC) (Heard et al., 2006); a type of forward selection. This starts with each of the N gene profiles in N separate clusters with fixed values of the hyperparameters. A sequence of new partitions is then obtained by sequentially merging two clusters, thus decreasing the number of clusters by one. The two clusters chosen to be combined are the ones that increase the score (here the marginal likelihood of the partition) by the most (or reduce it by the least). Clusters are combined in this way until the trivial partition is reached, with one cluster containing all N genes. We have now calculated the marginal likelihood for a selection of N promising partitions containing 1 to N clusters. Finally we choose the partition in this sequence with the highest score: i.e. with the highest posterior probability over the partitions searched. Examples of other more elaborate search algorithms also using adjacent moves either in conjunction with a deterministic or stochastic search are given in Anderson et al. (2006) and, in a slightly different context, Chipman et al. (1998, 2002).

For the remainder of the chapter we will study the geometry of $\Phi(\widehat{\sigma}_1^2, \widehat{\sigma}_2^2, \widehat{\boldsymbol{\beta}}_1, \widehat{\boldsymbol{\beta}}_2)$

as a function of the sufficient statistics $\{\widehat{\boldsymbol{\beta}}_j, \widehat{\sigma}_j^2 : j = 1, 2\}$ in order to understand the behaviour of MAP model selection methods using AHC. However, we note that the problems we identify with the consequent model selection also apply to more sophisticated local search algorithms that allow clusters to be split as well as combined.

4.3 Bayesian Model Selection over Partitions

4.3.1 Three Weaknesses of Uncalibrated Bayesian Model Selection

Uncalibrated model selection based on Bayes factors, like the one discussed above, can fail for a number of reasons. Firstly, we have noted that the Bayes factor acts as an implicit real-valued score function over the different cluster partitions. There is thus an inevitable implicit trade-off between the closeness of the variances of the two potentially combined clusters and the closeness of their mean profiles. For this and other reasons, it is now well recognised that the chosen values of prior hyperparameters have a marked effect on the characteristics of Bayesian model selection, and their influence on inference cannot be expected to automatically fade away as the sample size increases. In fact, in Section 4.6 we show how influential the selection of these hyperparameters is not only on the scale, but also on the *nature* of discrepancies that drive the selection. So there is great advantage to choose (whenever possible) prior values for hyperparameters not only so that the features of the selection algorithm match contextual knowledge, but also so that selection characteristics of the method are plausible a priori. As we discuss below, if this is not done, the properties of the induced selection algorithm can be absurd.

Secondly, as emphasised in Denison et al. (2002), the function Φ is not translation invariant. We demonstrate below that the optimal choice of partition is typically *critically* dependent on where we choose to set the prior mean vector of the profile –

here we select zero. Hence, unless there actually is a natural “preferred point”, as in Edwards et al. (2006), we cannot recommend the use of these methods. Henceforth we will assume, as is often the case in practice, that such a preferred point exists.

Thirdly, the assumption of conjugacy is usually an expedience and there are at least two questionable consequences. First, the tails of the conjugate marginal likelihoods are inverse polynomials. Although this helpfully limits the number of small clusters, it also finds "optimal" partitions that often contain clusters that include outlying profiles. Second, these conjugate models imply that the prior mean and variance of the cluster profiles are quite highly dependent: for a careful discussion of this see O'Hagan and Forster (2004). One implication is that clusters observed to have an estimated profile very different from zero – our preferred point – will be allocated a high prior variance: a property which, if not recognised and adjusted for, can distort any search algorithm in ways discussed below.

4.3.2 Selection as a Function of the Magnitude of the Mean Profile

From the comments above we might suspect model selection to be disrupted by outliers. Consider the effects of increasing the magnitude of a cluster profile away from zero whilst holding all other statistics fixed. Fix $\overline{z}_2$, $\mathbf{w}_j$, $\widehat{\sigma}_j^2$ and n_j for $j = 1, 2.$. Then, provided $0 < \rho_1 < 1$,

$$\lim_{|z_1| \to \infty} \Phi(\overline{z}_1, \overline{z}_2) = \infty.$$

Thus, whatever the values of prior hyperparameters, as we increase the magnitude $\overline{z}_1$ of the profile of the first cluster c_1 (provided $\overline{z}_1$ is large enough) our model will prefer to keep clusters c_1 and c_2 separate, as we might hope.

However, if *two* outlying clusters $(c_1,\ c_2)$ both have profiles $(\overline{z}_1, \overline{z}_2)$ far from zero then model selection can start to display strange properties. If $\overline{z}_2 = l\overline{z}_1^k$, l is fixed and $|\overline{z}_1| \to +\infty$, then

$$\begin{aligned}\Phi(\overline{z}_1, \overline{z}_2) &\to \log|\overline{z}_1^2| - \sum_{j=1,2} \rho_j \log|\overline{z}_j^2| \\ &\to 2(1 - \rho_1 - k\rho_2)\log|\overline{z}_1|\end{aligned}$$

diverges to $-\infty$ if $\rho_1 + k\rho_2 > 1$ and diverges to $+\infty$ if $\rho_1 + k\rho_2 < 1$. For example, Heard et al. (2006) recommend setting $\overline{a} = a_1 = a_2$. This implies that

$$\rho_1 + \rho_2 = \frac{4\overline{a} + rn_{12}}{2\overline{a} + rn_{12}} = 1 + \frac{2\overline{a}}{2\overline{a} + rn_{12}} > 1,$$

where n_{12} is the total number of observations associated with the two groups. In this case, by simultaneously increasing the magnitude of the two cluster profiles by the same amount $z_1 = z_2$, we will eventually reach a magnitude where two clusters are combined *irrespective* of how different the shapes of those clusters are: definitely not what we want to happen. This occurs because, when combined into one cluster, these two outliers become one outlier and, a priori, one outlier is assumed more probable than two.

Thus two clusters whose expression profiles are far from zero – and hence possibly biologically significant – will be combined in preference to any other pair: even clusters whose statistics are identical! The reason this unfortunate property is relatively rare in practice is that studies such as Heard et al. (2006) happen to suggest the use of a small value of $\overline{a}$. Therefore profiles have to be very different from zero before this phenomenon can be realised. However, this still happens even at the recommended settings of the parameters. In Fig. 4.1 we can see that genes with completely different profiles have been attracted into a cluster under an optimal MAP partition found under an AHC search. Note that when this phenomenon occurs early in an AHC search, the combined cluster can largely cancel out and then has the signature of the large variance cluster: something we term a junk cluster in Anderson et al. (2006). When such a cluster is formed under AHC it tends to act as an attractor to yet more disparate and biologically interesting clusters resulting in a cluster like the one depicted in Fig. 4.2.

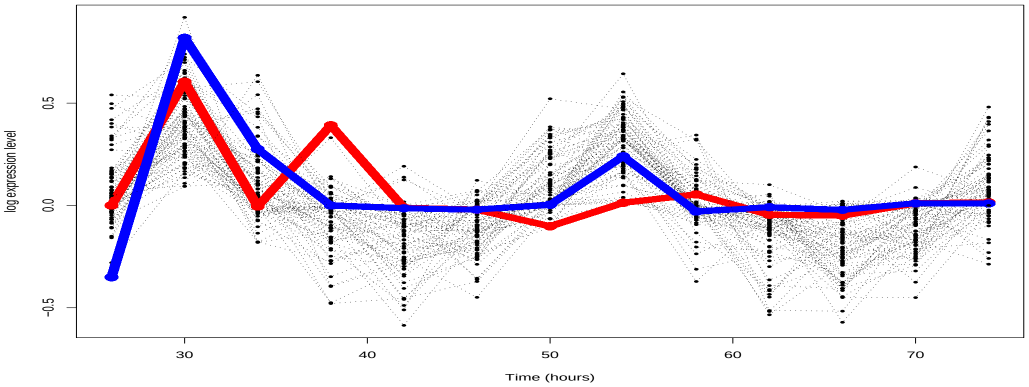


Figure 4.1: A cluster of 81 gene expression profiles from a early stage of the clustering performed in Edwards et al. (2006) using the default hyperparameter settings. The two highlighted genes are clearly outliers that do not belong in this cluster. This is a result of the AHC and the default settings.

If we differ from Heard et al. (2006) and choose a prior with $\rho_1 + \rho_2 < 1$, then $\Phi(\overline{z}_1, \overline{z}_2) \rightarrow \infty$ as $|\overline{z}_1| \rightarrow \infty$. This gives rise to an even more problematic property. Whatever our settings of prior hyperparameters, two profiles sufficiently far from zero will always be put in separate clusters even when $\widehat{\boldsymbol{\beta}}_1 = \widehat{\boldsymbol{\beta}}_2$, $\widehat{\sigma}_1^2 = \widehat{\sigma}_2^2$ and $n_1 = n_2$, i.e., even when these two clusters are identical in all respects! Note that the position of the prior mean (here the zero setting) is central to determining which profiles are outlying in the sense above.

The *only* case when the associated limit stays finite is when $\rho_1 + \rho_2 = 1$. Unless we set the hyperparameters to ensure this, on observing profiles far from zero the implications of the prior are unlikely to be faithful to contextual beliefs. Therefore, the Bayesian clustering algorithm will be prone to perform inappropriately, and combine profiles it was never meant to.

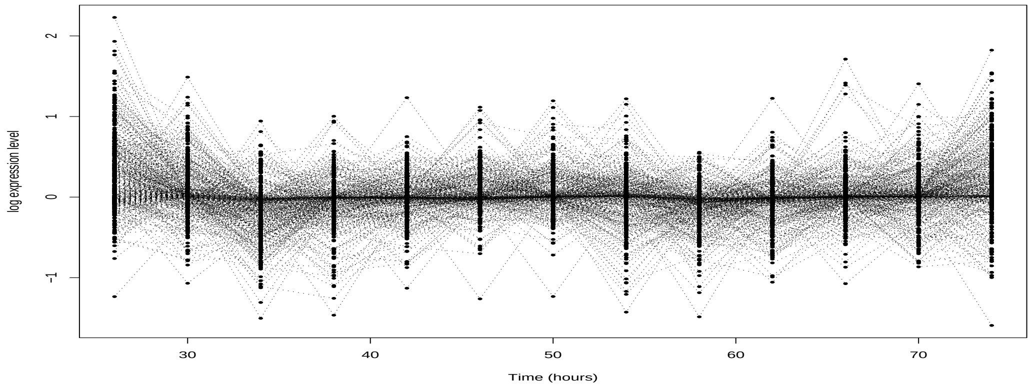


Figure 4.2: A cluster of 453 gene expression profiles from the same partition as Fig. 4.1. This so-called junk cluster is a by-product of AHC and contains a broad variety of profile shapes. Note that in this context log expressions outside $[-0.5, 0.5]$ are considered to be potentially of biological interest.

4.3.3 Models with $\rho_1 + \rho_2 = 1$

By setting hyperparameters so that $\rho_1 + \rho_2 = 1$ the characteristics of the resulting merging criterion are much more compelling. The demand that $\rho_1 + \rho_2 = 1$ is satisfied provided that the hyperparameters (a_1, a_2) of two clusters in a partition and the hyperparameter $\overline{a}$ of the combined cluster in an adjacent partition satisfy

$$a_1 + a_2 = \overline{a}.$$

For balanced priors, this implies that we set the corresponding hyperparameter $a_c = \breve{a} n_c$, where n_c is the number of profiles in c rather than require a_c to be independent of cluster size as is the case in Heard et al. (2006). Our suggestion would make the prior coefficient of variation of the precision of a cluster proportional to $n_c^{1/2}$. For example, in the context of gene clustering this would mean that 'genuine' clusters containing large numbers of gene profiles are expected to have smaller associated coefficients of variation in their precision. Thus we are a priori less certain about the value of the variance of big clusters: not an unreasonable assumption in this context. Note that under this setting

$\rho_j = n_j n_{12}^{-1}$, $j = 1, 2$.

4.4 Bayes Factors and Measures of Separation

Under balanced priors, each cluster c in a partition has a set of sufficient statistics $\mathbf{x}(c) = (n_c^{-1}\widehat{\boldsymbol{\beta}}_c, \widehat{\sigma}_c^2, n_c)$. Let $\kappa'' = \min \Phi$. Then, it is common (Denison et al., 2002) to interpret the function $\Delta = \Phi - \kappa''$ as a measure of the separation between the combined clusters c_1 and c_2 in two adjacent partitions that are identical except on $c_1 \cup c_2$. We have seen above that this interpretation may well not be correct. Whenever $\rho_1 + \rho_2 \neq 1$, two clusters c_1 and c_2 with identical sufficient statistics can be arbitrarily more separated – i.e. have an arbitrarily higher value of Δ – than two clusters that have very different sufficient statistics. In particular, under *any* search over the partition space, it is quite possible for two clusters with widely differing profiles to be combined in preference to two clusters with identical $\{\widehat{\boldsymbol{\beta}}_j, \widehat{\sigma}_j^2 : j = 1, 2\}$.

Although this phenomenon is much more dramatic when $\rho_1 + \rho_2 \neq 1$, the problem can still remain even when hyperparameters are set so as to ensure $\rho_1 + \rho_2 = 1$. In this section we investigate to what extent, with appropriate parameter settings, $\Delta(\mathbf{x}(c_1), \mathbf{x}(c_2))$ can be interpreted as a measure of separation between the clusters c_1 and c_2.

If $\Psi(\mathbf{x}(c_1), \mathbf{x}(c_2)) = f_1(\Delta(\mathbf{x}(c_1), \mathbf{x}(c_2))) + f_2(n_1, n_2)$ where f_1 is some strictly increasing function of $\Delta(\mathbf{x}(c_1), \mathbf{x}(c_2))$ and f_2 is an arbitrary penalty function on the size of clusters, then a property that would normally be required of a separation measure is that for any two clusters c_1 and c_2 that have identical characteristics, so that $\mathbf{x}(c_1) = \mathbf{x}(c_2)$, we have

$$\Delta(\mathbf{x}(c_1), \mathbf{x}(c_2)) = 0. \tag{4.2}$$

At this point it is convenient to re-parameterise Φ. Let

$$d = \frac{\overline{z}_1^2 + \overline{z}_2^2}{\overline{u}}$$

represent a normalised squared distance from zero of the two clusters, define

$$\alpha_j = \frac{\overline{z}_j^2}{d\overline{u}} \quad \text{with} \quad j = 1, 2$$

to be the corresponding relative squared distance from zero of the two clusters and let

$$v_j = \frac{u_j}{\overline{u}} \quad \text{with} \quad j = 1, 2$$

be approximately the relative sums of squares of the two clusters. Note that $\alpha_1, \alpha_2 \geq 0$ and

$$\alpha_1 + \alpha_2 = \frac{1}{d\overline{u}}(\overline{z}_1^2 + \overline{z}_2^2) = 1.$$

Then

$$\begin{aligned} \gamma &= \lambda_1\alpha_1 - 2\lambda_{12}\sqrt{\alpha_1\alpha_2} + \lambda_2\alpha_2, \\ \overline{z}_j^2 &= \alpha_j d\overline{u}, \end{aligned}$$

and

$$\begin{aligned} \Phi &= \log\left(\overline{u}(1 + \gamma d)\right) - \rho_1 \log\left(\overline{u}(v_1 + \alpha_1 d)\right) - \rho_2 \log\left(\overline{u}(v_2 + \alpha_2 d)\right) \\ &= \log(1 + \gamma d) - \rho_1 \log(v_1 + \alpha_1 d) - \rho_2 \log(v_2 + \alpha_2 d). \end{aligned}$$

Definition 4.4.1. *Define* $\Phi(\mathbf{x}(c_1), \mathbf{x}(c_2))$ *as* homogeneous *if, whenever* $\mathbf{x}(c_1) = \mathbf{x}(c_2)$, $\Phi(\mathbf{x}(c_1), \mathbf{x}(c_2)) = \Phi_0$ *is a function of* (n_1, n_2) *alone.*

Under the family of separations above, a necessary and sufficient condition for $\Psi(\mathbf{x}(c_1), \mathbf{x}(c_2))$ to satisfy the property leading to equation (4.2) is that $\Phi(\mathbf{x}(c_1), \mathbf{x}(c_2))$ is homogeneous.

Theorem 4.4.2. *If $\Phi(\mathbf{x}(c_1), \mathbf{x}(c_2))$ is homogeneous and a g-prior is employed then for any two identical clusters c_1 and c_2 such that $\bar{n} = 2n_1$,*

$$\bar{a} = 2a_1, \quad \bar{b} = 2b_1 \quad and \quad \bar{g} = 2g_1.$$

Furthermore, if these three conditions above hold, then $\Phi(\mathbf{x}(c_1), \mathbf{x}(c_2))$ will be homogeneous.

Proof. If $\mathbf{x}(c_1) = \mathbf{x}(c_2)$ then $\alpha_1 = \alpha_2 = 0.5$, $v_1 = v_2 = v$ (say), $n_1 = n_2 = n$ and $\rho_1 = \rho_2 = \rho$. Therefore,

$$\begin{aligned} \exp(\Phi) &= \frac{1+\gamma d}{(v_1+\alpha_1 d)^{\rho_1}(v_2+\alpha_2 d)^{\rho_2}} \\ &= 2^{2\rho}\frac{1+\gamma d}{(2v+d)^{2\rho}} \end{aligned}$$

where

$$\begin{aligned} \gamma &= 0.5\,(\lambda_1 - 2\lambda_{12} + \lambda_2) \\ &= \frac{(\overline{g}+n)(g_1+n)}{(\overline{g}+2n)g_1} - \frac{n(g_1+n)}{(\overline{g}+2n)g_1} \\ &= \frac{(g_1+n)\overline{g}}{\overline{g}+2n)g_1} \end{aligned}$$

because $\lambda_1 = \lambda_2$, $g_1 = g_2$ and $\cos\theta[\mathbf{z}_1, \mathbf{z}_2] = 1$. Clearly, Φ is a function of z unless $\rho = 0.5$ implying $\bar{a} = 2a_1$. Substituting gives

$$\exp(\Phi) = 2\frac{1+\gamma d}{2v+d}.$$

Since by definition v and d are functionally independent, we therefore must have

$$v = 0.5 \;\Leftrightarrow\; 2u_j = \overline{u} \;\Leftrightarrow\; \bar{b} = 2b_1$$

and also

$$\gamma = 1 \;\Leftrightarrow\; 1 + \frac{n}{g_1} = 1 + \frac{2n}{\overline{g}} \;\Leftrightarrow\; \overline{g} = 2g_1$$

as required. Finally, under these conditions when $\mathbf{x}(c_1) = \mathbf{x}(c_2)$, $\Phi(\mathbf{x}(c_1), \mathbf{x}(c_2)) = \log 2$. □

Note that the standard way of assigning a prior to a conjugate model is not homogeneous and so falls at the first hurdle. However there is an obvious family of conjugate Bayesian models which is homogeneous.

Corollary 1. *The* proportional model *which sets $a_c = \breve{a}n_c$, $b_c = \breve{b}n_c$ and $g_c = \breve{g}n_c$ for some values $\breve{a}, \breve{b}, \breve{g} > 0$ is homogeneous.*

For the proportional model, $\rho_j = n_j n_{12}^{-1}$, $u_j = (2\breve{b} + r\widehat{\sigma}_j^2)n_j$ and $\overline{u} = u_1 + u_2$ so that $v_1 + v_2 = 1$. Furthermore, let the value of γ when two profiles are identically oriented (so that $\theta[\mathbf{z}_1, \mathbf{z}_2] = 0$) be γ_0. Then, under the proportional model,

$$\begin{aligned}\gamma_0 &= \lambda_1\alpha_1 - 2\lambda_{12}\sqrt{\alpha_1\alpha_2} + \lambda_2\alpha_2\\ &= \left(1 + \frac{\rho_2}{\breve{g}}\right)\alpha_1 - 2\frac{1}{\breve{g}}\sqrt{\rho_1\rho_2\alpha_1\alpha_2} + \left(1 + \frac{\rho_1}{\breve{g}}\right)\alpha_2\\ &= \alpha_1 + \alpha_2 + \frac{1}{\breve{g}}\left(\rho_2\alpha_1 - 2\sqrt{\rho_1\rho_2\alpha_1\alpha_2} + \rho_1\alpha_2\right)\\ &= 1 + (\sqrt{\rho_2\alpha_1} - \sqrt{\rho_1\alpha_2})^2\breve{g}^{-1}.\end{aligned}$$

We can now derive some important properties of proportional clustering.

Theorem 4.4.3. *Under proportional clustering, for all possible values of $\mathbf{x}_1(c_1), \mathbf{x}_2(c_2)$,*

$$\Phi(\mathbf{x}_1(c_1), \mathbf{x}_2(c_2)) \geq I(\rho),$$

where $\rho_j = n_j n_{12}^{-1}, j = 1, 2$, and $I(\boldsymbol{\rho}) = -\sum_{j=1,2}\rho_j \log \rho_j$.

Proof.

$$\begin{aligned}\Phi &= \log(1 + \gamma d) - \rho_1 \log(v_1 + \alpha_1 d) - \rho_2 \log(v_2 + \alpha_2 d) \qquad (4.3)\\ &= \Delta^{(1)}(\gamma, d) + \Delta^{(2)}(v_1, \rho_1, \alpha_1, d) + I(\rho),\end{aligned}$$

where

$$\Delta^{(1)}(\gamma, d) = \log(1+\gamma d) - \log(1+d) \geq 0$$

since

$$\gamma = \lambda_1\alpha_1 - 2\lambda_{12}\sqrt{\alpha_1\alpha_2} + \lambda_2\alpha_2 \geq \gamma_0$$

with equality if and only if $\cos\theta[\mathbf{z}_1, \mathbf{z}_2] = 1$ where $\gamma_0 \geq 1$ is defined above, and

$$\Delta^{(2)}(v_1, \rho_1, \alpha_1, d) = \log(1+d) - \rho_1\log(v_1+\alpha_1 d) - \rho_2\log(v_2+\alpha_2 d) - I(\boldsymbol{\rho}).$$

Note that

$$\Delta^{(1)} = 0 \;\Leftrightarrow\; \cos\theta[\mathbf{z}_1, \mathbf{z}_2] = 1 \;\Leftrightarrow\; \rho_2\alpha_1 = \rho_1\alpha_2 \Leftrightarrow\; \frac{\alpha_1}{n_1} = \frac{\alpha_2}{n_2}.$$

Also, for fixed ρ_1, ρ_2 with $\rho_1 + \rho_2 = 1$,

$$-\rho_1\log(v_1+\alpha_1 d) - \rho_2\log(v_2+\alpha_2 d)$$

is minimised when $\sum_{j=1,2}\rho_j\log x_j$ is maximised w.r.t. $x_j = v_j + \alpha_j d$. Let $x_1 + x_2 = 1 + d$ when $x_j = (1+d)\rho_j$. So letting $v_j = \rho_j = \alpha_j$ and using Jensen's inequality for concave functions, that is,

$$\sum_{j=1,2}\rho_j\log x_j \leq \log\left(\sum_{j=1,2}\rho_j x_j\right)$$

then

$$\sum_{j=1,2}\rho_j\log\frac{x_j}{\rho_j(1+d)} \leq \log\sum_{j=1,2}\frac{x_j}{1+d} = 0$$

and

$$\sum_{j=1,2}\rho_j\log x_j \leq \sum_{j=1,2}\log\left[\rho_j(1+d)\right].$$

Therefore,

$$\begin{aligned}
\Delta^{(2)}(v_1, \rho_1, \alpha_1, d) &\geq \log(1+d) - \rho_1\log(\rho_1(1+d)) - \rho_2\log\left[\rho_2(1+d)\right] - I(\boldsymbol{\rho}) \\
&= 0 \\
&= \Delta^{(2)}(\rho_1, \rho_1, \alpha_1, d).
\end{aligned}$$

Therefore

$$\Phi(v_1, \rho_1, \alpha_1, d) \geq I(\rho) = \Phi(\rho_1, \rho_1, \alpha_1, d).$$

□

Corollary 2. *For any fixed (unordered) pair* $\mathbf{n} = (n_1, n_2)$

$$\Delta_{\mathbf{n}}(\mathbf{x}(c_1), \mathbf{x}(c_2)) = \Phi(\mathbf{x}_1(c_1), \mathbf{x}_2(c_2)) + \kappa''(\mathbf{n}),$$

where $\kappa''(\mathbf{n}) = -I(\boldsymbol{\rho})$ *is a separation measure. That is,*

1. *For all pairs* $(\mathbf{x}(c_1), \mathbf{x}(c_2))$

$$\Delta_{\mathbf{n}}(\mathbf{x}(c_1), \mathbf{x}(c_2)) \geq 0$$

   with equality if, and only if, $\mathbf{x}(c_1) = \mathbf{x}(c_2)$

2. *For all pairs* $(\mathbf{x}(c_1), \mathbf{x}(c_2))$

$$\Delta_{\mathbf{n}}(\mathbf{x}(c_1), \mathbf{x}(c_2)) = \Delta_{\mathbf{n}}(\mathbf{x}(c_2), \mathbf{x}(c_1)).$$

Proof. It follows from Theorem 4.4.3 that

$$\begin{aligned} \Delta_{\mathbf{n}}(\mathbf{x}(c_1), \mathbf{x}(c_2)) &= \Phi(\mathbf{x}_1(c_1), \mathbf{x}_2(c_2)) + \kappa''(\mathbf{n}) \\ &= \Delta^{(1)}(\mathbf{x}(c_1), \mathbf{x}(c_2)) + \Delta^{(2)}(\mathbf{x}(c_1), \mathbf{x}(c_2)). \end{aligned}$$

The first point is a direct consequence of the theorem on noting that $\Delta^{(1)} \geq 0$, and $\Delta^{(1)} = 0$ takes its maximum if, and only if, $\gamma = 1$ and $\frac{\alpha_1}{n_1} = \frac{\alpha_2}{n_2}$ so that the scaled distances of the two profiles from zero satisfy $n_1^{-1}\widehat{\boldsymbol{\beta}}_1 = n_2^{-1}\widehat{\boldsymbol{\beta}}_2$. Also, $\Delta^{(2)} \geq 0$ and $\Delta^{(2)} = 0$ if, and only if, $\alpha_j = \rho_j$ so that

$$v_j + \alpha_j d = \rho_j(1 + d) \Leftrightarrow v_j = \rho_j \Leftrightarrow \widehat{\sigma}_1^2 = \widehat{\sigma}_2^2.$$

The second point is immediate from the symmetry in $(\mathbf{x}(c_1), \mathbf{x}(c_2))$ of the three functions $\Phi(\mathbf{x}_1(c_1), \mathbf{x}_2(c_2))$, $I(\boldsymbol{\rho})$ and $\kappa''(\mathbf{n})$. □

So a sufficient and almost necessary condition for MAP selection to behave in a way that combines clusters in partitions with "close" statistics is that the hyperparameters are set as a proportional model. For most other settings, and in particular those advocated by other authors as defaults, this is not the case. It is interesting to note that to ensure consistency in different contexts various authors have suggested introducing a dependency of the parameter g on sample size. However, this suggested dependency demands that the prior variance of the proportional model decreases in the cluster size n whereas here it increases. This is not too disturbing for our applications. The natural type of consistency we might require here is associated with the length of profile – a function of the experimental design – not the number of genes of certain types which is determined by the technology of the gene chip and thus fixed. Note that with the hyperparameter settings recommended here, consistency is automatic under increasing profile length.

4.5 Comparison for Two Simple Simulation Studies

In order to illustrate the characteristics of cluster inference under the conventional settings of the hyperparameters as described by Heard et al. (2006) and our proportional setting, we have simulated from scenarios where the desired characteristics of the clustering algorithm are fairly transparent.

4.5.1 Outliers and Junk Clusters

First consider clustering just 7 points (profiles of length 1) simulated from 3 clusters of sizes $n_1 = 2$, $n_2 = 4$ and $n_3 = 1$. The two points in the first cluster are drawn from a distribution with a large negative mean expression $-s$, the points in the second cluster drawn have zero mean expression and the point in the fourth cluster has large expression

s. So the means of the 7 points cluster into the partition $A = \{(-s, -s), (0, 0, 0, 0), (s)\}$. In our notation

$$y^{(j)} = B\boldsymbol{\beta}_j + \varepsilon^{(j)},$$

where $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \boldsymbol{\beta}_3) = (-s, 0, s)'$, $B = 1$ and $\varepsilon^{(j)} \sim N(0, 0.05)$ for $j = 1, 2, 3$ and we set $s = 1,000$.

Note that whilst the undesirable partition $B = \{(-s, -s, s), (0, 0, 0, 0)\}$ appears as a candidate partition in both methods, as is typical, it appears earlier under the conventional settings than the proportional settings.

To compare the proportional scaling method with that of Heard et al. (2006) for simplicity we have subsequently set $\breve{g} = g$, $\breve{a} = a$, $\breve{b} = b$, so that the algorithms exactly correspond at the beginning. For comparability we use the same default prior as Heard et al. (2006) over the partition space.

We now compare the performance of the clustering algorithm by Heard et al. (2006) for different values of the prior parameters to ours in Fig. 4.4. A typical dendrogram of the combination under the conventional setting and default partition priors is given in Fig. 4.3 together with another dendrogram which is often produced by the algorithm by Heard et al. (2006). Note that in the second dendrogram in Fig. 4.3, as anticipated in Section 4.3.2, the first cluster combines the three outliers at an early step, a combination that under AHC can never be retrieved. Such unhelpful properties depend on the setting of the hyperparameters.

Obviously the precise combination reflected in such dendrograms depends on how the values of the hyperparameters are chosen. So in Fig. 4.4 we have determined which values of the simulated data sets correctly identified the true simulated partition for the conventional and our settings of hyperparameter (identified as above) and default choice of prior by Heard et al. (2006) over the partition space. Notice that our method appears

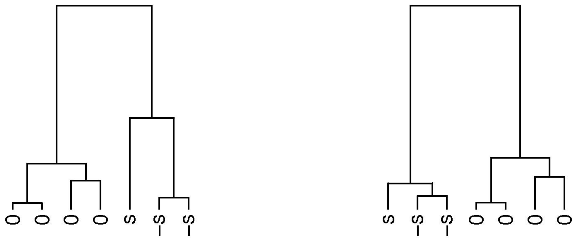


Figure 4.3: Two of the dendrograms produced by AHC using the algorithm with proportional parameters and the algorithm by Heard et al. (2006).

much more stable to misspecification of these three hyperparameters. The values used are $g = \breve{g}^{-1} \in [1, 10]$, $a = \breve{a} \in [0.01, 1]$ and $b = \breve{b} \in [0.01, 1]$.

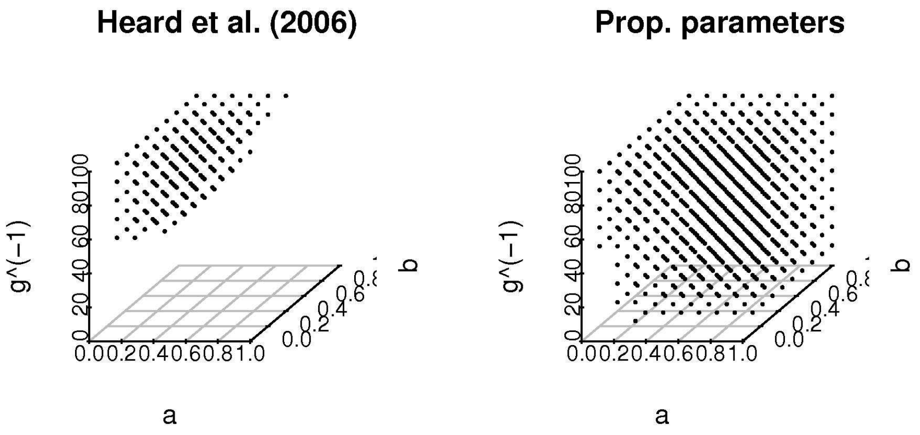


Figure 4.4: Result of the clustering algorithm by Heard et al. (2006) and our algorithm for different values of the prior parameters. Each dot corresponds to a combination of values of the prior parameters which generated the desired partition A of our dataset.

Finally in Fig. 4.5 we compare the number of times during the simulations the partitions A, B, the 'large variance' partition $\mathcal{C} = \{(-s, -s, 0, 0, 0, 0, s)\}$ and all other partitions D are chosen as optimal. Notice that the broad effect here is for the vast proportion of partitions misclassified as B under conventional clustering to be properly clustered as A under proportional clustering.

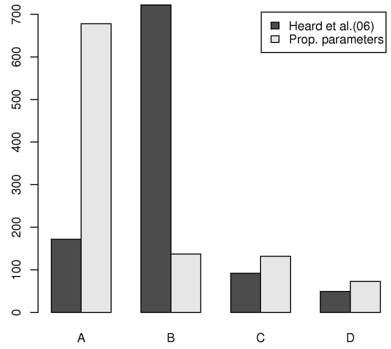


Figure 4.5: When different prior parameters are used, the algorithms produce different partitions of our dataset. The plot above shows the counts of each partition produced by the algorithm by Heard et al. (2006) and the algorithm with proportional parameters.

4.5.2 Merging of Complementary Profiles

A property of a clustering algorithm we would like to avoid is one where two complementary profiles (i.e. two profiles where one is approximately the negative of the other, each with high expression) are combined into a single large-variance approximately zero-mean cluster. In our second simulation we therefore created such a scenario. Typically for higher dimensional problems we introduce further tuning parameters on the prior over the partition. However, the parameters have no effect on the combination of clusters when they are all of the same cardinality - as they are at the beginning of the AHC algorithm. We can therefore compare our algorithm fairly with the conventional one with default prior if we focus on the behaviour of the algorithm on the *first combination* of the AHC.

Thus, consider the dataset formed by the following three clusters

$$\begin{aligned} y_k^{(1)} &= B\boldsymbol{\beta}^{(1)} + \varepsilon_k \qquad k = 1, \\ y_k^{(2)} &= B\boldsymbol{\beta}^{(2)} + \varepsilon_k \qquad k = 2, \ldots, 6, \\ y_k^{(3)} &= -B\boldsymbol{\beta}^{(1)} + \varepsilon_k \quad k = 7, \end{aligned} \tag{4.4}$$

where $\varepsilon_k \sim N(0,1)$ for $k = 1, \dots, 7$ and B is the Fourier design matrix as in Anderson et al. (2006).

Following the notation and vocabulary of the running example as in Anderson et al. (2006), our dataset, drawn in Fig. 4.6, has two complementary gene profiles and 5 gene profiles close to zero. Note that genes in cluster 1 and 3 have opposite complementary profiles, so it is critical not to combine genes from these two different profiles into a single cluster. Cluster 2 represents a set of unresponsive genes with a zero mean profile.

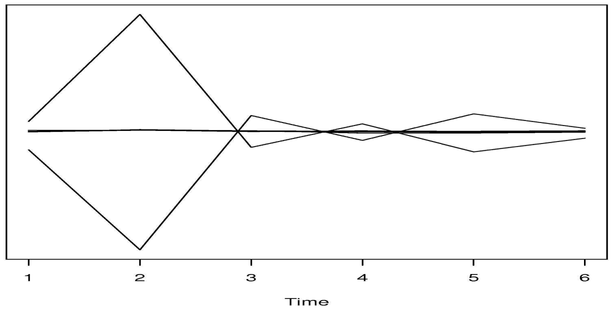


Figure 4.6: Data simulated as in model (4.4). The amplitude of the curves is variable and depends on the value of σ^2.

The worst case scenario happens when the observations in clusters 1 and 3 are combined together at the first iteration of the algorithm. When this happens under AHC the algorithm can never identify the desired partition, which therefore will not be identified no matter which priors we are using on the partitions. Consider the results in 4.1. The prior parameters used were $g = \breve{g} \in [1, 1000]$, $a = \breve{a} \in [0.01, 1]$ and $b = \breve{b} \in [0.01, 1]$. Again it is easy to see how our new settings improve on the original in this circumstance, particularly when expressions are large.

Table 4.1: The table shows the number of times (out of 432) that genes in cluster 1 and 3 are combined together at the first step of the algorithm by Heard et al. (2006) and the algorithm with proportional parameters as the amplitude of the curves increases.

