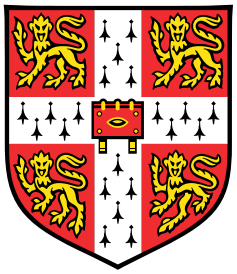
Disease subtyping and biomarker discovery using high-dimensional Bayesian mixture models with feature selection



Emma Prévot

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Master of Philosophy in Machine Learning and Machine Intelligence

Corpus Christi College August 2023

To my parents, who ignited my passion for learning and continuously fueled its flame, your unwavering support and encouragement are my enduring pillars of strength and resilience.

To my sister, for all the remarkable achievements that await you on your academic journey. Embrace every step.

To my best friend Elena, who has been with me through every high and low, standing by my side throughout the year.

Declaration

I, Emma Prévot of Corpus Christi College, being a candidate for the MPhil in Machine Learning and Machine Intelligence, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose.

All code used in this thesis is my own original work. The codebase contains Python code, in the form of Jupyter notebooks, for the implementation of the developed VBVarSel algorithm and all main experiments presented. There are also a few R scripts that contain code to test compared methods developed in R and some data visualisation code. I used standard Python libraries such as Numpy, Pandas, scikit-learn, and standard R packages together with *BiocManager* (Morgan and Ramos, 2023); *MCMCpack* (Martin et al., 2011); *dendextend* (Galili, 2015); *TCGAbiolinks* (Mounir et al., 2019); *PReMiuM* (Liverani et al., 2014); *coda* (Plummer et al., 2006). The code and the mathematical derivations may be found in the following Google drive folder.

All the data used in this thesis is either synthetically generated by myself or publicly available biomedical data. We extract an expression dataset for breast cancer from The Cancer Genome Atlas (TCGA) (School et al., 2012). More information about this dataset and how to access it can be found here. We also apply our algorithm on proteomic pan-cancer data from The Cancer Proteome Atlas (TCPA) (Akbani et al., 2014; Li et al., 2013). More information on the data and how to access it can be found here.

All the results presented were obtained with experiments I implemented, except the performance of the SUGSVarSel algorithm reported through Tables 4.1- 4.4 which was taken from its original publication by Crook et al. (2019b).

The word count is 14,951, excluding declarations, bibliography, photographs, and diagrams, but including tables, footnotes, figure captions, and appendices.

Emma Prévot
August 2023

Acknowledgements

It almost feels surreal to submit this work, as if I'm closing a chapter of my life. This year has been exceptionally challenging for various reasons, and looking back, I can't help but feel a profound sense of pride and accomplishment for having persevered through it all.

I begin by thanking my research supervisors, Dr Paul D. W. Kirk and Dr Filippo Pagani, for the unwavering support and guidance provided. Your invaluable assistance and active involvement throughout the learning process of this research project have been instrumental in shaping the outcome. I am honored to have had the opportunity to work with you.

A special thanks goes out to the MLMI staff for their excellent organisation of the course, especially to John and Anne for being so helpful, supportive, and kind.

Last but not least, I am very grateful for all the friends I made in this journey. We have been through it all together, and each one of you has been a vital part of this transformative chapter of my life.

Abstract

Precision medicine is transforming disease diagnosis, treatment, and prevention. Research in this field is fueled by the increasing availability of vast, high-dimensional biomedical data, with machine learning offering methods to analyse and interpret this complex landscape. One such method is model-based clustering to subtype diseases, which can be paired with feature selection to simultaneously extract informative biomarkers and enhance the quality and interpretability of the stratification. However, many existing approaches rely on computationally demanding inference methods and scale poorly with these expansive datasets, making them impractical unless dimensionality reduction or preprocessing is applied to the data. Though Variational Inference can offer an efficient alternative, it has not gained popularity in the field. Motivated by this gap, we develop a novel method to perform simultaneous clustering and variable selection, harnessing the computational efficiency of Variational Inference, while ensuring accuracy and reliability despite its approximate nature. We demonstrate empirically its superior speed, scalability, and robust performance both in simulated environments and with real biomedical data from The Cancer Genome Atlas. Furthermore, we tackle the *local-optima trap* by introducing annealing in the variational framework. Theoretically, this should help the optimiser navigate multi-modal posterior landscapes, which are common with biomedical data, and our results validate this with a stabilised and improved performance. Overall, this project provides a computationally affordable algorithm capable of rapidly analysing whole biomedical datasets, and introduces a previously unexplored enhancement to the inference process in this context through annealing.

Table of contents

List of figures

List of tables

Nomenclature

Acronyms / Abbreviations

ARI Adjusted Rand Index

ELBO Evidence LOwer-Bound

EM Expectation-Maximisation

GMM Gaussian Mixture Model

MCMC Markov Chain Monte Carlo

PCA Principal Component Analysis

TCGA The Cancer Genome Atlas

TCPA The Cancer Proteome Atlas

VB Variational Bayes

VI Variational Inference

Chapter 1

Introduction

By customising healthcare strategies and therapies to each person's unique characteristics, precision medicine is revolutionising how diseases are treated and prevented. The wealth of high-dimensional data generated in the biomedical field has played a pivotal role in advancing precision medicine, especially in the field of oncology (Cremin et al., 2022). Often, these datasets are used to identify disease subtypes and meaningful features, via clustering algorithms such as model-based clustering, with the goal of enhancing our understanding of disease and improving patient outcomes (Golub et al., 1999; Weinstein et al., 2013). However, this task is fraught with challenges, due to the high-dimensionality and heterogeneity of the data involved (Fop and Murphy, 2018; Kirk et al., 2023; Witten and Tibshirani, 2010).

Frequently, all available variables are used in the modeling process on the assumption that making use of all available information will improve the expected performance of a clustering algorithm (Law et al., 2004). However, in practice, this can be computationally expensive, as well as detrimental to the stratification task due to the inclusion of irrelevant, "noisy" variables (Bouveyron and Brunet-Saumard, 2014; Gnanadesikan et al., 1995; Hastie et al., 2004). Therefore, employing variable selection techniques can aid model fitting, simplify the interpretation of results, and enhance data classification quality (Fop and Murphy, 2018).

Bayesian methods, including Bayesian mixture models, have shown promise in enabling both stratification and feature selection in high-dimensional, unsupervised settings where there are no labels available to guide selection and subtyping (Fop and Murphy, 2018). Many existing algorithms perform inference using Markov Chain Monte-Carlo (MCMC) methods (Bensmail and Meulman, 1998) as, in principle, they can accurately quantify uncertainty and integrate automatic inference of the number of clusters. Nonetheless, MCMC is computationally demanding and scales very poorly in high dimensional settings, which makes its application to real biomedical datasets significantly slow, impractical, and inefficient (Kirk et al., 2023). In contrast, Variational Inference (VI) typically offers more scalable and

efficient inference, even if the results are only approximate (Blei et al., 2017). Nevertheless, VI has not gained popularity in the research field.

Importantly, a commonly overlooked challenge in both MCMC and VI is the *local optima trap*, which makes it very hard, if not impossible to escape a local minimum and find the global optimum in multi-modal landscapes. Therefore, performance heavily relies on the number of local minima of the objective function, the effectiveness of the chosen initial configuration, and the quality of the assumptions on the prior probability distributions. One way in which we can tackle the *local optima trap* and improve inference is via simulated annealing (Katahira et al., 2008; Rose et al., 1990; Ueda and Nakano, 1998). Annealing is based on principles of statistical mechanics. A temperature parameter is introduced in the objective function and eventually varied according to a time-dependent schedule. This effectively smooths the objective function, thereby preventing the optimisation process from becoming trapped in shallow local optima.

In this thesis, motivated by existing challenges, we provide a scalable and computationally efficient algorithm for simultaneous clustering and variable selection, leveraging Variational Inference. Our model is exceptionally faster than other popular methods, making it feasible for very high-dimensional and large datasets, while maintaining accuracy, reliability, and performance. Furthermore, we introduce annealing into our algorithm to enhance inference when dealing with multi-modal posterior distributions. To the author's knowledge, annealing remains largely unexplored in our problem setting.

1.1 Contributions

The main contributions of this thesis are:

- An extensive and systematic overview of model-based clustering, variable selection, and (annealed) Variational Inference, together with a review of similar established methods.
- Detailed derivation and mathematical formalisation of a Variational Bayes algorithm for simultaneous model-based clustering and feature selection, using Gaussian mixture models with diagonal covariance and a binary covariate selection indicator. Importantly, our method uniquely integrates annealing into the variational framework.
- The creation and implementation of a novel algorithm, which we name VBVarSel, as well as auxiliary functions to facilitate data generation, loading, visualisation, and results processing. This also includes extensive and detailed guidance on how to initialise parameters and how to fine-tune them to maximise performance.

- An exhaustive evaluation of our proposed algorithm and comparison against established methods on synthetic data. We showcase its superior efficiency and speed, which is achieved while maintaining accuracy and performance. We also demonstrate the benefits of annealing in this controlled environment.
- A thorough application and evaluation of the algorithm on real biomedical high-dimensional data. The algorithm achieved sensible results, in agreement with existing knowledge and literature, but in a drastically reduced runtime. The introduction of annealing enhanced and stabilised inference. We also provide insights into biomarkers for breast cancer stratification.