σ^2	Heard et al. (2006)	Our algorithm
10	3	0
100	8	0
1,000	32	0
10,000	62	0
100,000	95	0
1,000,000	102	0

4.6 Separation of Models: Separation of Statistics

4.6.1 Some Useful Parameters

Although we have found a separation measure corresponding to Bayesian selection, it remains to demonstrate that this induced measure is largely consistent with a separation measure with which we would be content predictively. We therefore next examine how the function $\Phi = \Delta^{(1)} + \Delta^{(2)} + I(\boldsymbol{\rho})$ (see Equation 4.3) where

$$\begin{aligned}\Delta^{(1)} &= \log(1+\gamma d) - \log(1+d) \\ \Delta^{(2)} &= \log(1+d) - \rho_1 \log(v_1 + \alpha_1 d) - \rho_2 \log(v_2 + \alpha_2 d) - I(\boldsymbol{\rho}),\end{aligned}$$

compares adjacent partitions for the proportional model as a function of the sufficient statistics of two profiles. This allows us both to confirm that the characteristics of the induced separation measure are largely desirable and guides us to settings of prior hyperparameters that ensure plausible predictive implications. Because we need to acknowledge that the Bayes factor clustering has an intrinsic structure that selects as a function of (n_1, n_2), in this section we will assume the cardinalities (n_1, n_2) of two candidate clusters – and hence (ρ_1, ρ_2) – are fixed. There are four statistics that are central to the combination rule: d, v_1 (defined above), η and ζ^2 (defined below).

1. The statistic $\eta = \sqrt{\alpha_1\alpha_2}(1 - \cos(\theta[\mathbf{z}_1, \mathbf{z}_2])$ is a measure of the *dissimilarity in orientation* of the two profiles.

2. A measure of the *differences in overall magnitudes* of squared differences in distance from zero relative to that expected under the given cluster size under the prior is ζ^2 where

$$\begin{aligned}
\zeta &= \sqrt{\rho_1\alpha_2} - \sqrt{\rho_2\alpha_1} \\
&= \sqrt{\rho_1}\sqrt{1-\alpha_1} - \sqrt{\alpha_1}\sqrt{1-\rho_1} \\
&= \cos x \sin y - \cos y \sin x \\
&= \sin\left(\sin^{-1}(\sqrt{\alpha_1}) - \sin^{-1}(\sqrt{\rho_1})\right)
\end{aligned}$$

where $\alpha_1 + \alpha_2 = \rho_1 + \rho_2 = 1$, $\rho_1 = \sin^2 x$, $\rho_2 = \cos^2 x$, $\alpha_1 = \sin^2 y$, $\alpha_2 = \cos^2 y$. Note that

$$0 \leq \zeta^2 \leq \max(\rho_1, \rho_2)$$

because

$$\begin{aligned}
\sqrt{\rho_1\alpha_2} - \sqrt{\rho_2\alpha_1} &\leq \sqrt{\rho_2} \\
\sqrt{\frac{\rho_1}{\rho_2}}\sqrt{\alpha_2} &\leq 1 + \sqrt{\alpha_1}
\end{aligned}$$

which is satisfied only if $\rho_2 \leq \rho_1$.

Now

$$\Delta_{\mathbf{n}}^{(1)}(\gamma(\eta,\zeta), d) = \log\left(1 + (\gamma - 1)(1 + d^{-1})^{-1}\right),$$

where

$$\begin{aligned}
1 \leq \gamma &= \lambda_1\alpha_1 - 2\lambda_{12}\sqrt{\alpha_1\alpha_2} + \lambda_2\alpha_2 \\
&= \left(1 + \frac{\rho_2}{\breve{g}}\right)\alpha_1 - 2\cos(\theta[\mathbf{z}_1, \mathbf{z}_2])\frac{\sqrt{\rho_1\rho_2}}{\breve{g}} + \left(1 + \frac{\rho_1}{\breve{g}}\right)\alpha_2 \\
&= 1 + \frac{\zeta^2 + 2\eta\sqrt{\rho_1\rho_2}}{\breve{g}} \qquad (4.5)
\end{aligned}$$

and

$$\Delta_{\mathbf{n}}^{(2)}(v_1, \zeta, d) = \log(1+d) - \sum_{j=1,2} \rho_j \log(v_j + \alpha_j d) - I(\boldsymbol{\rho}).$$

Note that $\Delta_{\mathbf{n}}^{(1)}$ is a function only of (η, ζ, d) and ignores (v_1, v_2) whilst $\Delta_{\mathbf{n}}^{(2)}$ is a function of relative variances, relative size ζ expressed as a function of α_1, and combined size d and also ignores η.

It is straightforward to verify that $\Delta_{\mathbf{n}}^{(1)}(\gamma(\eta, \zeta), d)$ is strictly increasing in η, ζ and d and bounded, with

$$\begin{aligned}
\lim_{\eta \to 0} \Delta_{\mathbf{n}}^{(1)} &= \log(1 + \zeta^2 (1 + d^{-1})^{-1} \breve{g}^{-1}) \geq 0, \\
\sup_{d} \Delta_{\mathbf{n}}^{(1)} &= \log(1 + (\max\{\rho_1, \rho_2\} + 2\sqrt{\rho_1 \rho_2}) \breve{g}^{-1}), \\
\lim_{d \to 0} \Delta_{\mathbf{n}}^{(1)} &= 0,
\end{aligned}$$

when the noise-to-signal parameter $\breve{g}$ is large, so that observational error is assumed to dominate the signal and the contribution of this term is negligible.

The second component,

$$\Delta_{\mathbf{n}}^{(2)}(v_1, \alpha_1, d) = \log(1+d) - \rho_1 \log(v_1 + \alpha_1 d) - \rho_2 \log(v_2 + \alpha_2 d) - I(\boldsymbol{\rho}),$$

is a function of the relative sums of squares and scaled relative distances from zero but not $\breve{g}$. Unlike $\Delta_{\mathbf{n}}^{(1)}(\gamma, d)$ it is unbounded above and, depending on the distance d of the two clusters from zero, can heavily penalise the combination of clusters with relatively very different associated estimated variances or different scaled lengths from the origin. When d is small and the two profiles are close to zero $\Delta_{\mathbf{n}}$ acts as a penalty mainly for divergent estimates of variance, taking close to its minimum value whenever $\widehat{\sigma}_1^2 = \widehat{\sigma}_2^2$. However when d is very large it penalises almost entirely on the basis of the difference in distance of the two clusters from the origin and ignores any divergence in their estimated variances.

Whatever the value of ρ_1 when $v_1 = \alpha_1$, $\Delta^{(2)}$ is not a function of d and takes the value

$$\Delta_{\mathbf{n}}^{(2)}(v_1, \zeta, d) = \sum_{j=1,2} \rho_j \log \frac{\rho_j}{v_j}.$$

Thus the characteristics of the induced separation measure of the proportional model seem eminently desirable, with the caveat that the conjugacy encourages outlying clusters with similar profiles but different variances to occasionally be combined when the two clusters are far from zero. However it is easily verified that when clusters are about the same cardinality, so that $\rho_1 \simeq \rho_2$, and ε, ω are small then this dependence on d is insignificant.

4.6.2 Combined Separation

Note that the geometry of Φ is simple because $\rho_1 + \rho_2 = 1$. When $\rho_1 + \rho_2 \neq 1$ it is easily verified that the stationary points lie on a quadratic, giving rise to a much richer geometry in Φ. This is the algebraic reason for much of the strangeness of the induced selection. This phenomenon is illustrated in the central column of Fig. 4.7 where we graph Φ for two clusters with identical orientation and cardinality for various settings of the hyperparameters. The magnitude of this measure represents the inclination of two clusters to merge, d is the mutual distance from zero of the two clusters, α_1 is the distance from zero of cluster 1 relative to this mutual distance, and v_1 is the relative variance of the profile of cluster 1 relative to that of the combined cluster for the proportional model. The central column shows the recommended setting when $\rho_1 + \rho_2 = 1$ whilst the left and right hand columns show $L(\Phi)$ when $\rho_1 + \rho_2 < 1$ and $\rho_1 + \rho_2 > 1$ respectively. Here, $v_1 = \{0.05, 0.3, 0.5\}, \rho_1 = \{0.3, 0.5, 0.7\}$ and $\alpha_1 \in [0.1, 0.9], \log(d) \in [-4, 6]$. We choose $\rho_1 = \rho_2$, $\eta = 0$ (since the two clusters are identically oriented) and $\breve{g}^{-1} = 100$ so that equation (4.5) implies that $\gamma = 1 +$

$100\rho_1(\sqrt{\alpha_1} - \sqrt{1-\alpha_1})^2$. Note that $\alpha_1 + \alpha_2 = 1$ and $v_1 + v_2 = 1$ are always satisfied. Equation (4.5) has singularities at $v_1, \alpha_1 = \{0, 1\}$. All nine plots have the same axes.

The dependence of Φ on d is only significant when Φ takes large values. In this case the clusters will usually be kept separate for other reasons anyway. Dependence on the relative distance from zero of the two clusters only occurs when d is of moderate magnitude. Furthermore the discrepancy in relative variances is only significant when their ratio is substantially different from one and then only when d is quite far from zero.

The left hand column of Fig. 4.7 illustrates the phenomenon discussed in Section 4.3.2 that when $\rho_1 + \rho_2 < 1$, clusters become increasingly large the further d is from zero: Φ eventually becoming very large regardless of how close the pair of cluster statistics are. On the other hand the right hand column (when $\rho_1 + \rho_2 > 1$) shows as d becomes large, the two clusters will become close regardless of the value of the other statistics. This illustrates why Bayes Factor selection can be badly behaved unless the hyperparameters are chosen carefully.

Further comments on parameters are in Smith et al. (2008a).

4.6.3 Setting Hyperparameters in Proportional Models

There are two complementary and fully Bayesian ways of setting the hyperparameters $(\breve{a}, \breve{b}, \breve{g})$. First, these parameters should be chosen so as to coincide with predictive beliefs about the individual cluster profiles we expect to see before incorporating the data. The value of $\breve{a}/\breve{b}$ is our prior expectation of the precision σ^{-2} of a typical cluster, whilst $\breve{a}$ can be calibrated to our coefficient of variation of this information $[\breve{a}n_c]^{-1}$ for a cluster c of a given cardinality n_c. The magnitude of $\breve{g}$ determines the relative strength of the prior information on each unit profile and governs the extent that the

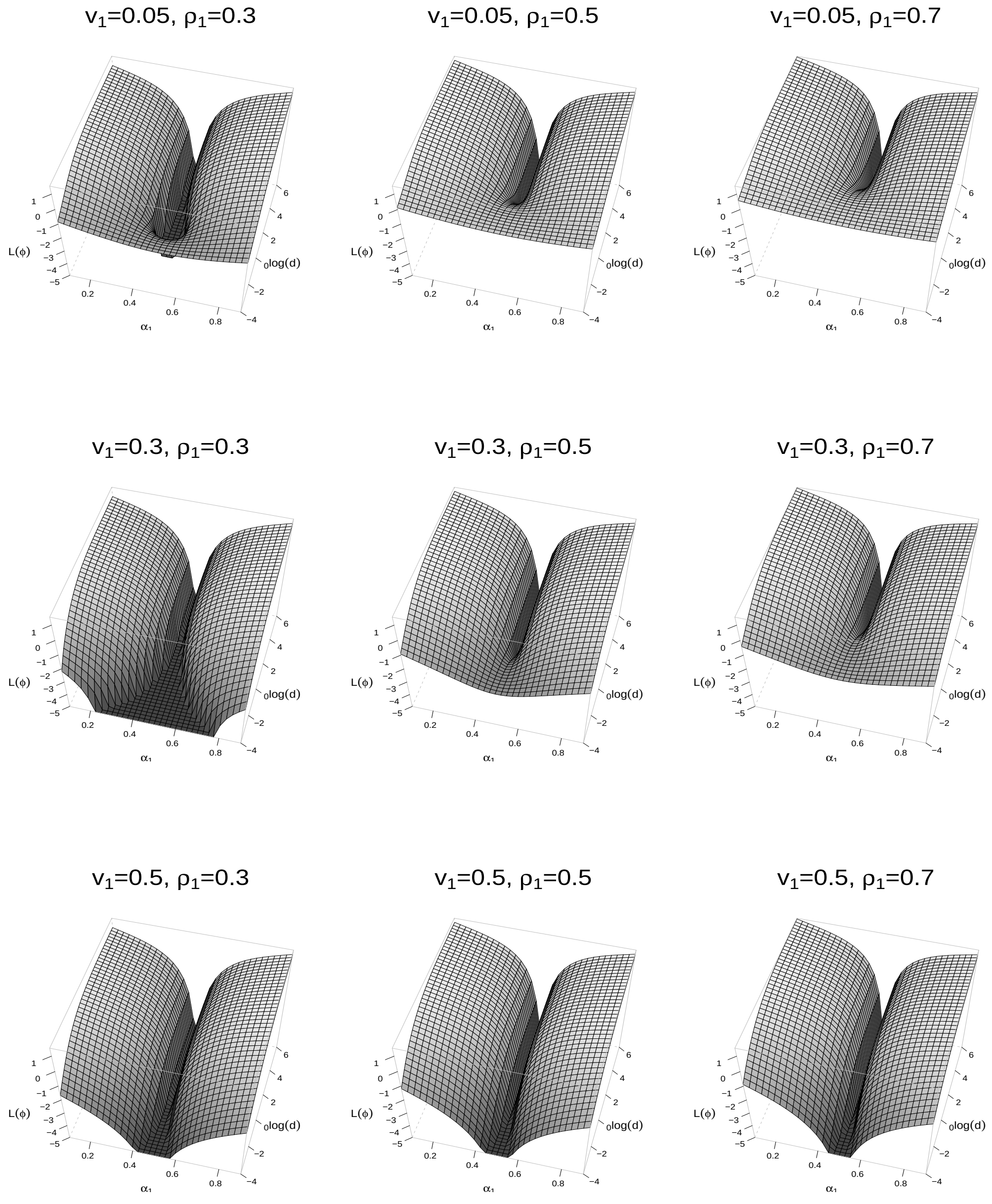


Figure 4.7: A plot of $L(\Phi) = \log(\max\{\Phi - \log(2), \exp(-5)\})$.

cluster posterior means shrink towards zero. Note that, in agreement with Wakefield et al. (2003), we recommend setting these prior parameters so that they calibrate to pre-posterior predictions of the variance of a particular cluster.

Second, it is important that the values $(\breve{a}, \breve{b}, \breve{g})$ calibrate hyperparameters to pre-posterior beliefs about the relative probabilities of adjacent partitions after realising certain hypothetical observations. Thus, the magnitude of parameter $\breve{g}$ solely influences the relative weight we place on two clusters having different orientations of profiles. The smaller this parameter, the more likely clusters – all of whose characteristics are the same but whose orientations are different – are kept separate. To fix an appropriate value of $\breve{g}$ we suggest calibrating to two expected profiles of different orientation distances from the value of $\breve{g}$ we suggest calibrating to two expected profiles of different orientation distances from the origin and asking the scientist which two profiles are most likely to come from the same cluster.

The effects of the setting of the value of $\breve{b}$ has a strong effect on the combination rule when clusters have profiles close to zero. If it is set very small so that $d \to 0$ then two clusters with small cardinality and a ratio of the sums of squares very different from unity will be kept apart. Within the context of our running example, such gene expression profiles are not in practice interesting enough to keep separate and this phenomenon can sometimes disrupt the AHC algorithm. So, at least pragmatically, there are good reasons for keeping this parameter well away from zero. This implicitly demands that the prior expectation on the precision σ^{-2} is not big: often a plausible assumption. Interestingly, this parameter is set by default to be very small in Heard et al. (2006) which may account for a different type of instability in their algorithm that sometimes occurs early in the AHC.

4.7 More Data Analyses

Our collaborators, biologists Andrew J. Millar (Centre for Systems Biology at Edinburgh and School of Biological Sciences) and François-Yves Bouget (Laboratoire Arago, Banyuls-sur-Mer, France), provided us with datasets to test our algorithm in real life problems.

We present here the analysis of a time-course microarray experiment on the tiny algae *Ostreococcus tauri* carried out using the algorithm described in this chapter and implemented in C++. The graphical output was obtained using R (R Development Core Team, 2009). The analysis presented in this section is my contribution to the paper Monnier et al. (2009), which also includes a full discussion of the biological meaning and implications of the findings.

4.7.1 The Experiment

François-Yves Bouget and his colleagues of the Laboratoire Arago in Banyuls-sur-Mer conducted have conducted a genome-wide analysis of gene expression in *Ostreococcus tauri* cells exposed to light/dark cycles. To identify genes with a diurnal rhythm, cells entrained under 12:12 light/dark (L/D) cycles were sampled every 3 hours for 24 hours with two overlapping time points at time 9 in 3 independent experiments. The light went on at time 9 and off at time 21. Under medium light conditions, cell division is synchronised, occurring at the onset of night and most of cell cycle genes are regulated by the diurnal cycle.

The expression of each time point was compared to a pool of all 27 time points. A 3-factor ANOVA identified 6822 probes corresponding to genes differentially expressed over time during the light/dark cycle with a Pvalue $< 10^{-3}$. Biological triplicate were highly reproducible as revealed by principal component analysis (PCA) performed on the

27 time points. PCA on the individual 6822 gene probes confirmed that the differential gene expression over the time course (day/nigh, evening/morning) accounts for most of the variability observed. Fewer genes had their phase of expression 3 hours after dusk (Time 0) suggesting a gap in transcription at that time.

For subsequent analysis, we selected the 2039 gene probes with best dispersion after PCA. Globally all genes selected after ANOVA, had rhythmical and highly reproducible profiles of expression over three 12:12 LD cycles. In our single LD 12:12 condition, all expressed genes in O. tauri display robust diurnal rhythms, consistent with a global regulation of transcription under light/dark cycles. Genome wide regulation of gene expression by the photoperiod is well known in cyanobacteria but to our knowledge, this is the first example in eukaryotic microalgae.

4.7.2 Analysis and Results

We use our Bayesian hierarchical clustering approach with a Fourier basis function to cluster genes according to their transcript waveform. See also Edwards et al. (2006). For computational reasons, only the first, third, sixth and ninth harmonics, along with the constant term, were included, yielding 9 parameters. A direct analysis could not be performed on all 6822 genes due to memory space limitation.

The clustering algorithm produced 138 clusters. Some of these clusters are given in Fig. 4.8 and 4.9. Since such a large number of clusters (138) was generated, the size of each cluster was relatively small (2 to 50 probes). Because the size of the clusters was small, each cluster was examined individually.

The results are also shown in Fig. 4.10 by a 24hr clock. Each squared dot represents a cluster, with the size of the dot proportional to the size of the cluster. The clusters are positioned around the clock depending on when they peak during the

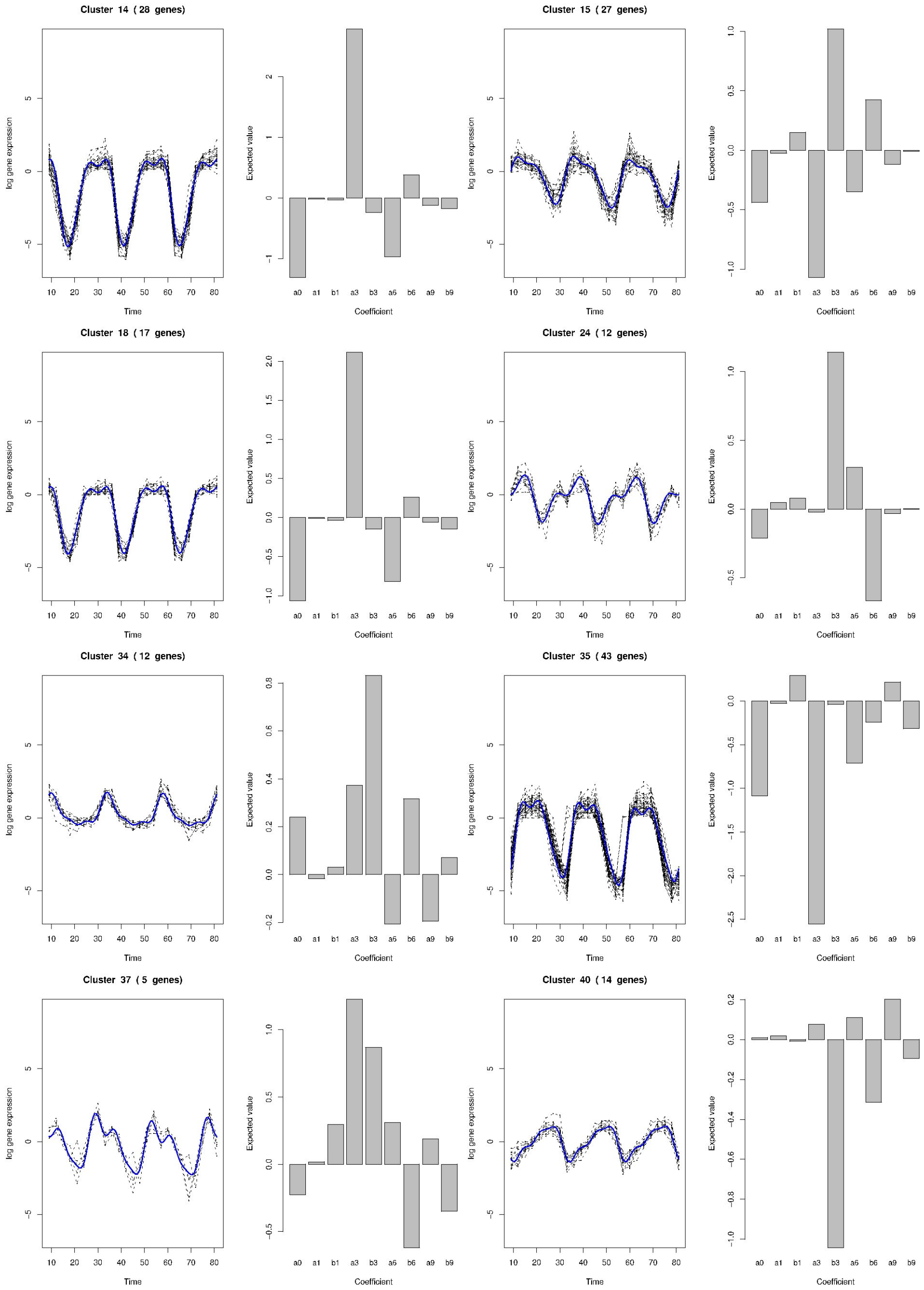


Figure 4.8: Some of the clusters for the *Ostreococcus* experiment.

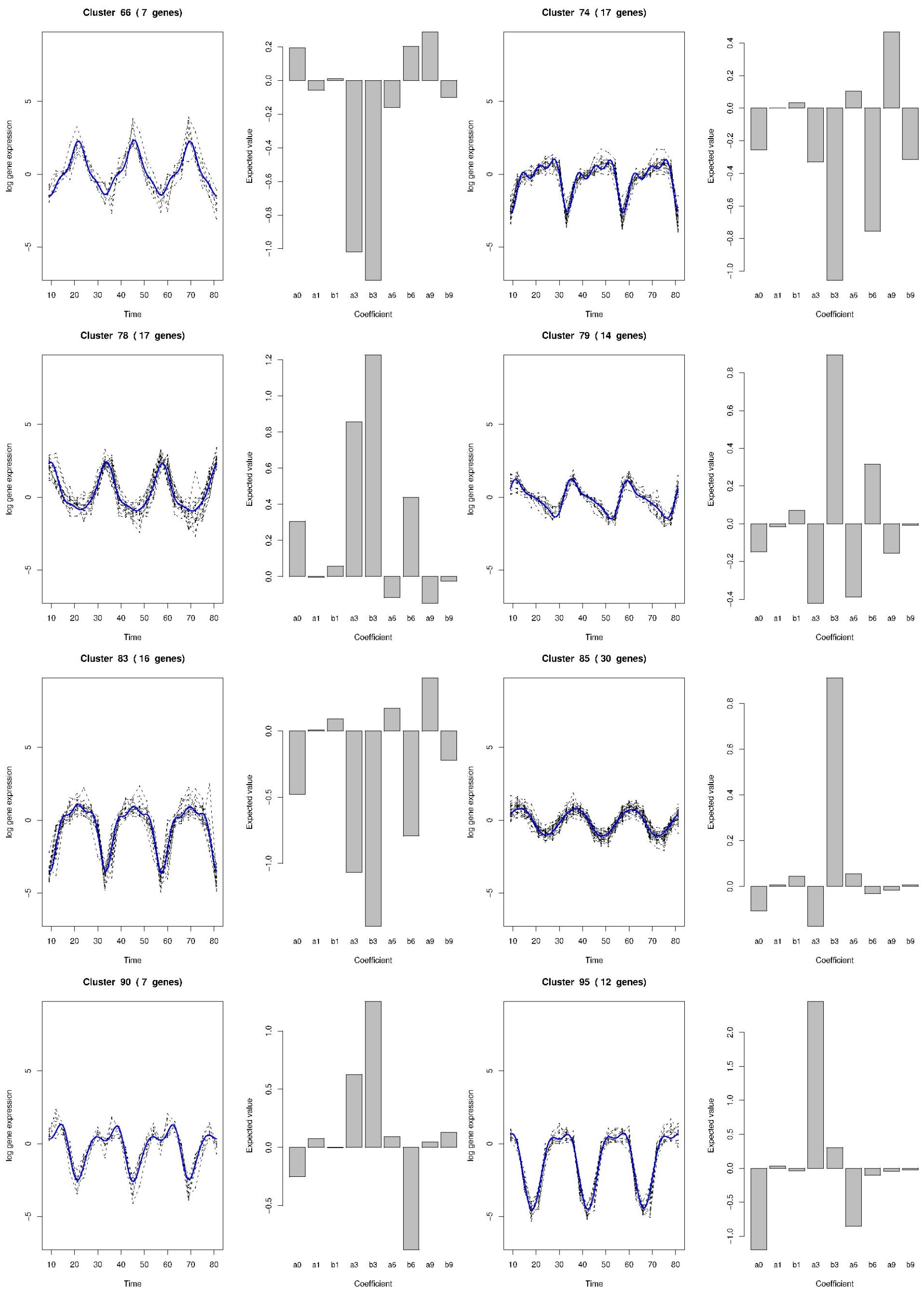


Figure 4.9: Some of the clusters for the *Ostreococcus* experiment.

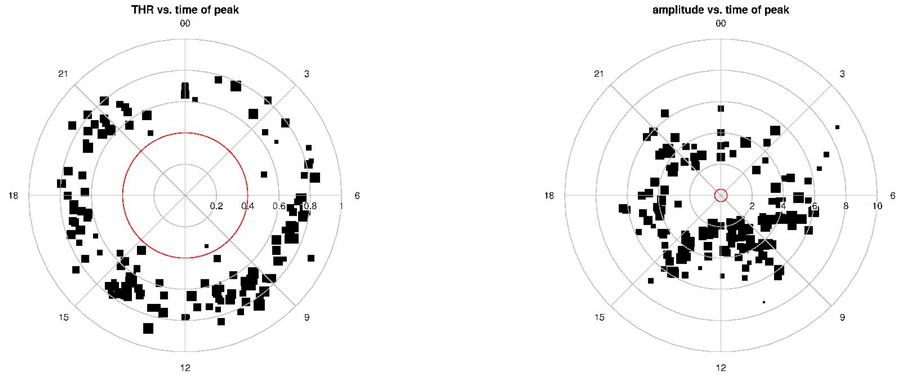


Figure 4.10: Radial plots where each squared dot represents a cluster, with the size of the dot proportional to the size of the cluster. The clusters are positioned around the clock depending on when they peak during the day. The distance from the centre depends on the value of the THR and amplitude respectively.

day. The distance from the centre depends on the value of the Third Harmonic Ratio (THR) and amplitude respectively. The amplitude is the difference between the highest value of the cluster and the lowest, whilst the THR represents the strength of the third harmonics, which is somehow measuring how strong the 'circadianity' is in each cluster. The THR is given by

$$\mathrm{THR} = \frac{(a_3^2 + b_3^2)^{\frac{1}{2}}}{\sum_{i=1,3,6,9}(a_i^2 + b_i^2)^{\frac{1}{2}}}, \tag{4.6}$$

where a_i and b_i for $i = 1, 3, 6, 9$ are the Fourier parameters corresponding to the first, third, sixth and nineth harmonics respectively. Only few genes are in clusters around time 0, similarly to what was observed by PCA, confirming a gap in transcription at this time of the day.

Our clustering algorithm revealed biological processes associated with specific clusters. Transcriptional coregulation of genes encoding mitochondrial/plastidial ribosomal protein is one the most striking example of a transcriptional network regulated by the photoperiod. For example, cluster 14 contains (26 probes) 11 genes encoding 70S plastid/mitochondria ribosomal protein. Cluster 18 (9 probes), which has an almost identical profile as cluster 14 has 3 plastid/mitochondria proteins and a chloroplast related IF2 translation initiation factor. Several genes involved in 80S ribosome biogenesis including RNA polymerase III, were overrepresented in Clusters 50 and 21. See Appendix A for figures and tables.

Classification of gene clusters according to their phase of expression revealed that transcript involved in specific biological processes, were associated with defined times windows during the day (see Fig. 4.10). For example, most genes implicated in protein synthesis had their phase of expression around dawn while replication genes peaked 3 hours before night and mitosis genes at dusk. To our knowledge, such a coregulation of biological processes has never been observed to this extent under 24hr light/dark cycle. These results are summarised in Fig. 4.11. This gave us a unique opportunity to get insight into the biology and physiology of O. tauri based on transcription profiling of the genome expression under light/dark conditions where most genes are expressed.

4.8 Conclusion

Our experiences suggest that simply getting hyperparameters in the right ball park as described above can dramatically improve the characteristics of these search algorithms. This is also shown in Chapter 5. Conjugate models with proportional parameter settings are not only fast but, if reasonably calibrated, behave appropriately. Even the occasional outlier can be identified and easily separated from the body of a cluster, iterating on the search algorithm if this is then necessary. The inconvenience in having to do this appears to us a small price to pay for the fast conjugate algorithm.

One useful spin-off of this analysis is that we have noted that for gene regulation, after a MAP partition has been found, the between-cluster statistic η is a useful summary. Thus clusters of genes that are potentially co-regulated can be expected to

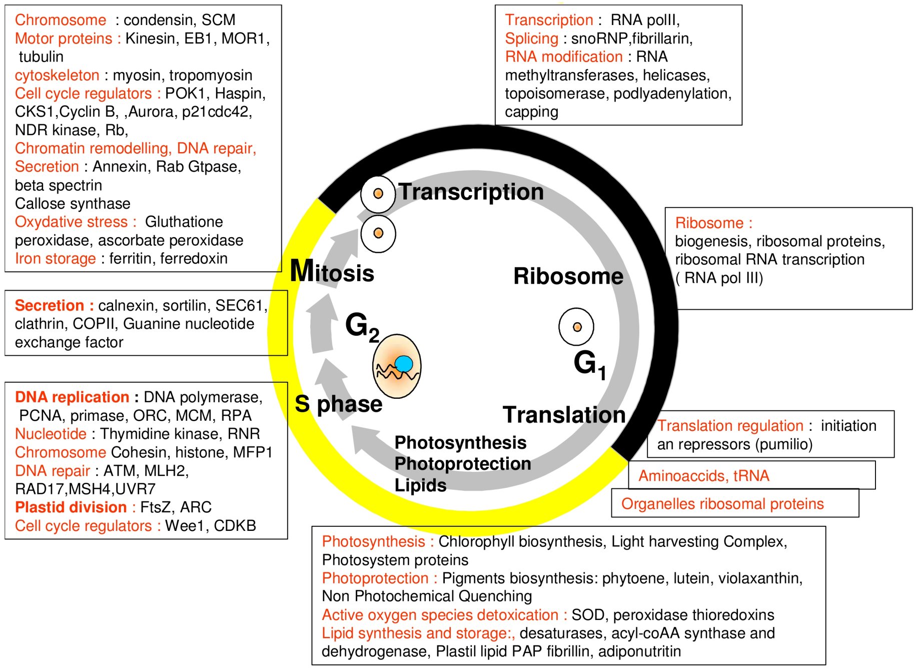

Chromosome : condensin, SCM
Motor proteins : Kinesin, EB1, MOR1, tubulin
cytoskeleton : myosin, tropomyosin
Cell cycle regulators : POK1, Haspin, CKS1,Cyclin B, ,Aurora, p21cdc42, NDR kinase, Rb,
Chromatin remodelling, DNA repair,
Secretion : Annexin, Rab Gtpase, beta spectrin
Callose synthase
Oxydative stress : Gluthatione peroxidase, ascorbate peroxidase
Iron storage : ferritin, ferredoxin
Secretion : calnexin, sortilin, SEC61, clathrin, COPII, Guanine nucleotide exchange factor
DNA replication : DNA polymerase, PCNA, primase, ORC, MCM, RPA
Nucleotide : Thymidine kinase, RNR
Chromosome Cohesin, histone, MFP1
DNA repair : ATM, MLH2, RAD17,MSH4,UVR7
Plastid division : FtsZ, ARC
Cell cycle regulators : Wee1, CDKB
Transcription : RNA polII,
Splicing : snoRNP,fibrillarin,
RNA modification : RNA methyltransferases, helicases, topoisomerase, podlyadenylation, capping
Ribosome : biogenesis, ribosomal proteins, ribosomal RNA transcription (RNA pol III)
Transcription
Mitosis
Ribosome
G_2
G_1
S phase
Translation
Photosynthesis
Photoprotection
Lipids
Translation regulation : initiation an repressors (pumilio)
Aminoaccids, tRNA
Organelles ribosomal proteins
Photosynthesis : Chlorophyll biosynthesis, Light harvesting Complex, Photosystem proteins
Photoprotection : Pigments biosynthesis: phytoene, lutein, violaxanthin, Non Photochemical Quenching
Active oxygen species detoxication : SOD, peroxidase thioredoxins
Lipid synthesis and storage:, desaturases, acyl-coAA synthase and dehydrogenase, Plastil lipid PAP fibrillin, adiponutritin

Figure 4.11: Overview of the transcriptional regulations of the main biological processes during a light/dark cycle. During the night actors of transcription, translation and protein synthesis are sequentially transcribed. During the light period, genes of Photosynthesis and lipid metabolism are transcriptionally coregulated. DNA repair and photoprotection genes are found in midday clusters. At the end of the day, specific transcriptional networks are associated with DNA replication and mitosis.

have similar profile shapes whilst the extent of the expressions, as measured by (ζ, d), is less biologically significant. Note that under Bayesian selection, provided search is extensive, all subsets of genes in a cluster will have similar associated values of η to other clusters and so this parameter not only characterises differences between clusters but also differences between collections of genes *within* clusters. This stability is important in this application since certain subsets of genes within clusters are of known biological function and therefore of more interest than others and would not be accounted for by other more ad-hoc methods. Note that the separation η between any two clusters is trivial to calculate given the previously computed statistics associated with the clusters in the MAP partition.

Finally, it is important to point out that although the problems addressed in this chapter are easy to *demonstrate* using a conjugate analysis, many are not simply a *consequence* of conjugacy but actually derive from a misinterpretation of a Bayes factor as a separation measure. There is every reason to believe that other non-conjugate selection based on Bayes factors and routinely chosen prior hyperparameters will also exhibit analogous unfortunate properties. Indeed, conjugate analysis has much useful symmetry which is destroyed by incorporating different priors. The effect of introducing this lack of symmetry through the use of non-conjugate models is likely to be influential to the selection, but very difficult to characterise so that the inevitably influential hyperparameters can be set appropriately. We speculate that most current numerical analogues of the models discussed here which exhibit the same qualitative hierarchical structure will be prone not to act as if guided by a separation measure. The same care is needed to ensure genuine prior predictive beliefs are specified, otherwise the *formal* selection (and not just its numerical approximation) is likely to be unstable in this more general setting.

In the next chapter we demonstrate how MAP selection can be further improved by localising the search for an optimal partition.

Chapter 5

Efficient Utility-based Clustering over High Dimensional Partition Spaces

Many Bayesian model selection procedures are based on the posterior probability distribution over models. Two very common methods are MAP selection, where the most a posteriori probable model is selected (Heard et al., 2006), and model mixing, where posterior probability distributions are calculated over the most promising candidate models and the results then mixed over these models (Fraley and Raftery, 1998). Here, for simplicity we will focus on the former, as in the previous chapter. In either case, a full exploration of the partition space is not possible when, as in our case, the number of elements in a cluster is in the order of tens of thousands, even when using fast conjugate modelling. The number of partitions of a set of n elements grows quickly with n. For example, there are 5.1×10^{13} ways to partition 20 elements.