1.2 Thesis Plan

The remainder of this report is divided in chapters, which are structured as follows:

- In **Chapter 2** we formalise the problem context and discuss relevant and necessary concepts to understand the theoretical as well as technical background of the project. Additionally, we provide a review of relevant literature.
- In **Chapter 3** we provide an overview of our methodology, focusing on the technical and mathematical details of our proposed model. We tie together the foundational concepts introduced in Chapter 2 into a unique algorithmic framework, which we name VBVarSel, and provide details on parameter initialisation, model selection, and performance evaluation.
- In **Chapter 4** we perform an in-depth investigation of VBVarSel's performance in a comprehensive series of simulations with synthetic data.
- In **Chapter 5** we consider a real-world biomedical application, applying the algorithm to breast cancer transcriptomic data from The Cancer Genome Atlas (TCGA). We execute a wide array of experiments to provide a robust and realistic analysis of the developed algorithm, including challenges and limitations.
- In **Chapter 6** we apply VBVarSel to pan-cancer proteomic data from The Cancer Proteome Atlas (TCPA) to analyse the algorithm performance on a real dataset that is inherently different in terms of dimensionality and correlation.
- In **Chapter 7** we conclude our investigation with a summary of our findings and explore future directions.

Chapter 2

Background

In this chapter, we set the scene by framing our problem setting in the context of precision medicine and motivating its relevance. We explain pivotal concepts for our model, such as model-based clustering and feature selection, and our variational inference process, and try to conceptually tie them together in a cohesive manner. Throughout the chapter, we provide sparse literature on existing approaches.

2.1 Problem setting

2.1.1 What is precision medicine?

Historically, population averages and generalised standards have been used to develop medical diagnoses and treatments, which is what we define as conventional therapy (Figure 2.1). However, given the heterogeneity and variability both in disease manifestations and individuals, conventional standardised treatments can have different outcomes for different patients.

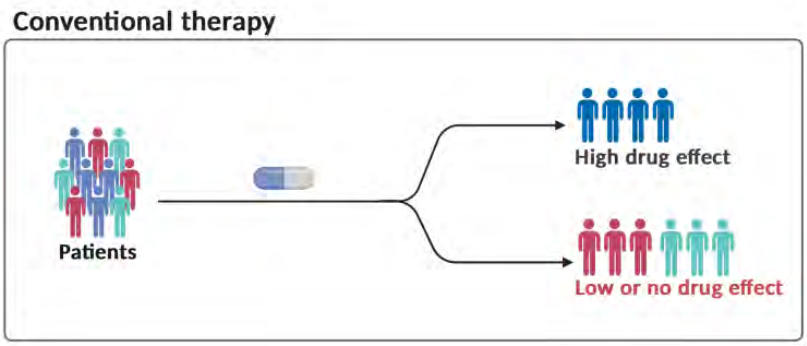


Fig. 2.1 An overview of conventional medicine.

In contrast, precision medicine (Figure 2.2), often referred to as personalised or individualised medicine, customises medical interventions and treatments based on a patient's particular genetic make-up, environment, lifestyle, and other characteristics. The end goal is to deliver the appropriate treatment to the right patient at the right time in order to maximise effectiveness and yield the best outcome for each individual.

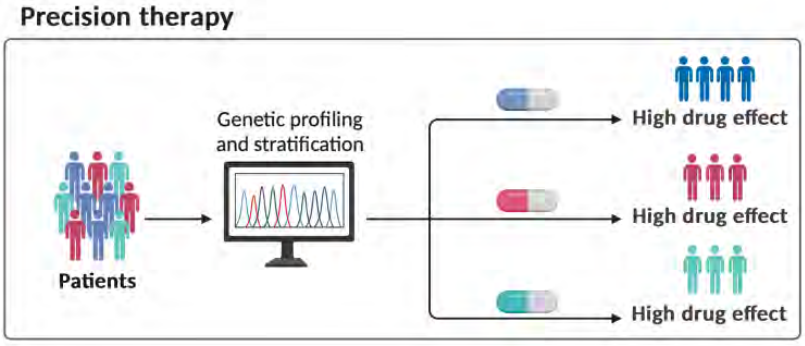


Fig. 2.2 An overview of precision medicine.

The role of Machine Learning

Precision medicine acknowledges and recognises heterogeneity and variability, which most often involves dealing with large amounts of data that is also high-dimensional in nature. To make sense of this vast and complex landscape, advanced Machine Learning (ML) approaches have become essential resources. ML algorithms are able to find complex relationships and patterns in highly dimensional data, facilitating the identification of previously elusive connections, biomarkers, and treatment responses that are difficult to identify using conventional statistical methods. For instance, ML allows us to stratify patients into disease subtypes, and identify biomarkers driving this stratification to ultimately develop different diagnoses and treatments.

2.1.2 Problem formulation

We now provide a conceptual overview of this project, mentioning technical concepts that will be explained later in this chapter. The general aim of our research is to stratify patient populations into subgroups while simultaneously extracting relevant biomarkers that characterise the specific clustering structure. This can provide insights not only into disease heterogeneity but also into the biomarkers that drive the variability. Moreover, given the high-dimensionality of biomedical data, we show that incorporating feature selection can improve computational scalability.

To accomplish this, we will model our data as a finite Gaussian Mixture Model (GMM), making several independence assumptions to streamline the computations. Consequently, the task of stratifying the patients becomes an unsupervised model-based clustering task, which will be executed simultaneously with feature selection. The parameters of our model will be inferred using Variational Inference. We propose annealing as a way to improve inference, specifically the exploration of the multi-modal posterior distribution.

2.2 Unsupervised model-based clustering

After outlining the problem formulation, we begin our model discussion focusing on the subtyping task. We are formulating this through an unsupervised model-based clustering approach, which is an elegant yet accessible way of analysing intricate, high-dimensional data. The clustering task is framed within a modeling context, and data generation is represented as a finite mixture of probability distributions, each of which characterises a distinct cluster (Fraley and Raftery, 2002; Lau and Green, 2007; McNicholas, 2016). Unlike conventional techniques like k-means (MacQueen et al., 1967) or hierarchical clustering (Ward Jr, 1963), this methodology offers a robust statistical framework that permits a probabilistic interpretation of cluster allocations (McNicholas, 2016). This is particularly relevant in biomedical applications as it enhances the interpretability of results.

Importantly, as we take an unsupervised approach, the model is tasked with discovering hidden structures (clusters) within unlabeled data.

2.2.1 Finite mixture model

The key statistical tool underlying model-based clustering is finite mixture models which fall under the umbrella of probabilistic models. Using a notation similar to Bishop (2006), in an unsupervised setting, we have the data $X = \{\mathbf{x}_n\}_{n=1}^{N}$ where $\mathbf{x}_n$ is a J-dimensional vector of random variables, J being the number of features. We define the K-components, or clusters, generative finite mixture models as,

$$p(X|\Phi,\pi) = \prod_{n=1}^{N}\prod_{k=1}^{K}\pi_k f_X(\mathbf{x}_n|\Phi_k) \tag{2.1}$$

Where π_k is the mixing coefficient for the kth component, i.e. the prior probability of the k^{th} cluster being the one that generated observation $\mathbf{x}_n$, and $f_X(\mathbf{x}_n|\Phi_k)$ is the model which serves as the probability density function for each individual cluster or component k. It characterises the statistical properties of the observations within the cluster, with Φ_k denoting

the set of parameters that govern its distribution. The set of $\mathbf{x}_n$ generated from the same model $f_X(\mathbf{x}_n|\Phi_k)$ is called a cluster.

How do we determine K?

A persistent challenge in clustering methods, including mixture models, is determining the optimal number of clusters. While we can often assume to have some knowledge of the number of components in the mixture K, generally speaking, model selection is largely about determining K. One way to address this is by simulating different clustering models with different finite K, eventually in a hierarchical manner, as in Yau and Holmes (2011), and selecting between those via Bayesian Model Selection or Averaging (Claeskens and Hjort, 2008; McLachlan and Rathnayake, 2014; Raftery and Dean, 2006). Another approach is to use overfitted mixtures, where K is large but finite, and throughout the inference process, the mixture weights of the "extra" components are shrunk towards zero (Rousseau and Mengersen, 2011). This is similar to what we implement in our model.

Alternatively, one can fit mixture models with an unknown number of components involving the use of stochastic processes, such as the reversible-jump MCMC (Richardson and Green, 1997; Tadesse et al., 2005) or the continuous-time Markov birth-death processes (Stephens, 2000). These methods allow for the creation and removal of mixture components, providing greater flexibility in the model fitting process. Another Bayesian, non-parametric alternative to finite K involves using Dirichlet process priors on the mixture proportions (Ferguson, 1973; Tadesse et al., 2005), which is equivalent to fitting mixture models with countably infinite components. This approach was first proposed in this context by Kim et al. (2006), and then gained more popularity in works such as Kirk et al. (2023) and Papathomas et al. (2012).

2.3 Feature selection for clustering

Determining the number of underlying components in the data is but one of the challenges that arise in the clustering process. In fact, when for instance model-based clustering is applied to a variety of molecular variables, the task becomes exceedingly complex due to the high-dimensionality and heterogeneity of the data involved (Kirk et al., 2023). Frequently, all available variables are used in the modeling process as in principle, the greater the amount of information available for each data point, the better the expected performance of a clustering algorithm (Law et al., 2004). Nonetheless, in practice, this approach can lead to unnecessary computation. Indeed, some features can amount to little more than noise, and may not carry any valuable clustering information, making them unhelpful or even detrimental to

the clustering process (Hancer et al., 2020; Miao and Niu, 2016). The increasing number of dimensions introduces the curse of dimensionality (Bellman, 1957) and the inclusion of extraneous variables may result in identifiability issues and over-parametrisation (Bouveyron and Brunet-Saumard, 2014; Gnanadesikan et al., 1995; Hastie et al., 2004). Hence, employing variable selection techniques can aid model fitting, simplify the interpretation of results, and enhance data classification quality (Fop and Murphy, 2018).