In this chapter we assume that each cluster has a signature defining how scientifically interesting each cluster is. This context is quite common and in our case it was motivated by the need to cluster data from time course microarray data.

In our running example the scientists were only interested in discovering those genes whose expression profiles over two days exhibited circadian rhythms: other expression profiles were irrelevant, as we discussed in Chapter 2. Because of the enormous size of the search space, for the sake of efficiency, it looked promising to try to customize the search algorithms so that they reflect the scientific inquiry by focusing an algorithm to refine only clusters containing potentially interesting genes and not to waste time refining parts of a partition of no interest to the scientist. The question we answer in this chapter is: can this sort of procedure be formalized within a Bayesian framework?

MAP selection has a utility based justification. Bernardo and Smith (1994) introduce several perspectives on model comparison and in particular the $\mathcal{M}$-closed perspective. This scenario corresponds to believeing that one of the models available to us corresponds to the 'true' model. This seems reasonable in our context where we are searching the finite partition space. Then we define a zero-one utility function, equal to one when the true model is chosen. It follows that the optimal decision is to choose the model that has the highest posterior probability.

Bayesian selection techniques with specific priors have been limited to different classes of score functions (Denison et al., 2002). However there is a more pertinent literature - albeit specifically for Bayesian Networks, e.g. Tatman and Shachter (1990), springing from a subclass of multiattribute utility functions. This describes how, when the decision maker's utility function is separable, then, with the appropriate structure of prior, the search for an optimal decision can be localized, facilitating fast optimization. Using a utility led approach we will demonstrate how a modification of this algorithm also allows us to focus search on parts of the parameter space of importance to the scientist.

This chapter is organized as follows. In Section 5.1 we briefly discuss the class of conjugate Gaussian regression models introduced in Chapters 3 and 4: one of the types of model to which our methodology can apply. In Section 5.2 we introduce a formal framing of this genre of clustering problems in terms of multiattribute decision theory and discover a set of assumptions that will lead us to formally explain only parts of the underlying partition space. In Section 5.3 we show that if the product utility function is used, local search algorithms, widely used for conventional model exploration, are equally valid within this general framework. This means that the new utility based method is easy to implement. In Section 5.4 we briefly illustrate these methods through two examples. These concern a recent microrray experiment on the plant model organism *Arabidopsis thaliana*, designed to detect genes whose expression levels, and hence functionality, might be connected with circadian rhythms. The examples describe how our utility methods can be quickly applied to a very large dataset: here over 22,000 13-dimensional profiles were clustered.

5.1 A Clustering for Time-course Data

For the sake of simplicity, in this we illustrate our utility based approach in conjuction with a conjugate model developed by Heard et al. (2006), which we found particularly appealing, used in conjunction with the hyperpriors as defined in Chapter 4. We will use the notation outlined in Section 3.3.

Although other priors can be used in this context, in Chapter 4 we recommended the use of coherence priors over the partition space. Under these cohesion priors both the prior and posterior probability $\pi(\mathcal{C})$, where the generating partition is $\mathcal{C}$, has the form

$$\pi(\mathcal{C}|y) = A \prod_{c \in \mathcal{C}} \pi(c|y) \tag{5.1}$$

where A is a constant ensuring the probabilities of different possible partitions all sum to one.

Assuming the parameters of different clusters are independent, because the likelihood separates, it is straightforward to check (see Chapter 4) that the log marginal likelihood score S for any partition $\mathcal{C}$ with clusters $c \in \mathcal{C}$ is given by

$$S = \sum_{c \in \mathcal{C}} \log p_c(y) + \log \pi(\mathcal{C}) \tag{5.2}$$

where $\log \pi(\mathcal{C})$ is given in (5.1).

As discussed in Chapter 4, an essential property of the search for MAP models - dramatically increasing the efficiency of the partition search - is that with the right family of priors the search is *local*. That is, if $\widehat{\mathcal{C}}^+$ and $\widehat{\mathcal{C}}^-$ differ only in the sense that the cluster $\widehat{c}^+ \in \widehat{\mathcal{C}}^+$ is split into two clusters $\widehat{c}_1^-, \widehat{c}_2^- \in \widehat{\mathcal{C}}^-$ then the log marginal likelihood score is a linear function only of the posterior cluster probabilities on $\widehat{c}^+, \widehat{c}_1^-$ and $\widehat{c}_2^-$. We show in Section 5.3 that this local property is preserved when we use our utility based clustering method provided a product utility search is employed.

As mentioned in Chapter 3, the simplest search method using local search is agglomerative hierarchical clustering (AHC). Let us review it here briefly to clarify the notation used in this chapter. AHC starts with all the genes in separate clusters, our original $\mathcal{C}_0$, and evaluates the score of this partition. Each cluster is then compared with all the other clusters and the two clusters which increase the log likelihood in (5.2) by the most are combined to produce a new partition $\mathcal{C}_1$. We now substitute $\mathcal{C}_1$ for $\mathcal{C}_0$ and repeat this procedure to obtain a partition $\mathcal{C}_2$. We continue in this way until we have evaluated the logmarginal score $\Sigma(\mathcal{C}_i)$ for each partition $\{\mathcal{C}_i, 1 \leq i \leq N\}$. We then choose the partition which maximizes the score $\Sigma(\mathcal{C}_i)$.

A drawback of this method and ones like it is that the set of searched partitions is an extremely small subset of the set of all partitions. Moreover, no regard is taken by the algorithm of whether there is any scientific inferential merit in combining two clusters together. In our context an automatic search algorithm like AHC will spend the vast majority of its time examining the efficacy of combining two non-circadian gene clusters, an activity quite worthless from the scientific perspective. The motivation of this chapter is to try to find formal and efficient ways of addressing this obvious inadequacy of simple deterministic search.

5.2 Utility over Partitions

5.2.1 A Useful Class of Utilities

Our idea in this chapter is to use a utility function expressing the nature of the scientific interest to guide the search for the partition focusing on finding the partition with the highest posterior expected utility.

Let us generalize the notation introduced earlier for our running example. Let $\boldsymbol{\theta}_c$ be the vector of parameters associated with a cluster c. In our running example $\boldsymbol{\theta}_c$ is the vector of regression coefficients $\boldsymbol{\beta}_c$ and the variance term σ_c^2 . Let $\boldsymbol{\theta}(\mathcal{C}) = \{\boldsymbol{\theta}_c : c \in \mathcal{C}\}$ denote the vector of parameters associated with a given partition. Recall that under the usual model assumptions - both a priori and a posteriori - the density $\pi_c(\boldsymbol{\theta}_c)$ of $\boldsymbol{\theta}_c$ depends on the cluster index c but not on the partition $\mathcal{C}$ and that the vectors $\{\boldsymbol{\theta}_c : c \in \mathcal{C}\}$ are mutually independent of each other. Using this more general notation, it follows that the density $\pi(\boldsymbol{\theta}(\mathcal{C})|\mathcal{C}, y)$ can be written in the form

$$\pi(\boldsymbol{\theta}(\mathcal{C})|\mathcal{C}, y) = \prod_{c \in \mathcal{C}} \pi_c(\boldsymbol{\theta}_c|\mathcal{C}, y) = \prod_{c \in \mathcal{C}} \pi_c(\boldsymbol{\theta}_c|y) \tag{5.3}$$

The most complex family $\mathbb{U}$ of utility functions over many attributes in current use consists of utility functions $U(\widehat{\mathcal{C}}|\mathcal{C}, \boldsymbol{\theta}(\mathcal{C}))$ which exhibit mutually utility independent attributes (Keeney and Raiffa, 1976; Keeney and von Winterfeldt, 2007; French and Rios Insua, 2000). In our context, when each attribute is the expression profile of each gene $i \in \Omega$, by definition these utilities have the functional form

$$U\left(\widehat{\mathcal{C}}|\mathcal{C}, \boldsymbol{\theta}(\mathcal{C})\right) + 1 = \prod_{i \in \Omega}(1 + \kappa_i u_i(\widehat{c}|c, \boldsymbol{\theta}_c)) \tag{5.4}$$

where the conditional utility $u_i(\widehat{c}|c, \boldsymbol{\theta}_c)$ is the utility score of gene i when placed in cluster $\widehat{c}$ when in the generating partition $\mathcal{C}$ gene i lies in cluster $c \in \mathcal{C}$ and κ_i is the scaling constant for the single attribute utility function. See Keeney and Raiffa (1976) and French and Britain) (1989) for a more extensive discussion on utility functions and the derivation of additive and multiplicative utility function from the multi-attribute utility function.

The choice of a multiplicative utility function is the most appropriate in our context because it links in with the MAP model selection proposed by Heard et al. (2006), as we will show in Section 5.3. However, note there are other possible choices of utility independent attributes, such as the additive utility function (Keeney and Raiffa, 1976) given by

$$U(\mathcal{C}) = \sum_{i \in \Omega} \kappa_i u_i(\widehat{c}|c, \boldsymbol{\theta}_c) \tag{5.5}$$

if the score on each partition is defined as a function *separable* over the set of clusters. According to the definition by Tatman and Shachter (1990), a function $g(x)$ is separable if we can write it as

$$g(x) = \sum g_i(x_i) \quad \text{or} \quad g(x) = \prod g_i(x_i). \tag{5.6}$$

This is a nice property but the results presented in this chapter can only be obtained for product separable utilities over partitions. However, note that the additive utility function can only be applied when the restrictive condition of additive independence holds. See Keeney and von Winterfeldt (2007) and Keeney and Raiffa (1976) for causes of nonadditivity. We do not believe that teh additivity independence holds for our applications so we restrict ourselves to the study of product utility functions.

The relative magnitude of κ_i to κ_j reflects the importance the scientist places on gene i relative to gene j, and that as $\max \kappa_i \to 0$ this utility function tends to a linear one, whilst as $\min \kappa_i \to \infty$ we only score partitions which succeed in classifying *all* genes partially well. We now identify a subclass $\mathbb{V} \subseteq \mathbb{U}$ that on the one hand can plausibly embody the preference structure of a typical biologist investigating gene profiles and on the other provides a framework for more focused search algorithms over the partition space.

Thus suppose the scientist is prepared to state whether each given gene $i \in \Omega$ is potentially interesting - henceforth written $i \in I$ - or uninteresting - denoted here by $i \in \overline{I}$. When $U \in \mathbb{U}$, the implication of the above is that the scientist should set $\kappa_i = 0$ whenever $i \in \overline{I}$. Note that sometimes it will be appropriate to set $I = \Omega$. Let n_I denote the number of potentially interesting genes.

Definition 5.2.1. *Say a partition $\mathcal{C}$ of Ω is $I-$*simple *if all of its clusters c either have the property $c \cap I = c$ or $c \cap I = \emptyset$. Denote the set of all $I-$simple partitions by $\mathcal{S}(I)$.*

A partition $\mathcal{C}$ is $I-$simple if and only if I can be expressed in the form

$$I = \bigcup_{c \in \mathcal{C}(I)} c \tag{5.7}$$

where $\mathcal{C}(I)$ is a subset of the clusters c of $\mathcal{C}$ such that $c \cap I = c$. Obviously all partitions are $I-$simple when $I = \Omega$. Clearly for any partition $\widehat{\mathcal{C}_1}$ there is a partition $\widehat{\mathcal{C}_2} \in \mathcal{S}(I)$ such that $U(\widehat{\mathcal{C}_1}|\mathcal{C}, \boldsymbol{\theta}(\mathcal{C})) = U(\widehat{\mathcal{C}_2}|\mathcal{C}, \boldsymbol{\theta}(\mathcal{C}))$. We henceforth restrict our search to the partitions $\mathcal{C}$ that belong to $\mathcal{S}(I)$.

Let π_I denote the probability under the mass function (5.1) that the generating partition $\mathcal{C} \in \mathcal{S}(I)$. Then, if the scientist believes that $\pi_I = 1$, for any cluster c that does not satisfy $c \cap I = c$ or $c \cap I = \emptyset$, $\pi(c) = \pi(c|y) = 0$. A scientist making this assumption a priori believes that with probability one the generating partition will contain only clusters that inherit the label of being unambiguously interesting (i.e. containing only interesting genes) or unambiguously uninteresting (i.e. containing only uninteresting genes). This is a substantive but often plausible assumption. It embodies the belief that the definition of the term interesting is consistent with the underlying generating partition. If the scientist were not to hold this belief then it would bring into question whether a partition model should be used at all in the decision analysis. Note that this assumption simplifies the analysis because it allows the focus of the problem to switch from the individual units to the more coarse clusters of a partition.

It follows that we can write

$$\pi(\mathcal{C}|y) = \pi(\mathcal{C}|y, \mathcal{C} \in \mathcal{S}(I)) = \pi_1(\mathcal{C}(\overline{I})|y)\pi_2(\mathcal{C}(I)|y) \tag{5.8}$$

where $\mathcal{C}(\overline{I})$ is a partition of $\overline{I}$ and $\mathcal{C}(I)$ is a partition of I and

$$\begin{aligned} \pi_1(\mathcal{C}(\overline{I})|y) &= A_1 \prod_{c \notin \mathcal{C}(I)} \pi(c|y) \\ \pi_2(\mathcal{C}(I)|y) &= A_2 \prod_{c \in \mathcal{C}(I)} \pi(c|y) \end{aligned} \tag{5.9}$$

where $A_1$and A_2 are proportionality constants ensuring $\pi_1(\mathcal{C}(\overline{I})|y)$ and $\pi_2(\mathcal{C}(I)|y)$ are probability mass functions. So in particular any function of $\mathcal{C}$ depending only on the configuration of clusters in the partition $\mathcal{C}(I)$ of the interesting genes I and not those in $\mathcal{C}(\overline{I})$ of $\overline{I}$ will be independent of $\mathcal{C}(\overline{I})$.

Say that preferences are *cluster critical* if whenever $i \in \widehat{c} \neq c$ for all values of $\boldsymbol{\theta}_c$

$$u_i(\widehat{c}|c, \boldsymbol{\theta}_c) = 0 \tag{5.10}$$

A biologist's preferences will be consistent with this if for any gene i in the cluster $\widehat{c} \in \widehat{\mathcal{C}}(I)$ of interesting genes to contribute to the utility score, it is necessary for i to be classified correctly so that $\widehat{c} = c$. When the conditional utilities are cluster critical write

$$w_i(\widehat{c}|\boldsymbol{\theta}_{\widehat{c}}) \triangleq u_i(\widehat{c}|\widehat{c}, \boldsymbol{\theta}_{\widehat{c}}) \tag{5.11}$$

In this chapter we will also assume that the scientist's preferences over interesting genes within the same cluster are exchangeable. Thus assume that genes in I are *cluster exchangeable* meaning that

$$\kappa_i = \begin{cases} \kappa\phi(\widehat{c}) & \text{when } i \in I \cap \widehat{c} \\ 0 & \text{when } i \in \overline{I} \end{cases} \tag{5.12}$$

and that the genes in I that are cluster critical are *utility exchangeable* meaning that whenever $i, j \in \widehat{c}$

$$w_i(\widehat{c}|\boldsymbol{\theta}_{\widehat{c}}) = w_j(\widehat{c}|\boldsymbol{\theta}_{\widehat{c}}) \triangleq w_{\widehat{c}}(\boldsymbol{\theta}_{\widehat{c}}) \tag{5.13}$$

where the functions of the conditional utilities $0 \leq w_{\widehat{c}}(\boldsymbol{\theta}_{\widehat{c}}) \leq 1$ reflect how highly the gene $i \in \widehat{c}$ scores when i really lies in the cluster $\widehat{c}$ - with associated parameters θ_c - of the generating partition $\mathcal{C}$. Note that a least preferable estimate $\widehat{c}$ of c has $w_{\widehat{c}}(\boldsymbol{\theta}_{\widehat{c}}) = 0$ and the most preferable $w_{\widehat{c}}(\boldsymbol{\theta}_{\widehat{c}}) = 1$.

In Section 5.4 we use the functions $w_{\widehat{c}}(\boldsymbol{\theta}_{\widehat{c}})$ to preferentially weight the score of some genes in a potentially interesting cluster in terms of the values of the parameters $\boldsymbol{\theta}_{\widehat{c}}$: for example those with high expression and/or parameter values that suggest a clear diurnal pattern that would be associated with circadian regulatory genes. In particular, in Section 5.4 we will approximate this utility function by using a measure of the circadianity of genes over time. Further discussion on $w_{\widehat{c}}(\boldsymbol{\theta}_{\widehat{c}})$ is given in Section 5.3.2.

Definition 5.2.2. *Say a utility function $U \in \mathbb{U}$ is* amenable *if it is cluster critical and* *cluster and utility exchangeable - i.e. if the three conditions (5.10), (5.12) and (5.13) all hold. Denote the class of amenable utility functions by* $\mathbb{V}$*.*

Henceforth assume $U \in \mathbb{V}$. Then by definition, from (5.4) for decision $\widehat{\mathcal{C}} \in \mathcal{S}(I)$

$$U\left(\widehat{\mathcal{C}}|\mathcal{C}, \boldsymbol{\theta}(\mathcal{C})\right) + 1 = \prod_{i \in I}(1 + \kappa_i u_i(\widehat{c}|c, \boldsymbol{\theta}_c)) \tag{5.14}$$

which by cluster criticality can be written

$$U\left(\widehat{\mathcal{C}}|\mathcal{C}, \boldsymbol{\theta}(\mathcal{C})\right) + 1 = \prod_{i \in J(\widehat{\mathcal{C}})}(1 + \kappa_i w_i(\widehat{c}|\boldsymbol{\theta}_{\widehat{c}})) \tag{5.15}$$

where $J(\widehat{\mathcal{C}})$ is the set of genes correctly classified by $\widehat{\mathcal{C}}$ i.e.

$$J(\widehat{\mathcal{C}}) = \{i : i \in \widehat{c} = c\} \tag{5.16}$$

By cluster and utility exchangeability this now reduces to the form

$$U\left(\widehat{\mathcal{C}}|\mathcal{C}, \boldsymbol{\theta}(\mathcal{C})\right) = \prod_{\widehat{c} \in \widehat{\mathcal{C}}(\mathcal{C}, I)}(1 + \kappa\phi(\widehat{c})w_{\widehat{c}}(\boldsymbol{\theta}_{\widehat{c}}))^{n_{\widehat{c}}} - 1 \tag{5.17}$$

where $\widehat{\mathcal{C}}(\mathcal{C}, I)$ is the set of clusters correctly classified in $\widehat{\mathcal{C}}(I)$.

For reasons that will become apparent later we will call a scientist's preference weights $\phi(\widehat{c})$ *balanced* if $\phi(\widehat{c}) = n_{\widehat{c}}^{n_{\widehat{c}}^{-1}}$.

5.2.2 Marginal Search

When $U \in \mathbb{V}$, from (5.3) and (5.17) the expected utility $\overline{U}(\widehat{\mathcal{C}}|\mathcal{C})$ of choosing the partition $\widehat{\mathcal{C}} \in \mathcal{S}(I)$, when the true generating partition is $\mathcal{C}$, is given by

$$\begin{aligned}\overline{U}(\widehat{\mathcal{C}}|\mathcal{C}) &= \int U(\widehat{\mathcal{C}}|\mathcal{C}, \boldsymbol{\theta}(\mathcal{C}))\pi\left(\boldsymbol{\theta}(\mathcal{C})|\mathcal{C}, y\right) d\boldsymbol{\theta}(\mathcal{C}) \\ &= \prod_{\widehat{c} \in \widehat{\mathcal{C}}(\mathcal{C}, I)} \overline{u}(\widehat{c}) - 1 \qquad (5.18)\end{aligned}$$

where for each $\widehat{c} \in \widehat{\mathcal{C}}(I)$

$$\overline{u}(\widehat{c}) = \int (1 + \kappa\phi(\widehat{c}) w_{\widehat{c}}(\boldsymbol{\theta}_{\widehat{c}}))^{n_{\widehat{c}}} \pi_{\widehat{c}}(\boldsymbol{\theta}_{\widehat{c}}|y)\, d\boldsymbol{\theta}_{\widehat{c}} \tag{5.19}$$

Thus when the generating cluster is known, one plus the score of a simple partition is the product over scores obtained from each correct potentially interesting cluster.

Recall that $\widehat{\mathcal{C}}(\mathcal{C}, I) \subseteq \mathcal{C}(I)$ is the set of clusters correctly classified in $\widehat{\mathcal{C}}(I)$ and $\mathcal{C}(I)$ is a partition of I. Because by definition $\widehat{\mathcal{C}}(\mathcal{C}, I) \subseteq \mathcal{C}(I)$ note the score $\overline{U}(\widehat{\mathcal{C}}|\mathcal{C})$ depends on $\mathcal{C}$ only through $\mathcal{C}(I)$ and is independent of $\mathcal{C}(\overline{I})$ because I and $\overline{I}$ are disjoint sets. By the comments made after Equation (5.9) if the scientist a priori believes $\pi_I = 1$ then the expectation $\overline{U}(\widehat{\mathcal{C}})$ of $\overline{U}(\widehat{\mathcal{C}}|\mathcal{C})$ over $\mathcal{C}$ - the score we assign to $\widehat{\mathcal{C}}$ - only depends on our choice of $\widehat{\mathcal{C}}(I)$. In this scenario, investigating splits and combinations of clusters in $\mathcal{C}(\overline{I})$ is pointless since such moves cannot improve the score. Thus when $U \in \mathbb{V}$ and $\pi_I = 1$ there is no loss in restricting our moves between partitions $\widehat{\mathcal{C}}^+$ and $\widehat{\mathcal{C}}^-$ whose differential clusters $\widehat{c}^+ \in \widehat{\mathcal{C}}^+$ and $\widehat{c}_1^-, \widehat{c}_2^- \in \widehat{\mathcal{C}}^-$ lie in I.

Under the assumptions above we can therefore, without loss, simply search the partition space over the space I. However, in general, standard local search algorithms cannot be used for expected utility maximization because the local properties of this score function are lost. Nevertheless, in the next section we prove that the product utility function, which is a limit of the usual class of utility independent utilities, *does* retain this important property.

5.3 Properties of the Product Utility

5.3.1 Product Utilities and Local Moves

The product utility function is closely linked to the family $\mathbb{V}$ and also admits the simple evaluation of relative scores under local search.

Definition 5.3.1. *The* product utility function $U_I(\widehat{\mathcal{C}}|\mathcal{C},\boldsymbol{\theta}(\mathcal{C}))$ *on a set* $I \subseteq \Omega$ *has the form*

$$U_I(\widehat{\mathcal{C}}|\mathcal{C},\boldsymbol{\theta}(\mathcal{C})) = \prod_{c\in\mathcal{C}(I)} \phi(\widehat{c})^{n_{\widehat{c}}} u_{\widehat{c}}(\widehat{c}|c,\boldsymbol{\theta}_c)^{n_{\widehat{c}}} \tag{5.20}$$

where the conditional utilities $u_{\widehat{c}}(\widehat{c}|c,\boldsymbol{\theta}_c)$ *are cluster critical. Denote the set of product utility functions on* I *by* $\mathbb{V}_I$*.*

Note that in the notation developed above we can write a product utility function in the simplified form

$$U_I\left(\widehat{\mathcal{C}}|\mathcal{C},\boldsymbol{\theta}(\mathcal{C})\right) = \begin{cases} \prod_{\widehat{c}\in\widehat{\mathcal{C}}(I)} \{\phi(\widehat{c}) w_{\widehat{c}}(\boldsymbol{\theta}_{\widehat{c}})\}^{n_{\widehat{c}})} & \text{when } \widehat{\mathcal{C}}(I) = \mathcal{C}(I) \\ 0 & \text{otherwise} \end{cases} \tag{5.21}$$

where $w_{\widehat{c}}(\boldsymbol{\theta}_{\widehat{c}})$ is defined in (5.11).

The product utility function $U_I \in \mathbb{V}_I$ is a limit of a utility function $U \in \mathbb{V}$ in the following sense. For any partition $\mathcal{C} \in \mathcal{S}(I)$ write

$$\overline{U}_I(\widehat{\mathcal{C}}|\mathcal{C}) = \prod_{\widehat{c}\in\widehat{\mathcal{C}}(I)} v(\widehat{c}) \tag{5.22}$$

where for each $\widehat{c} \in \widehat{\mathcal{C}}(I)$

$$v(\widehat{c}) = \phi(\widehat{c})^{n_{\widehat{c}}} \int u_{\widehat{c}}(\widehat{c}|c,\boldsymbol{\theta}_c)^{n_{\widehat{c}}} \pi_c\left(\boldsymbol{\theta}_c|y\right) d\boldsymbol{\theta}_c \tag{5.23}$$

Recalling that n_I is the number of genes in interesting clusters, using the notation above we see that as $\kappa \to \infty$, and holding weights so that $\min\{\phi(\widehat{c}) : \widehat{c} \subseteq I\} \geq M$

$$\begin{aligned} \kappa^{-n_I}\left\{\overline{U}\left(\widehat{\mathcal{C}}|\mathcal{C}\right)+1\right\} &= \prod_{\widehat{c}\in\widehat{\mathcal{C}}(I)} \int \left(\kappa^{-1} + \phi(\widehat{c}) u_{\widehat{c}}\left(\widehat{c}|\widehat{c},\boldsymbol{\theta}_c\right)\right)^{n_{\widehat{c}}} \pi_c\left(\boldsymbol{\theta}_c|y\right) d\boldsymbol{\theta}_{\widehat{c}} \\ &= \overline{U}_I(\widehat{\mathcal{C}}|\mathcal{C}) + 0(\kappa^{-1}) \end{aligned} \tag{5.24}$$

provided $u_{\widehat{c}}(\widehat{c}|\widehat{c},\boldsymbol{\theta}_c) \geq \varepsilon > 0$ for all $c \in \mathcal{C}$. So a linear transformation of the expected utility score of a sequence of $U \in \mathbb{V}$ tends to that of a corresponding product utility score as the criterion weight on all the potentially interesting genes becomes large.

In addition to inheriting the interpretability of its parameters from $\mathbb{V}$ the $U_I \in \mathbb{V}_I$ also exhibits the property that its scoring is local. Because $U_I(\widehat{\mathcal{C}}|\mathcal{C}, \boldsymbol{\theta}(\mathcal{C})) = 0$, whenever $\widehat{\mathcal{C}} \neq \mathcal{C}$ letting

$$\overline{U}_I(\widehat{\mathcal{C}}) \triangleq \mathbb{E}\{\overline{U}_I(\widehat{\mathcal{C}}|\mathcal{C})\} \tag{5.25}$$

- the expected value of $\overline{U}_I(\widehat{\mathcal{C}}|\mathcal{C})$ over the possible generating partitions - we note that, since $\widehat{\mathcal{C}} \in \mathcal{S}(I)$, with any cohesion prior (5.1) on the partitions

$$\begin{aligned}
\overline{U}_I(\widehat{\mathcal{C}}) &= \overline{U}_I(\widehat{\mathcal{C}}|\mathcal{C})\pi(\widehat{\mathcal{C}} = \mathcal{C}, \mathcal{C} \in \mathcal{S}(I)|y) \\
&= \overline{U}_I(\widehat{\mathcal{C}}|\mathcal{C})\pi(\widehat{\mathcal{C}} = \mathcal{C}|y, \mathcal{C} \in \mathcal{S}(I))\pi_I|y \\
&= \prod_{\widehat{c} \in \widehat{\mathcal{C}}(I)} v(\widehat{c})\pi_2(\widehat{\mathcal{C}}(I)|y)\pi_I|y
\end{aligned}$$

So in particular on comparing the adjacent partitions $\widehat{\mathcal{C}}^+$ and $\widehat{\mathcal{C}}^- \in \mathcal{S}(I)$, $\overline{U}_I(\widehat{\mathcal{C}}^+) \geq \overline{U}_I(\widehat{\mathcal{C}}^-)$ if and only if

$$\log v(\widehat{c}^+) + \log \pi(\widehat{c}^+|y) \geq \log v(\widehat{c_1^-}) + \log \pi(\widehat{c_1^-}|y) + \log v(\widehat{c_2^-}) + \log \pi(\widehat{c_2^-}) \tag{5.26}$$

or equivalently

$$\log \pi(\widehat{c}^+|y) - \log \pi(\widehat{c_1^-}|y) - \log \pi(\widehat{c_2^-}|y) \geq \zeta \tag{5.27}$$

where

$$\zeta = \log v(\widehat{c_1^-}) + \log v(\widehat{c_2^-}) - \log v(\widehat{c}^+) \tag{5.28}$$

Whenever the parameter κ is large this property provides a vehicle for efficiently comparing the efficacy of adjacent $I-$simple partitions. Note that any optimal I-simple partition $\widehat{\mathcal{C}}$ will maximize

$$\mathbb{E}(\kappa^{-n_I}\{\overline{U}\left(\widehat{\mathcal{C}}|\mathcal{C}\right) + 1\}) = \overline{U}_I(\widehat{\mathcal{C}}) + 0(\kappa^{-1}). \tag{5.29}$$

So if

$$\log \pi(\widehat{c}^+|y) - \log \pi(\widehat{c_1^-}|y) - \log \pi(\widehat{c_2^-}|y) > \zeta \tag{5.30}$$

then there is a κ for which $\overline{U}(\widehat{\mathcal{C}}^{+}) > \overline{U}(\widehat{\mathcal{C}}^{-})$ and conversely if

$$\log \pi(\widehat{c}^{+}|y) - \log \pi(\widehat{c_1}^{-}|y) - \log \pi(\widehat{c_2}^{-}|y) < \zeta \tag{5.31}$$

then there is a κ for which $\overline{U}(\widehat{\mathcal{C}}^{+}) < \overline{U}(\widehat{\mathcal{C}}^{-})$.

Note that under this subfamily of utilities we do not need to assume that $\mathcal{C}$ is $I-$simple, just that we only search over $\widehat{\mathcal{C}}$ that are $I-$simple. This is because, if this $\widehat{\mathcal{C}}$ is $I-$simple but $\mathcal{C}$ is not, then $\mathcal{C} \neq \widehat{\mathcal{C}}$ which - unlike in the more general scenario - in turn implies $U_I(\widehat{\mathcal{C}}|\mathcal{C}, \boldsymbol{\theta}(\mathcal{C})) = 0$.

5.3.2 Relationships between Product Utility and MAP

The implications of the results are the following:

1. From the comments in Section 5.2.2, to find the utility maximising partition we need only find the utility maximising partition over the potentially interesting genes I.

2. Under the product utility function, like the MAP score, the expected utility score decomposes making it possible to use simple standard search algorithms to explore the space for high scoring partitions.

From (5.27) if the combined cluster in the coarser partition is more interesting than the two smaller clusters in the finer partition then we are more prepared to choose the coarser partition than under MAP. In this sense the local algorithm associated with a product utility can be seen as exactly a MAP search but over the genes in I and with adjusted priors over the partition space: the original prior cohesions $\pi_0(\widehat{c}^{+}), \pi_0(\widehat{c_1}^{-}), \pi_0(\widehat{c_2}^{-})$

in Equation (5.26) are simply replaced by the adjusted prior cohesions

$$\begin{aligned} \pi_0'(\widehat{c}^+|y) &= v(\widehat{c}^+)\pi_0(\widehat{c}^+|y) \\ \pi_0'(\widehat{c_1^-}|y) &= v(\widehat{c_1^-})\pi_0(\widehat{c_1^-}|y) \\ \pi_0'(\widehat{c_2^-}|y) &= v(\widehat{c_2^-})\pi_0(\widehat{c_2^-}|y) \end{aligned} \tag{5.32}$$

So, from an algorithmic perspective, searching for a simple partition maximising $\overline{U_I}\left(\mathcal{C}\right)$ is almost identical to searching for a MAP model over the subclass of potentially interesting genes, except that the most interesting clusters are given a higher prior weight than the less interesting ones. To simplify our notation henceforth write c for $\widehat{c}$.

It is interesting to note that under appropriate conditions we can now find prior densities and $U_I \in \mathbb{V}_I$ where an optimal partition $\mathcal{C}$ under U_I is a MAP optimal partition on I. Thus assume $\phi(c)$ are balanced. The weights defined in (5.23) are then of the form

$$v(c) = n_c \int w_c(\boldsymbol{\theta}_c)^{n_c} \pi_c\left(\boldsymbol{\theta}_c|y\right) d\boldsymbol{\theta}_c \tag{5.33}$$

Second assume that the marginal utilities on the parameters are indicators so that when $\boldsymbol{\theta}_c \in \Psi$ where Ψ is a particular region of the parameter space the scientist is satisfied whilst otherwise she is not. Then

$$w_c(\boldsymbol{\theta}_c) = \begin{cases} 1 & \text{when } \boldsymbol{\theta}_c \in \Psi \\ 0 & \text{when } \boldsymbol{\theta}_c \notin \Psi. \end{cases} \tag{5.34}$$

Note that the space Ψ can be defined as appropriate and it can incorprorate more than just one class of interesting genes. For example, it can be defined as to include gene expression profiles that are differentially expressed over time but do not have a circadian profile, as well as potentially circadian genes. In general the interesting genes are simply ones with weight different from zero and this is reasonable because in most problems there are signatures that the scientists are not interested in. On the other hand, the interesting genes can be given different weights and hence different degrees depending on the weights. However, for simplicity, we concentrate the discussion of the paper on the simpler case.

Under these conditions the threshold ζ defined in (5.28) can be written as

$$\zeta = \log P(\boldsymbol{\theta}_{c_1^-} \in \Psi|y) + \log P(\boldsymbol{\theta}_{c_2^-} \in \Psi|y) - \log P(\boldsymbol{\theta}_{c^+} \in \Psi|y) \tag{5.35}$$

In particular if we assume we have certainty ,i.e.

$$P(\boldsymbol{\theta}_{c_1^-} \in \Psi|y) = P(\boldsymbol{\theta}_{c_2^-} \in \Psi|y) = P(\boldsymbol{\theta}_{c^+} \in \Psi|y) = 1 \tag{5.36}$$

then $\zeta = 0$. So we recover MAP search but now restricted to I rather than Ω.

It is common practice in this context to first preselect genes that lie in a set I and then search for an optimal partition using MAP. So note that with the assumptions above this is a specific case of our method. Therefore our development above can be seen as providing a formal check about whether or not clustering combined with a particular preselection technique is valid and it also provides a way of adjusting this procedure when this is not so.

Note that we have shown that the preselection method is valid if the generating partition $\mathcal{C}$ is I-simple. When the scientist is not looking for specific structures, the sorts of routine preselection methods commonly used may well often be plausible. However when the scientist has strong beliefs about what shapes of data she is looking for, routine preselection will often not be consistent with the I-simple hypothesis. Furthermore, I will certainly be larger than it is needed for the analysis and so the search unnecessarily computational hungry.

Summarising we therefore have the following result.

Theorem 5.3.2. *The relative score between adjacent $I-$simple partitions under a $U \in \mathbb{V}$ (5.20) score and cohesion priors is the relative MAP score over all partitions of I using the adjusted cohesion priors over the partition space given by (5.32). In the particular case when conditions (5.33), (5.34) and (5.36) hold then this relative product utility score over I-simple partitions is exactly the MAP score on all partitions of I using the original priors on the partition space.*

Thus in the very special case when (5.33), (5.34) and (5.36) hold the optimal MAP partition found by local search on I will also be optimal under product utility score over the space of simple partitions. It is simply that there are other optimal partitions under the product utility: namely those that differ from the MAP partition but only on the clustering of the uninteresting genes. The fact that there are so many more optimal partitions under product utility means that we are more likely, with an efficacious search algorithm, to find a high scoring partition more quickly. So when $I = \Omega$ our utility based search inherits all the search efficiency of local MAP search, whilst when $I \subset \Omega$ and we are content to search only for $I-$simple partitions, our search algorithm can focus on partitions optimal with respect to I. Then, in fact, the utility based search is *much quicker* than MAP.

We also can conclude that if $I \subseteq J$ and under U_J a $J-$simple locally optimal partition $\mathcal{C}_J^*$ is also $I-$simple then $\mathcal{C}_J^*$ is also locally optimal under the utility function U_I. In this sense, if we include too many genes in our potentially interesting set this will affect the efficiency of our search but not the optimality. So there is a robustness to misspecification of the set I, provided we err on the side of caution and include genes in I if we think they might be interesting. An illustration of this methodology is given in the second example of Section 5.4.3.

5.3.3 Robustness of the Utility Weighted Score

Of course for most statistical models the certainty condition (5.36) will hold at best only approximately. Thus suppose that the scientist's utility has the form given by (5.33) and (5.34) but that we only know that, for all $c \in I$, $P(\boldsymbol{\theta}_c \in \Psi|y) \geq 1 - \alpha$ for some small value of α. If the interesting genes are discovered by thresholding then this rather than (5.36) may well be the type of condition we might have (see Section 5.4). Note that

$$-2\alpha \doteq 2\log(1-\alpha) \leq \zeta \leq -\log(1-\alpha) \doteq \alpha \tag{5.37}$$

It follows that when α is small and $I = \Omega$ then in this scenario our utility based search will closely approximate MAP search. So the method only performs significantly differently from MAP search in this context when either at least some of the clusters have weights α that are not small or when $I \subset \Omega$ which will be illustrated below.

Suppose therefore that, under the notation above

$$P(\boldsymbol{\theta}_{c_1^-} \in \Psi|y) \doteq P(\boldsymbol{\theta}_{c_2^-} \in \Psi|y) \doteq P(\boldsymbol{\theta}_{c^+} \in \Psi|y) = 1 - \alpha \tag{5.38}$$

Then $\zeta \doteq \log(1-\alpha)$ would mean that for large α we would combine clusters much more often than under MAP: i.e. the partition will be coarser over the less interesting genes in I.

5.3.4 Some Practical Issues

With balanced criterion weights we have that

$$v(c) = n_c \int u_c^0(\boldsymbol{\theta}_c)\pi_c\,(\boldsymbol{\theta}_c|y)\,d\boldsymbol{\theta}_c \tag{5.39}$$

where $u_c^0(\boldsymbol{\theta}_c) = w_c(\boldsymbol{\theta}_c)^{n_c}$. Note that $u_c^0(\boldsymbol{\theta}_c) = w_c(\boldsymbol{\theta}_c)$ if $w_c(\boldsymbol{\theta}_c)$ is an indicator function. In order to implement our search algorithm to be comparably fast with MAP search we need to be able to quickly evaluate $v(c)$. This then provides the thresholds ζ determining whether or not we move to an adjacent partition. We could approximate this function using summary statistics calculated already and so obtain an approximately optimal partition. Alternatively we could try to find functions $u_c^0(\boldsymbol{\theta}_c)$ which on the one hand reflect the preferences of the biologist and on the other admit the explicit calculation of $v(c)$.

In our running example we need to find expedient $u_c^0(\boldsymbol{\theta}_c)$ - when $\pi(\boldsymbol{\theta}_c|y)$ which has a product Gaussian - Inverse Gamma form - making $v(c)$ an explicit function of the hyperparameters of $\pi(\boldsymbol{\theta}_c|y)$. The second alternative is clearly more elegant, but we have found that the first option is more flexible and appears to be robust to the approximations we take.

5.4 Examples

To investigate the efficacy of this method we studied the circadian rhythms of the plant *Arabidopsis thaliana*. The experimental results were provided by Kieron D. Edwards and Andrew J. Millar and have been published in Edwards et al. (2006), although the analysis performed below is more refined than the original. We compare standard MAP methods used with AHC local search to our expected utility based search using adapted AHC on the same datasets.

We will illustrate our proposed method and its efficacy on a smaller example of 18 genes. Then we will show how an approximation of the methodology presented can be used on a larger example of tens of thousands of genes. For simplicity we will concentrate on an analysis where conditions (5.33), (5.34) and (5.36) are assumed to hold and we use an approximation rather than an exact evaluation of (5.39).

5.4.1 Data

The gene expression of 22,810 genes was measured by Affymetrix microarrays at 13 time points over two days. The aim was to identify the genes (of order 1,000) which may be connected with the circadian clock of the plant. After training the plants for two weeks to cycles of day and night, constant white light was shone on the plants for 26 hours before the first microarray was taken, with samples every four hours. The light remained on for the rest of the time course. Thus, there are two cycles of data (13 time points) for each of the 22,810 genes available on the *Arabidopsis* microarray chip. Subjective dawn occurs at about the 24th and 48th hours – this was when the plant has been trained to expect light after 12 hours of darkness. An exposition of the whole dataset, together with a discussion of its biological significance is given in Edwards et al. (2006) and subsequently by Michael et al. (2008).

The cluster profiles at time t, $y(t)$, over a 48 hour time course are given by

$$y(t) = \beta_0 + \sum_{i=1}^{6} \left[\beta_{2i-1} \cos\left(2\pi i t/48\right) + \beta_{2i} \sin\left(2\pi i t/48\right)\right] \tag{5.40}$$

5.4.2 A Simple Example of How Direct Use of AHC Fails

Here 15 genes, known to be circadian from the dataset above, were contaminated with 3 outliers. By outliers, in this case, we refer to genes that have a much higher amplitude and do not have a shape that can be associated with the genes that are usually involved in the circadian clock. Using AHC on the 18 genes produces two clusters (see Fig. 5.2). The second cluster contains all potentially interesting genes.

To compare this with our utility based approach we simply specified our set of interesting genes I as those whose individual first harmonic is a larger than the expected proportion of the total variation, here set to 0.25. Preselecting the set I in this coarse

Table 5.1: The score of the best partition of the interesting genes obtained with direct AHC and AHC but applied to potentially interesting gene only.

	$\Sigma(C)$
Direct AHC	64.896
AHC on I	68.295

way identifies the 15 genes in the second cluster in Fig. 5.1. However now using AHC on I leads to the further discrimination of the 15 genes into the two clusters in Fig. 5.2.

It is easy to see that both in terms of their Bayes factor scores and visually these new clusters discriminate profiles much better than AHC used directly. AHC is disrupted by outliers in larger problems in similar ways. See Chapter 4 for reasons for this. When potentially interesting profiles can be defined then even using a crude filter like the one illustrated above and then using a simple local search algorithm like AHC on I can greatly enhance the discovery process and classify interesting genes more precisely. We have seen earlier that proceeding in this way is formally justified provided the scientist has a utility as defined in Section 5.3 with equal utility weights.

5.4.3 A Simple Approximate Guided Learning Algorithm

We have shown in the previous example that AHC does not always succeed in identifying the best scoring partition and that our proposed utility method can enhance the clustering algorithm by formally allowing for the search to focus on interesting areas of the partition space. This is particularly important when dealing with high dimensional datasets, such as the whole dataset of 22,810 gene expression profiles of Arabidopsis.

It is usual to prefilter using either a simple expression threshold (Zhou et al., 2006) or a naive simple filter, such as the first harmonic in the Cosopt software (Straume, 2004) or prefiltering techniques as in Eisen et al. (1998), Tamayo et al. (1999), Wakefield et al. (2003) and Zhou et al. (2006). However in our context we found that prefiltering

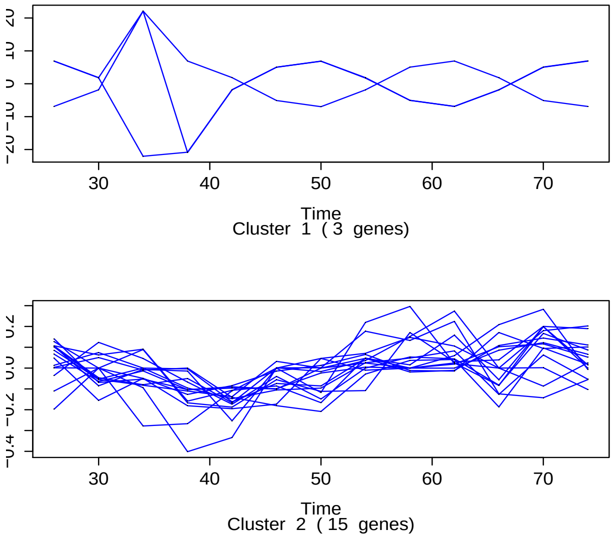


Figure 5.1: Clusters obtained on 18 genes of *Arabidopsis thaliana* running AHC only once on the whole dataset ($\Sigma(\mathcal{C}) = 64.896$). The y-axis is the log of gene expression. Note the different y-axis scale for the two clusters. This is the partition with the lowest score.

in this way removed a high proportion of genes whose profiles looked interesting to the biologist, because it was special shapes of harmonic, often non sinusoidal, profiles and sometimes relatively lowly expressed profiles which experience had suggested had biological regulatory importance. By performing the more refined filters of preclustering we were able to reduce the variance of estimated flexibly shaped profiles when these were replicated, helping to ensure that circadian but lowly expressed genes appeared in the set I we subsequentially searched.

First of all, we preclustered using the Bayes factors associated to a full Bayesian clustering algorithm on subsets of genes using the usual conjugate analyses by Heard et al. (2006) but adapted to a Fourier domain. We then treated the cluster parameter distributions as providing rough estimates of the profile of the *individual* genes contained in each particular cluster. We noticed that although cluster containment could be very sensitive to the setting of hyperparameters the estimates of individual gene profiles

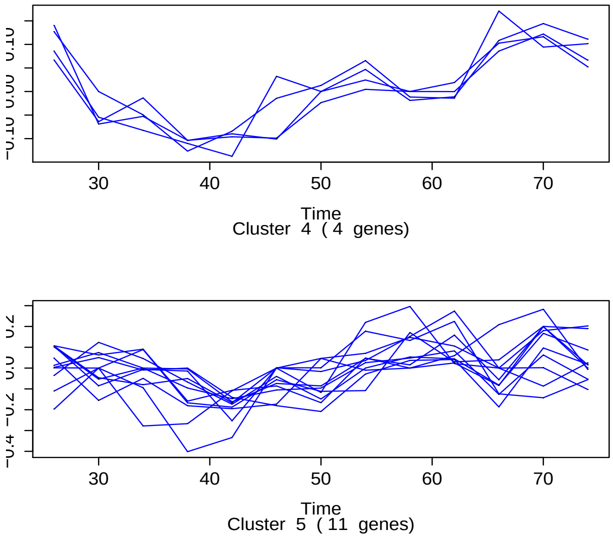


Figure 5.2: Running AHC again on the genes in I we do not search the partition space around the 3 outliers, but we find a higher scoring partition for the other 15 genes. The score of this new partition is $\Sigma(\mathcal{C}_{\text{Iterative}}) = 68.295$.

was remarkably robust to our hyperparameter settings: see Appendix B. The only exception to this was that, because of certain technical difficulties described in Chapter 4, these algorithms occasionally produced ‘junk’ clusters containing many genes with highly or moderately highly expressed, but heterogeneous, profiles. The cautious approach advocated in Section 5.3.2 therefore suggested we included these genes into the class I of interesting genes. So the set I constituted genes with well estimated profile means in interesting areas of the parameter space together with genes whose profile estimated variance was large.

There were several options for defining regions of potential interest. One was to use the posterior distribution of a measure of the interestingness of a cluster profile being greater than a given threshold using the MAP estimate of each gene. In our particular context biological expert judgement suggested that an interesting cluster is one whose second harmonic is high relative to the third, fourth, fifth and sixth harmonics.We define the *second harmonic ratio* (SHR) as

$$\mathsf{SHR} = \left(\beta_3^2 + \beta_4^2\right)^{\frac{1}{2}} \Bigg/ \sum_{i=1,3,5,7,9,11} \left(\beta_i^2 + \beta_{i+1}^2\right)^{\frac{1}{2}} \tag{5.41}$$

Because the distribution of this measure was not in closed form, provided the estimated variance of the regression parameters was not large, we approximated this by substituting the posterior means for their actual value in the thresholding formulae, as suggested in Section 5.3.4.

Once AHC has been used to discriminate the set I it is possible to use more refined search techniques on smaller sets. However, for the purpose of this illustration in this chapter we simply ran AHC again but now restricted to I. By doing this, we found that the contribution to the marginal likelihood over the set I of the final pass was much greater than that associated with the marginal likelihood over interesting genes found in the final run because outlying genes were largely sieved out through the iteration process. From Section 5.3 this means that the utility score for these new partitions was also greater. Full results of this final pass are given in Appendix B, where the clustering can be seen to be tight.

An example of the practical as well as theoretical usefulness of our utility based algorithm is illustrated in Fig. 5.3 using the unguided standard AHC method, whatever the values of the hyperparameters, the regulatory PKS1-like gene was always classified in a high variance no signal cluster like the one depicted in the first graph of Fig. 5.3. However, by first identifying the subset I of interesting genes the profile of this possibly regulatory gene is reclassified into a new cluster which is clearly circadian. Potentially useful possible homologues of the PKS1-like gene can now be identified as those genes whose profile lie in this cluster.

Note that our method does not use the data twice. It uses the early AHC just to provide coarse estimates of the individual gene profiles and then we implement

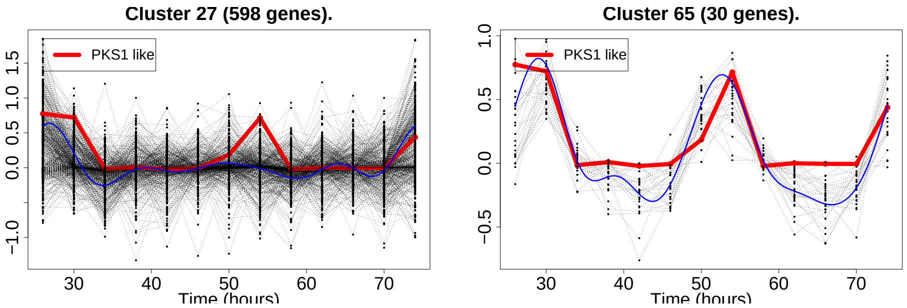


Figure 5.3: Demonstration of the advantages of the utility-based search algorithm. On the left is a potentially not interesting cluster from the penultimate step containing PKS1-like which biologists believe may be involved in the clock. The cluster on the right shows that PKS1-like ends up reclassified as a potentially circadian cluster after AHC has been reused on the subset I alone.

conservative bounds to ensure all possible interesting candidates are included. By doing so we increase the robustness of the method at a cost of efficiency. As I state in the chapter this step can be omitted but there is a trade off between catching all the genes and computational efficiency. The study of appropriate enaction of this trade-off is beyond the scope of this thesis but it could be studied using techniques of crossvalidation. However, through my informal experiences, I believe that, at least in the context of the given examples, the method described is very robust.

5.5 Discussion

Guided Bayesian clustering methods like the simple one described here clearly enhance the performance of Bayesian clustering algorithms for longitudinal time series. Our proposed methods can explore much larger relevant regions of the partition space and provide a useful, practical and formally defensible tool for the search of high dimensional partition spaces where the units in the partition are not exchangeable. Note that our techniques apply outside the narrow context of clustering gene profiles. Any clustering of large numbers of units can benefit from the approach discussed above, provided that the domain expert can be specific enough about her priorities to specify relative utility weights.

Of course there are significant further improvements that can be made to the methodology above. A first improvement is to weight the interesting genes, as described in the last section, rather than use a simple indicator discriminant. It is easy to do this if we approximate using the continuous score on the SHR obtained as a function of the means of the parameters in the penultimate iteration. Moreover, in the example we used an approximated guided algorithm, but those approximations are unnecessary for some expedient choices of utility functions. Instead of SHR, measures with a known distribution could have been used for precision rather than speed. Our results so far, though, showed that the practical gain in such exact methods, although measurable, was not great.

We have demonstrated in this chapter that MAP selection can be improved by localising the search for an optimal partition. In the next two chapters we focus on the refining of the search itself and we propose new algorithms that reformulate our clustering problem as a well-known problem studied in Artificial Intelligence (Chapter 6) and redefine the search over partitions in a general formal framework (Chapter 7).

Chapter 6

Searching a multivariate partition space using weighted MAX-SAT

The methodology presented so far has been based on the use of AHC. A full exploration of the partition space is not possible when, as in our case, the number of elements is in the order of tens of thousands, even when using fast conjugate modelling. The number of partitions of a set of n elements grows quickly with n, as pointed out at the beginning of Chapter 5. The intelligent search of the partition space is a challenge and in this chapter we demonstrate how to explore a partition space using weighted MAX-SAT. The SAT problem, which addresses whether a given set of propositional clauses is satisfiable, can be extended to the weighted MAX-SAT problem where weights are added to each clause and the goal is to find an assignment that maximises the sum of the weights of satisfied clauses. This problem setting has been used by Cussens (2008) for model search over Bayesian networks, a class of models which shares some similarities with the search over partitions. For example, in both scenarios, models are scored using a marginal likelihood which is *local* in the sense of Chapter 5 and *decomposable* (see

Section 5.1).

The advantage of algorithms encoding the weighted MAX-SAT methodology over many greedy search algorithms such as agglomerative hierarchical clustering (AHC) is that they are not intrinsically sequential. Under AHC once a decision to combine to clusters is made it cannot be reversed. This is not the case with weighted MAX-SAT solvers generally. In our illustrative examples this is a big advantage since under Bayes factor search via AHC early combinations of clusters are prone to be distorted by the presence of outliers (Chapter 4). On the other hand the advantage weighted MAX-SAT has over random search algorithms is that it is typically more efficient and finds local maxima of the Bayes score function for sure in a sense explained later in the chapter. Thus in small problems weighted MAX-SAT can be used to find an optimal partition for sure, whilst in large problems it can be used to enhance the performance of faster but less refined and adaptable algorithms.

Provided the appropriate local prior structure over the partition space is used a weighted MAX-SAT algorithm can be very flexible and can be used to search all spaces its competitors can. Here we will illustrate how this method can be used to cluster a class of time-course experiments known to exhibit circadian rhythms (Edwards et al., 2006).

This chapter is organised as follows. In Section 6.1 we illustrate the model used to score partitions and review the current methods used to search the partition space. Section 6.2 describes how the search on the partition space is encoded as a weighted MAX-SAT problem. We discuss some examples in Section 6.3 and present ongoing work on the reduction of the cluster scores in Section 6.4 .

6.1 Evaluating Partitions

The main contribution of this chapter is to encode the formal Bayes factor search on partitions as a weighted MAX-SAT problem and use well-known solvers for that problem to search over a multivariate partition space.

We use weighted MAX-SAT in conjunction with a conjugate Gaussian regression model developed by Heard et al. (2006). This model has a wide applicability because it can be customised through the choice of a given design matrix X. Conjugacy ensures the fast computation of scores for a given partition because these can be written explicitly and in closed form as functions of the data and the chosen values of the hyperparameters of the prior. Applications range from one-dimensional data points to multidimensional datasets with time dependence among points or where the points are obtained by applying different treatments to the units.

In this chapter we are going to refer to the notation introduced in Section 3.3. Note that in this chapter we use the hyperpriors as they were defined in Chapter 4. However, we do not include the utility function defined in Chapter 5, as we discuss later in Section 6.5.

6.1.1 Choosing an Appropriate Prior over Partitions

As discussed in Section 3.3.3, there are many possible choices for a prior over partitions. Partition priors can usually be separated into priors for each cluster and a constant term. The constant is often a function of the hyperparameters of the partition prior and the number of clusters, that is, it is a constant for partitions with the same number of clusters. For example, the Multinomial-Dirichlet prior 3.23 used by Heard et al. (2006)

can be rewritten as

$$\begin{aligned}\log p(\mathcal{C}) &= \log(N-1)! - \log n - \log(n+N-1)! + \sum_{k=1}^{N} \log n_k! \\ &= f(N, n, \alpha_1 = 1, \ldots, \alpha_N = 1) + \sum_{k=1}^{N} g(n_k, \alpha_1 = 1, \ldots, \alpha_N = 1)\end{aligned}$$

where f is a function of the number of clusters N, the number of observations to cluster n and the hyperparameters α_k while g is a function of the size of each cluster n_k and the hyperparameters α_k.

However, the implementation of our clustering algorithm in the context of MAX-SAT solvers requires the use of a partition prior whose $f(.)$ function defined above does not depend on the number of clusters N. An appropriate choice in this scenario is the Crowley partition prior $p(\mathcal{C})$ (Crowley, 1997; McCullagh and Yang, 2006; Booth et al., 2008) for partition $\mathcal{C}$,

$$p(\mathcal{C}) = \frac{\Gamma(\lambda)\lambda^N}{\Gamma(n+\lambda)} \prod_{i=1}^{N} \Gamma(n_i) \tag{6.1}$$

where $\lambda > 0$ is the parameter of the partition prior, N is the number of clusters and n is the total number of observations, with n_i the number of observations in cluster c_i. This prior is *consistent* in the sense of McCullagh and Yang (2006). The authors argue that this property is extremely desirable for any partition process to hold. Conveniently if we use a prior from this family then the score in (5.2) decomposes. Thus

$$\begin{aligned}\Sigma(\mathcal{C}) &= \log p(N, n_1, \ldots, n_N | y) \\ &= \log p(N, n_1, \ldots, n_N) + \sum_{i=1}^{N} \log p(y_i) \\ &= \log \Gamma(\lambda) - \log \Gamma(n+\lambda) + \sum_{i=1}^{N} S_i\end{aligned}$$

where

$$S_i = \log p(y_i) + \log \Gamma(n_i) + \log \lambda$$

Thus, the score $\Sigma(\mathcal{C})$ is *decomposable* into the sum of the scores S_i over individual clusters plus a constant term. This is especially useful for weighted MAX-SAT which needs the score of an object to be expressible as a sum of component scores. The choice of the Crowley prior in (6.1) ensures that the score of a partition is expressible as a linear combination of scores associated with individual sets within the partition. It is this property that enables us to find straightforward encoding of the MAP search as a weighted MAX-SAT problem.

Note that a particular example of a Crowley prior is the Multinomial-Dirichlet prior used by Heard et al. (2006), where λ is set so that $\lambda \in (1/n, 1/2)$.

6.1.2 Searching the Partition Space

The simplest search method using the *local* property is agglomerative hierarchical clustering (AHC) that chooses the partition which maximises the score $\Sigma(\mathcal{C}_i)$.

A drawback of this method is that the set of partitions searched is an extremely small subset of the set of all partitions. The number of partitions of a set of elements n grows quickly with n. For example, there are 5.1×10^{13} ways to partition 20 elements, and the AHC evaluates only 1331 of them!

As discussed also in Chapter 5, despite searching only a small number of partitions, AHC is surprisingly powerful and often finds good partitions of clusters, especially when used for time-course profile clustering as in the context of our example section. It is also very fast. However one drawback is that the final choice of optimal partition is completely dependent on the early combinations of elements into clusters. This initial part of the combination process is subject to be sensitive and can make poor initial choices, especially in the presence of outliers or poor choices of hyperparameters when used with Bayes factor scores in a way carefully described in Chapter 4.

Analogous instabilities in search algorithms over similar model spaces have prompted some authors to develop algorithms that devote time to early refinement of the initial choices in the search (Chipman et al., 2002) or to propose alternative stochastic search (Lau and Green, 2007). The latter method appears very promising but is difficult to implement within our framework due to the size of the datasets.

We propose an enhancement of the widely used AHC with weighted MAX-SAT. This is simple to use in this context provided a prior such as (6.1) is used over the model space which admits a decomposable score. Weighted MAX-SAT is able to explore many more partitions and different regions of the partition space and is not nearly as sensitive to the instabilities that AHC, used on its own, is prone to exhibit.

6.2 Encoding the Clustering Algorithm

Cussens (2008) showed that for the class of Bayesian networks a decomposition of the marginal likelihood score allowed weighted MAX-SAT algorithms to be used. The decomposition was in terms of child-parent configurations $p(x_i|\mathrm{Pa}_{x_i})$ associated with each random variable x_i in the Bayesian network. Here our partition space under the Crowley prior exhibits an analogous decomposition into cluster scores.

In the following Section 6.2.1 we introduce some concepts of propositional logic, the SAT problem, the weighted MAX-SAT and in Section 6.2.2 we present our encoding of clustering as a weighted MAX-SAT problem.

6.2.1 Weighted MAX-SAT

SAT (for satisfiability) is the problem of satisfying a set of clauses in propositional logic. MAX-SAT is the problem of maximising the number of clauses satisfied and it is an important and widely studied combinatorial optimisation problem with applications in

Artificial Intelligence and other areas of computing science. Below we introduce some concepts of propositional logic and then use them to introduce the SAT and MAX-SAT problems. This section is based on Hoos and Stützle (2005) and Büning and Lettmann (1999).

Propositional Clauses

A *propositional clause* is a *disjunction*: it states that at least one of a number of propositions is true. In logic the symbol for 'or' is $\vee$. So, for example,

$$x_1 \vee x_2$$

is a statement that one of x_1 or x_2 is true. Moreover, x_1 and x_2 are called *atomic formulae* or, for short, *atoms*, and they are the simplest statements in propositional logic: they assert that something is true and are not constructed from other formulae. The set of all atoms created for a particular problem defines what is called a *propositional language*.

Propositional logic also allows one to state that something is not true. The formula $\overline{x_1}$ asserts that x_1 is not true. Now for some terminology: a formula which is either a single atom or a negation of a single atom is called a *literal*. Evidently, literals come in exactly two types: *positive literals* (which are just atoms) and *negative literals* which are negations of atoms.

Clauses can have both positive and negative literals. The clause

$$x_1 \vee \overline{x_2}$$

states that either x_1 is true or $\overline{x_2}$ is true (or both). This clause is called a two-literal clause.

Assignments

An *assignment* associates a *truth value* to every atom in our chosen language. There are two truth values: TRUE and FALSE. In this way atoms can be seen as nothing more than binary variables. An assignment will either satisfy or fail to satisfy a clause. For example, the assignment $x_1 =$ TRUE, $x_2 =$ TRUE satisfies the clause

$$x_1 \vee \overline{x_2}$$

but the assignment $x_1 =$ FALSE, $x_2 =$ TRUE fails to satisfy it. Clearly, there are 2^n possible assignments for a language with n atoms. A clause can be seen as a constraint on assignments: assignments which fail to satisfy the clause are ruled out.

Conjunctive Normal Form

The symbol for 'and' is $\wedge$ so that the formula

$$(x_1 \vee \overline{x_2}) \wedge (x_2 \vee x_3) \tag{6.2}$$

is an assertion that the clause $(x_1 \vee \overline{x_2})$ and the clause $(x_2 \vee x_3)$ are both true. This is called a *conjunction*. A formula which is a conjunction of clauses (that is, a conjunction of disjunctions) is said to be in conjunctive normal form (CNF). It is a basic theorem of propositional logic that any formula in propositional logic can be reformulated to be in CNF.

Often CNF formulae are presented more informally as simply a list of clauses, so that the symbol $\wedge$ is just implicit. In this informal notation 6.2 can be rewritten as

$$\begin{array}{ccc} x_1 & \vee & \overline{x_2} \\ x_2 & \vee & x_3 \end{array}$$

Recall that a clause can be seen as a constraint on satisfying assignments. So, adding a clause to a CNF amounts to adding an extra constraint on assignments and the more clauses, the harder it is to satisfy a CNF.

The SAT Problem

There are many assignments that would satisfy the CNF in 6.2. Setting all three atoms to TRUE would do, for example. On the other hand, there are no satisfying assignments for this CNF with 3 clauses

$$\begin{aligned} x_1 \vee \overline{x_2} \\ \overline{x_1} \\ x_2 \end{aligned}$$

To satisfy the last two clauses we must have $x_1 =$ FALSE, $x_2 =$ TRUE, but this fails to satisfy the first clause. Note that the last two clauses have only one literal; such clauses are often called unit clauses, or facts.

Given a CNF formula, the SAT problem is to determine whether there exists an assignment satisfying it. All known algorithms for solving the SAT problem have running time exponential in the size of the CNF formula. It is widely believed that there is no fast (i.e. polynomial time) algorithm for solving SAT. Note, however, that checking whether any given assignment satisfies a CNF formula can be done very quickly. This asymmetry between checking a solution versus finding is the hallmark of NP (non-deterministic polynomial time) problems. A P-problem (whose solution time is bounded by a polynomial) is always also NP. If a problem is known to be NP, and a solution to the problem is somehow known, then demonstrating the correctness of the solution can always be reduced to a SAT is the classic NP problem (Hoos and Stützle, 2005).

Despite the intractability of the SAT problem in the worst case, many SAT problems have a structure which clever algorithms (SAT solvers) can exploit. By encoding a problem as a SAT problem one has access to these algorithms.

Weighted MAX-SAT

A weighted CNF formula is where each clause has a positive weight attached. This weight should be interpreted as a cost which is incurred by an assignment if that assignment does not satisfy the clause. The total cost of any assignment is the sum of the costs of clauses that assignment fails to satisfy.

The weighted MAX-SAT problem is an optimisation problem: find an assignment with minimal cost. Evidently, if all clauses can be satisfied by an assignment then that assignment is optimal: it has cost zero. It is useful to allow some clauses to be hard clauses. Such clauses have infinite cost - they must be satisfied if at all possible. Other clauses are soft - it would be nice to satisfy them but an optimal assignment might break them.

In practice it is not possible to represent infinite weights, so hard clauses are just given a weight which is sufficiently big that it gets treated as if it were infinite.

As with the SAT problem, there are many solvers available for the weighted MAX-SAT problem. In particular, UBCSAT (Tompkins and Hoos, 2005) is an implementation and experimentation environment for Stochastic Local Search (SLS) algorithms for SAT and MAX-SAT. UBCSAT provides implementations of numerous well-known and widely used SLS algorithms for SAT and MAX-SAT, including GSAT, WalkSAT, and SAPS; these implementations generally match or exceed the efficiency of the respective original reference implementations. UBCSAT is implemented in C and runs on numerous platforms and operating systems; it is publicly and freely available.

We can now encode the clustering problem as a weighted MAX-SAT problem as follows.

6.2.2 Weighted MAX-SAT Encoding for Clustering

For each considered cluster c_i, a propositional atom, also called c_i, is created. In what follows no distinction is made between clusters and the propositional atoms representing them. Propositional atoms are just binary variables with two values: TRUE and FALSE. A partition is represented by setting all of its clusters to TRUE and all other clusters to FALSE.

However, most truth-value assignments for the c_i do not correspond to a valid partition, and so such assignments must be ruled out by constraints represented by logical clauses. To rule out the inclusion of overlapping clusters we assert clauses of the form:

$$\overline{c_i} \vee \overline{c_j} \tag{6.3}$$

for all non-disjoint pairs of clusters c_i, c_j. (A bar over a formula represents negation.) Each such clause is logically equivalent to $\overline{c_i \wedge c_j}$: both clusters cannot be included in a partition.

In general, it is also necessary to state that each data point must be included in some cluster in the partition. Let $\{c_{y_1}, c_{y_2}, \ldots, c_{y_{i(y)}}\}$ be the set of all clusters containing data point y. For each y a single clause of the form:

$$c_{y_1} \vee c_{y_2} \vee \cdots \vee c_{y_{i(y)}} \tag{6.4}$$

is created.

The 'hard' clauses in (6.3) and (6.4) suffice to rule out non-partitions; it remains to ensure that each partition has the right score. This can be done by exploiting the decomposability of the partition score into cluster scores and using ‘soft’ clauses to represent cluster scores. If S_i, the score for cluster c_i, is positive the following weighted clause is asserted:

$$S_i : c_i \tag{6.5}$$

Such a clause intuitively says: “We want c_i to be true (i.e. to be one of the clusters in the partition) and this preference has weight S_i.” If a cluster c_j has a negative score S_j then this weighted clause is asserted:

$$-S_j : c_j \tag{6.6}$$

which states a preference for c_j not to be included in the partition. Given an input composed of the clauses in (6.3)–(6.6) the task of a weighted MAX-SAT solver is to find a truth assignment to the c_i which respects all hard clauses and maximises the sum of the weights of satisfied soft clauses. Such an assignment will encode the highest scoring partition constructed from the given clusters.

Note that if a given cluster c_i can be partitioned into clusters $c_{i_1}, c_{i_2}, \dots c_{i_{j(i)}}$ where $S_i < S_{i_1} + S_{i_2} + \dots + S_{i_{j(i)}}$, then due to the decomposability of the partition score, c_i cannot be a member of any optimal partition: any partition with c_i can be improved by replacing c_i with $c_{i_1}, c_{i_2}, \dots c_{i_{j(i)}}$. Removing such clusters prior to the logical encoding reduces the problem considerably and can be done reasonably quickly: for example, one particular collection of 1023 clusters which would have generated 495,285 clauses was reduced to 166 clusters with 13,158 clauses using this approach. The filtering process took 25 seconds using a Python script. This cluster reduction technique was used in addition to those mentioned in the sections immediately following.

6.2.3 Reducing the Number of Cluster Scores

To use weighted MAX-SAT algorithms effectively in this context, the challenge in even moderately sized partition spaces is to identify promising clusters that might be components of an optimal partition. The method in Cussens (2008) of evaluating the scores only of subsets of less than a certain size is not ideal to this context since in our applications many good clusters appear to have a high cardinality.

However there are more promising techniques formulated in other contexts to address this issue. One of these, which we use in the illustrative example, is outlined below and others presented in Section 6.4.

Reduction by Iterative Augmentation

A simple way to reduce the number of potential cluster scores for weighted MAX-SAT is to evaluate all the possible clusters containing a single observation and to iteratively augment the size of the plausible clusters only if their score increases too, thanks to the nice decomposability of our score function. We will focus our discussion in this chapter to an algorithm, the iterative augmentation algorithm described below.

Step 1 Compute the cluster score for all n observations as if each belonged to a different cluster. Save these scores as input for weighted MAX-SAT. Set $k \leftarrow 0$ and $c \leftarrow \emptyset$.

Step 2 Set $k \leftarrow k+1$, $j \leftarrow k+1$ and $c \leftarrow \{k\}$. Exit the algorithm when $k = n$.

Step 3 Add element j to cluster c and compute the score for this new cluster c'. If $S_{c'} > S_c + S_j$, then

- Save the score for cluster c'
- If $j = n$, go to Step 2.

- $c \leftarrow c'$ and $j \leftarrow j+1$
- Go to Step 3

else

- If $j = n$, go to Step 2.
- Set $j \leftarrow j+1$
- Go to Step 2.

The main advantage of this algorithm is that it evaluates the actual cluster scores, never approximating them by pairwise dissimilarities or in any other way. Furthermore, this method does not put any restriction on the maximum size of the potential clusters.

Hybrid AHC Algorithm

Even though this algorithm performs extremely well when the number of clustered units $n < 100$, it slows down quickly as the number of observational vectors increases. However this deficiency disappears if we use it in conjunction with the popular AHC search to refine clusters of less than 100 units. When used to compare partitions of profiles as described in Section 5.1, AHC performs extremely well when the combined clusters are large. So to improve its performance we use weighted MAX-SAT to reduce dependence on poor initialisation. By running a mixture of AHC together with weighted MAX-SAT we are able to reduce the dependence whilst retaining the speed of AHC and its efficacy with large clusters. AHC is used to initialise a candidate partition. Then weighted MAX-SAT is used as a 'split' move to refine these clusters and find a new and improved partition on which to start a new AHC algorithm. The hybrid algorithm is described below.

Step 1 Initialise by running AHC to find best scoring partition $\mathcal{C}_1$ on this search.

Step 2 *(Splitting step)* Take each cluster c in $\mathcal{C}_1$. Score promising subsets of c and run a weighted MAX-SAT solver to find the highest scoring partition of c. Note that, because our clusters are usually several orders of magnitude smaller than the whole set, this step will be feasible at least for interesting clusters.

Step 3 Substitute all the best sub-clusters of each cluster c in $\mathcal{C}_1$ to form next partition $\mathcal{C}_2$.

Step 4 If $\mathcal{C}_1 = \mathcal{C}_2$ (i.e. if the best sub-cluster for each cluster in $\mathcal{C}_1$ is the cluster itself) then stop.

Step 5 *(Combining step)* If this is not the case then by the linearity of the score $\mathcal{C}_2$ must be higher scoring than $\mathcal{C}_1$. Now take $\mathcal{C}_2$ and - beginning with this starting partition to test combinations of clusters in $\mathcal{C}_2$ - using AHC. (Note we could alternatively use weighted MAX-SAT here as well). This step may combine together spuriously clustered observations that initially appeared in different clusters of $\mathcal{C}_1$ and were thrown out of these clusters in the first weighted MAX-SAT step. Find the optimal partition $\mathcal{C}_3$ doing this.