In our proposed approach, model-based clustering is performed concurrently with the selection of relevant variables. Starting from the mixture model in Equation (2.1), we introduce a latent binary variable $\gamma_j \in \{0,1\}$ indicating whether feature j should be used to infer the clustering structure ($\gamma_j = 1$) or not ($\gamma_j = 0$). We name γ_j as a covariate selection indicator. We make one important assumption which is the independence between covariates, given the cluster allocation. This allow us to factorise the functional form $f(\mathbf{x}_n|\Phi_k)$ as follows:

$$f(\mathbf{x}_n|\Phi_k) = \prod_{j=1}^{J} f_j(x_{nj}|\Phi_{kj}), \tag{2.2}$$

Therefore, we have a univariate functional form $f_j(x_{nj}|\Phi_{kj})$ for each dimension, i.e. covariate. Introducing now the selection indicator γ_j, we write:

$$f(\mathbf{x}_n|\Phi_k, \gamma) = \prod_{j=1}^{J} f_j(x_{nj}|\Phi_{kj})^{\gamma_j} f_j(x_{nj}|\Phi_{0j})^{1-\gamma_j}, \tag{2.3}$$

where Φ_{0j} denotes parameter estimates obtained under the null assumption that there is no clustering structure present in the j^{th} covariate, i.e. the j^{th} feature is not relevant.

Defining variable saliency

The key to performing variable selection is to define the saliency, or importance, of each variable. We can distinguish between relevant and irrelevant covariates. Relevant variables are those that contain essential clustering information. Conversely, irrelevant variables do not convey any useful information to the clustering process, their inclusion does not improve the accuracy of the clusters and it may occasionally even detract from it. The structure of any given cluster will be dependent on relevant covariates, but independent of irrelevant ones (Fop and Murphy, 2018).

2.3.1 Bayesian Wrapper methods

The interaction between the variable selection algorithm and the model fitting process characterises the overall approach to the problem. We can distinguish between filter methods, where the selection is performed as a pre-processing (or occasionally post-processing) step, and wrapper methods, which integrate learning and variable selection concurrently. In the latter, the selection procedure is "enveloped" around the learning algorithm (Fop and Murphy, 2018; Hancer et al., 2020). Filter approaches offer ease of implementation and computational efficiency. Nevertheless, wrapper methods often deliver better outcomes, despite being more complex (Fop and Murphy, 2018). We focus our attention on wrapper methods, as we aim to simultaneously infer the latent clustering structure and relevant covariates.

Within the category of wrapper methods, we can further distinguish 3 main sub-categories, depending on the statistical approach used. We have Bayesian approaches, penalisation approaches, and model-selection approaches, but most existing methods show some overlap between these (Fop and Murphy, 2018). We restrict our focus to Bayesian approaches, which assume the existence of latent variables encoding whether a covariate is relevant or not, and the cluster assignments and inference is made about the posterior distribution of these variables.

Related work

We report key studies pioneering the research around Bayesian wrapper methods, and later enhancing the baseline, and we address the reader to the cited extensive reviews (Celeux et al., 2013; Fop and Murphy, 2018; Hancer et al., 2020; Miao and Niu, 2016; Steinley and Brusco, 2008).

Liu et al. (2003) proposed a "hard" variable selection, the *anchor mode model*, which begins with dimensionality reduction such as Principal Component Analysis (PCA) and retains only the first k_0 factors of the data as relevant variables. The inference on the number of components to retain is carried out via MCMC. While this method shows a good performance, it comes at the risk of losing important information given the dimensionality reduction. The first to introduce the concept of *feature saliency* was Law et al. (2004). Saliency is represented via a binary variable γ_j such that $\gamma_j = 1$ if the j^{th} covariate is relevant, $\gamma_j = 0$ otherwise. The saliency of the j^{th} covariate is expressed as the probability that $\gamma_j = 1$. The authors used Expectation-Maximisation (EM) for maximum a posteriori (MAP) estimation. Constantinopoulos et al. (2006) later extended this work by implementing VI. The idea of a binary covariate selection indicator gained popularity in other studies such as the work of Tadesse et al. (2005), in which posterior samples for the inferred parameters

are taken using Metropolis moves and the reversible-jump MCMC, embedded within a Gibbs sampler. Shortly after, Kim et al. (2006) proposed a similar model that used instead split-merge MCMC and a Dirichlet process mixture model.

More recently, the literature focused on enhancing or expanding baseline Bayesian wrapper methods, as we attempt to do. Swartz et al. (2008) proposed an approach to improve the latent clustering imposing a known substructure within the data, i.e. incorporating prior knowledge on subgroups in the inference process. Kirk et al. (2023) identified and addressed two issues with traditional approaches: the first is that omics datasets often define multiple clustering structures, or views, depending on the subset of variables selected; the second is the task of selecting among these different views. Their proposed implementation is a semi-supervised multi-view Bayesian clustering model which extracts different mixture models during inference and then decides between them using a left-out measurable variable such as survival time. Notably, Crook et al. (2019b) instead focused on making the algorithm more efficient by using a different fast approximate implementation, namely the Sequential Updating and Greedy Search (SUGS) algorithm (Wang and Dunson, 2011; Zhang et al., 2014). This was also attempted by Liverani et al. (2014) for MCMC-based sampling methods.

This encapsulates the essential knowledge required for the present discussion. As we progress further in the subsequent chapter, we will unfold additional layers of complexity and delve into the specifics of our model. We now introduce the core of our methodology: Variational Inference (VI).

2.4 The inference process

In our model implementation, we have introduced different distributions, parametrised by unknown parameters and latent variables. Inference is about estimating those parameters by evaluating their posterior distributions, which is essentially the probability of the parameters taking certain values, given the observed data. However, for complex probabilistic models, evaluating these distributions or computing expectations with respect to them may be infeasible due to high dimensionality or intricate form (Bishop, 2006). Hence, we resort to approximation techniques, which are broadly categorised into two classes: stochastic or deterministic. Stochastic techniques such as MCMC methods operate on the principles of sampling. While these methods can generate exact results given infinite computational time, they tend to be computationally demanding and resource-intensive (Bishop, 2006). This makes them impractical for large-scale computation problems (Kirk et al., 2023). Conversely, deterministic methods like VI rely on analytical approximations but they scale better to large

problems, making them suitable for high-dimensional datasets such as those encountered in molecular medicine (Bishop, 2006; Constantinopoulos et al., 2006).

Despite VI's computational advantages, it remains less popular in the field. Established literature predominantly includes studies using MCMC or EM methods for inference, therefore typically implementing pre-processing or dimensionality reduction for handling large, high-dimensional datasets. Motivated by this gap and our problem's complexity, we choose to operate with VI and focus our efforts on providing a significantly more efficient and faster method, without sacrificing accuracy, reliability, and ease of use.

2.4.1 Variational Inference

Variational Inference, or Variational Bayes (VB), is a family of deterministic approximation techniques aimed at finding an analytical approximation for the posterior distribution. As an indispensable computational tool in our research, VI allows us to bridge the model-based clustering task and feature selection in an efficient and scalable way.

Mean-field variational methods

We work in the Bayesian formalism, in which parameters are assigned prior distributions. Let X denote the set of all observed variables, and θ the set of all parameters and latent variables. Our probability model specifies the joint distribution $p(X,\theta)$, and we seek the posterior distribution $p(\theta|X)$, and also (perhaps) the marginal likelihood, $p(X)$. The main idea behind VI is to approximate the true posterior distribution $p(\theta|X)$ with a simpler distribution $q(\theta)$, usually found by minimising a divergence measure between the two. We note that for any distribution $q(\theta)$, the following equality holds:

$$\ln p(X) = \mathscr{L}(q) + KL(q||p), \tag{2.4}$$

where

$$\mathscr{L}(q) = \int q(\theta) \ln \left(\frac{p(X,\theta)}{q(\theta)} \right) d\theta \tag{2.5}$$

$$KL(q||p) = -\int q(\theta) \ln \left(\frac{p(\theta|X)}{q(\theta)} \right) d\theta \tag{2.6}$$

and $KL(q||p)$ is the Kullback-Leibler (KL) divergence between $q(\theta)$ and $p(\theta|X)$ (Kullback and Leibler, 1951). Since $KL(q||p) \geq 0$, with equality if and only if $q(\theta) = p(\theta|X)$, it fol-

lows that $\mathscr{L}(q)$ is a lower bound for $\ln p(X)$, with equality if and only if $q(\theta) = p(\theta|X)$. We define $\mathscr{L}(q)$ as the Evidence Lower-BOund (ELBO). By maximising $\mathscr{L}(q)$ via optimisation of $q(\theta)$, we minimise the KL divergence between $q(\theta)$ and $p(\theta|X)$.

If we allow any possible form of $q(\theta)$, then the ELBO is maximised when the KL divergence vanishes for $q(\theta) = p(\theta|X)$. However, in VI we work under the assumption that the true posterior distribution is intractable. Hence, we restrict $q(\theta)$ to a family of distributions that yields only tractable solutions, while still being sufficiently rich and flexible to provide a good approximation. We choose the family of distributions that can be factorised as:

$$q(\theta) = \prod_{i=1}^{M} q_i(\theta_i). \tag{2.7}$$

where M is the total number of parameters and latent variables in the model. This factorised approach to VI is known as *mean field theory*, which is an approximation framework originated in physics by Parisi (1979).

Among all potential distributions $q(\theta)$, we seek the one maximising the ELBO, which requires us to conduct a free-form (variational) optimisation of $\mathscr{L}(q)$ with respect to all the distributions $q_i(\theta_i)$. We achieve this through iterative optimisation of individual factors. We rewrite the ELBO equation using this factorised form for q, focusing on the contribution of the factor $q_l(\theta_l)$ to give,

$$\mathscr{L}(q) = \int \prod_i q_i \left(\ln p(X,\theta) - \sum_i \ln q_i \right) d\theta \tag{2.8}$$

$$= \int q_l \ln \tilde{p}(X,\theta_l) d\theta_l - \int q_l \ln q_l d\theta_l + C, \tag{2.9}$$

where q_i denotes $q_i(\theta_i)$, C is a term that does not depend on $q_l(\theta_l)$, and we defined a new distribution $\tilde{p}(X,\theta_l)$ such that

$$\ln \tilde{p}(X,\theta_l) = \mathbb{E}_{i\neq l}[\ln p(X,\theta)] + const. \tag{2.10}$$

Here *const* is a constant that ensures $\tilde{p}(X,\theta_l)$ integrates to 1, and the notation $\mathbb{E}_{i\neq l}[f(X,\theta)]$ is defined as follows: $\mathbb{E}_{i\neq l}[f(X,\theta)] := \int f(X,\theta) \prod_{i\neq l} q_i d\theta_i$. Note that Equation (2.9) is simply the negative KL divergence between $q_l(\theta_l)$ and $\tilde{p}(X,\theta_l)$ (plus a constant). Thus, maximising $\mathscr{L}(q)$ with respect to $q_l(\theta_l)$ is equivalent to minimising the KL divergence between $q_l(\theta_l)$

and $\tilde{p}(X, \theta_l)$, which occurs when $q_l(\theta_l) = \tilde{p}(X, \theta_l)$. Hence the optimal solution, $q_l^*(\theta_l)$, satisfies:

$$\ln q_l^*(\theta_l) = \mathbb{E}_{i \neq l}[\ln p(X, \theta)] + const. \tag{2.11}$$

In words, this says that the natural logarithm of the optimal solution for the factor q_l is obtained by considering the natural logarithm of the joint distribution over all θ and X, and then taking the expectation with respect to all of the other factors q_i. We can write the solution for $q_l^*(\theta_l)$ explicitly as follows:

$$q_l^*(\theta_l) = \frac{\exp\left(\mathbb{E}_{i \neq l}[\ln p(X, \theta)]\right)}{\int \exp\left(\mathbb{E}_{i \neq l}[\ln p(X, \theta)]\right) d\theta_l}. \tag{2.12}$$

The collection of equations provided by Equation (2.12) establish a set of consistency conditions for the maximisation of the lower bound, given the factorisation constraint. Nevertheless, these do not present an explicit solution since each factor l depends on the expectations computed with the other factors $i \neq l$. Within the VI machinery we therefore first initialise all the factors $q_i(\theta_i)$ and sequentially update them according to Equation (2.12) based on the current estimate of all the other factors. This iterative process will converge to a solution, as the ELBO is convex with respect to each factor $q_l(\theta_l)$ (Boyd and Vandenberghe, 2004).

2.4.2 Annealed Variational Inference

As we draw this introductory exploration of VI to a close, we must address one final hurdle - local optima in the posterior density. When VI reaches a local minimum during optimisation, it is very hard, if not impossible to escape and find the global optimum. Therefore, performance heavily relies on the number of local minima within the objective function, the chosen initial configuration, and the quality of the assumptions on the prior probability distributions (Rose et al., 1990; Tadesse et al., 2005). In the realm of clustering and biomarker identification, it is crucial to understand that there is no single best solution. In fact, there are different plausible - maybe even equally optimal and valid - clustering structures depending on the set of features selected (Kirk et al., 2023). Hence, our posterior distribution is very multi-modal, which increases the likelihood of getting trapped in local optima. In an attempt to enhance the exploration of this multi-modal space, we implement annealing. Annealing is an approach based on principles of statistical mechanics and maximum entropy which can help navigate intricate posterior landscapes (Katahira et al., 2008; Rose et al., 1990; Ueda and Nakano, 1998). The idea is to introduce a temperature parameter T into the

ELBO to gradually transition from the prior to the posterior. The annealed version of the ELBO can be written as:

$$\mathscr{L}(q) = \int q(\theta) \ln p(X, \theta) d\theta - T \int q(\theta) \ln q(\theta) d\theta \tag{2.13}$$

When $T = 1$ this reduces to the standard ELBO, which encourages exploitation. When $T > 1$ the KL term, which is the *entropy term*, is scaled up while the log of the joint distribution is penalised. This makes the variational distribution closer to the prior and encourages exploration as the approximation is more dispersed.

Similarly to standard VI, we can derive the solution for $q_l^*(\theta_l)$ explicitly for annealed VI (AVI) as follows:

$$q_l^*(\theta_l) = \frac{\exp\left(\frac{1}{T}\mathbb{E}_{i \neq l}[\ln p(X, \theta)]\right)}{\int \exp\left(\frac{1}{T}\mathbb{E}_{i \neq l}[\ln p(X, \theta)]\right) d\theta_l}. \tag{2.14}$$

To the author's knowledge, no existing literature has provided a comprehensive annealing framework in our problem context. Tadesse et al. (2005) mentions the use of parallel tempering (Earl and Deem, 2005) in their MCMC method but they lack mathematical and empirical details. In contrast, Ruffieux et al. (2020) provides a detailed annealed VI framework, but it is focused on variable selection in regression tasks with numerous predictors and multiple outcomes. Yet, their results show that annealing yields a more robust and stable inference.

Chapter 3

Variational mixture of Gaussians with feature selection

In this chapter, we tie together the elements introduced in the previous chapter into a unique algorithmic framework, which we name VBVarSel, and explain how the model is constructed and operates. Our goal is ambitious, yet clear: building a model to perform clustering and feature selection simultaneously, which can excel in terms of efficiency, complexity, and scalability without sacrificing reliability and accuracy. In the following sections, we discuss our model's core features, its underlying inference process, and practical implementation, including the pseudocode.

3.1 Model definition

The first step in understanding the model is to grasp how the mathematical components interact with each other, providing the framework for simultaneous clustering and variable selection. We commence from the foundational aspects and progressively add up, gradually unfolding the complexities of VBVarSel layer by layer.

3.1.1 Product of univariate - multivariate - Gaussians

In Section 2.2.1 we introduced finite mixture models. Given the data $X = \{\mathbf{x}_n\}_{n=1}^{N}$ where x_n is J-dimensional vector of random variables, we define the K-components generative mixture models as,

$$p(X|\Phi,\pi) = \prod_{n=1}^{N}\prod_{k=1}^{K}\pi_k f_X(\mathbf{x}_n|\Phi_k) \tag{3.1}$$

where $f_X(\mathbf{x}_n|\Phi_k)$ is the functional form for component k, parametrised by Φ_k. In our model, we focus on GMM, which are linear combinations of Gaussian distributions, mathematically presented as follows:

$$p(X|\Phi,\pi) = \prod_{n=1}^{N}\prod_{k=1}^{K}\pi_k \mathcal{N}(\mathbf{x}_n|\mu_k,\Lambda_k^{-1}) \tag{3.2}$$

and

$$\mathcal{N}(\mathbf{x}_n|\mu_k,\Lambda_k^{-1}) = \frac{\sqrt{|\Lambda_k|}}{(2\pi)^{J/2}}\exp\left(-\frac{1}{2}(\mathbf{x}_n-\mu_k)^T\Lambda_k(\mathbf{x}_n-\mu_k)\right) \tag{3.3}$$

We used multivariate Gaussian distributions with parameters $\Phi_k = \{\mu_k,\Lambda_k\}$, respectively mean vector μ_k and precision matrix Λ_k. For each observation $\mathbf{x}_n$ we introduce a latent variable $\mathbf{z}_n$, representing cluster assignment, which is a "1-of-K" binary vector of length K which has precisely one non-zero element (*one-hot* encoding). If $z_{nk} = 1$, then $\mathbf{x}_n$ is associated with the k^{th} component. By conditioning on the latent variable Z, we can decompose the joint distribution as follows:

$$p(X,Z,\pi,\Phi) = p(X|Z,\Phi)p(Z|\pi)p(\pi)p(\Phi) \tag{3.4}$$

where $p(\pi)$ and $p(\Phi)$ are priors on the mixture weights and component-specific parameters (respectively). We can write down the conditional distribution of Z as

$$p(Z|\pi) = \prod_{n=1}^{N}\prod_{k=1}^{K}\pi_k^{z_{nk}}; \tag{3.5}$$

and similarly the conditional distribution of the observed data as

$$p(X|Z,\Phi) = \prod_{n=1}^{N}\prod_{k=1}^{K} f_X(\mathbf{x}_n|\Phi_k)^{z_{nk}}. \tag{3.6}$$

In this formulation, we have implicitly assumed independence between observations and components, which allowed us to factorise in n and k. We make another critical assumption on the independence between covariates j, given the component allocations Z, which allows us to further factorise our functional form as follows,

$$f_X(\mathbf{x}_n|\Phi_k) = \prod_{j=1}^{J} f_j(x_{nj}|\Phi_{kj}) = \prod_{j=1}^{J}\mathcal{N}_j(x_{nj}|\mu_{kj},\tau_{kj}^{-1}) \tag{3.7}$$

Where x_{nj} denotes the j^{th} dimension of $\mathbf{x}_n$, $\Phi_{kj} = \left\{\mu_{kj},\tau_{kj}\right\}$ denotes the parameters associated with the k^{th} mixture component, restricted to the j^{th} covariate. This factorisation is

equivalent to having Λ_k as a diagonal matrix with diagonal entries τ_{kj}. Our functional form is now a univariate Gaussian distribution:

$$f_j(x_{nj}|\Phi_{kj}) = \mathcal{N}_j(x_{nj}|\mu_{kj}, \tau_{kj}^{-1}) = \left(\frac{\tau_{kj}}{2\pi}\right)^{1/2} \exp\left(-\frac{1}{2}\tau_{kj}(x_{nj}-\mu_{kj})^2\right) \tag{3.8}$$

We refer to our mixture model as a product of univariate - multivariate - Gaussians. This conditional independence assumption is fundamental to the functionality of VBVarSel and plays a significant role in its ability to efficiently perform clustering and feature selection simultaneously.