Step 6 If $\mathcal{C}_3 = \mathcal{C}_2$ stop, otherwise go to Step 2.

This hybrid algorithm obviously performs at least as well as AHC and is able to undo any early erroneous combination of AHC. The shortcomings of AHC, discussed in Chapter 4, are overcome by checking each cluster running weighted MAX-SAT to identify outliers. Note that the method is fast because weighted MAX-SAT is only run on subsets of small cardinalities. We note that at least in the applications that we have encountered most clusters of interest appear to contain less than a hundred units.

6.3 An Application to a Time-course Microarray Experiment

We will illustrate the implementation of weighted MAX-SAT for clustering problems in comparison to and in conjunction to the widely used AHC.

Here we demonstrate that weighted MAX-SAT can be used to cluster time-course gene expression data. The cluster scores are computed in C++ using the algorithm that we implemented and used in Section 4.7, modified for the use of the Crowley prior. The graphical output is obtained using R (R Development Core Team, 2009). All runs of weighted MAX-SAT were conducted using the C implementation available from the UBCSAT home page http://www.satlib.org/ubcsat.

The literature on weighted MAX-SAT solvers is extensive and many solvers have been proposed and improved over the years. The interest of researchers in computer science and artificial intelligence focuses on the solvers' efficiency and their ability to output the solution which maximises the sum of the weights. However, it should be noted that the proofs in this context are empirical rather than mathematical. Over the years there have been many competitions all over the world to compare weighted MAX-SAT solvers, evaluate them, and study their performance in different settings. See for example McAllester et al. (1997) and Heras et al. (2008). For our clustering problem we have compared the results obtained with several of the solvers available from the UBCSAT web page. We have used their implementation of WalkSat in this chapter.

6.3.1 Data

Our algorithm will be illustrated by an example on a recent microarray experiment on the plant model organism *Arabidopsis thaliana*. This experiment was designed to detect genes whose expression levels, and hence functionality, might be connected with circadian rhythms. The aim is to identify the genes (of order 1,000) which may be connected with the circadian clock of the plant. A full analysis and exposition of this data, together with a discussion of its biological significance is given in Edwards et al. (2006).

We will illustrate our algorithms on genes selected from this experiment. The gene expression of $n = 22,810$ genes was measured at $r = 13$ time points over two days by Affymetrix microarrays. Constant white light was shone on the plants for 26 hours before the first microarray was taken, with samples every four hours. The light remained on for the rest of the time course. Thus, there are two cycles of data for each of the *Arabidopsis* microarray chip. Subjective dawn occurs at about the 24th and 48th hours – this is when the plant has been trained to expect light after 12 hours of darkness.

6.3.2 Hybrid AHC Using Weighted MAX-SAT

Although our clustering algorithms apply to a huge space of over 22,000 gene profiles, to illustrate the efficacy of our hybrid method it is sufficient to show results on a small subset of the genes: here a proxy for two clusters. Thus we will illustrate how our hybrid algorithm can outperform AHC and how it rectifies partitions containing genes clustered spuriously in an initial step. In the example below we have therefore selected 15 circadian genes from the dataset above and contaminated these with 3 outliers that we generated artificially.

We set the parameters $v = 10$, $a = 0.001$, $b = 0.001$ and $\lambda = 0.5$ and ran AHC which obtained the partition formed by 2 clusters shown in Fig. 6.1. AHC is partially successful: the 15 circadian genes have been clustered together, and so have the 3 outliers. The latter cluster is a typical example of misclassification in the sense of Chapter 4 in that it is rather coarse with a relatively high associated variance. The score for this partition is $\Sigma(\mathcal{C}_{\textsf{AHC}}) = 64.89565$.

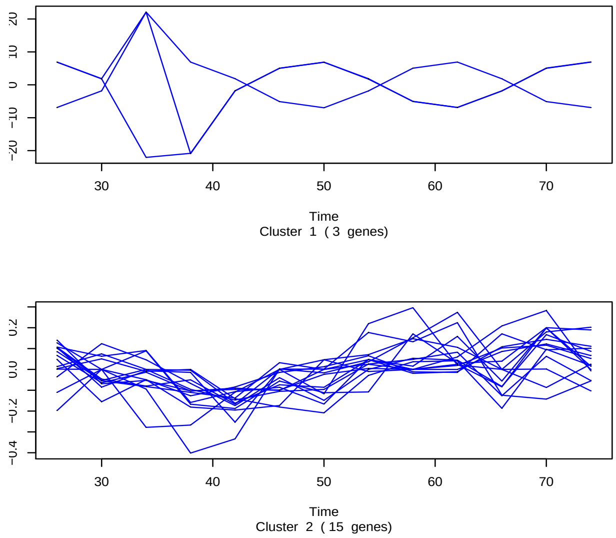


Figure 6.1: Clusters obtained on 18 genes of *Arabidopsis thaliana* using AHC ($\Sigma(\mathcal{C}_{\textsf{AHC}}) = 64.89565$). The y-axis is the log of gene expression. Note the different y-axis scale for the two clusters.

Following the hybrid AHC algorithm we then ran MAX-SAT on both the clusters obtained by AHC. The clusters obtained are shown in Fig. 6.2 and 6.3. Both the clusters obtained by AHC have been split up by MAX-SAT. The score of the partition formed by these 5 clusters, including the constants, is now $\Sigma(\mathcal{C}_{\textsf{MAX-SAT}}) = 79.43005$. This is the log of the marginal likelihood and taking the appropriate exponential, in terms of Bayes factor, this represents a decisive improvement for our model. Note that the increase in the log marginal likelihood is supported also by the visual display. The outliers are very different between themselves and from the real data and it seems reasonable that each one would generate a better cluster on its own - note the different scale of the y-axis. The other 15 genes have a more similar shape and it seems visually reasonable to cluster them together, as AHC does initially, but MAX-SAT is able to identify a more

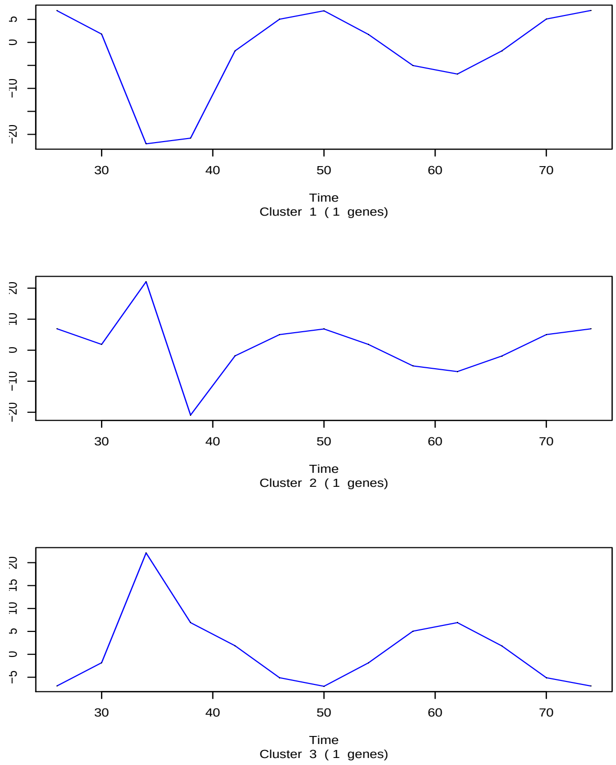


Figure 6.2: Clusters obtained on 3 outliers of *Arabidopsis thaliana* using AHC (1 cluster, $S_1 = -156.706$) and weighted MAX-SAT (3 cluster, $S_1 = -145.571$).

subtle difference between 2 shapes contained in that cluster. It was not necessary in our case to run AHC again to combine clusters, given the nature of our data. A single iteration of the loop described in our hybrid algorithm identified the useful refinement of the original partition.

This example shows how, as discussed in Chapter 4, AHC can be unstable especially when dealing with outliers at an early stage in the clustering. The weighted MAX-SAT is helpful to refine the algorithm, and obtain a higher scoring partition.

It is clear that in larger examples involving thousands of genes the improvements above add up over all moderate sized clusters of an initial partition, by simply using weighted MAX-SAT over each cluster in the partition, as described in our algorithm and

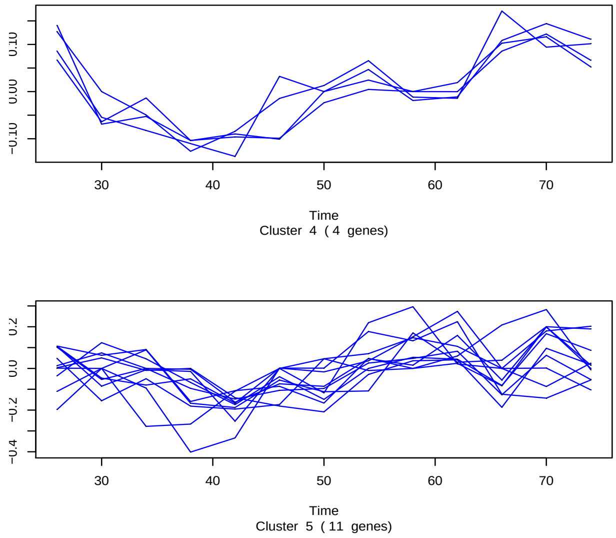


Figure 6.3: Clusters obtained on 15 genes of *Arabidopsis thaliana* using AHC (1 cluster, $S_2 = 255.973$) and weighted MAX-SAT (2 clusters, $S_2 = 259.372$).

illustrated above.

6.4 Further Work on Cluster Scores for Large Clusters

In the approach taken in this chapter clusters are explicitly represented as propositional atoms in the weighted MAX-SAT encoding and so it is important to reduce the number of clusters considered as much as possible. The hybrid method with iterative augmentation that we have described in Section 6.2.3 works very efficiently for splitting clusters with cardinality smaller than 100. However it slows down dramatically for greater cardinalities. It would be useful to generalise the approach so that it can also be employed to split up larger clusters. The main challenge here is to identify good candidate sets. Two methods that we are currently investigating are outlined below.

Reducing Cluster Scores Using Cliques

One promising method for identifying candidate clusters is to use a graphical approach based on pairwise proximity between the clustered units. Ben-Dor et al. (1999) - a well known and highly cited paper - proposes the CAST algorithm to identify the clique graph which is closest to the graph obtained from the proximity matrix. A graph is called a clique graph if it is a disjoint union of complete graphs. The disjoint cliques obtained by the CAST algorithm define the partition.

We suggest using an approach similar to Ben-Dor et al. (1999), enhanced by the use of weighted MAX-SAT and a fully Bayesian model.

We focus on maximal cliques, instead of clique graphs as in Ben-Dor et al. (1999), to identify possible clusters to feed into weighted MAX-SAT. A maximal clique is a set of vertices that induces a complete subgraph, and that is not a subset of the vertices of any larger complete subgraph. The idea is to create an undirected graph based on the adjacency matrix obtained by scoring each pair of observations as a possible cluster and then use the maximal cliques of this graph to find plausible clusters. It is reasonable to assume that a group of elements is really close and should belong to the same cluster when it forms a clique. This considerably reduces the number of clusters that need to be evaluated and are the input for weighted MAX-SAT, which will then identify the highest scoring partition.

The first step is to calculate the proximity between observations i and j $(i, j = 1, \ldots, n)$ such as

$$D = \{d_{ij}\} = S_{ij} - (S_i + S_j)$$

which gives a matrix of adjacencies A

$$A = \{a_{ij}\} = \begin{cases} 1 & \text{if } d_{ij} > K \\ 0 & \text{otherwise} \end{cases}$$

from which we can draw a graph (S_{ij} is the score for the cluster of 2 elements, i and j). Each vertex represents an observation. Two vertices are connected by an edge according to matrix D. The adjacency matrix defines an undirected graph. The maximal cliques, the intersections between maximal cliques and the union of maximal cliques with common elements define the potential cluster scores for weighted MAX-SAT.

Although such methods are deficient in the sense that they use only pairwise relationships within putative clusters, they identify potentially high scoring clusters quickly. Of course, it does not matter whether some of these clusters turn out to be low scoring within this candidate set, because each is subsequently fully scored for weighted MAX-SAT and their deficiency identified. This is in contrast to the method of Ben-Dor et al. (1999) which is completely based on pairwise dissimilarities. So the only difficulty with this approach is induced by those clusters which are actually high scoring but nevertheless are not identified as promising.

Other advantages of this method are that all the scores that are calculated are used as weights in the weighted MAX-SAT and it does not induce any artificial constraint on cluster cardinalities.

We are currently investigating ways of improving this. It still needs some other simplifying ideas before MAX-SAT can be a viable alternative to AHC rather than a way of refining partitions previously discarded.

Reducing Cluster Scores by Approximating

An alternative to the method described above is to represent the equivalence relation given by a partition directly: for each distinct pair of data points y_i, y_j, an atom $a_{i,j}$ would be created to mean that these two data points are in the same cluster. Only $O(n^2)$ such atoms are needed. Hard clauses ($O(n^3)$ of them) expressing the transitivity of the equivalence relation would have to be added. With this approach it might be possible to indirectly include information on cluster scores by *approximating* cluster scores by a quadratic function of the data points in it. A second-order Taylor approximation is an obvious choice. Such an approach would be improved by using a different approximating function for each cluster size.

6.5 Discussion

WalkMaxSat appears to be a promising algorithm for enhancing partition search. It looks especially useful to embellish other methods such as AHC to explore regions around the AHC optimal partition and to find close partitions with better explanatory power. We demonstrated above that this technique can enhance performance on small subsets of the data and on large datasets too, in conjunction with AHC.

Although we have not tested this algorithm in the following regard, the algorithm can also be used as a useful exhaustive local check of a MAP partition found by numerical search (Lau and Green, 2007). Also, note that weighted MAX-SAT can be used not just for MAP identification, but also by following the adaptation suggested by Cussens (2008) in model averaging, using to identify all models that are good.

There are many embellishments of the types of methods described above that will potentially further improve our hybrid search algorithm. The main issue is the choice of cluster scores, as discussed in Section 6.4. This limited the size of our applications in this chapter.

However, we believe that the use of scientific knowledge through utility functions instead of scores is a promising possibility, as we have already demonstrated in Chapter 5 that a utility-based search maintains the properties of MAP selection. This approach would reduce the cluster scores by localising the search for an optimal partition according to the scientific interest of the clusters. The great difficulty that we found in running weighted MAX-SAT for large examples might have been directly overcome by using a utility-based approach to start with. However, a utility-based approach can be used only when a utility function can be defined for the problem at hand. Hence, we decided to start by encoding a general clustering problem using weighted MAX-SAT. Encoding utility functions instead of scores is the next step, even though in hindsight it might have been a better starting point.

However, in this chapter we have demonstrated that in circumstances where the Crowley priors are appropriate weighted MAX-SAT solvers can provide a very helpful addition to the tool box of methods for MAP search over a partition space. In the following chapter we continue to focus on new search methods for the space of partitions with a theoretical approach that defines a general formal framework for search algorithms.

Chapter 7

Search on the Lattice of Partitions

7.1 Introduction

A common problem for clustering algorithms is that only a few partitions are considered when searching for the best one. Another issue is how to move from one partition to another in the space of partitions.

In the literature, moves between clusterings are obtained by adding or deleting branches to dendrograms. See, for example, Chipman et al. (1998), Denison et al. (2002) and Wu et al. (2007). However, in this chapter we view clusterings as elements of a lattice and we try to define a general framework for the search of the partition space. The lattice of partitions has been studied in the context of clustering by Meilă (2005) and Meilă (2007) with a focus on the comparison of clusterings by defining distance measures on the lattice.

However, the Bayesian clustering algorithm proposed by Heard et al. (2006) has been shown (see Chapter 4) to have some properties, similar to those identified by Meilă (2005), desirable for moves on the lattice of partitions. Therefore we can define moves on the lattice of partitions when we are searching the space for the optimal partition in the sense of Heard et al. (2006).

Another perspective on our approach is that it is an *internal criterion* for clustering comparison. Meilă (2005) focuses on *external* evaluation, that is, an evaluation that simply measures how close the obtained clustering is to a gold standard. This evaluation method is independent of the algorithm or the way clusterings were obtained. However, when the clustering algorithm is defined as by Heard et al. (2006), we can also evaluate clusterings by *internal criteria*, such as distortion, likelihood, etc.

In Sections 7.2 and 7.4 we define the lattice of partitions and its properties. In Section 7.5 we propose methods to search the partition space.

7.2 The Lattice of Partitions

A partition, or clustering, $\mathcal{C}$ is the division of a dataset D into sets c_1, c_2,. . . ,c_N, called clusters, such that

$$c_k \cap c_l = \emptyset \quad \text{for} \quad k \neq l \quad \text{and} \quad \bigcup_{k=1}^{N} c_k = D. \tag{7.1}$$

Let N and n_k be the number of data points in D and c_k respectively. Therefore,

$$\sum_{k=1}^{N} n_k = n \tag{7.2}$$

and we also assume that $n_k \geq 1$, that is, there are no empty sets. We will use the terms data points and observations interchangeably throughout.

The space of partitions of a dataset D can be expressed using a directed graph, as the Hasse diagram shows in Fig. 7.1 for $n = 4$, called the *lattice of partitions* (Stanley, 1997). In this graph, an edge between two partitions $\mathcal{C}$ and $\mathcal{C}'$ will be present if $\mathcal{C}'$ can be obtained by splitting a cluster of $\mathcal{C}$ into two parts, or vice-versa if $\mathcal{C}$ can

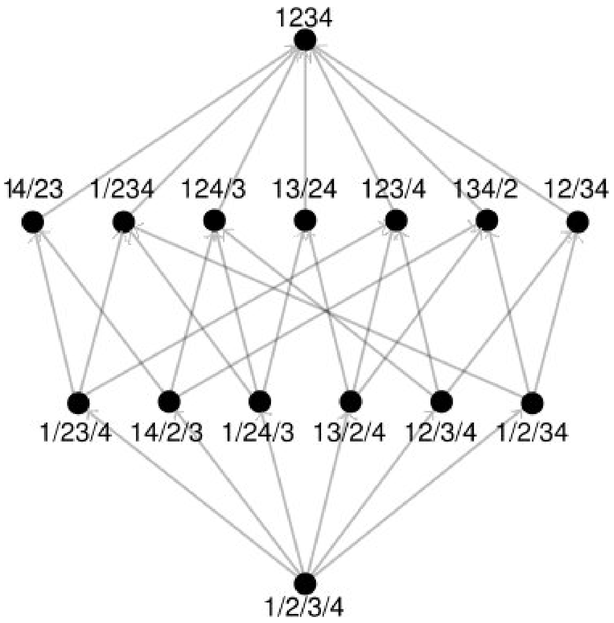


Figure 7.1: The lattice of partitions. In the case $n = 4$, the partial order of the set of all 15 partitions is depicted in this Hasse diagram, where 12/34 represents a partition formed by two clusters: $\{1, 2\}$ and $\{3, 4\}$.

be obtained by merging two clusters of $\mathcal{C}'$. In this representation the edges are directed from $\mathcal{C}'$ to $\mathcal{C}$, that is, clusters are merged when directed edges are followed. This lattice has a lower bound $C_0 = \{\{1\}, \{2\}, \ldots, \{n\}\}$ (partition of all singletons) and an upper bound $C_D = \{1, 2, \ldots, n\}$ (partition with all genes in the same cluster). The directed edges form paths from $\mathcal{C}_0$ to $\mathcal{C}_D$

The number of partitions for n data points is finite but huge: superexponential (Stanley, 1997). The Bell number, B_n, is the number of different partitions of a set with n elements. Starting with $B_0 = B_1 = 1$, the first few Bell numbers are

$$1, 1, 2, 5, 15, 52, 203, 877, 4140, 21147, 115975, \ldots$$

The number of partitions of a set increases quickly as the number of elements in the set increases. The Bell numbers satisfy the recursive formula

$$B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k. \tag{7.3}$$

Also, the Bell numbers can be written in terms of Stirling numbers of the second kind as given by

$$B_n = \sum_{k=1}^{n} S(n,k). \tag{7.4}$$

The Stirling numbers of the second kind,

$$S(n,k) \qquad k,n \in \mathbb{N}, \qquad 1 \leq k \leq n,$$

are a doubly-indexed sequence of natural numbers. The Stirling number $S(n,k)$ is the number of ways to partition a set of n objects into k groups. The following properties hold.

$$S(n,n) = S(n,1) = 1 \tag{7.5}$$

$$S(n,k) = kS(n-1,k) + S(n-1,k-1) \tag{7.6}$$

$$S(n,k) = \frac{1}{k!} \sum_{j=0}^{k} (-1)^{k-j} \binom{k}{j} j^n \tag{7.7}$$

$$S(n,2) = 2^{n-1} - 1 \tag{7.8}$$

The Stirling numbers of the second kind are also useful in the context of lattices of partitions because it is easily verified that there are $S(n,k)$ different partitions of the n given observations into k clusters. If we think of the lattice as a 'ball' with layers, where the first layer is $\mathcal{C}_0$ and the external layer (layer n) is $\mathcal{C}_D$, then layer k has $S(n,n-k)$ different partitions.

Moreover, there are

$$\binom{N_{\mathcal{C}}}{2} \tag{7.9}$$

edges departing from partition $\mathcal{C}$ of which $\mathcal{C}$ is a one-edge refinement, with $N_{\mathcal{C}}$ the number of subsets in $\mathcal{C}$. Also, there are

$$\sum_{i=0}^{N_{\mathcal{C}}} S(n_i, 2) = \sum_{i=0}^{N_{\mathcal{C}}} 2^{n_i - 1} - N_{\mathcal{C}} \tag{7.10}$$

refinements of partition $\mathcal{C}$, where $S(\cdot, \cdot)$ is the Stirling number of the second kind defined above.

7.3 Lattices and MAP Search

As already discussed in Chapters 3, 4 and 6 we use the lattice of partitions as a partition search tool in conjunction with the conjugate Gaussian regression model developed by Heard et al. (2006) for clustering. This model has a wide applicability because it can be customised through the choice of a given design matrix X. Conjugacy ensures the fast computation of scores for a given partition because these can be written explicitly and in closed form as functions of the data and the chosen values of the hyperparameters of the prior. Applications range from one-dimensional data points to multidimensional datasets with time dependence among points or where the points are obtained by applying different treatments to the units.

Let $y_i \in \mathbf{R}^r$ for $i = 1, \ldots, n$ represent the r-dimensional units to cluster. Let $D = (y_1, \ldots, y_n)$ and $y = \text{vec}(D)$ satisfy

$$y = X\boldsymbol{\beta} + \varepsilon,$$

where $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)' \in \mathbf{R}^p$ and $\varepsilon \sim N(\mathbf{0}, \sigma^2 I)$ is a vector of independent error terms with $\sigma^2 > 0$. Using the posterior Normal Inverse-Gamma joint density of the parameters $(\boldsymbol{\beta}, \sigma^2)$ denoted by $NIG(\mathbf{0}, V, a, b)$, and assuming the parameters of different clusters are independent then, because the likelihood separates, it is straightforward to check (see Chapter 4) that the log marginal likelihood score $\Sigma(\mathcal{C})$ for any partition $\mathcal{C}$ with clusters $c \in \mathcal{C}$ is given by

$$\Sigma(\mathcal{C}) = \log p(\mathcal{C}) + \sum_{c \in \mathcal{C}} \log p_c(y). \tag{7.11}$$

Here the prior $p(\mathcal{C})$ is often chosen from the class of cohesion priors over the partition space (Quintana and Iglesias, 2003) which assigns weights to different models in a plausible and convenient way: see e.g. Chapter 4.

This class of models is used for clustering in conjunction with an agglomerative hierarchical clustering (AHC). If we tried to replicate the AHC partition search on the lattice, we would explore the partitions on one path $\mathcal{C}_0$ to $\mathcal{C}_D$. However, this implies that, if we had 20 elements, only 1331 out of the $B_{20} = 5.1 \times 10^{13}$ partitions of the lattice would be explored. Note that AHC starts at C_0 and selects the partition with the highest score between all those linked by an edge. It then moves on to the selected partition and repeats. Therefore, the path from $\mathcal{C}_0$ to $\mathcal{C}_D$ includes 20 partitions but 1331 are evaluated during the course of the algorithm. This is a small number compared to the 5.1×10^{13} partitions for 20 elements!

An essential property of the search for MAP models - dramatically increasing the efficiency of the partition search - is that with the right family of priors the search is *local*. That is, if $\mathcal{C}^+$ and $\mathcal{C}^-$ differ only in the sense that the cluster $c^+ \in \mathcal{C}^+$ is split into two clusters $c_1^-, c_2^- \in \mathcal{C}^-$ then the log marginal likelihood score is a linear function only of the posterior cluster probabilities on c^+, c_1^- and c_2^-. Moreover, if we use an appropriate prior (Crowley, 1997) then the score in equation (7.11) decomposes. Thus

$$\begin{aligned}\Sigma(\mathcal{C}) &= \log p(N, n_1, \ldots, n_N | y) \\ &= \log p(N, n_1, \ldots, n_N) + \sum_{i=1}^{N} \log p(y_i) \\ &= \log \Gamma(\lambda) - \log \Gamma(n+\lambda) + \sum_{i=k}^{N} S(c_k),\end{aligned}$$

where

$$S(c_k) = \log p(y_i) + \log \Gamma(n_i) + \log \lambda.$$

Thus, the score $\Sigma(\mathcal{C})$ is *decomposable* into the sum of the scores $S(c_k)$ over individual clusters plus a constant term. The choice of the Crowley prior ensures that the score of a partition is expressible as a linear combination of scores associated with individual sets within the partition. It is this property that enables us to find straightforward encoding of the MAP search as a lattice search problem.

Recall that we have defined directed edges over the partition space. The formulation of the MAP search above allows us to focus only on *weights* which are associated to each of the directed edges of the lattice. These weights represent the edges of the lattice of partitions and therefore the possible moves in the partition space. In particular the weight $w(\mathcal{C}, \mathcal{C}')$ is defined as

$$w(\mathcal{C}, \mathcal{C}') = \Sigma(\mathcal{C}) - \Sigma(\mathcal{C}'). \tag{7.12}$$

Note that it holds that $w(\mathcal{C}, \mathcal{C}') = -w(\mathcal{C}', \mathcal{C})$. This is due to the directionality of the edges, so that when moving against the direction of the edge the weight is $w(\mathcal{C}, \mathcal{C}')$ and when moving along the direction of the edge the weight associated with the edge is $w(\mathcal{C}', \mathcal{C})$. Then, when the partitions $\mathcal{C}$ and $\mathcal{C}'$ are adjacent, that is, they are connected by an edge, the weight $w(\mathcal{C}, \mathcal{C}')$ simplifies to

$$w(\mathcal{C}, \mathcal{C}') = \log(n+\lambda+1) - \log(n+\lambda) + S(c_i) - S(c_l) - S(c_j), \tag{7.13}$$

where $c_k \in \mathcal{C}$, $c_l, c_j \in \mathcal{C}'$ and $c_i = c_l \cup c_j$. The reason for the name *weight* will be apparent later, when all the possible edges departing from a vertex are considered according to their weight. Note that the vertices and edges on the lattice of partitions are identical for every dataset of size n while the weights on the edges are dataset-dependent.

7.4 Properties of the Lattice of Partitions

Meilă (2005) gives formal definitions for three additivity properties that are desirable for a lattice of partitions. The paper focuses on the comparison of clusterings so these additivity properties concern the distance measure d that Meilă (2005) defines. We propose here a re-formulation of these properties with respect to the weights $w(\mathcal{C}, \mathcal{C}')$. Note that the weights do not define a *distance*, and it holds that

$$w(\mathcal{C}, \mathcal{C}') = -w(\mathcal{C}', \mathcal{C}). \tag{7.14}$$

Additivity with respect to refinement If $\mathcal{C}'$ is obtained from $\mathcal{C}$ by splitting one or more clusters, then we say that $\mathcal{C}'$ is a refinement of $\mathcal{C}$. For example, $\mathcal{C}' = \{\{a, b\}, \{c\}\}$ is a refinement of $\mathcal{C} = \{\{a, b, c\}\}$. The weight w is additive with respect to refinement if, and only if, for any clusterings $\mathcal{C}$, $\mathcal{C}'$, $\mathcal{C}''$ such that $\mathcal{C}'$ is a refinement of $\mathcal{C}$ and $\mathcal{C}''$ is a refinement of $\mathcal{C}'$, it holds that

$$w(\mathcal{C}, \mathcal{C}'') = w(\mathcal{C}, \mathcal{C}') + w(\mathcal{C}', \mathcal{C}''). \tag{7.15}$$

This property corresponds to steps down the Hasse diagram and it states that the weight $w(\mathcal{C}, \mathcal{C}'')$ corresponds to the sum of the weights of two successive refinements.

Additivity with respect to the join Meilă (2005) defines the *join* between two clusterings $\mathcal{C}$ and $\mathcal{C}'$ as

$$\mathcal{C} \times \mathcal{C}' = \{c_k \cap c'_{k'} | c_k \in \mathcal{C}, c'_{k'} \in \mathcal{C}', c_k \cap c'_{k'} \neq \emptyset\}, \tag{7.16}$$

that is, the join of two clusterings in the clustering formed from all the nonempty intersections of clusters from $\mathcal{C}$ with clusters from $\mathcal{C}'$. A weight w is additive with respect to the join if, and only if, for any clustering $\mathcal{C}$ and $\mathcal{C}'$,

$$w(\mathcal{C}, \mathcal{C}') = w(\mathcal{C}, \mathcal{C} \times \mathcal{C}') + w(\mathcal{C} \times \mathcal{C}', \mathcal{C}'). \tag{7.17}$$

The property of additivity with respect to the join is a more general form of the property of additivity with respect to refinement, for when clusterings $\mathcal{C}$ and $\mathcal{C}'$ are not a refinement of one another. It can be seen as steps down the Hasse diagram followed by steps up the Hasse diagram. There are often many possible paths between two clustering on the Hasse diagram, but note that the $w(\mathcal{C}, \mathcal{C}')$ will be the same for all of them.

Convex additivity Let $\mathcal{C} = \{c_1, \ldots, c_N\}$ be a clustering and $\mathcal{C}'$ and $\mathcal{C}''$ be two refinements of $\mathcal{C}$. Denote by $\mathcal{C}'_k$ ($\mathcal{C}''_k$) the partitioning induced by $\mathcal{C}'$ (respectively $\mathcal{C}''$) on c_k. Let $P(k)$ represent the proportion of data points that belong to cluster c_k. Then

$$w(\mathcal{C}', \mathcal{C}'') = \sum_{k=1}^{N} P(k) w(\mathcal{C}'_k, \mathcal{C}''_k). \tag{7.18}$$

This property expresses additivity over the sublattices corresponding to the individual clusters. It also implies that the weight $w(\mathcal{C}, \mathcal{C}')$ does not depend on those clusters that are identical in both partitions $\mathcal{C}$ and $\mathcal{C}$, but only on those that have been partitioned differently. This is a desirable property for clustering, as non-local algorithms can be counter-intuitive. A non-local algorithm implies that a change inside a single cluster counts differently depending on how the rest of the data is clustered.

7.5 Search on the Partition Space

The weights on the lattice edges defined above can be used for the search on the partition space. This is equivalent to an internal evaluation criteria for clustering algorithms (Meilă, 2005).

In what follows we will use the terms partition and vertex interchangeably, as each vertex of the lattice corresponds to a partition. Moreover, we will refer to downward (upward) edges from a vertex $\mathcal{C}$ according to the Hasse diagram in Fig. 7.1.

We aim to identify the partition $\mathcal{C}$ that maximises the score function $\Sigma(\mathcal{C})$. Anderson et al. (2006) notes that the score function $\Sigma(\mathcal{C})$ as defined by Heard et al. (2006) is a monotonically increasing function until its maximum, and then monotonically decreasing, when the AHC path on the lattice is followed. This leads us to believe that similar paths can be found on the lattice and to the definition of modal vertices and lattices.

Definition 7.5.1. *A vertex $\mathcal{C}$ of the lattice of partitions $\mathcal{L}$ is called modal if all of its upward edges have a strictly positive weight and all downward edges have negative weight. Denote the set of modal vertices by M. Call lattice $\mathcal{L}$ modal for a given dataset if there is exactly one element in M.*

In other words, a partition $\mathcal{C}$ is called modal if all the other partitions that are a connected to it by an edge have a lower score. This implies that the score $\Sigma(\mathcal{C})$ has a local mode in $\mathcal{C}$.

Definition 7.5.2. *For a given dataset, call $\mathcal{L}$ weakly modal if the edge scores on all source-to-sink paths $\lambda = \{e_1, \ldots, e_{n-1}\}$ are such that $w(e_i) \geq 0$ for $1 \leq i < g(\lambda)$ and $w(e_i) < 0$ for $g(\lambda) \leq i \leq n-1$, for some $1 \leq g(\lambda) \leq n-1$.*

Given a weakly modal lattice $\mathcal{L}$, call

$$U = \{\mathcal{C} | \exists \mathcal{C}^-, \mathcal{C}^+ : w(\mathcal{C}, \mathcal{C}^+) < 0 \quad \text{and} \quad w(\mathcal{C}^-, \mathcal{C}) > 0\} \tag{7.19}$$

with a directed edge from $\mathcal{C}$ to $\mathcal{C}^+$ and from $\mathcal{C}^-$ to $\mathcal{C}$, that is, $\mathcal{C}$ is a one-edge refinement of $\mathcal{C}^+$ and $\mathcal{C}^-$ is a one-edge refinement of $\mathcal{C}$.

Theorem 7.5.3. *If no scores on vertices are equal, i.e.* $w(\mathcal{C}, \mathcal{C}') \neq 0 \; \forall \mathcal{C} \neq \mathcal{C}'$, *then the partition with the highest score must be modal. So, in particular, if* $\mathcal{L}$ *is modal then its unique modal partition is the highest scoring partition.*

Proof. The partition with highest score satisfies

$$\exists \mathcal{C}^* : \Sigma(\mathcal{C}^*) > \Sigma(\mathcal{C}) \quad \forall \mathcal{C} \in \mathcal{L}/\{\mathcal{C}^*\} \tag{7.20}$$

because there are no vertices with equal scores and we assume that all paths from $\mathcal{C}_0$ to $\mathcal{C}_D$ are increasing and then decreasing. So the partition with the highest score is modal, because its ingoing edges are strictly positive and all outgoing edges have negative weight. Therefore, $\#M \geq 1$. In particular, if $\#M = 1$, $M = \{\mathcal{C}^*\}$, with $\#M$ the cardinality of set M. □

Theorem 7.5.4. *When Ł is weakly modal then* $M \subseteq U$.

Proof. Say that $\mathcal{C} \in M$. Therefore, all the directed edges from $\mathcal{C}$ are such that $w(\mathcal{C}, \mathcal{C}^+) < 0$, $\forall \mathcal{C}^+$, and all the directed edges to $\mathcal{C}$ are such that $w(\mathcal{C}^-, \mathcal{C}) > 0$, $\forall \mathcal{C}^-$. That implies that $M \subseteq U$, from the definition of U given above. □

Define as parents of a vertex $\mathcal{C}$ all the vertices from which an edge directed to $\mathcal{C}$ departs. Define as children of a vertex $\mathcal{C}$ all the vertices to which edges departing from $\mathcal{C}$ are directed.

Once a partition $\mathcal{C}$ which belongs to U on a modal lattice $\mathcal{L}$ is found, using, for example, an Agglomerative Hierarchical Clustering algorithm (AHC), there are two possible cases.

- Case 1 - All the incoming edges have a positive weight and all the outgoing edges have a negative weight. This also implies that $\mathcal{C} \in M$.
- Case 2 - All the outgoing edges have negative weight and at least one of the ingoing edges has a positive weight.

In Case 1 we have found the modal vertex and if $\mathcal{L}$ is modal then we have found the highest scoring partition. In Case 2, we are in U and by moving on the partition space within U we aim to reach a modal vertex in M.

We propose the *Iterative local Search Algorithm* (ISA). Note that we assume here that Ł is a modal lattice.