3.1.2 Covariate selection model

In our approach, model-based clustering is performed concurrently with the selection of relevant variables. We stress the fact that this is not a two-step approach but we simultaneously infer each cluster's parameters and allocations as well as the selection indicators. In our formulation, we introduce a latent binary variable $\gamma_j \in \{0,1\}$ indicating whether feature j should be used to infer the clustering structure ($\gamma_j = 1$) or not ($\gamma_j = 0$). We name γ_j as a covariate selection indicator. We extend Equation (3.7) as follows:

$$f(\mathbf{x}_n|\Phi_k, \gamma) = \prod_{j=1}^{J} f_j(x_{nj}|\Phi_{kj})^{\gamma_j} f_j(x_{nj}|\Phi_{0j})^{1-\gamma_j} \tag{3.9}$$

$$= \prod_{j=1}^{J} \mathcal{N}_j(x_{nj}|\mu_{kj}, \tau_{kj}^{-1})^{\gamma_j} \mathcal{N}_j(x_{nj}|\mu_{0j}, \tau_{0j}^{-1})^{1-\gamma_j} \tag{3.10}$$

where Φ_{0j} denotes parameter estimates obtained under the null assumption that there is no clustering structure present in the j^{th} covariate. We pre-compute these estimates before starting the inference procedure by Maximum Likelihood Estimate (MLE) as the mean and the precision of the data. Given the data $X = \{\mathbf{x}_n\}_{n=1}^N$ where $\mathbf{x}_n$ is J-dimensional vector of random variables, for each dimension j we compute:

$$\mu_{0j}^{MLE} = \frac{1}{N}\sum_{n=1}^{N} x_{nj} \tag{3.11}$$

$$\tau_{0j}^{MLE} = \left(\frac{1}{N}\sum_{n=1}^{N}(x_{nj}-\mu_{0j}^{MLE})^2\right)^{-1} \tag{3.12}$$

Given the introduction of the latent variable γ, we update the decomposed joint distribution as follows:

$$p(X,Z,\pi,\Phi,\gamma) = p(X|Z,\Phi,\gamma)p(Z|\pi)p(\pi)p(\Phi)p(\gamma) \tag{3.13}$$

where $p(\gamma)$ is the prior on the covariate selection indicators. We can write the conditional distribution of the observed data in Equation (3.6) as

$$p(X|Z,\Phi,\gamma) = \prod_{n=1}^{N}\prod_{k=1}^{K}\left[\prod_{j=1}^{J} f_j(x_{nj}|\Phi_{kj})^{\gamma_j} f_j(x_{nj}|\Phi_{0j})^{1-\gamma_j}\right]^{z_{nk}} \tag{3.14}$$

where the functional form f_j is given in Equation (3.8). In the next sections, we will define the prior distributions $p(\pi)$, $p(\Phi)$, and $p(\gamma)$.

3.2 Variational framework

We now focus our discussion on how we apply the variational machinery to infer the parameters of our model and obtain cluster allocations, cluster parameters, and covariate selection indicators. Our starting point is the model joint distribution at Equation (3.13).

3.2.1 Prior distributions

The choice of an appropriate prior is a delicate task as it is crucial to strike a balance between incorporating existing knowledge or belief and letting the data "speak" for itself. As we proceed to introduce the priors over the parameters π, $\Phi = \{\mu, \tau\}$, and γ, we strategically choose to work with conjugate prior distributions. The inherent properties of conjugate distributions ensure that the posterior distribution over the parameters is of the same family as the prior. Given that our primary task in VI is to accurately approximate the posterior distribution, knowing already its expected form streamlines calculations.

We take our prior on π to be a symmetric Dirichlet distribution with fixed concentration parameter α_0 for each cluster, not subject to inference.

$$p(\pi) = \mathrm{Dir}(\pi \mid \alpha_0) = C(\alpha_0)\prod_{k=1}^{K}\pi_k^{\alpha_0-1} \tag{3.15}$$

where $C(\alpha)$ is just the normalisation constant. This distribution is a common choice in this context. The α_0 parameter can be interpreted as *pseudocounts*, i.e. the effective prior number of observations associated with each mixture component. Think of it this way: the α_{0k}

parameter for each cluster k is asserting that before observing any data, we already believe that there are around α_{0k} instances of the k^{th} component. Given we do not want to impose a strong preliminary belief of how the proportions should be distributed, we set it to be the same for every component, meaning that all components are equally likely *a priori*. The role of α_0 is also crucial to automatically infer the number of clusters K. By setting $0 < \alpha_0 < 1$, we effectively favor clustering structures in which some of the mixing coefficients are zero, i.e. some clusters are shrunk to zero assignments.

We then proceed to discuss the prior distribution on $\Phi = \{\mu, \tau\}$. Each mixture component k is modeled as a product of independent univariate Gaussian distributions with parameters $\Phi_{kj} = \{\mu_{kj}, \tau_{kj}\}$. We take independent Gaussian-Gamma priors for all μ_{kj}, τ_{kj}, so that:

$$p(\Phi_{kj}) = p(\mu_{kj}, \tau_{kj}) = p(\mu_{kj}|\tau_{kj})p(\tau_{kj}) \tag{3.16}$$

$$= \mathcal{N}(\mu_{kj}|m_{0kj}, (\beta_{0kj}\tau_{kj})^{-1})\Gamma(\tau_{kj}|a_{0kj}, b_{0kj}) \tag{3.17}$$

and,

$$\Gamma(\tau_{kj}|a_{0kj}, b_{0kj}) = \frac{b_{0kj}^{a_{0kj}}}{\Gamma(a_{0kj})}\tau_{kj}^{a_{0kj}-1}\exp(-b_{0kj}\tau_{kj}) \tag{3.18}$$

where Γ is the Gamma distribution. Together these distributions constitute a Gaussian-Gamma conjugate prior distribution and their conjugacy guarantees that the posterior will take the form of a Gaussian-Gamma. This type of prior is a common choice when both parameters of a Gaussian distribution, μ and τ, are unknown (Bishop, 2006) and it comprises the product of a Gaussian distribution for the mean μ, whose precision is proportional to τ, and a Gamma distribution over τ. We have introduced 4 hyperparameters. The mean parameter m_{0kj} influences the center of the corresponding Gaussian distribution in the mixture, while the shrinkage parameter β_{0kj} influences the tightness and spread of the cluster, with smaller shrinkage leading to tighter clusters. The degrees of freedom, a_{0kj}, controls the shape of the Gamma distribution, the higher the degree of freedom, the more peaked (i.e. less dispersed) the Gamma distribution will be. Hence, a_{0kj} directly influences the variability of the clusters and their overlap in the feature space. Finally, the scale parameter b_{0kj} scales the Gamma distribution, the larger b_{0kj}, the broader the range of potential precisions, which influences the spread of the corresponding cluster. We set β_0, a_0 to be equal for every j^{th} dimension and every k^{th} cluster, while we set a m_{0j} and b_{0j} for every j^{th} dimension.

For the covariate selection indicators γ, we introduce another parameter δ, on which we condition to allow conjugacy. Indeed, for each γ_j, we take an independent Bernoulli conditional prior with parameter δ_j, so that:

$$p(\gamma_j|\delta_j) = \delta_j^{\gamma_j}(1-\delta_j)^{1-\gamma_j}, \tag{3.19}$$

The conjugate prior of a Bernoulli distribution is the Beta distribution. Hence, we take independent symmetric Beta priors for δ_j, so that:

$$p(\delta_j) = \text{Beta}(d_0). \tag{3.20}$$

The value of δ_j represents the probability of $\gamma_j = 1$. We use a symmetric Beta distribution with fixed shape parameter d_0, equal across every dimension j. The symmetry around 0.5 implies no prior preference for either $\gamma_j = 1$ or $\gamma_j = 0$. When $d_0 = 1$, the Beta distribution turns into a uniform distribution. For $d_0 < 1$, the Beta distribution is "U-shaped" and δ_j is more likely to take "extreme" values (0 or 1). For $d_0 > 1$ it is instead "bell-shaped", and middle values (≈ 0.5) are preferred.

3.2.2 Variational distribution

Having defined the variational framework in which we operate, we are ready to derive our complete variational distribution. The complete joint distribution of all variables is given by:

$$p(X,Z,\pi,\mu,\tau,\gamma,\delta) = p(X|Z,\mu,\tau,\gamma)p(Z|\pi)p(\pi)p(\mu|\tau)p(\tau)p(\gamma|\delta)p(\delta) \tag{3.21}$$

This decomposition is also shown graphically in Figure 3.1.