1. Set $i = 1$.
2. Follow any clustering algorithm to find an optimum partition. For instance, follow the Agglomerative Hierarchical Clustering (AHC) similarly to Heard et al. (2006), until the optimum partition $\mathcal{C}_i$ is found or the edge weight is negative.
3. Check whether $\mathcal{C}_i$ is modal by examining all the edges arriving and departing from it. If $\mathcal{C}_i$ is modal (that is, all the incoming edges are positive and all the outgoing edges are negative), go to Step 6. If $\mathcal{C}_i$ is not modal then choose one of the refinement partitions $\mathcal{C}^-$ among the
$$\sum_{i=0}^{N_{\mathcal{C}_i}} S(n_i, 2) = \sum_{i=0}^{N_{\mathcal{C}_i}} 2^{n_i-1} - N_{\mathcal{C}_i} \tag{7.21}$$
edges linking $\mathcal{C}_i$ to its refinements. Choose one of these randomly, but accept the proposed partition only if the weight $w(\mathcal{C}^-, \mathcal{C}_i)$ is negative.

4. Set $\mathcal{C}_{i+1} = \mathcal{C}^-$ and $i = i + 1$.

5. Run steps 1-4 until $\mathcal{C}_i$ is modal, or as long as possible otherwise.

6. The optimal partition $\mathcal{C}^*$ is the partition with the highest associated score among those explored by the algorithm, or the modal partition found, if any.

The idea behind this algorithm is that the highest scoring partition $\mathcal{C}^* \in U$, so we focus our partition search on the subset U of the huge partition space $\mathcal{L}$. Moreover, in certain scenarios, as in Chapter 5, it is possible to further restrict the area of the partition space that requires exploration as, for example, a utility function might guide the algorithm through specific areas of interest. Note also that the search algorithm proposed is fast because the score functions (and the weights) only depend on the clusters involved in the merge/split moves at each step.

Finally, the following theorem guarantees that the optimal solution can be found in a finite number of steps, and therefore, time.

Theorem 7.5.5. *If $\mathcal{L}$ is modal, and you move on the U boundary you'll find partition $\mathcal{C}^*$ where $\Sigma(\mathcal{C}^*) > \Sigma(\mathcal{C})\ \forall \mathcal{C}$ in a finite number of steps.*

Proof. Each path from $\mathcal{C}_0$ to $\mathcal{C}_D$ has to go through each level of k. If all the paths have the maximum score for level k then the number $N_{\mathcal{C}}$ of partitions $\mathcal{C} \in U$ is at most equal to

$$N_{\mathcal{C}} \leq \arg\max_k S(n,k) = K_n, \tag{7.22}$$

where K_n is the maximum Stirling number of the second kind given n and it can be approximated by $K_n \sim n/\log(n)$ (Harper, 1967). Note that K_n is not always unique (Canfield and Pomerance, 2002). □

7.6 Discussion

This chapter aims to build a general framework for the search over partitions by using the lattice to define the moves over such space. Meilă (2005) defined distances over partitions and she was then able to compare many distance measures using the lattice. In a similar way, we believe that we can use the lattice as a general framework for all the search algorithms. This has several advantages.

First of all, it allows us to define all the search algorithms on the same space. This implies that properties of different classes of methods can be defined and then used to evaluate and compare each technique. By defining the most common search algorithms in terms of the lattice, we therefore provide a general formal framework for partition space search. For example, this is straightforward for AHC, as mentioned in Section 7.3. A particular focus of further work would be on the definition of MAX-SAT solvers on the lattice, if at all possible, allowing us to use well-known and highly-efficient algorithms to search the space.

Second, further study of the properties of the lattice could be used to refine search algorithms allowing splits and long jumps, currently used in other domains, such as stochastic search (Chipman et al., 2002; Lau and Green, 2007).

Finally, this rigorous outlook on the search over the partition space and its properties could encourage the developments of new methods, such as the ISA that we propose in this chapter.

Chapter 8

Conclusion and Further Work

This thesis is concerned with the study of a Bayesian clustering algorithm, proposed by Heard et al. (2006), and used successfully for microarray experiments over time.

The first results, presented in Chapter 4, focus on the study of hyperparameters and how they can affect the stability of the algorithm and inference, especially when outliers are present. In this chapter we propose new ways of setting hyperparameters, proportionally to the size of the clusters, rather than the default settings used by Heard et al. (2006). Moreover, we derive explicit characterisations ensuring that Bayes factor selection prefers partitions that combine clusters when they are close with respect to a certain separation measure. This chapter includes examples on simulated data, as well as an example on a biological dataset. The examples demonstrate numerically the desiderability of proportional settings of the hyperparameters and also the danger of missclassification that is more likely to occur when using default settings. We use the proportional settings in all the remaining chapters of this thesis.

Then we focus on the huge partition space, design new intelligent algorithms for the search over such space and further improve MAP selection.

In Chapter 5, in particular, we note that the search for an optimal decision can be localised facilitating fast optimisation and we show that, when it is appropriate to costumise the search so that it reflects the scientific enquiry, the local algorithm associated with a product utility can be seen as exactly a MAP search but over the genes of interest to the scientist. Moreover, it can be shown that this is equivalent to a MAP search with adjusted priors. In several examples we show how our guided algorithm allows us to retrieve misclassified genes both in a simulated example (by introducing outliers) and in a real example (where a known circadian gene was originally misclassified using the method by Heard et al. (2006)).

However, speed and efficiency are persistent issues in clustering because a full exploration of the partition space is usually impossible. Therefore in Chapter 6 we investigate the use of weighted MAX-SAT solvers for clustering, aiming to improve the performance of our algorithm by drawing expertise on computational issues from a different discipline. Weighted MAX-SAT is an important and widely studied combinatorial optimisation method with applications in Artificial Intelligence and other areas of computer science. By defining an appropriate local prior structure over the partition space we successfully encode our problem and run weighted MAX-SAT for clustering and therefore open possibilities for statisticians to take advantage of the fast algorithms that researchers in Artificial Intelligence have been working on for decades.

We originally decided to investigate the performance of MAX-SAT for general clustering problems, not only those where scientific knowledge was available and a utility function could be defined appropriately. Therefore in Chapter 6 we did not use the results on the MAP search obtained in Chapter 5. However, the overall performance of weighted MAX-SAT solvers is inferior to the one presented in Chapter 5. Therefore, it is promising to combine these methods and further work on the use of MAX-SAT for clustering will focus on the combination of the methods developed in Chapters 5 and 6 by defining clusters utilities rather than scores.

Finally, in Chapter 7 we present work on the definition of a general framework that views partitions as elements of a lattice and aims to provide a new outlook on the properties and the evaluation of search algorithms. By formulating the problem of partition search in such a general form we believe that we can obtain more general results that allow us to study further properties of the space and therefore methods to explore it.

First of all, by re-defining all the available search techniques – AHC, weighted MAX-SAT, utility-based search, stochastic search (Lau and Green, 2007) – on the lattice we can directly compare different methods and develop criteria for their evaluation. This has been successfully done by Meilă (2005) to obtain general results on clustering evaluation criteria and we believe that this can be extended to the evaluation of search algorithms too.

Moreover, the study of the lattice and its property can also inspire the creation of new intelligent search algorithms that move over the most promising areas of the partition space. We include in this chapter preliminary work on one such new algorithm defined on the lattice.

Chapters 5, 6 and 7 all contribute to the definition of a fast search algorithm: in Chapter 7 we define and study the space and its properties in a general form, in Chapter 5 we prove that the search can be localised while maintaining all of its properties, and in Chapter 6 we propose the use of fast methods to implement the search. The combination of all of these results is extremely powerful, and it is the objective of further work on this topic.

At the end of each chapter we have included a discussion and conclusion section on the topics raised in that chapter along with further improvements and work that could be done. However, we include here a few additional comments on the assumptions made throughout this thesis, sensitivity, evaluation criteria and uncertainty.

This thesis is based on several assumptions. A fundamental one is that clusters are independent. This is usual in partition search but unrealistic for the biological applications presented here. However, it is this assumption that allows us to reduce the dimensionality of datasets of tens of thousands of genes and relaxing this hypothesis is beyond the scope of this thesis. However, Section 8.1 for guidelines for further work in this direction.

Another strong assumption is made when choosing the basis function. This is an assumption about the results and the shapes that we expect from our clustering. For the examples presented in this thesis, following discussion with biologists, we believe that the Fourier basis was an appropriate choice. However, it is important to note that this strongly affects the results, especially when a full basis cannot be implemented for computational reasons, such as in Section 4.7. This is further discussed in Section 3.3.2.

Moreover, note also the sensitivity of the methods presented to the choice of hyperparameters, an important issue that we discuss throughout the thesis. First of all, in Chapter 4 we show the misclassification that can occur using the default settings of the hyperparameters and their influence on the results. We also include some guidelines for the choice of hyperparameters and in practice we run our algorithms multiple times changing the hyperparameters. We select the hyperparameters by choosing those that maximise the log marginal likelihood and also by discussing with the biologists regarding the within-cluster variance that they expect. Moreover, in Appendix B, we discuss the robustness of the results to changes in the hyperparameters in the example presented in Chapter 5. Figures B.17 and B.18 illustrate the robustness of the initial gene estimates to misspecification of the hyperparameters. We note that changing v we obtain fewer, but larger, clusters. Even though we obtain different solutions, the estimated profiles of most genes do not radically differ under changes in v. Therefore, after a careful study of sensitivity to hyperparameters presented in Chapter 4, we have relayed on the robustness shown in Appendix B for the choice of hyperparameters.

The comments above on sensitivity naturally introduce the broader topic of evaluation criteria and how this has been addressed in this thesis. Validating clustering algorithms and comparing the performance of different algorithms is complex because it is difficult to find an objective measure of the quality of clusters. We reviewed evaluation criteria in Section 3.3 and we have used several of the methods mentioned there throughout the thesis. We have used biological interpretation and visual comparisons as standard methods for evaluation, but we have also used several of the methods suggested by Rand (1971) in Chapter 4 (retrieval of the originating structure) and Chapters 5 and 6 (perturbation with outliers). We believe that evaluation criteria are fundamental to the comparison of clustering methods but it is not always immediately obvious how to proceed in such comparisons when, as in our applications, there is no absolute scheme with which to measure clusterings.

Finally, however, it is appropriate to note that for stochastic search algorithms evaluation criteria play an intrinsic role. This is a trade-off. These methods are usually inferior in terms of speed and efficiency but they have the advantage of including a measure of uncertainty in the selection of the output model. This poses additional problems regarding how this uncertainty can be incorporated into the final result, but it describes the underlying process more faithfully than the deterministic search algorithms used throughout this thesis.

In this thesis we have contributed to the development of Bayesian clustering methods, and in particular to the study of the geometry of Bayes factors for selection and the influence of hyperparameters. We have shown that MAP selection can maintain certain properties when used in conjunction with a utility function, and we have developed new methods for the search of the partition space, using a utility function, weighted MAX-SAT solvers and the lattice of partitions.

We would like to conclude this thesis by presenting a proposal for further research on a topic that moves away from clustering in the context of biological regulatory networks: the development of a method for model selection for high-dimensional dependent time series. A usual modelling assumption in Bayesian partition search and throughout this thesis is that genes with different profiles are independent of each other. However, this is unrealistic, especially in the context of regulatory networks, and therefore we would like to focus on the development of types of partition models over short time series that explicitly model dependence structures.

8.1 Further Work on Model Selection for Dependent Time Series

Despite the success of the class of partition models in this problem, when used to search over potential regulatory pathways, the class of partition models searched is too restrictive in this domain to be fully realistic. The search effectively identifies the co-expressing clusters and their shared associated expression profiles. However, scientists are interested in how clusters of co-expressing genes might excite or inhibit one another, but the hypothesis that these regulatory relationships exist contradicts a modelling assumption made in Bayesian partition search: namely that genes with different profiles are independent of each other. Thus, until such dependence is recognised within the model space searched, any Bayesian methodology is therefore necessarily limited in its scope. Further work could focus on the development of types of partition models over short time series that explicitly model dependence structures reflecting different sorts of excitation and inhibition hypotheses between different clusters of genes whilst retaining the efficiency and scope of the exploratory process.

We note that within the domain of gene expression there has been massive activity in hypothesising, estimating and diagnostically checking the topology of various regulatory networks (de Jong, 2002; Newman, 2003; van Someren et al., 2002). Many graphical technologies such as Bayesian Networks, undirected graphs, and mixed graphs have all been used in this connection. These graphs are nearly always only fitted on a small subset of the genes: either those already suspected of having a regulatory role; or where the set is determined by pre-filtering using analogous models to those described above.

However, it has now been established (see Chapter 4) that the relationship between the types of dependence expressed by the adjacency of genes in these graphs and any hypothesis of co-regulation is not a strong one. Even if the correspondence between regulatory networks and the conditional independence relationships expressed by the missingness of edge in the graph were to exist, it is clear from other domains that the direction of arrows in these graphs *cannot* be interpreted causally (Pearl, 2000). Thus the actual descriptive power of graphs obtained through a search across this type of model is limited.

More recently, the existence of longitudinal profiles has enabled researchers to develop search over dynamic Bayesian networks and related models. These provide a much more solid class of statistical model around which to express and model co-expression relationships. However, due to computational issues, such dynamics dependence models are usually only applied to a selected small number of identified genes (Koller and Lerner, 2001; Dean and Kanazawa, 1988).

Further work would focus on the generalisation of the class of regression partition models searched by MAP techniques to include models that exhibit the type of dependence between clusters that might be expected were one cluster to regulate another. The statistical models added to the search space need not only to be parsimonious so that they can be estimated and scored at speed, utilising properties of conjugacy, but will also need to mirror closely the types of regulatory dependence expected within this context. The fact that each dependency model within the class can be associated with a particular network of relationships means that it will be possible to depict the MAP model graphically using the semantics of excitation and inhibition familiar to the scientist rather than through conditional independence graphs whose meanings are easily misunderstood. In the context of our main example each node in these new graphs will be indexed by a set of clusters and their profiles that are potentially regulatory. Each edge will be either directed - representing excitation between these sets - or blocked - representing inhibition. The genes within the sets of clusters will of course be annotated allowing the scientist to explore their signatures and how these might relate to the actual postulated message passing mechanisms. This search methodology, if employed on a new microarray experiment, would be able to provide possible explanatory regulatory graphs together and we could provide such a schemata back to the biologists.

The simple idea behind this methodology will be to allow the vector of Gaussian variables associated with one cluster profile to depend in a simple linear way on another. Throughout this device, conjugacy, conditional on the values of a few hyperparameters, will be maintained - so that a closed form score can still be quickly calculated, but also dependence will be explicitly modelled. Since the dependence model will typically have far fewer parameters than the original independence models and due to the familiar properties of Bayes Factor search, if the dependence model is consistent with the data it will be preferred to an alternative more heavily parametrised independence model.

8.1.1 Simple but Plausible Classes of Dependence Models

The sorts of relationships we will initially explore will assume homogeneity between clusters where these clusters are defined relative to partitions explored with between-cluster independence assumed. Therefore, the dimension of the problem immediately reduces in complexity by approximately a square root reducing a potentially enormous search space. For example, in our Fourier clustering the number of possible relationships reduces to hundreds rather than tens of thousands of units. Furthermore, because clusters are associated by profiles, it is clear that most clusters could not be candidates as regulators or inhibitors of others which again reduces the problem.

It is common for the mean profiles of two different clusters of a MAP partition, obtained using the hypothesis of independence, to be almost exactly proportional to one another (see Chapter 4). In such circumstances, from a regulatory point of view, it is helpful to represent these sets of clusters as a single entity: either the whole set or the most highly expressive cluster being linked to the regulatory genes. We will call these clusters *hyperclusters* and it is these hyperclusters which will be the nodes of any grap and the objects through which we define our putative regulatory mechanism. There are different ways of formally performing these initial aggregations, with different methods corresponding to different classes of models. The initial step is therefore to discover the most scientifically sensitive and computationally fast method of performing this initialisation step.

The relationships between regulating and regulated hyperclusters can be expressed in terms of a simple transition relationship between the cluster regression parameters. In this context a parsimonious class of dependence models is straightforward to propose. For example, in the case of excitation, the forms of the transition corresponds to a small forward phase shift together with a possible small (scaled) damping. Inhibition, another important mechanism, can be analogously expressed by switching the signs of some of the parameters in the transition functions. Assuming shared variances, or plugging in estimated variances for large clusters, will allow us to calculate a conjugate score for the combined cluster that can be compared to the component scores. This will allow us to perform a conjugate analysis over a sort grid of 2, or perhaps 3, hyperparameters and then score and select good interpretable parsimonious dependence models.

8.1.2 Sparse Dependence Structures for Regulation

Even with the caveats above, search over the dependence space would be infeasible in general. However, it is fortunate in the context of the given problem that the number of relationships between clusters can be safely hypothesised to be sparse. There are two reasons it is possible to make this assumption, at least for our running example. First, it makes biological sense that only certain sets of co-expressing genes regulate those in another set. Second, even if this were not the case, if a model were to describe too many simultaneous regulators on a single cluster, the associated science of the model becomes impenetrable. In particular, the dense graphs associated with such complex mechanisms cannot be usefully communicated through graphs. By limiting the space of models to have a sparse set of relationships, where each cluster has no more than a certain small number of regulators, dramatically simplifies the problem and makes search feasible.

Fast search is also helped in the sense that each cluster/hypercluster has its own signature profile. Since the hypotheses of excitation and inhibition of one hypercluster on another could only be plausible if the two hypercluster profiles were related in a small number of very specific ways, a simple course search for possible regulatory pairs of hyperclusters based on the mean profiles alone will enable us to presort hyperclusters. As well as providing a list of the most promising candidates before formal evaluation of excitation and inhibition the sorting will discover which pairs have no chance of being in a useful direct relationship with one another.

8.1.3 Intelligent Fast Search over Dependent Clusters

One feature which can significantly enhance the speed of the search is if the score function of different partitions can be expressed as simple functions of scores of their component clusters. Sadly, a naive implementation of dependence models, whilst admitting explicit scores over partitions, loses this decomposition property. However, conjugate Gaussian structures admit fast graphically-based propagation algorithms to sequentially update the joint distribution over the clusters. The simplest case occurs when the cliques of the Gaussian dependence model over the set of all regression parameters form a decomposable joint density where fast updating code is available from a number of sources.

Interestingly, in the class of models we consider, the long cyclic regulatory patterns existing in this domain mean that more complicated models can have a non-decomposable and sometimes non-graphical representation and lie in the class of non-recursive structural equation models (Koster, 1996).

8.1.4 Graphical Interface for Communication with Scientists

To be fully useful the results of the search over putative mechanisms needs to be communicated in a way which is in as familiar a format as possible for the scientist. This graph will need to be a logically coherent representation of the regulatory process sharing, as far as it is possible, the semantics of the graphs currently in use in this domain. Fortunately, we have access not only to scientists using these semantics but also, through Andrew Millar, to those currently formalising the representations of these graphically-based systems. The developed graphical framework will have a Bayesian-network-like structure and semantics similar to a Bayesian network but with a variety of different types of edges and different semantics to read from them. The nodes will be linked to their expected profiles and the links embellished with information about important features, such as phase delays and statistical strength of evidence.

In this thesis we focused on Bayesian clustering algorithms and we contributed to their development by not only designing new fast search algorithms of the vast heterogeneous space, but also by developing new ways of setting the model hyperparameters so that inferences both reflect the scientific needs and contribute to the inferential stability of the search. In particular, our experience demonstrates the value of talking to experimentalists and this is the reason that, despite the success of the class of partition models discussed in this thesis, we now propose to continue our research by including regulatory relationships in our Bayesian partition search.

Appendix A

Supplementary Figures and Tables for Ostreococcus tauri

We include here supplementary figures and tables to support the results discussed in Section 4.7. A comprehensive discussion of these results is presented in Monnier et al. (2009). The figures in this appendix show the biological processes of the genes belonging to some of the clusters obtained by BFC (Bayesian Fourier Clustering).

Figure S1 Coregulation of genes involved in basic transcription machinery during the night;

Figure S2 Clusters of genes involved protein synthesis including translation regulators, tRNA and amino biosynthesis around dawn;

Figure S3 Coregulation of DNA replication and DNA repair genes at the end of the light period;

Figure S4 Clusters of genes involved in photoprotection, defence against oxidative stress and DNA repair around midday;

Figure S5 Coregulation of genes involved in mitosis at dusk;

Figure S6 Coregulation of several genes involved in secretion;

Figure S7 Late night clusters of genes involved in chloroplast biogenesis, pigment biosynthesis, lipid biosynthesis and metabolism.

These results contributed to Fig. 4.11.

Figure S1

Transcription and mRNA processing (RNA polymerase, snoRP, RNA splicing, RNA methylase, RNA helicase), Ribosome biogenesis, tRNA and translation

Feat	Num	BFC	Gene description
3519	61		40S ribosomal protein S10 (RPS10A)
7616	61		DNA-directed RNA polymerase
3985*	61		KOG0272 U4/U6 small nuclear ribonucleoprotein Prp4 (contains WD40 repeats)
8033*	61		KOG0272 U4/U6 small nuclear ribonucleoprotein Prp4 (contains WD40 repeats)
7455	61		KOG2574 mRNA splicing factor PRP31
6169	61		60S ribosomal protein L10 (RPL10B)
4427	61		KOG0110 RNA-binding protein (RRM superfamily)
3081	61		KOG3503 H/ACA snoRNP complex, subunit NOP10
4169	61		KOG1596 Fibrillarin and related nucleolar RNA-binding proteins
7875	61		KOG2574 mRNA splicing factor PRP31
6369	61		KOG2409 KRR1-interacting protein involved in 40S ribosome biogenesis
6832	61		DEAD box RNA helicase (RH26)
487	61		KOG1253 tRNA methyltransferase
7804	61		DNA binding / DNA-directed RNA polymerase
1770	61		KOG0121 Nuclear cap-binding protein complex, subunit CBP20
5731	61		KOG3073 Protein required for 18S rRNA maturation and 40S ribosome biogenesis
1456	54		KOG0838 RNA Methylase, SpoU family
4501	54		KOG0924 mRNA splicing factor ATP-dependent RNA helicase
3457	54		KOG3064 RNA-binding nuclear protein (MAK16) containing a distinct C4 Zn-finger
423	54		KOG0433 Isoleucyl-tRNA synthetase
5462	54		KOG0603 Ribosomal protein S6 kinase
4689	54		KOG0333 U5 snRNP-like RNA helicase subunit
7951	54		KOG0048 Transcription factor, Myb superfamily
6894	54		KOG2054 Nucleolar RNA-associated protein (NRAP)
5351	54		KOG1801 tRNA-splicing endonuclease positive effector (SEN1)
4745	54		RNA binding
3183	54		KOG4213 RNA-binding protein La
5690	111		DNA-directed RNA polymerase II, putative
6054	111		pseudouridine synthase family protein
7993	111		KOG0302 Ribosome Assembly protein
1475	111		KOG1896 mRNA cleavage and polyadenylation factor II complex, subunit CFT1
4021	67		GL3 (GLABRA 3); transcription factor
6485	67		TRFL6 (TRF-LIKE 6); DNA binding / transcription factor
5403	67		DNA topoisomerase family protein
3045	67		PAB5 (POLY(A)-BINDING PROTEIN); RNA binding
2097	67		SPDS1 (SPERMIDINE SYNTHASE 1)
6271	67		KOG4008 rRNA processing protein RRP7
7209	29		KOG0147 Transcriptional coactivator CAPER (RRM superfamily)
6152	29		KOG3293 Small nuclear ribonucleoprotein (snRNP)
836	29		KOG0466 Translation initiation factor 2, gamma subunit (eIF-2gamma; GTPase)
4540	29	8	KOG3045 Predicted RNA methylase involved in rRNA processing
5349	29	9	pseudouridylate synthase
1258	29	11	KOG2102 Exosomal 3'-5' exoribonuclease complex, subunit Rrp44/Dis3
6259	29	13	tRNA/rRNA methyltransferase (SpoU) family protein
158	29	16	nonsense-mediated mRNA decay NMD3 family protein
6253	29	17	DRH1 (DEAD box RNA helicase 1)
2563	29	18	KOG2553 Pseudouridylate synthase
1140	29	20	KOG0343 RNA Helicase
7060	42	1	KOG2738 Putative methionine aminopeptidase
5502	42	1	KOG2809 Telomerase elongation inhibitor/RNA maturation protein PINX1
7032	42	1	KOG1855 Predicted RNA-binding protein
5987	42	2	KOG3909 Queuine-tRNA ribosyltransferase
7474	42	4	KOG2793 Putative N2,N2-dimethylguanosine tRNA methyltransferase
874	42	5	ADK1 (ADENYLATE KINASE 1); adenylate kinase
6533	42	7	KOG2809 Telomerase elongation inhibitor/RNA maturation protein PINX1
312	42	7	DEAD box RNA helicase, putative (RH20)
6888	42	7	RNA cyclase family protein
211	42	12	DEAD/DEAH box helicase, putative (RH10)
4980	92	2	APUM23 (ARABIDOPSIS PUMILIO 23); RNA binding
4018	92	3	nucleolar protein, putative
2930	92	3	KOG2187 tRNA uracil-5-methyltransferase
1629	92	5	KOG3492 Ribosome biogenesis protein NIP7
3311	92	10	KOG1070 rRNA processing protein Rrp5
7136	92	10	DEAD/DEAH box helicase, putative
5801	92	11	APUM24 (ARABIDOPSIS PUMILIO 24); RNA binding
5824	92	14	KOG2038 CAATT-binding transcription factor
1732	92	14	KOG0272 U4/U6 small nuclear ribonucleoprotein Prp4

6082	80	KOG3115 Methyltransferase-like protein
5955	80	KOG0148 Apoptosis-promoting RNA-binding protein TIA-1/TIAR
486	80	RBM28, rna-binding protein 28 (rna-binding motif protein 28)
2247	80	KOG0343 RNA Helicase
2224	80	KOG1098 Putative SAM-dependent rRNA methyltransferase SPB1
157	28	RNA binding / tRNA (guanine-N2-)-methyltransferase
376	28	dihydrouridine synthase family protein
2911	28	KOG1926 Predicted regulator of rRNA gene transcription (MYB-binding protein)
365	28	KOG1882 Transcriptional regulator SNIP1, contains FHA domain
2070	28	KOG1562 Spermidine synthase
7415	28	40S ribosomal protein S12 (RPS12C)
3297	28	KOG3051 RNA binding/translational regulation protein of the SUA5 family
5306	28	KOG2102 Exosomal 3'-5' exoribonuclease complex, subunit Rrp44/Dis3
7507	12	KOG4661 Hsp27-ERE-TATA-binding protein
3581	12	KOG2529 Pseudouridine synthase
2944	12	KOG1919 RNA pseudouridylate synthases
47	12	KOG0924 mRNA splicing factor ATP-dependent RNA helicase
7680	12	KOG2793 Putative N2,N2-dimethylguanosine tRNA methyltransferase

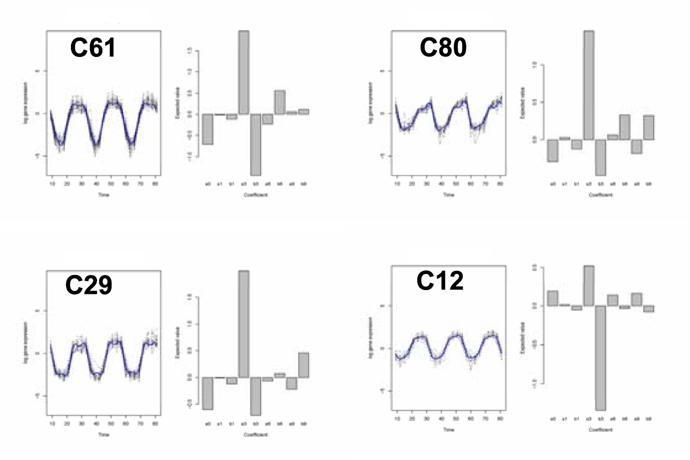


Figure S2

Translation factors , amino acid synthesis, tRNA synthesis, ribosome biogenesis and transcription

FeatNum	BFC	Gene description
4992	39	methionine--tRNA ligase, putative / methionyl-tRNA synthetase, putative
4471	39	KOG2240 RNA polymerase II general transcription factor BTF3
8027	39	KOG0628 Aromatic-L-amino-acid/L-histidine decarboxylase
932	39	KOG4492 Chorismate synthase
3437	39	KOG0401 Translation initiation factor 4F,(eIF-4G)
5511	39	translation initiation factor IF-2, chloroplast, putative
6979	39	KOG0433 Isoleucyl-tRNA synthetase
3921	39	translation initiation factor 3 (IF-3) family protein
6395	39	aspartate/glutamate/uridylate kinase family protein
5130	39	KOG2072 Translation initiation factor 3, subunit a (eIF-3a)
2278	39	KOG1801 tRNA-splicing endonuclease positive effector (SEN1)
6326	39	KOG1801 tRNA-splicing endonuclease positive effector (SEN1)
5015	39	KOG0556 Aspartyl-tRNA synthetase
7079	39	KOG1885 Lysyl-tRNA synthetase (class II)
5302	39	KOG3677 RNA polymerase I-associated factor - PAF67
6634	39	KOG1560 Translation initiation factor 3, subunit h (eIF-3h)
4772	39	KOG4655 U3 small nucleolar ribonucleoprotein (snoRNP) component
1197	39	KOG2436 Acetylglutamate kinase/acetylglutamate synthase
6767	39	KOG4163 Prolyl-tRNA synthetase
5791	39	KOG2314 Translation initiation factor 3, subunit b (eIF-3b)
3201	39	60S ribosomal protein L23 (RPL23B)
203	39	SSR16 (ribosomal protein S16); structural constituent of ribosome
474	39	tRNA pseudouridine synthase family protein
1675	107	KOG2975 Translation initiation factor 3, subunit f (eIF-3f)
7489	107	KOG0401 Translation initiation factor 4F, ribosome/ (eIF-4G)
1289	107	KOG2072 Translation initiation factor 3, subunit a (eIF-3a)
5888	107	ribosomal protein L1 family protein
4107	107	KOG3499 60S ribosomal protein L38
6788	107	APG3 (ALBINO AND PALE GREEN); translation release factor
2545	107	APUM12 (ARABIDOPSIS PUMILIO 12); RNA binding
5547	115	KOG0257 Kynurenine aminotransferase, glutamine transaminase K
5963	115	KOG1579 Homocysteine S-methyltransferase
6463	115	KOG4175 Tryptophan synthase alpha chain
5339	115	KOG1637 Threonyl-tRNA synthetase

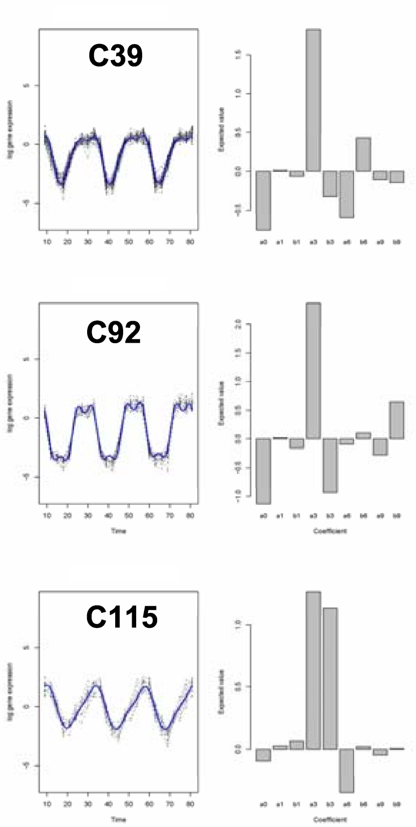


Figure S3

DNA replication, DNA repair, chromosome structure, histone and chromatin remodelling , cell cycle control, organelles division, transcription

FeatNum	BFC	Gene description
7024	41	KOG2011 Sister chromatid cohesion complex Cohesin, subunit STAG/IRR1/SCC3
6006	41	KOG0386 Chromatin remodeling complex SWI/SNF, component SWI2)
7990	41	KOG2543 Origin recognition complex, subunit 5
4965	41	KOG1513 Nuclear helicase MOP-3/SNO (DEAD-box superfamily)
5441	41	KOG4603 TBP-1 interacting protein
2614	41	KOG1978 DNA mismatch repair protein - MLH2/PMS1/Pms2 family
5851	41	FTSZ1-1 (FtsZ1-1); structural molecule
2271	41	KOG1625 DNA polymerase alpha-primase complex, polymerase-associated subunit B
3595	41	KOG3108 Single-stranded DNA-binding replication protein A (RPA), medium (30 kD) subunit
6779	41	nucleotide binding
5453	64	KOG3125 Thymidine kinase
3411	64	KOG0481 DNA replication licensing factor, MCM5 component
347	64	KOG3265 Histone chaperone involved in gene silencing
7195	64	PCNA1 (PROLIFERATING CELLULAR NUCLEAR ANTIGEN)
3821	64	ATR (ATAXIA TELANGIECTASIA-MUTATED AND RAD3-RELATED)
5298	64	EMB2411 (EMBRYO DEFECTIVE 2411); ATP-dependent DNA helicase
6286	64	KOG0969 DNA polymerase delta, catalytic subunit
748	64	KOG2299 Ribonuclease HI
7630	64	ARID/BRIGHT DNA-binding domain-containing protein
7526	87	KOG0481 DNA replication licensing factor, MCM5 component
7654	87	KOG2851 Eukaryotic-type DNA primase, catalytic (small) subunit
1998	57	MFP1 (MAR BINDING FILAMENT-LIKE PROTEIN 1)
8049	57	KOG1525 Sister chromatid cohesion complex Cohesin, subunit PDS5
2893	57	ATRAD17 (RADIATION SENSITIVE)
305	57	KOG2807 RNA polymerase II transcription initiation/nucleotide excision repair factor TFIIH
8051	57	KOG0968 DNA polymerase zeta, catalytic subunit
1266	57	MYB3R-5 (myb domain protein 3R-5); DNA binding / transcription factor
2012	116	KOG0996 Structural maintenance of chromosome protein 4 (Condensin, subunit C)
4534	116	WEE1 (Arabidopsis wee1 kinase homolog); kinase/ protein kinase
1785	116	KOG3786 RNA polymerase II assessory factor Cdc73p
3244	116	KOG4373 Predicted 3'-5' exonuclease
6438	116	ERCC1 (UV REPAIR DEFICIENT 7)
5055	46	KOG1112 Ribonucleotide reductase, alpha subunit
4903	46	ARC5 (ACCUMULATION AND REPLICATION OF CHLOROPLAST 5); GTP binding / GTPase
5585	46	KOG1744 Histone H2B
3730	46	CDKB (CDC2-LIKE GENE) kinase
4507	122	histone H3.2
1916	122	MFP1 (MAR BINDING FILAMENT-LIKE PROTEIN 1)
7336	122	kinesin motor protein-related
3193	129	KOG0642 Cell-cycle nuclear protein, contains WD-40 repeats
7574	129	KOG0220 Mismatch repair ATPase MSH4 (MutS family)
2589	129	KOG4109 Histone H3 (Lys4) methyltransferase complex, subunit CPS25/DPY-30
3608	129	KOG0851 Single-stranded DNA-binding replication protein A (RPA), large (70 kD)
3410	109	ARC6 (ACCUMULATION AND REPLICATION OF CHLOROPLASTS 6)
2185	109	KOG0386 Chromatin remodeling complex SWI/SNF, component SWI2 (DNA/RNA helicase)
6362	109	KOG2519 5'-3' exonuclease
6318	109	KOG0481 DNA replication licensing factor, MCM5 component
1466	109	KOG2097 Predicted N6-adenine methylase involved in transcription regulation
1228	102	cell division cycle protein 48-related / CDC48-related
3242	102	KOG1513 Nuclear helicase MOP-3/SNO (DEAD-box superfamily)

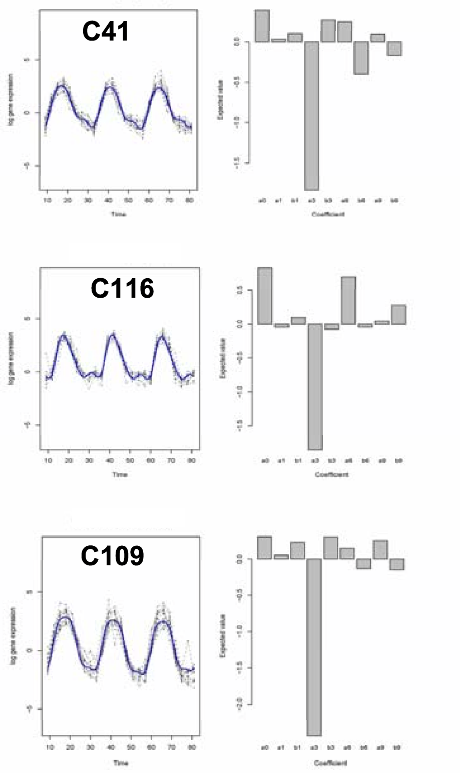


Figure S4

Oxidative stress, carotenoids biosynthesis, DNA repair and remodelling, photosynthesis and lipid metabolism

Feat Num	BFC	Gene description
2303	85	thioredoxin-related
4249	85	ACCELERATED CELL DEATH 1, PHEOPHORBIDE A OXYGENASE
5831	85	ATF1/TRXF1 (THIOREDOXIN F-TYPE 1)
2506	85	KOG0191 Thioredoxin/protein disulfide isomerase
3791	85	thylakoid lumen 18.3 kDa protein
2186	85	NPQ1 (NON-PHOTOCHEMICAL QUENCHING 1)
3897	27	UVR3 (UV REPAIR DEFECTIVE 4)
6562	27	photosystem II protein M
2550	27	ATMINE1 (ARABIDOPSIS HOMOLOGUE OF BACTERIAL MINE 1)
143	27	CSD2 (COPPER/ZINC SUPEROXIDE DISMUTASE 2)
7017	7	FORMAMIDOPYRIMIDINE-DNA GLYCOSYLASE 1,
807	7	DNA cross-link repair protein-related
2184	7	CHR17 (CHROMATIN REMODELING FACTOR17); DNA-dependent ATPase
2538	7	DEAD/DEAH box helicase, putative (RH22)
3014	7	ATR2 (ARABIDOPSIS P450 REDUCTASE 2)
7850	7	violaxanthin de-epoxidase-related
7938	7	GPPS (GERANYLPYROPHOSPHATE SYNTHASE)
6921	7	ATERS/ERS/OVA3 (OVULE ABORTION 3); glutamate-tRNA ligase
3607	7	PGR5 (PROTON GRADIENT REGULATION 5)
7870	24	KOG4720 Ethanolamine kinase
7800	24	KOG4254 Phytoene desaturase
6238	24	ABA1 (ABA DEFICIENT 1); zeaxanthin epoxidase
4791	24	HEMA1; glutamyl-tRNA reductase
4877	24	KOG1336 Monodehydroascorbate/ferredoxin reductase
7861	24	KOG4232 Delta 6-fatty acid desaturase/delta-8 sphingolipid desaturase
7480	76	FAD2 (FATTY ACID DESATURASE 2); delta12-fatty acid dehydrogenase
6483	76	KOG1737 Oxysterol-binding protein
6947	76	KOG1285 Beta, beta-carotene 15,15'-dioxygenase
4703	76	CLB6 (CHLOROPLAST BIOGENESIS 6)
1285	76	PDE149 (PIGMENT DEFECTIVE 149)
5231	76	FAD2 (FATTY ACID DESATURASE 2); delta12-fatty acid dehydrogenase

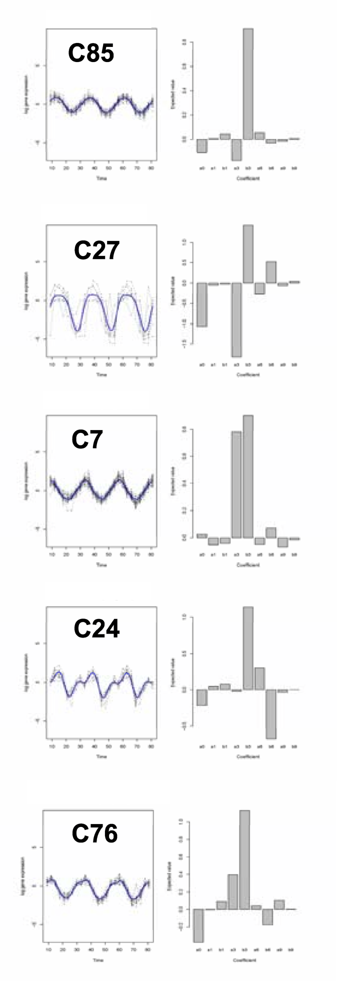


Figure S5

Cytoskeleton, regulatory kinases, DNA replication and repair, chromosome structure, secretion, oxidative stress and iron

FeatNum	BFC	Gene description
6489	71	KOG2328 Chromosome condensation complex Condensin, subunit H
4976	71	KOG0414 Chromosome condensation complex Condensin, subunit D2
7432	71	ATORC1B/ORC1B/UNE13 (ORIGIN OF REPLICATION COMPLEX 1B)
4436	71	CHR24 (chromatin remodeling 24); ATP binding / DNA binding / helicase
388	71	KOG1068 Exosomal 3'-5' exoribonuclease complex, subunit Rrp41 and related exoribonucleases
3924	71	ATK2 (ARABIDOPSIS THALIANA KINESIN 2); microtubule motor
5956	71	KOG0517 Beta-spectrin
7691	71	KOG2464 Serine/threonine kinase (haspin family)
788	71	POK1 (PHRAGMOPLAST ORIENTING KINESIN 1); microtubule motor
790	71	KOG1433 DNA repair protein RAD51/RHP55
4287	71	ATK5 (Arabidopsis thaliana kinesin 5); microtubule motor
4259	71	KOG0192 Tyrosine kinase specific for activated (GTP-bound) p21cdc42Hs
4784	71	ATEB1C (MICROTUBULE END BINDING PROTEIN 1); microtubule binding
6717	71	KOG2084 Predicted histone tail methylase containing SET domain
1967	49	CYCB2;2 (CYCLIN B2;2); cyclin-dependent protein kinase regulator
1146	49	KOG0161 Myosin class II heavy chain
7911	49	KOG2058 Ypt/Rab GTPase activating protein
2707	49	KOG2025 Chromosome condensation complex Condensin, subunit G
6733	104	KOG2510 SWI-SNF chromatin-remodeling complex protein
5593	104	KOG0161 Myosin class II heavy chain
1088	104	CKS1 (CDK-SUBUNIT 1); cyclin-dependent protein kinase
5330	104	ATAUR1 (ATAURORA1); histone serine kinase(H3-S10 specific) / kinase/ protein serine/threonine kinase
7755	104	KOG0192 Tyrosine kinase specific for activated (GTP-bound) p21cdc42Hs
7463	36	KOG0239 Kinesin (KAR3 subfamily)
3385	36	SMC2 (STRUCTURAL MAINTENANCE OF CHROMOSOMES 2)
8020	77	KOG1651 Glutathione peroxidase
5409	77	KOG2065 Gamma-tubulin ring complex protein
5342	77	KOG0916 1,3-beta-glucan synthase/callose synthase catalytic subunit
3275	77	KOG0605 NDR and related serine/threonine kinases
4264	77	KOG0819 Annexin
5069	77	MOR1 (MICROTUBULE ORGANIZATION 1)
2312	62	KOG0239 Kinesin (KAR3 subfamily)
7847	62	APX1 (ASCORBATE PEROXIDASE 1, MATERNAL EFFECT EMBRYO ARREST 6)
2795	62	KOG1003 Actin filament-coating protein tropomyosin
1181	25	KOG1010 Rb (Retinoblastoma tumor suppressor)-related protein
2326	25	ferredoxin family protein
2886	25	KOG0578 p21-activated serine/threonine protein kinase
4001	25	LEM3 (ligand-effect modulator 3) family protein / CDC50 family protein
5633	25	ATFER3 (FERRITIN 3); ferric iron binding
1318	25	MSH6 (MUTS HOMOLOG 6-1)

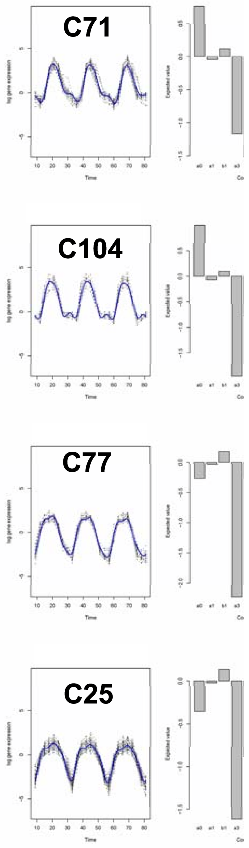
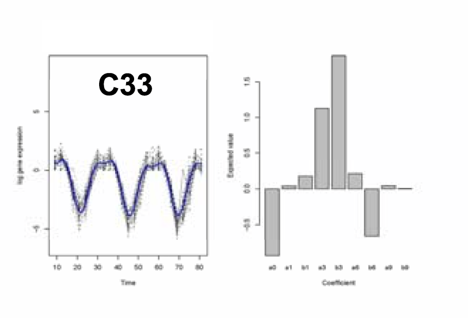


Figure S6

Chloroplast biogenesis and chlorophyll biosynthesis/degradation, lipid metabolism carotenoids biosynthesis, metabolism

Feat Num	BFC	Gene description
6521	33	3-oxoacyl-(acyl-carrier protein) reductase, chloroplast
4009	33	CHLI2; magnesium chelatase
4579	33	geranylgeranyl reductase
5495	33	KAS I (3-KETOACYL-ACYL CARRIER PROTEIN SYNTHASE I)
3943	33	KOG4232 Delta 6-fatty acid desaturase/delta-8 sphingolipid desaturase
5411	33	PRK (PHOSPHORIBULOKINASE)
897	33	LPD1 (LIPOAMIDE DEHYDROGENASE 1)
1506	33	fructose-bisphosphate aldolase, putative
5664	33	GC1 (GIANT CHLOROPLAST 1); catalytic/ coenzyme binding
6389	33	CH1 (CHLORINA 1); chlorophyll a oxygenase
4564	33	cytochrome b6
5066	33	TIM (TRIOSEPHOSPHATE ISOMERASE)
2820	33	GcpE (CHLOROPLAST BIOGENESIS 4)
3625	33	KOG3773 Adiponutrin and related vesicular transport proteins
1001	33	ACD1 (ACCELERATED CELL DEATH 1, PHEOPHORBIDE A OXYGENASE)
1021	33	GGPS1 (GERANYLGERANYL PYROPHOSPHATE SYNTHASE 1)
6294	33	APE1 (ACCLIMATION OF PHOTOSYNTHESIS TO ENVIRONMENT)
6278	33	FAB1 (FATTY ACID BIOSYNTHESIS 1); fatty-acid synthase
202	33	ATSS3 (STARCH SYNTHASE 3); starch synthase/ transferase
3207	33	KOG1448 Ribose-phosphate pyrophosphokinase
182	33	KOG1470 Phosphatidylinositol transfer protein PDR16 and related proteins
875	6	KOG1458 Fructose-1,6-bisphosphatase
7194	6	RPE (EMBRYO DEFECTIVE 2728); ribulose-phosphate 3-epimerase
1808	6	LUT1 (LUTEIN DEFICIENT 1); oxygen binding
267	6	KOG4153 Fructose 1,6-bisphosphate aldolase
3775	6	PDS3 (PHYTOENE DESATURASE)
5757	6	3,8-divinyl protochlorophyllide a 8-vinyl reductase



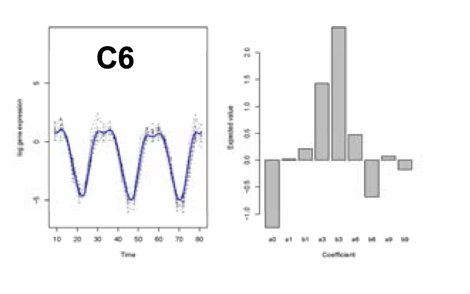
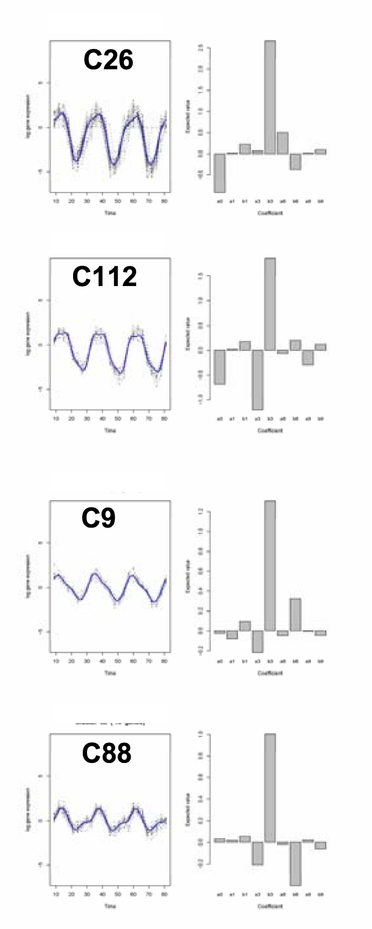


Figure S7

Chlorophyll and photosystem biosynthesis, lipid metabolism

Feat Num	BFC	Gene description
450	26	KOG2872 Uroporphyrinogen decarboxylase
6703*	26	HEMB1; porphobilinogen synthase
7529*	26	HEMB1; porphobilinogen synthase
6976	26	LIN2 (LESION INITIATION 2); coproporphyrinogen oxidase
6175	26	lil3 protein, putative
1865	26	CHLD/PDE166 (PIGMENT DEFECTIVE 166); magnesium chelatase
5212	26	PSBO-2/PSBO2 (PHOTOSYSTEM II SUBUNIT O-2); oxygen evolving
5099	26	GSA2 (GLUTAMATE-1-SEMIALDEHYDE 2,1-AMINOMUTASE 2)
1047	26	GSA2 (GLUTAMATE-1-SEMIALDEHYDE 2,1-AMINOMUTASE 2)
4368	13	PORA (Protochlorophyllide reductase A)
6791	13	CH1 (CHLORINA 1); chlorophyll a oxygenase
67	13	HEME1; uroporphyrinogen decarboxylase
2951	112	LHCA2 (Photosystem I light harvesting complex gene 2)
4633	112	LHCA2 (Photosystem I light harvesting complex gene 2)
4297	112	PSBY (photosystem II BY)
2720	112	PSBP-2 (photosystem II subunit P-2); calcium ion binding
4520	31	LHCA1
691	31	PSAD-2 (photosystem I subunit D-2)
166	31	LHCB4.3 (LIGHT HARVESTING COMPLEX PSII); chlorophyll binding
3323	9	FLU (FLUORESCENT IN BLUE LIGHT); binding
4040	9	LPA1 (LOW PSII ACCUMULATION1); binding
1049	9	thylakoid lumen 15.0 kDa protein
1440	88	photosystem II reaction center W (PsbW) protein-related
4983	88	plastid-lipid associated protein PAP / fibrillin family protein
7663	88	photosystem II family protein
6306	88	CIP1 (COP1-INTERACTIVE PROTEIN 1)
1270	88	family II extracellular lipase, putative



Appendix B

Supplementary Figures for Arabidopsis thaliana

In this appendix we give the results of our I-MAP optimally found clusters of the final AHC partition search presented in Chapter 5. Because of the speed of our method, many iterations could be performed over a grid of hyperparameters. The results given below correspond to the clustering giving the highest score with the choice of variance matrix $V = vI$ where $v = 0.498$.

After the clustering process, it was found helpful to the biologists to classify the posterior mean profiles into various shapes. The first five classifications cover the clusters identified as circadian over both 24-hour periods; the last five those that aren't. Types I, VI and VII are delineated by objective criteria whilst the remaining types are classified by eye. This is nevertheless useful as a guideline to the broad classes of behaviour that are displayed.

Further details on the analysis can be found in Anderson et al. (2006).

I. Sinusoidal: those clusters with SHR>0.65 and more than 11 genes.

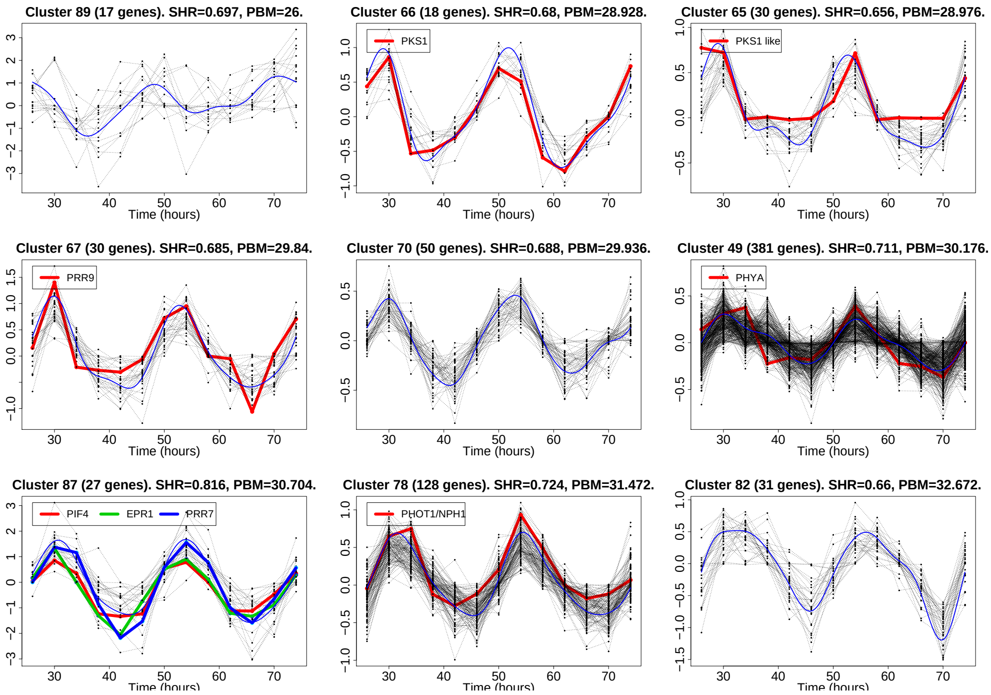


Figure B.1: Type I clusters: sinusoidal. The black dots are the data points (joined by dotted lines) and the blue line is the posterior mean for the cluster. Genes marked as interesting by the biologists are highlighted in thicker lines of red, green and blue.

II. Sharply rising then sharply falling.

III. Sharply falling then sharply rising.

IV. Sharply rising then drifting back to zero.

V. Sharply falling then drifting back to zero.

VI. Clusters classified as potentially not interesting by the algorithm.

VII. Clusters classified as O by the algorithm.

VIII. Potentially circadian, but not accurately repeated: clusters with a peak or trough in one 24 hour period, but not in the other.

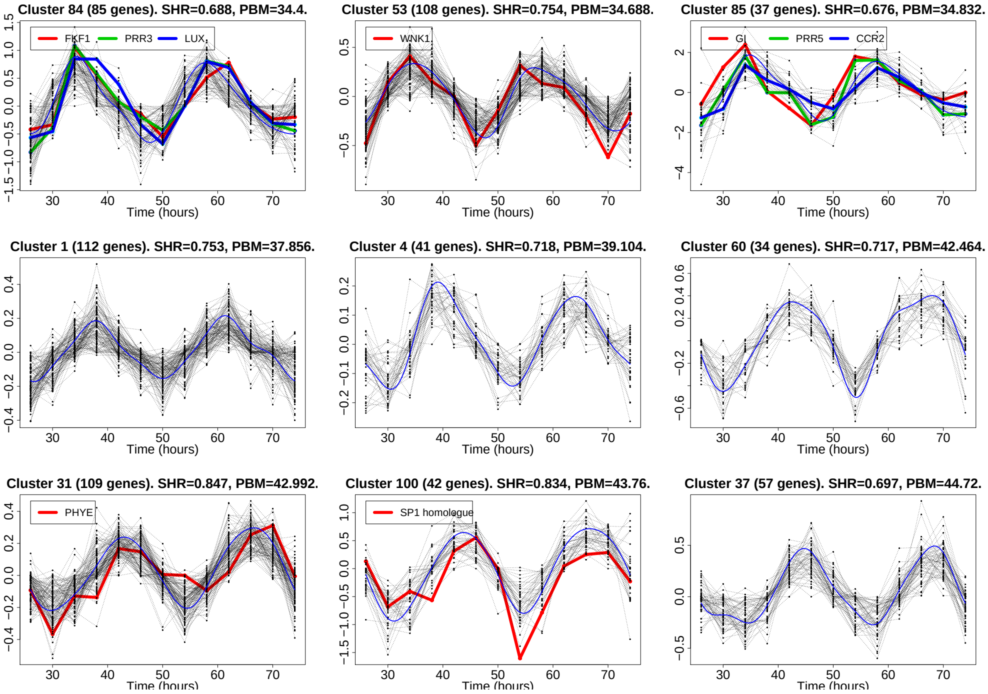


Figure B.2: Type I clusters: sinusoidal. The black dots are the data points (joined by dotted lines) and the blue line is the posterior mean for the cluster. Genes marked as interesting by the biologists are highlighted in thicker lines of red, green and blue.

IX. Outliers: clusters containing less than 11 genes.

X. ‘Junk’: clusters with expressions close to zero and non-circadian profiles.

The profiles of each of the 100 clusters identified among the interesting genes are shown in figures B.1 to B.16 classified in order according to the ten types above. Within each type, the clusters are sorted by their phase by maximum (the maximum value of the posterior mean in the first 24 hours). The second harmonic ratio (SHR) and phase by maximum (PBM) are given on each plot.

Figures B.17 and B.18 illustrate the robustness of the initial gene estimates to misspecification of hyperparameters. The clustering is very different when $v = 10,000$

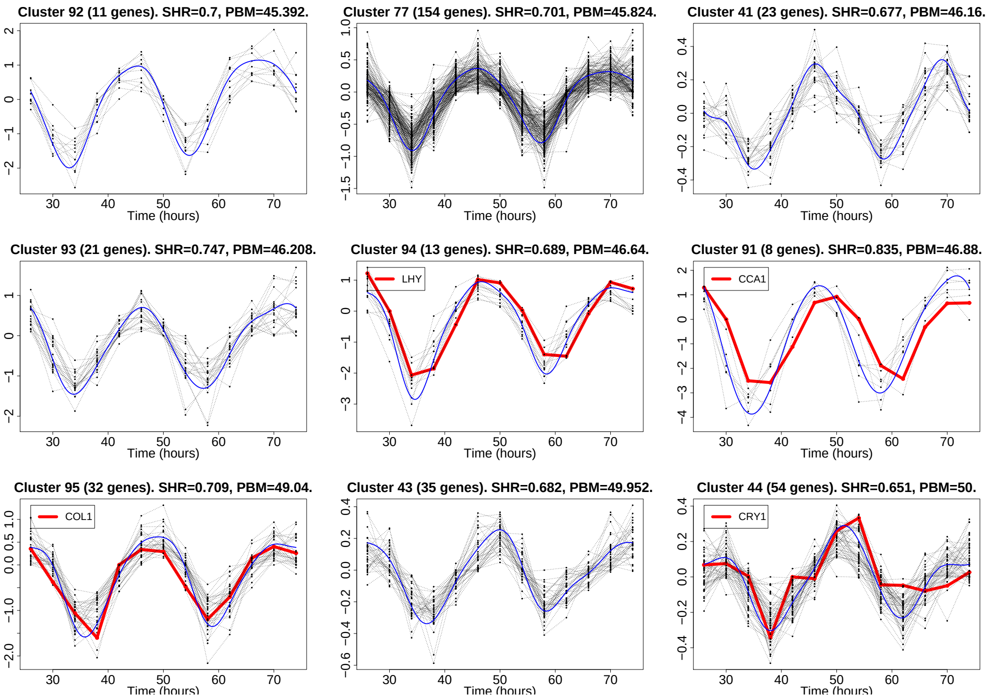


Figure B.3: Type I clusters: sinusoidal. The black dots are the data points (joined by dotted lines) and the blue line is the posterior mean for the cluster. Genes marked as interesting by the biologists are highlighted in thicker lines of red, green and blue.

rather than $v = 0.498$ giving us far fewer, but larger, clusters (34 instead of 100) so that we get intrinsically different solutions. Despite this, the estimated profiles of most genes do not radically differ under changes in v. This means that the sets of genes identified to have interesting profiles do not change greatly over large ranges of v. Furthermore, genes having similar profiles remain close.

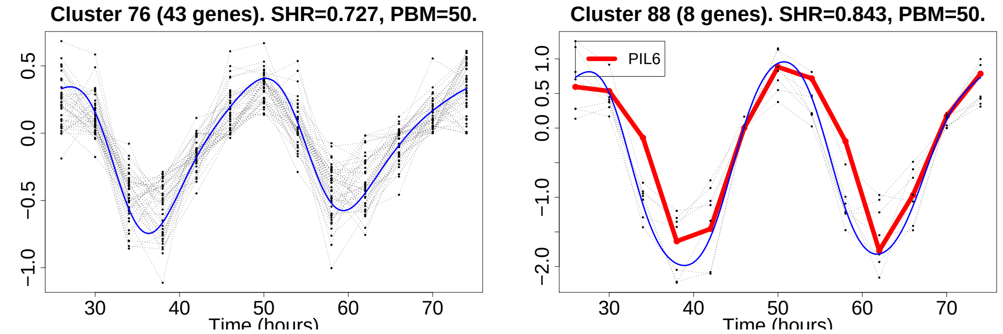


Figure B.4: Type I clusters: sinusoidal. The black dots are the data points (joined by dotted lines) and the blue line is the posterior mean for the cluster. Genes marked as interesting by the biologists are highlighted in thicker lines of red, green and blue.

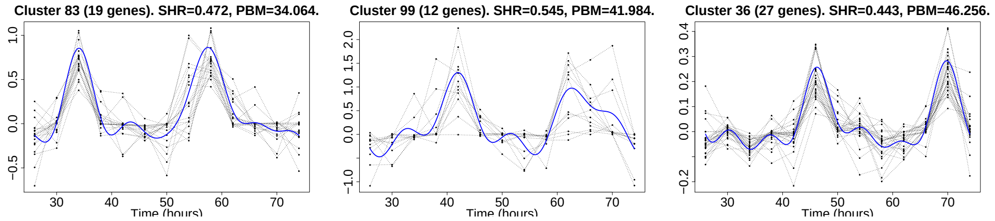


Figure B.5: Type II clusters: sharply rising then sharply falling. The black dots are the data points (joined by dotted lines) and the blue line is the posterior mean for the cluster.

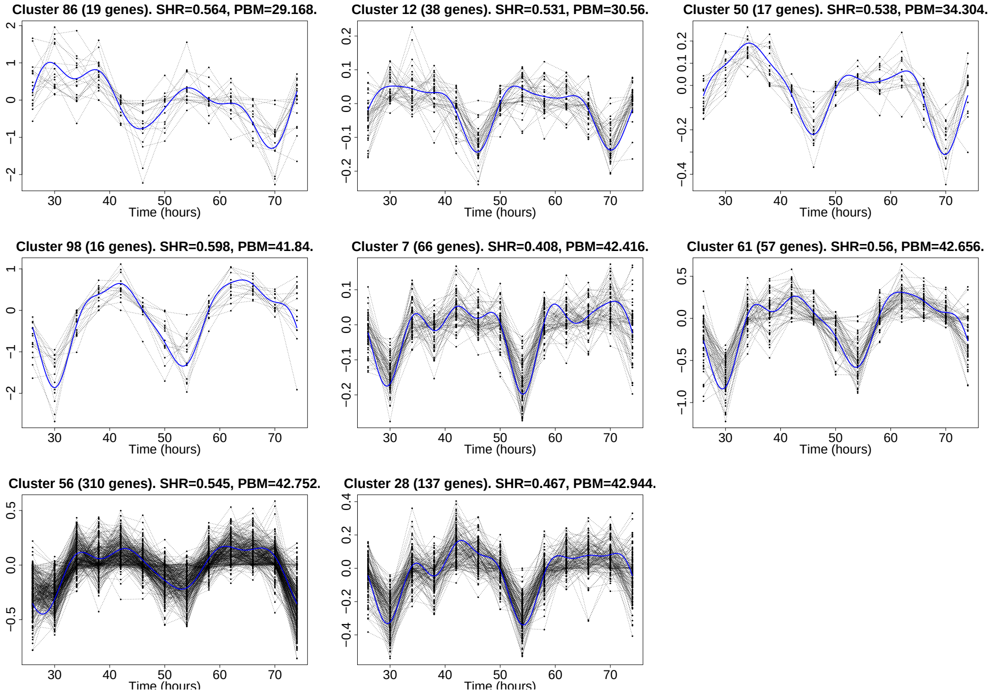


Figure B.6: Type III clusters: sharply falling then sharply rising. The black dots are the data points (joined by dotted lines) and the blue line is the posterior mean for the cluster.

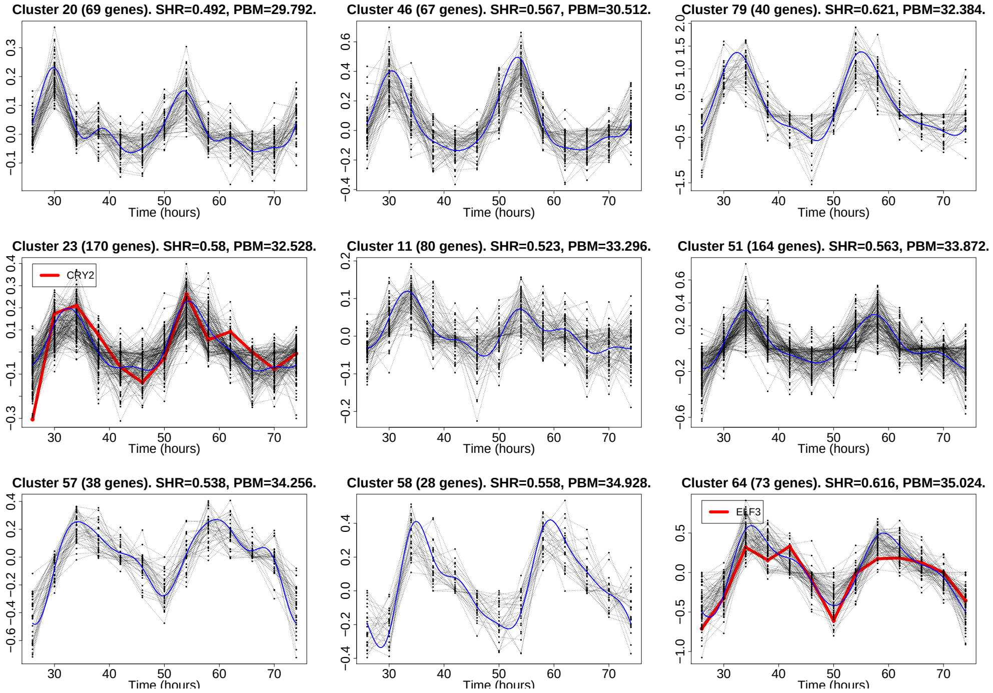


Figure B.7: Type IV clusters: sharply rising then drifting back to zero. The black dots are the data points (joined by dotted lines) and the blue line is the posterior mean for the cluster. Genes marked as interesting by the biologists are highlighted in thicker lines of red, green and blue.

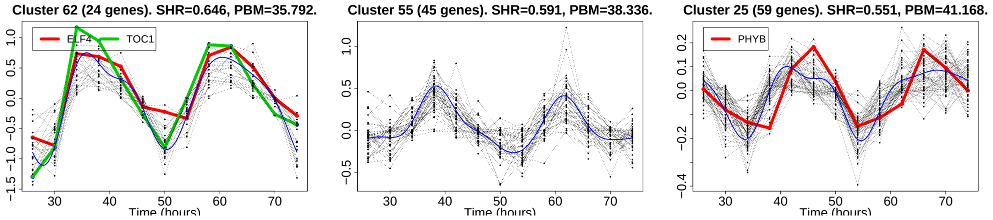


Figure B.8: Type IV clusters: sharply rising then drifting back to zero. The black dots are the data points (joined by dotted lines) and the blue line is the posterior mean for the cluster. Genes marked as interesting by the biologists are highlighted in thicker lines of red, green and blue.

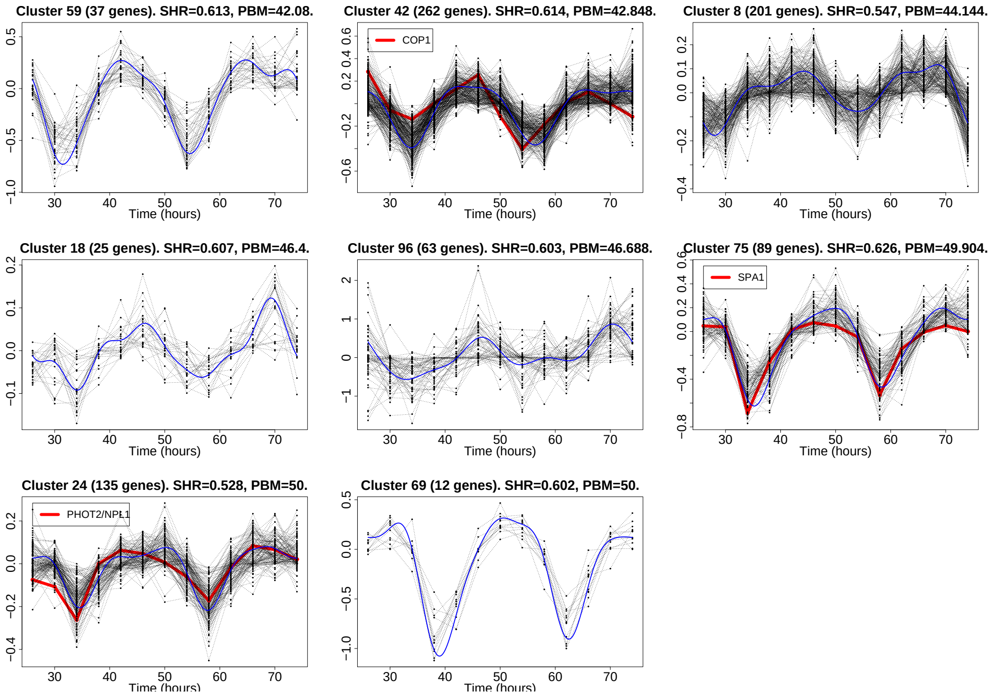


Figure B.9: Type V clusters: sharply falling then drifting back to zero. The black dots are the data points (joined by dotted lines) and the blue line is the posterior mean for the cluster. Genes marked as interesting by the biologists are highlighted in thicker lines of red, green and blue.

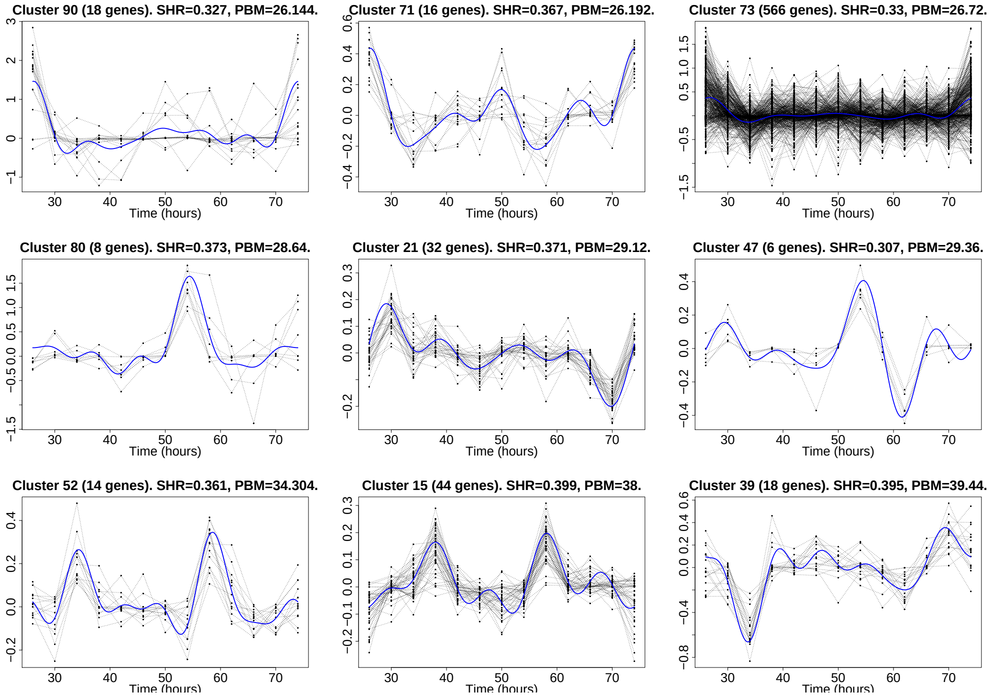


Figure B.10: Type VI clusters: potentially not interesting. The black dots are the data points (joined by dotted lines) and the blue line is the posterior mean for the cluster.