For the variational distribution, we obtain the following factorisation between parameters and latent variables:

$$q(Z,\pi,\mu,\tau,\gamma,\delta) = q(Z)q(\pi)\prod_{j=1}^{J} q(\gamma_j|\delta_j)q(\delta_j)\prod_{k=1}^{K} q(\mu_{kj}|\tau_{kj})q(\tau_{kj}) \tag{3.22}$$

Each factor will be updated iteratively as we minimise the KL divergence between the variational distribution and the actual posterior distribution. To derive the update equations, we utilise the foundational formula presented earlier at Equation (2.12). For convenience, we reserve examples of the full derivations to Appendix A. In the main body of this chapter we only present the final update equations implemented within our algorithm.

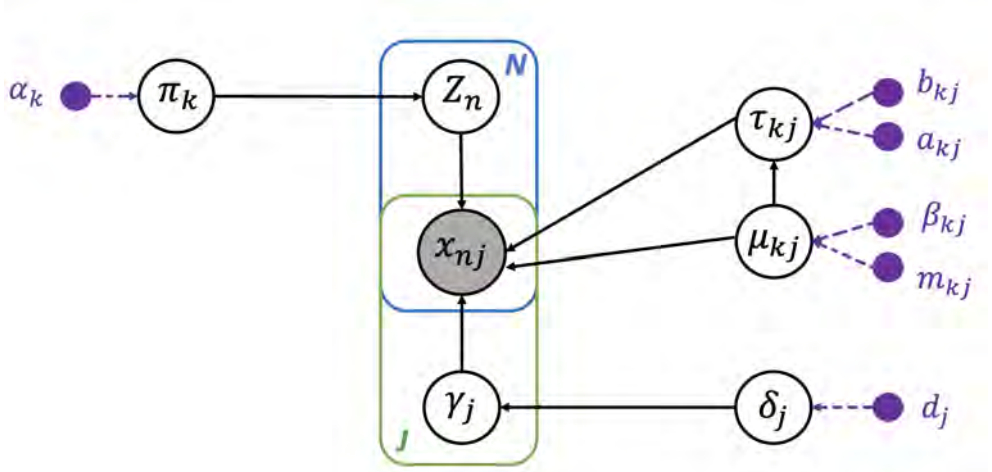


Fig. 3.1 Directed acyclic graph representing the complete model in Equation (3.21). The grey shade corresponds to observed variables. The boxes denote a set of i.i.d. observations and latent variables. The purple dots represent the hyperparameters in the corresponding posterior (or prior) distributions.

Updating Z

Starting with the latent cluster assignments Z, we derive the following:

$$\ln q^*(Z) = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk} \ln \rho_{nk} + const \tag{3.23}$$

where we define

$$\ln \rho_{nk} = \mathbb{E}_{\pi}[\ln \pi_k] + \mathbb{E}_{\Phi,\gamma}[\ln f(\mathbf{x}_n|\Phi_k)] \tag{3.24}$$

Note that:

$$\mathbb{E}_{\Phi,\gamma}[\ln f(\mathbf{x}_n|\Phi_k)] = \mathbb{E}_{\gamma}[\mathbb{E}_{\Phi}[\sum_{j=1}^{J}(\gamma_j \ln f_j(x_{nj}|\Phi_{kj}) + (1-\gamma_j)\ln f_j(x_{nj}|\Phi_{0j}))] \tag{3.25}$$

$$= \sum_{j=1}^{J}\left(c_j \mathbb{E}_{\Phi}[\ln f_j(x_{nj}|\Phi_{kj})] + (1-c_j) f_j(x_{nj}|\Phi_{0j})\right) \tag{3.26}$$

where $c_j = \mathbb{E}_{\gamma}[\gamma_j]$. We introduce r_{nk}, the responsibility of the k^{th} component for the n^{th} observation.

$$r_{nk} = \frac{\rho_{nk}}{\sum_{k=1}^{K}\rho_{nk}} = \mathbb{E}[z_{nk}] \tag{3.27}$$

Further, we make the following definition:

$$N_k = \sum_{n=1}^{N} r_{nk} \tag{3.28}$$

which is the expected number of observations associated with the k^{th} component (note that N_k need not be a whole number).

Updating π

Next, we consider the mixing proportions π. Unsurprisingly, given the conjugate prior on this parameter is a Dirichlet distribution, we recognise $q^*(\pi)$ as an asymmetric Dirichlet distribution with parameter $\alpha = [\alpha_1, \ldots, \alpha_k]$, where

$$\alpha_k = \alpha_0 + N_k \tag{3.29}$$

Recall that N_k is a function of the responsibilities, r_{nk}, as given in Equation (3.28). Hence, the contribution of the covariate selection indicators on π occurs via the responsibilities, r_{nk}.

Updating Φ

We derive the following expression for $q^*(\Phi)$:

$$\ln q^*(\Phi_{kj}) = \sum_{n=1}^{N} r_{nk} c_j \ln f_j(x_{nj}|\Phi_{kj}) + \ln p(\Phi_{kj}) + const \tag{3.30}$$

Note that we weight the contribution of the log-likelihood, $\ln f_j(x_{nj}|\Phi_{kj})$, by the factor $c_j = \mathbb{E}_\gamma[\gamma_j]$. Hence if the j^{th} covariate does not contribute to the clustering structure (i.e. $c_j \approx 0$), then Φ_{kj} will be dominated by the prior.

Given the form of the conjugate prior on Φ_{kj} (Equation (3.16)) and the functional form $f_j(x_{nj}|\Phi_{kj})$, we derive:

$$\begin{aligned} q^*(\Phi_{kj}) =& q^*(\mu_{kj}|\tau_{kj}) q^*(\tau_{kj}) && (3.31) \\ =& \mathcal{N}(\mu_{kj}|m_{kj}, (\beta_{kj}\tau_{kj})^{-1}) \Gamma(\tau_{kj}|a_{kj}, b_{kj}) && (3.32) \end{aligned}$$

We introduced the following parameters (β_{kj}, m_{kj}, a_{kj}, b_{kj}), and statistics ($\bar{x}_{kj}$, S_{kj}) of the observed data, with respect to the responsibilities:

$$\beta_{kj} = c_j \sum_{n=1}^{N} r_{nk} + \beta_0 \tag{3.33}$$

$$m_{kj} = \frac{1}{\beta_{kj}} \left(c_j \sum_{n=1}^{N} r_{nk} x_{nj} + m_{0j} \beta_0 \right) \tag{3.34}$$

$$a_{kj} = \frac{1}{2} c_j \sum_{n=1}^{N} r_{nk} + a_0 \tag{3.35}$$

$$b_{kj} = b_{0j} + \frac{1}{2} \left[c_j N_k S_{kj} + \frac{\beta_0 c_j N_k}{\beta_0 + c_j N_k} \left(\bar{x}_{kj} - m_{0j} \right)^2 \right] \tag{3.36}$$

$$\bar{x}_{kj} = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} x_{nj} \tag{3.37}$$

$$S_{kj} = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} (x_{nj} - \bar{x}_{kj})^2 \tag{3.38}$$

Updating γ_j

For the covariate selection indicator γ_j we unsurprisingly obtain a Bernoulli distribution,

$$q^*(\gamma_j | \delta_j) = c_j^{\gamma_j} (1 - c_j)^{1 - \gamma_j} \tag{3.39}$$

where

$$c_j = \frac{\eta_{1j}}{\eta_{1j} + \eta_{2j}} = \mathbb{E}_\gamma[\gamma_j], \tag{3.40}$$

and

$$\ln \eta_{1j} = \mathbb{E}_{\delta_j}[\ln(\delta_j)] + \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \mathbb{E}_\Phi[\ln f_j(x_{nj} | \Phi_{kj})] \tag{3.41}$$

$$\ln \eta_{2j} = \mathbb{E}_{\delta_j}[\ln(1 - \delta_j)] + \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \ln f_j(x_{nj} | \Phi_{0j})]. \tag{3.42}$$

Updating δ_j

Next, we consider δ_j, for which we obtain an asymmetric Beta distribution:

$$q^*(\delta_j) = \text{Beta}(c_j + d_0, 1 - c_j + d_0). \tag{3.43}$$

Evaluating r_{nk} and c_j

Having derived update equations for the variational distributions, we are left to evaluate r_{nk} and c_j, which are the expected value of the cluster allocations and the covariate selection indicators respectively. Recall that we have:

$$r_{nk} = \frac{\rho_{nk}}{\sum_{k=1}^{K} \rho_{nk}} = \mathbb{E}[z_{nk}] \quad \text{and} \quad c_j = \frac{\eta_{1j}}{\eta_{1j} + \eta_{2j}} = \mathbb{E}_{\gamma}[\gamma_j], \tag{3.44}$$

where to evaluate ρ_{nk}, η_{1j} and η_{2j} as in Equations (3.24), (3.41), and (3.42) respectively, we require expressions for $\mathbb{E}_{\pi}[\ln \pi_k]$, $\mathbb{E}_{\Phi}[\ln f_j(x_{nj}|\Phi_{kj})]$, $\mathbb{E}_{\delta_j}[\ln \delta_j]$, and $\mathbb{E}_{\delta_j}[\ln(1-\delta_j)]$. We can easily write the value for $\mathbb{E}_{\pi}[\ln \pi_k]$ from standard properties of the Dirichlet distribution:

$$\mathbb{E}_{\pi}[\ln \pi_k] = \psi(\alpha_k) - \psi\left(\sum_{k=1}^{K} \alpha_k\right), \tag{3.45}$$

where ψ denotes the digamma function.