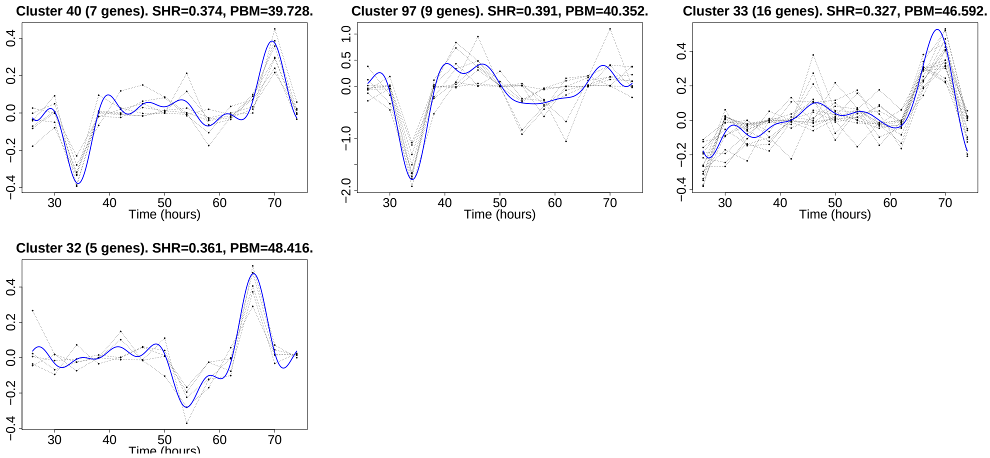


Figure B.11: Type VI clusters: potentially not interesting. The black dots are the data points (joined by dotted lines) and the blue line is the posterior mean for the cluster.

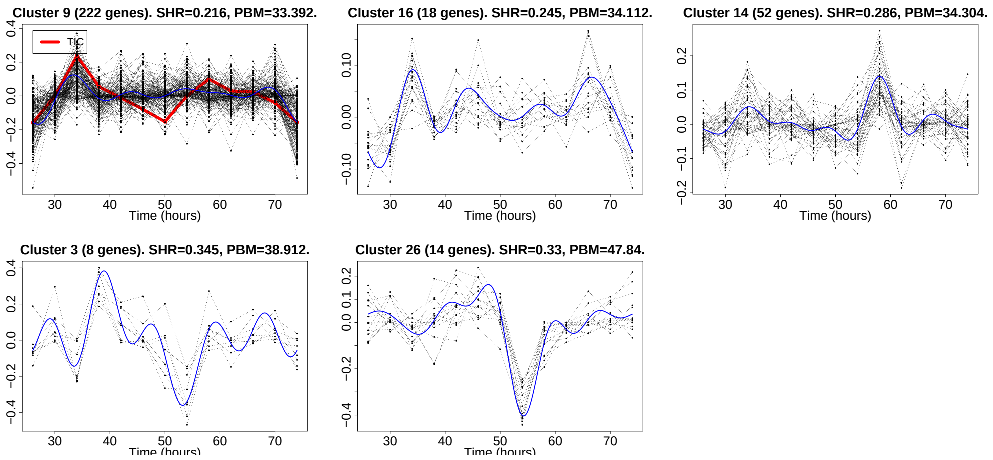


Figure B.12: Type VII clusters: other. The black dots are the data points (joined by dotted lines) and the blue line is the posterior mean for the cluster. Genes marked as interesting by the biologists are highlighted in thicker lines of red, green and blue.

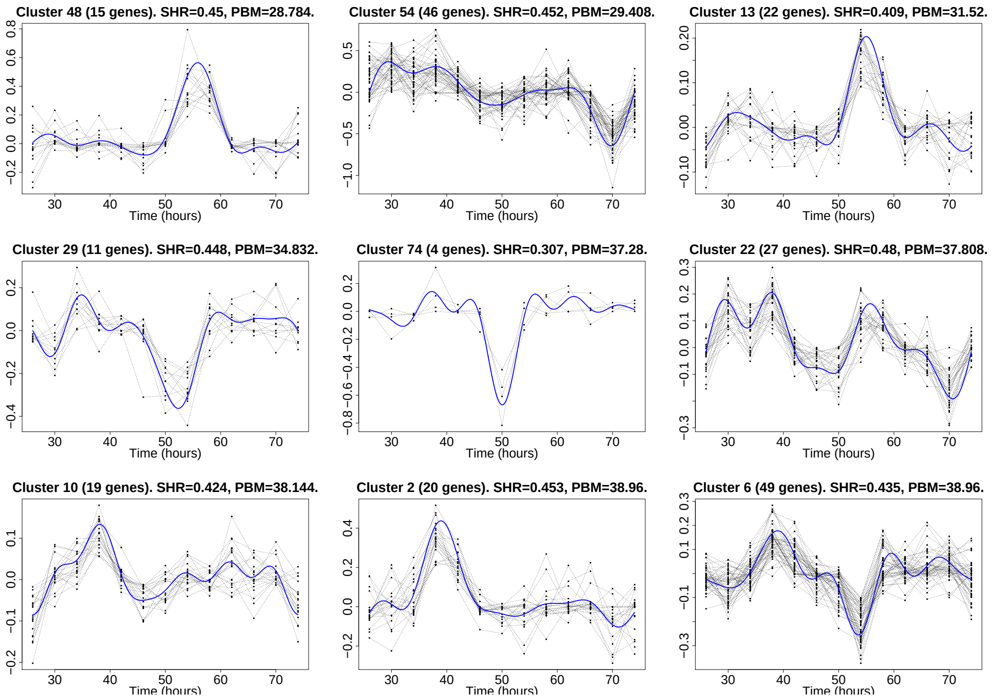


Figure B.13: Type VIII clusters: potentially circadian, but not repeated. The black dots are the data points (joined by dotted lines) and the blue line is the posterior mean for the cluster.

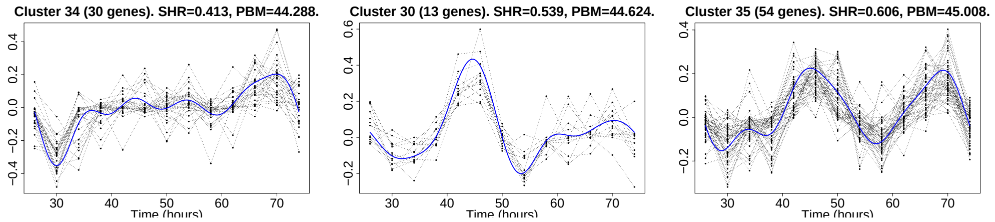


Figure B.14: Type VIII clusters: potentially circadian, but not repeated. The black dots are the data points (joined by dotted lines) and the blue line is the posterior mean for the cluster.

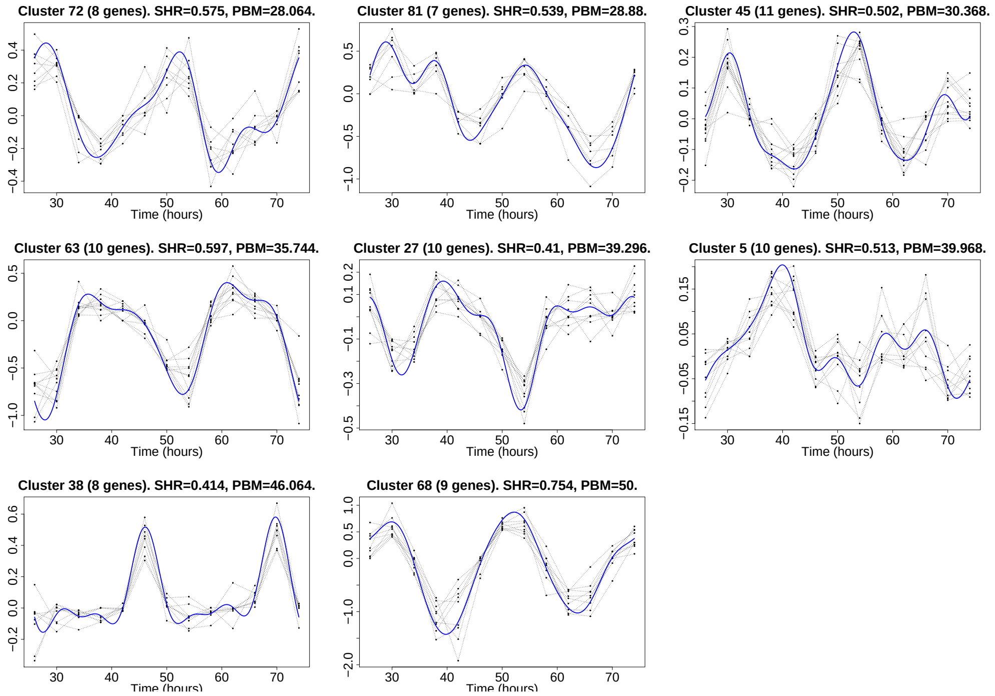


Figure B.15: Type IX clusters: outliers. The black dots are the data points (joined by dotted lines) and the blue line is the posterior mean for the cluster.

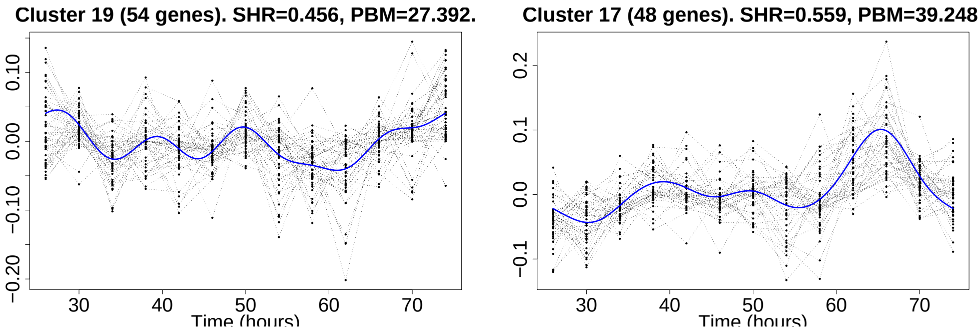


Figure B.16: Type X clusters: not interesting. The black dots are the data points (joined by dotted lines) and the blue line is the posterior mean for the cluster.

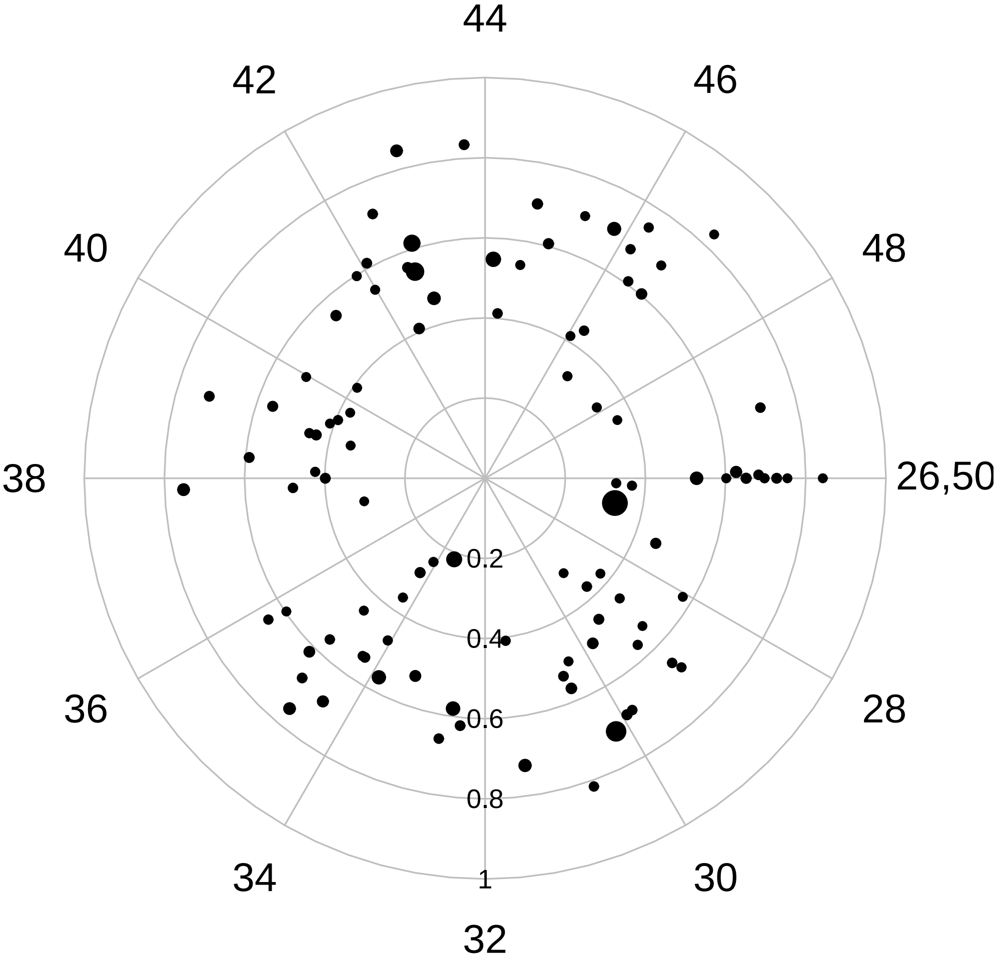


Figure B.17: Phase plot of the clusters of the final gene set with $v = 0.498$. Each dot is one cluster, its radius is proportional to the number of genes it contains. Its distance from the origin gives its second harmonic ratio, and the angle indicates the phase by maximum of the posterior mean profile.

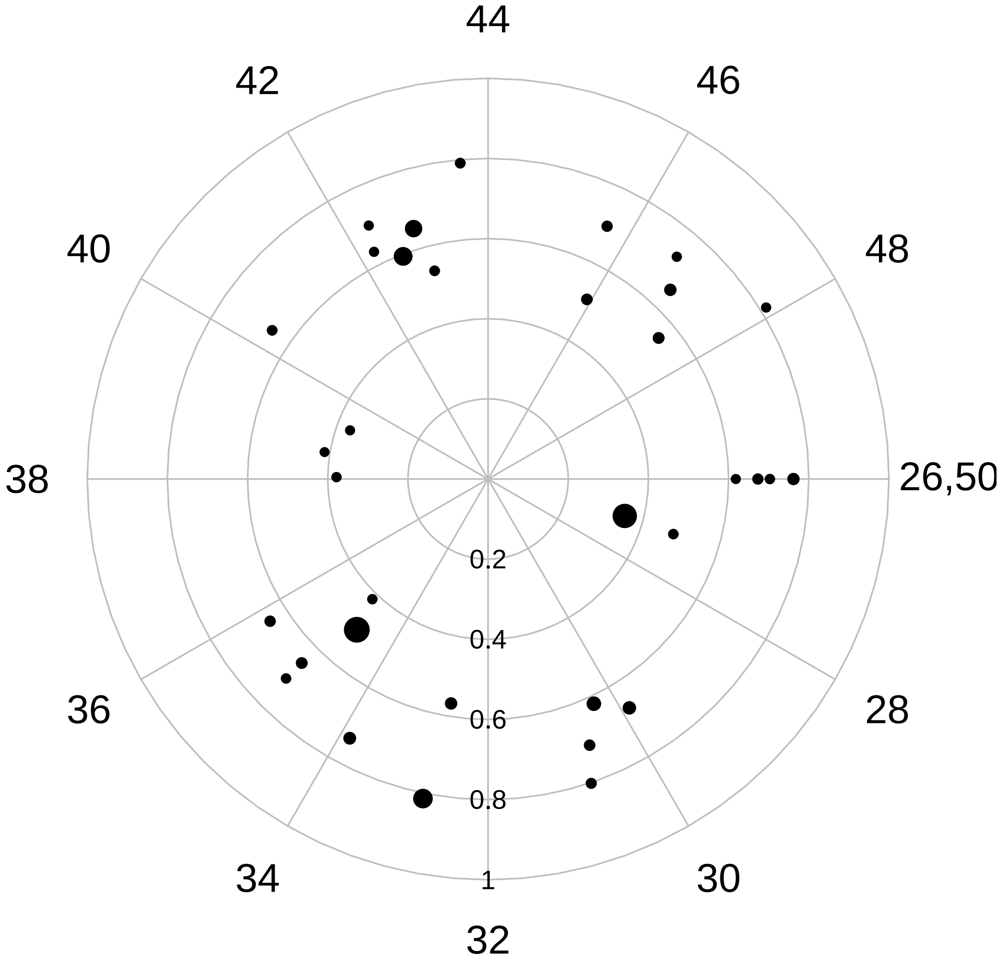


Figure B.18: Phase plot of the clusters of the final gene set with $v = 10,000$. Each dot is one cluster, its radius is proportional to the number of genes it contains. Its distance from the origin gives its second harmonic ratio, and the angle indicates the phase by maximum of the posterior mean profile. The structure is broadly similar to that of figure B.17.

Bibliography

Amaratunga, D. and Cabrera, J. (2003). *Exploration and analysis of DNA microarray and protein array data*. Wiley-IEEE.

Anderson, P. E., Smith, J. Q., Edwards, K. D., and Millar, A. J. (2006). Guided Conjugate Bayesian Clustering for Uncovering Rhythmically expressed Genes. Technical Report 07, CRiSM Working Paper, University of Warwick, UK.

Angelini, C., De Canditiis, D., Mutarelli, M., and Pensky, M. (2007). A Bayesian Approach to Estimation and Testing in Time-course Microarray Experiments. *Statistical Applications in Genetics and Molecular Biology*, 6(1):1299.

Banfield, J. D. and Raftery, A. E. (1993). Model-Based Gaussian and Non-Gaussian Clustering. *Biometrics*, 49(3):803–821.

Bar-Joseph, Z., Gifford, D., Jaakkola, T., and Simon, I. (2002). A new approach to analyzing gene expression time series data. *Proceedings of the 6th Annual International Conference on Computational Biology*, pages 39–48.

Barry, D. and Hartigan, J. A. (1992). Product Partition Models for Change Point Problems. *Annals of Statistics*, 20(1):260–279.

Ben-Dor, A., Shamir, R., and Yakhini, Z. (1999). Clustering Gene Expression Patterns. *Journal of Computational Biology*, 6(3–4):281–297.

Ben-Hur, A., Elisseeff, A., and Guyon, I. (2002). A stability based method for discovering structure in clustered data. In *Pacific Symposium on Biocomputing 2002: Kauai, Hawaii, 3-7 January 2002*, page 6. World Scientific Publishing Company.

Bernardo, J. M. and Smith, A. F. M. (1994). *Bayesian Theory*. Chichester: Wiley.

Bezdek, J. C. (1981). *Pattern recognition with fuzzy objective function algorithms*. Kluwer Academic Publishers Norwell, MA, USA.

Blangiardo, M. and Richardson, S. (2008). A Bayesian calibration model for combining different pre-processing methods in Affymetrix chips. *BMC bioinformatics*, 9(1):512.

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

Booth, J. G., Casella, G., and Hobert, J. P. (2008). Clustering using objective functions and stochastic search. *Journal of the Royal Statistical Society, Series B*, 70(1):119–139.

Bowler, C. and Allen, A. E. (2007). The contribution of genomics to the understanding of algal evolution. In Brodie, J. and Lewis, J., editors, *Unravelling the Algae: The Past, Present, and Future of Algal Systematics*, page 331. CRC Press.

Brettschneider, J., Collin, F., Bolstad, B. M., and Speed, T. P. (2008). Quality assessment for short oligonucleotide microarray data. *Technometrics*, 50(3):241–264.

Büning, H. and Lettmann, T. (1999). *Propositional logic: deduction and algorithms*. Cambridge University Press.

Canfield, E. R. and Pomerance, C. (2002). On the problem of uniqueness for the maximum Stirling number (s) of the second kind. *Electronic Journal of Combinatorial Number Theory*, 2(A01):2.

Castelo, R. (2002). *A Discrete Acyclic Digraph Markov Model in Data Mining*. PhD thesis, Faculteit Wiskunde en Informatica, Univeriteit Utrecht.

Chatfield, C. (2003). *The analysis of time series: an introduction*. Chapman and Hall, 6th edition.

Chiou, J. and Li, P. (2007). Functional clustering and identifying substructures of longitudinal data. *Journal of the Royal Statistical Society, Series B*, 69(4):679–699.

Chipman, H., George, E., and McCulloch, R. (1998). Bayesian CART Model Search. *Journal of the American Statistical Association*, 93:935–960.

Chipman, H., George, E. I., and McCulloch, R. E. (2001). The practical implementation of Bayesian model selection. *Notes Monographic Series (Model Selection)*, 38:67–116.

Chipman, H. A., George, E. I., and McCulloch, R. E. (2002). Bayesian treed models. *Machine Learning*, 48(1–3):299–320.

Cope, L., Irizarry, R., Jaffee, H., Wu, Z., and Speed, T. (2004). A benchmark for Affymetrix GeneChip expression measures. *Bioinformatics*, 20(3):323–331.

Covington, M., Maloof, J., Straume, M., Kay, S., and Harmer, S. (2008). Global transcriptome analysis reveals circadian regulation of key pathways in plant growth and development. *Genome Biology*, 9(8).

Cox, D. R. (1957). Note on grouping. *Journal of the American Statistical Association*, pages 543–547.

Crowley, E. M. (1997). Product Partition Models for Normal Means. *Journal of the American Statistical Association*, 92(437):192–198.

Cussens, J. (2008). Bayesian network learning by compiling to weighted MAX-SAT. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence (UAI 2008)*.

de Jong, H. (2002). Modeling and Simulation of Genetic Regulatory Systems: A Literature Review. *Journal of Computational Biology*, 9(1):67–103.

Dean, T. and Kanazawa, K. (1988). Probabilistic Temporal Reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 524–528.

Denison, D. G. T., Holmes, C. C., Mallick, B. K., and Smith, A. F. M. (2002). *Bayesian Methods for Nonlinear Classification and Regression*. Wiley Series in Probability and Statistics. John Wiley and Sons.

Deshmukh, S. R. and Purohit, S. G. (2007). *Microarray data: statistical analysis using R*. Alpha Science International Ltd.

Dunn, J. C. (1973). A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Cybernetics and Systems*, 3(3):32–57.

Edwards, K. D., Anderson, P. E., Hall, A., Salathia, N. S., Locke, J. C. W., Lynn, J. R., Straume, M., Smith, J. Q., and Millar, A. J. (2006). FLOWERING LOCUS C Mediates Natural Variation in the High-Temperature Response of the *Arabidopsis* Circadian Clock. *Plant Cell*, 18:639–650.

Eisen, M., Spellman, P., Brown, P., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 95(25):14863–14868.

Emmert-Streib, F. and Dehmer, M. (2008). *Analysis of Microarray Data: A Network Based Approach*. Wiley-VCH-Verl.

Fernández, C., Ley, E., and Steel, M. F. J. (2001). Benchmark priors for Bayesian model averaging. *Journal of Econometrics*, 100(2):381–427.

Fisher, W. D. (1958). On grouping for maximum homogeneity. *Journal of the American Statistical Association*, pages 789–798.

Fowlkes, E. B. and Mallows, C. L. (1983). A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, pages 553–569.

Fraley, C. and Raftery, A. E. (1998). How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis. *The Computer Journal*, 41:578–588.

Fraley, C. and Raftery, A. E. (2002). Model-Based Clustering, Discriminant Analysis and Density Estimation. *Journal of the American Statistical Association*, 97:611–631.

French, S. and Britain), O. R. S. G. (1989). *Readings in decision analysis*. Chapman and Hall London.

French, S. and Rios Insua, D. (2000). *Statistical Decision Theory*. Arnold, London.

Gnanadesikan, R. (1989). Discriminant Analysis and Clustering: Panel on Discriminant Analysis, Classification, and Clustering. *Statistical Science*, 4(1):34–69.

Gordon, A. (1999). *Classification*. Chapman and Hall, CRC Press, London, 2nd edition.

Gordon, A. D. (1987). A Review of Hierarchical Classification. *Journal of the Royal Statistical Society, Series A*, 150(2):119–137.

Green, P. J. (1995). Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4):711–732.

Gupta, A. K. and Nagar, D. K. (2000). *Matrix Variate Distributions*. Chapman & Hall/CRC.

Harper, L. H. (1967). Stirling Behavior is Asymptotically Normal. *Annals of Mathematical Statistics*, 38(2):410–414.

Hartigan, J. A. (1975). *Clustering Algorithms*. John Wiley & Sons, Inc., New York.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer-Verlag, New York.

Heard, N. A., Holmes, C. C., and Stephens, D. A. (2006). A Quantitative Study of Gene Regulation Involved in the Immune Response of Anopheline Mosquitoes: An Application of Bayesian Hierarchical Clustering of Curves. *Journal of the American Statistical Association*, 101(473):18–29.

Heras, F., Larrosa, J., and Oliveras, A. (2008). MiniMaxSAT: An efficient weighted Max-SAT solver. *Journal of Artificial Intelligence Research*, 31(1):1–32.

Hoos, H. and Stützle, T. (2005). *Stochastic local search: Foundations and applications*. Morgan Kaufmann.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2(1):193–218.

Hubert, L. J. (1974). Some applications of graph theory to clustering. *Psychometrika*, 39(3):283–309.

Kass, R. and Wasserman, L. (1995). A reference Bayesian test for nested hypotheses and its relationship to the Schwarz criterion. *Journal of the American Statistical Association*, pages 928–934.

Keeney, R. and Raiffa, H. (1976). *Decision with multiple objectives: Preferences and value tradeoffs*. John Wiley & Sons, New York.

Keeney, R. and von Winterfeldt, D. (2007). Practical Value Models. In Edwards, W., Miles, R. F., and von Winterfeldt, D., editors, *Advances in Decision Analysis: From Foundations to Applications*, pages 232–252. Cambridge University Press.

Koller, D. and Lerner, U. (2001). Sampling in factored dynamic systems. In Doucet, A., de Freitas, N., and Gordon, N., editors, *Sequential Monte Carlo Methods in Practice*, pages 445–464. Springer.

Koster, J. T. A. (1996). Markov properties of nonrecursive causal models. *Annals of Statistics*, 24:2148–2177.

Lau, J. W. and Green, P. J. (2007). Bayesian Model-Based Clustering Procedures. *Journal of Computational and Graphical Statistics*, 16(3):526.

Li, H., Luan, Y., Hong, F., and Li, Y. (2002). Statistical methods for analysis of time course gene expression data. *Frontiers in Bioscience*, 7:90–98.

Liverani, S., Anderson, P. E., Edwards, K. D., Millar, A. J., and Smith, J. Q. (2009a). Efficient Utility-based Clustering over High Dimensional Partition Spaces. *Journal of Bayesian Analysis*, 4(3):539–572.

Liverani, S., Cussens, J., and Smith, J. Q. (2009b). Searching a multivariate partition space using weighted MAX-SAT. In *Proceedings of 6th International Meeting of Computational Intelligence Methods for Bioinformatics and Biostatistics (CIBB2009)*.

Lu, X., Zhang, W., Qin, Z., Kwast, K., and Liu, J. (2004). Statistical resynchronization and Bayesian detection of periodically expressed genes. *Nucleic acids research*, 32(2):447.

Luan, Y. and Li, H. (2003). Clustering of time-course gene expression data using a mixed-effects model with B-splines. *Bioinformatics*, 19(4):474–482.

Ma, P., Castillo-Davis, C., Zhong, W., and Liu, J. (2006). A data-driven clustering method for time course gene expression data. *Nucleic Acids Research*, 34(4):1261–1269.

MacQueen, J. B. (1967). Some Methods for classification and Analysis of Multivariate Observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press.

McAllester, D., Selman, B., and Kautz, H. (1997). Evidence for invariants in local search. In *Proceedings of the national conference on artificial intelligence*, pages 321–326.

McCullagh, P. and Yang, J. (2006). Stochastic classification models. In *Proceedings International Congress of Mathematicians*, volume 3, pages 669–686.

McShane, L., Radmacher, M., Freidlin, B., Yu, R., Li, M., and Simon, R. (2002). Methods for assessing reproducibility of clustering patterns observed in analyses of microarray data. *Bioinformatics*, 18(11):1462–1469.

Meilă, M. (2005). Comparing clusterings – an axiomatic view. In *International Conference on Machine Learning*, volume 22, page 577.

Meilă, M. (2007). Comparing clusterings – an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895.

Michael, T., Mockler, T., Breton, G., McEntee, C., Byer, A., Trout, J., Hazen, S., Shen, R., Priest, H., Sullivan, C., Givan, S., Yanovsky, M., Hong, F., Kay, S., and Chory, J. (2008). Network Discovery Pipeline Elucidates Conserved Time-of-Day–Specific cis-Regulatory Modules. *PLoS Genetics*, 4(2):e14.

Monnier, A., Liverani, S., Bouvet, R., Jesson, B., Mosser, J., Corellou, F., Smith, J. Q., and Bouget, F.-Y. (2009). Light-regulated transcriptional networks in Ostreococcus provides insight into the biology and physiology of the marine eukaryotic picophytoplancton. *submitted to Genome Biology*.

Muller, P., Quintana, F., and Rosner, G. (2008). Bayesian clustering with regression. Technical report, Working paper.

Newman, M. E. J. (2003). The Structure and Function of Complex Networks. *SIAM Review*, 45(2):167–256.

O'Hagan, A. and Forster, J. (2004). *Bayesian Inference: Kendall's Advanced Theory of Statistics*. Arnold, 2nd edition.

O'Hagan, A. and Le, H. (1994). Conflicting Information and a Class of Bivariate Heavy-tailed Distributions. In Freeman, P. R. and Smith, A. F. M., editors, *Aspects of Uncertainty*, pages 311–327. Wiley.

Pearl, J. (2000). *Causality: models, reasoning and inference*. Cambridge University Press, Cambridge.

Peay, E. R. (1975). Nonmetric grouping: Clusters and cliques. *Psychometrika*, 40(3):297–313.

Poirier, D. J. (1995). *Intermediate Statistics and Econometrics: a comparative approach*. MIT Press, Cambridge, Mass.

Quintana, F. A. and Iglesias, P. L. (2003). Bayesian Clustering and Product Partition Models. *Journal of the Royal Statistical Society, Series B*, 65(2):557–574.

R Development Core Team (2009). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Ramoni, M. F., Sebastiani, P., and Kohane, I. S. (2002). Cluster Analysis of Gene Expression Dynamics. *Proceedings of the National Academy of Sciences*, 99(14):9121–9126.

Rand, W. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850.

Ray, S. and Mallick, B. (2006). Functional clustering by Bayesian wavelet methods. *Journal of the Royal Statistical Society, Series B*, 68(2):305–332.

Refinetti, R. (2006). *Circadian physiology*. CRC Press.

Richardson, S. and Green, P. (1997). On Bayesian analysis of mixtures with an unknown number of components. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 731–792.

Robertson McClung, C., Salome, P. A., and Michael, T. P. (2002). The Arabidopsis Circadian System. In Somerville, C. R. and Meyerowitz, E. M., editors, *The Arabidopsis Book*. American Society of Plant Biologists, Rockville, MD.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of statistics*, pages 461–464.

Scott, A. J. and Symons, M. J. (1971). Clustering Methods Based on Likelihood Ratio Criteria. *Biometrics*, 27(2):387–397.

Sebastiani, P., Gussoni, E., Kohane, I., and Ramoni, M. (2003). Statistical challenges in functional genomics. *Statistical Science*, pages 33–60.

Sehgal, A. (2004). *Molecular biology of circadian rhythms*. Wiley-IEEE.

Seidel, C. (2008). Introduction to DNA Microarrays. In Emmert-Streib, F. and Dehmer, M., editors, *Analysis of Microarray Data: A Network-Based Approach*, pages 1–26. Wiley-VCH.

Smith, J. Q., Anderson, P. E., and Liverani, S. (2008a). Clustering with Proportional Scaling. Technical Report 04, CRiSM Working Paper, University of Warwick, UK.

Smith, J. Q., Anderson, P. E., and Liverani, S. (2008b). Separation Measures and the Geometry of Bayes Factor Selection for Classification. *Journal of the Royal Statistical Society, Series B*, 70(5):957–980.

Smith, M. and Kohn, R. (1996). Non-parametric Regression using Bayesian Variable Selection. *Journal of Econometrics*, 75:317–343.

Song, J. J., Lee, H. J., Morris, J. S., and Kang, S. (2007). Clustering of time-course gene expression data using functional data analysis. *Computational Biology and Chemistry*, 31(4):265–274.

Stanford, D. and Raftery, A. (2002). Approximate Bayes factors for image segmentation:

The pseudolikelihood information criterion (PLIC). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1517–1520.

Stanley, R. P. (1997). *Enumerative Combinatorics*. Cambridge University Press, Cambridge.

Stone, M. (1979). Comments on model selection criteria of Akaike and Schwarz. *Journal of the Royal Statistical Society, Series B*, 41:276–278.

Straume, M. (2004). DNA microarray time series analysis: automated statistical assessment of circadian rhythms in gene expression patterning. *Methods in Enzymology*, 383:149–66.

Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E., and Golub, T. (1999). Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proceedings of the National Academy of Sciences*, 96(6):2907–2912.

Tatman, J. and Shachter, R. (1990). Dynamic programming and influence diagrams. *Systems, Man and Cybernetics, IEEE Transactions on*, 20(2):365–379.

Thalamuthu, A., Mukhopadhyay, I., Zheng, X., and Tseng, G. (2006). Evaluation and comparison of gene clustering methods in microarray analysis. *Bioinformatics*, 22(19):2405.

Tompkins, D. A. D. and Hoos, H. H. (2005). UBCSAT: An Implementation and Experimentation Environment for SLS Algorithms for SAT and MAX-SAT. In Hoos, H. H. and Mitchell, D. G., editors, *Theory and Applications of Satisfiability Testing: Revised Selected Papers of the Seventh International Conference (SAT 2004, Van-*

couver, BC, Canada, May 10–13, 2004), volume 3542 of *Lecture Notes in Computer Science*, pages 306–320, Berlin, Germany. Springer Verlag.

van Someren, E. P., Wessels, L. F. A., Backer, E., and Reinders, M. J. T. (2002). Genetic network modeling. *Pharmacogenomics*, 3(4):507–525.

Vogl, C., Sanchez-Cabo, F., Stocker, G., Hubbard, S., Wolkenhauer, O., and Trajanoski, Z. (2005). A fully Bayesian model to cluster gene-expression profiles. *Bioinformatics*, 21(2).

Wakefield, J., Zhou, C., and Self, S. (2003). Modelling gene expression over time: curve clustering with informative prior distributions. *Bayesian Statistics*, 7:721–732.

Wang, L., Ramoni, M., and Sebastiani, P. (2006). Clustering short gene expression profiles. *Lecture Notes in Computer Science*, 3909:60.

Weigel, D. and Glazebrook, J. (2002). *Arabidopsis: a laboratory manual*. CSHL Press.

Wu, Y., Tjelmeland, H., and West, M. (2007). Bayesian CART: Prior specification and posterior simulation. *Journal of Computational and Graphical Statistics*, 16(1):44–66.

Yeung, K. Y., Fraley, C., Murua, A., Raftery, A. E., and Ruzzo, W. L. (2001). Model-based clustering and data transformations for gene expression data. *Bioinformatics*, 17(10):977–987.

Zellner, A. (1971). *An Introduction to Bayesian Inference in Econometrics*. Wiley, New York.

Zhang, F. (1999). *Matrix theory: basic results and techniques*. Springer.

Zhou, C., Wakefield, J. C., and Breeden, L. L. (2006). Bayesian Analysis of Cell-Cycle Gene Expression Data. In Do, K.-A., Müller, P., and Vannucci, M., editors, *Bayesian*

Inference for Gene Expression and Proteomics, pages 177–200. Cambridge University Press.