We evaluate $\mathbb{E}_{\Phi}[\ln f_j(x_{nj}|\Phi_{kj})]$ as

$$\mathbb{E}_{\Phi}[\ln f_j(x_{nj}|\Phi_{kj})] = -\frac{1}{2}\ln 2\pi + \frac{1}{2}\mathbb{E}_{\tau_{kj}}[\ln \tau_{kj}] - \frac{1}{2}\mathbb{E}_{\mu_{kj},\tau_{kj}}[(x_{nj}-\mu_{kj})^2 \tau_{kj}], \tag{3.46}$$

and

$$\mathbb{E}_{\tau_{kj}}[\ln \tau_{kj}] = \psi(a_{kj}) - \ln b_{kj} \tag{3.47}$$

$$\mathbb{E}_{\mu_{kj},\tau_{kj}}[(x_{nj}-\mu_{kj})^2 \tau_{kj}] = \frac{a_{kj}}{b_{kj}}(x_{nj}-m_{kj})^2 + (\beta_{kj})^{-1}. \tag{3.48}$$

Finally, from standard properties of the Beta distribution we evaluate:

$$\mathbb{E}_{\delta_j}[\ln \delta_j] = \psi(c_j + d_0) - \psi(2d_0 + 1) \tag{3.49}$$

$$\mathbb{E}_{\delta_j}[\ln(1-\delta_j)] = \psi(1 - c_j + d_0) - \psi(2d_0 + 1). \tag{3.50}$$

3.2.3 Inference

The inference process itself, which concerns the optimisation of the variational posterior distribution, can be divided into two steps, much like the EM algorithm. It begins with a *variational E-step*, during which the current distributions and the current estimate of the parameters, are used to evaluate the expected values in Equations (3.45), (3.47), (3.48), (3.49), and (3.50). These are then used to evaluate the current estimate of the cluster assignments

$\mathbb{E}[z_{nk}] = r_{nk}$, and the covariate selection indicators $\mathbb{E}[\gamma_j] = c_j$. Then, in the *variational M-step*, r_{nk} and c_j are kept fixed and used to re-compute an estimate of the posterior variational distributions. The algorithm cycles through E and M steps until convergence is achieved.

Variational lower bound and convergence

In our variational framework, we evaluate the ELBO as[1]:

$$\begin{aligned}\mathscr{L} &= \sum_{\mathbf{Z}} \iiint q(\mathbf{Z},\pi,\mu,\tau,\gamma,\delta) \ln \left\{ \frac{p(\mathbf{X},\mathbf{Z},\pi,\mu,\tau,\gamma,\delta)}{q(\mathbf{Z},\pi,\mu,\tau,\gamma,\delta)} \right\} \mathrm{d}\pi\,\mathrm{d}\mu\,\mathrm{d}\tau\,\mathrm{d}\gamma\,\mathrm{d}\delta \\ &= \mathbb{E}[\ln p(\mathbf{X},\mathbf{Z},\pi,\mu,\tau,\gamma,\delta)] - \mathbb{E}[\ln q(\mathbf{Z},\pi,\mu,\tau,\gamma,\delta)] \\ &= \mathbb{E}[\ln p(\mathbf{X} \mid \mathbf{Z},\mu,\tau,\gamma,\delta)] + \mathbb{E}[\ln p(\mathbf{Z} \mid \pi)] + \mathbb{E}[\ln p(\pi)] + \mathbb{E}[\ln p(\mu,\tau)] + \mathbb{E}[\ln p(\gamma,\delta)] \\ &\quad - \mathbb{E}[\ln q(\mathbf{Z})] - \mathbb{E}[\ln q(\pi)] - \mathbb{E}[\ln q(\mu,\tau)] - \mathbb{E}[\ln q(\gamma,\delta)].\end{aligned} \tag{3.51}$$

The lower-bound has two crucial properties aiding in convergence assessment. First, it monotonically increases at every iteration. Secondly, it converges in a finite number of iterations. Hence, we can use it to check the correctness of our algorithm and to decide when to stop iterating. However, note that a converged ELBO does not guarantee that the variational distribution has converged to the true posterior distribution. Indeed, the algorithm might have simply reached a local optimum in the posterior space.

The topic of convergence in a multi-modal posterior space is paramount for our model and motivates our choice to explore the effects of annealing. The quality of the VI approximation depends on several factors, among which we highlight the flexibility of the variational distribution, the quality of the priors and parameter initialisation. As an optimisation method, it also faces the problem of getting stuck in local optima. Indeed, the optimisation starts from an initial specification of the parameters and latent variables to then iteratively refine those towards the true posterior distribution. However, when dealing with a multi-modal posterior distribution, if the initial specification is too far from the global maximum in the ELBO, and perhaps closer to a local maximum, we may converge there and escape is almost impossible. We aim to address this issue with annealing. The introduction of a temperature parameter "smooths" the posterior landscape, making the optimisation easier to perform.

[1]To keep the notation easier, given the equation is already involved itself, we have omitted the subscripts on the expectation operator. In reality, each expectation is taken with respect to all the variables in its argument.

3.3 Annealed Variational framework

Previous derivations followed the standard variational inference procedure for a general model with latent variables. To introduce annealing in the framework, we proceed as before but start from the annealed foundational formula in Equation (2.13). In most cases, this only yields an additional $1/T$ factor in the parameter update. We report only the equations for the parameters updates that are directly influenced by the temperature parameter. For the latent variables Z and γ, we get:

$$\ln \rho_{nk} = \frac{1}{T}\mathbb{E}_{\pi}[\ln \pi_k] + \frac{1}{T}\mathbb{E}_{\Phi,\gamma}[\ln f(\mathbf{x}_n|\Phi_k)] \tag{3.52}$$

$$\ln \eta_{1j} = \frac{1}{T}\mathbb{E}_{\delta_j}[\ln(\delta_j)] + \frac{1}{T}\sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\mathbb{E}_{\Phi}[\ln f_j(x_{nj}|\Phi_{kj})] \tag{3.53}$$

$$\ln \eta_{2j} = \frac{1}{T}\mathbb{E}_{\delta_j}[\ln(1-\delta_j)] + \frac{1}{T}\sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\ln f_j(x_{nj}|\Phi_{0j})] \tag{3.54}$$

$$\mathbb{E}_{\delta_j}[\ln \delta_j] = \psi\left(\frac{1}{T}(c_j + d_0 + T - 1)\right) - \psi\left(\frac{1}{T}(2d_0 + 2T - 1)\right) \tag{3.55}$$

$$\mathbb{E}_{\delta_j}[\ln(1-\delta_j)] = \psi\left(\frac{1}{T}(T - c_j + d_0)\right) - \psi\left(\frac{1}{T}(2v + 2T - 1)\right) \tag{3.56}$$

which are then used to evaluate r_{nk} and c_j as in Equations (3.44). The annealed posterior distributions over π and Φ are parametrised by:

$$\alpha_k = \frac{1}{T}\left(N_k + \alpha_0 + T - 1\right) \tag{3.57}$$

$$\beta_{kj} = \frac{1}{T}\left[c_j \sum_{n=1}^{N} r_{nk} + \beta_{0j}\right] \tag{3.58}$$

$$m_{kj} = \frac{1}{T\beta_{kj}}\left(c_j \sum_{n=1}^{N} r_{nk}x_{nj} + m_{0j}\beta_{0j}\right) \tag{3.59}$$

$$a_{kj} = \frac{1}{T}\left[\frac{1}{2}c_j \sum_{n=1}^{N} r_{nk} + a_{0j} + T - 1\right] \tag{3.60}$$

$$b_{kj} = \frac{1}{T}b_{0j} + \frac{1}{2T}\left[c_j N_k S_{kj} + \frac{\beta_{0j}c_j N_k}{\beta_{0j} + c_j N_k}\left(\bar{x}_{kj} - m_{0j}\right)^2\right] \tag{3.61}$$

And the annealed posterior distribution of δ becomes:

$$q^*(\delta_j) = \text{Beta}\left(\frac{1}{T}(c_j + d_0 + T - 1), \frac{1}{T}(T - c_j + d_0)\right) \tag{3.62}$$

The annealed variational lower-bound is as in Equation (3.51) but with an additional T factor in front of the negative terms and is also indirectly affected by the updated annealed parameters. Importantly, when $T = 1$, we retrieve the standard (non-annealed) variational inference.

Temperature schedule

One crucial aspect of annealing is the temperature schedule, i.e. how to initialise, set, and eventually vary the temperature throughout inference. There is generally no consensus on the type of schedule to use, and not much literature exploring the benefits of one approach over the other. Hence, everything is determined empirically. We follow Ruffieux et al. (2020) and Kirkpatrick et al. (1983) in the use of a geometric schedule, and Katahira et al. (2008) and Mandt et al. (2016) for the use of a fixed temperature. We also implement a harmonic schedule, which could be preferred for a slower, more gradual decline in temperature.

In the geometric schedule implemented in VBVarSel, the temperature T at every iteration i is evaluated as $T_i = T_0\alpha^i$ where T_0 is the initial temperature, and α is the cooling rate. We evaluate the cooling rate as $\alpha = (1/T_0)^{1/(i_a-1)}$, where i_a is the number of annealed iterations. This ensures that across the optimisation we slowly and consistently reduce the temperature until we retrieve the non-annealed model, i.e. $T = 1$, at the set i_a. In doing so, we implement a "balancing act" between exploration and exploitation. In the early iterations, we keep a relatively high temperature to encourage exploration, during which the algorithm will find various configurations, some more optimal than others. As the iterations progress, we gradually shift from exploration to exploitation, which encourages the algorithm to refine and optimise the best solution it found so far. We implement the same "balancing act" for the harmonic schedule, in which we evaluate $T_i = T_0/(1 + \alpha \cdot i)$, and $\alpha = (T_0 - 1)/i_a$.

Importantly, when using fixed temperature greater than 1, the inference is targeting an *annealed* (approximate) posterior. In contrast, when using either geometric or harmonic schedule, we ultimately retrieve the same (approximate) posterior as the non-annealed inference since we gradually decrease the temperature to $T = 1$. This will be accounted for during empirical comparison.

3.4 The VBVarSel algorithm

Having thoroughly discussed all the intricate aspects of our algorithm, we now encapsulate all these concepts into pseudocode, shown in Algorithm 1. It is important to note that during our implementation process, we addressed several potential numerical instabilities to prevent underflow/overflow and ensure the robustness of our algorithm.

Algorithm 1: The VBVarSel algorithm

Input: Data $X = \{x_n\}_{n=1}^N$, maximum number of clusters K, temperature schedule, initial temperature T_0, maximum iterations itr_{max}, convergence threshold ε

Output: Cluster allocations $Z = \{z_n\}_{n=1}^N$
Variable selection indicators $\gamma = \{\gamma_j\}_{j=1}^J$

Initialise α_0, a_0, b_{0j}, β_0, m_{0j}, δ_0, d_0, C, Z;
Calculate parameter estimates for Φ_{0j} according to Eq. (3.11);

$converged \leftarrow False$;
$i \leftarrow 0$;

while $i < itr_{max}$ **do**
  if *T_schedule is geometric OR harmonic* **then**
    $T \leftarrow$ eval_temp_schedule(T_0, i, i_a)
  else
    $T \leftarrow T_0$
  end
  Update parameters α_k, a_{kj}, b_{kj}, β_{kj}, m_{kj}, $\bar{x}_{kj}$, S_{kj}, and δ_j
  Evaluate Z and γ
  Compute ELBO according to Eq. (3.51)
  $improve \leftarrow \text{ELBO}[i] - \text{ELBO}[i-1]$
  if $i > 0$ *and* $0 < improve < \varepsilon$ **then**
    $converged \leftarrow$ True
    break
  end
  $i \leftarrow i + 1$
end

3.4.1 Parameter initialisation and tuning

Parameter initialisation significantly influences the performance of our model. We tried several initialisation approaches, from random values to more sophisticated strategies. Nonetheless, with 7 parameters to initialise, excluding the annealing temperature and latent variables,

it was impractical to test even a considerable amount of possible combinations. Therefore, we draw on the literature to narrow down the search space, particularly on the studies from Fraley and Raftery (2007) and McLachlan and Rathnayake (2014).

For the Gaussian-Gamma conjugate prior in Equation (3.18), we introduced 4 hyper-parameters, namely the mean m_{0j}, the shrinkage β_0, the degrees of freedom a_0, and the scale b_{0j}. We make the following choices:

- m_{0j}: As in Fraley and Raftery (2007), we set m_{0j} as the mean of j^{th} dimension of the data X.
- β_{0j}: We explore fixed values between 10^{-2} and 10^{-10}.
- a_{0j}: As we are modelling univariate Gaussian distributions, the dimensionality is $XDim = 1$. We follow Fraley and Raftery (2007) setting fixed $a_{0j} = XDim + 2 = 3$, and explore values of the same order of magnitude.
- b_{0j}: this is one of the most important parameters. We explore different specifications, both fixed within the range $[0.01, 2]$, or j-dependent as $var(X_j)/(K^2)$.

For the Bernoulli and Beta distributions in Equations (3.19)-(3.20), we only introduced the shape parameter d_0, for which we explore different values in the range $d_0 = [0.1, 10]$. We initialise the δ latent variables sampling from the symmetric Beta distribution parametrised by d_0. The covariate selection latent variables C are instead initialised either all as 1, 0.5, or sampling from the Bernoulli distribution parametrised by δ.

For the Dirichlet distribution in Equation (3.15), we introduced a concentration parameter α_0. We explore values $0 < \alpha_0 < 1$ to encourage the model to empty "extra" components. We initialise the cluster assignment latent variables Z sampling from a Dirichlet distribution parametrised by α_0.

Finally, for the maximum number of clusters K, we set it as large and finite to simulate an "overfitted" mixture (Rousseau and Mengersen, 2011), and ensure α_0 is small enough to empty the spurious components.

3.4.2 Model selection

After narrowing down the search space for parameter initialisation, we perform model selection to extract a range of configurations that work well for a given dataset. For some parameters, we leverage prior knowledge about the data, if any. For instance, if we know roughly the number of clusters in the dataset, we set K slightly greater but close to that value. Or if we know the proportion of relevant/irrelevant variables, we can adjust d_0 accordingly.

Furthermore, having prior expectations about performance can help us to better assess the impact that parameters have on results.

Within the reduced search space, for each parameter configuration under consideration, we run the VBVarSel algorithm 10-20 times across different data randomisations to mitigate the influence of stochastic elements. We then look at the ELBO shapes, ensuring they are monotonically increasing, and take the average of the convergence ELBOs. Finally, we compare those final averages between different configurations to extract the parameter initialisation effectively maximising the ELBO, which is then employed in our experiments. In Table B.1 we report the parameter initialisation derived from our model selection strategy on the datasets used in an attempt to promote experimental reproducibility.

3.4.3 Evaluating performance

After finding the parameter initialisation maximising the ELBO for a specific experiment, we run VBVarSel 10-20 times on the chosen configuration, across different randomisations of the covariates in the data to disperse the predictors, yield a distribution of scores, and mitigate the influence of stochastic elements inherent in the inference process. We then evaluate performance qualitatively, looking for instance at the scatter plots or heatmaps of the inferred stratification, and quantitatively, taking into account the number of selected and discarded covariates and the adjusted Rand index (ARI) (Hubert and Arabie, 1985; Rand, 1971). The Rand index (RI) is a measure of similarity between two data clustering evaluated as the number of pairs of observations that are either in the same or different clusters in both partitions. The RI is *adjusted* to account for the fact that some agreement can occur by chance. This is done as:

$$\text{ARI} = \frac{\text{RI} - \mathbb{E}[\text{RI}]}{\max(\text{RI}) - \mathbb{E}[\text{RI}]} \tag{3.63}$$

The ARI ranges from -1 to 1, where 1 indicates perfect agreement, 0 indicates random agreement, and < 0 worse than random. For the quantitative evaluation of our experiments, we report median scores, together with upper and lower quartiles, evaluated on those 10-20 runs.

Chapter 4

Simulations on synthetic data

We begin our evaluation of the VBVarSel algorithm using synthetic data to allow comparison and bench-marking against alternative algorithms. Subsequently, we demonstrate the benefits of annealing in different "corrupted" simulations.

4.1 Overview of the compared methods

Before showcasing results, we describe the compared methods. The work of Crook et al. (2019b) focuses on the Sequential Updating and Greedy Search (SUGS) algorithm (Wang and Dunson, 2011; Zhang et al., 2014), which they extend to perform variable selection for clustering (SUGSVarSel), avoiding the use of computationally costly MCMC methods. To automatically infer the number of clusters, they use a Dirichlet process prior on the mixture model. Then, the algorithm proceeds sequentially, greedily allocating the i^{th} observation to a cluster, given the allocations of the previous $i-1$ observations and the previous variable selection structure, and simultaneously greedily updating the covariate selection indicators, given the cluster allocations. Both our method and SUGSVarSel are computationally efficient relative to MCMC methods, but the mechanisms by which they achieve this efficiency are different. SUGSVarSel proceeds in a greedy and sequential manner, while VBVarSel approximates the full posterior distribution in order to make the problem more tractable. In addition to reporting SUGSVarSel performance, we tested the PReMiuM algorithm (Liverani et al., 2014), which is based on MCMC methods and uses a Dirichlet process mixture model to automatically infer the number of clusters. Despite the use of MCMC, the authors claim this algorithm should be relatively efficient. We choose this method as it implements the same model as ours: a mixture of multivariate Gaussians with diagonal covariance. For more details on the underlying model of PReMiuM and SUGSVarSel we address the reader to the cited papers.

Importantly, VBVarSel is a Python algorithm while both PReMiuM and SUGSVarSel are R packages with efficient underlying C++ code, which should give them a speed advantage.

4.2 Crook et al. (2019b) simulation

We recreate the Crook et al. (2019b) simulation of a combination of three Gaussian distributions with mixing ratios of 0.5, 0.3, and 0.2. These are centred at (0, 0, .., 0), (2, 2, ... , 2), and (-2, -2, ... , -2) respectively. The variance-covariance matrix for each is set as the identity matrix. The irrelevant variables are generated following a standard Gaussian distribution. We generate either $n = 100$ or 1000 data points, with a total of 200 variables. In both scenarios, we vary the percentage of relevant variables simulating 5%, 10%, 25%, or 50%.

For VBVarSel we explore different prior specifications following the general guidance in Section 3.4.1 and we report sensible configurations in Table B.1, experiment *synthetic*. Results are showcased in Table 4.1 through Table 4.4, consistent with the format in Crook et al. (2019b) and what we explained in Section 3.4.3. Across all tables, we include median scores, with the upper and lower quartiles across 20 runs, for the runtime, the percentage of both relevant and irrelevant variables that were accurately identified, and the ARI between the inferred clustering and the ground-truth labels from the generated data.

Method	n	Time, secs	Relevant	Irrelevant	ARI
VBVarSel	100	**4.2 [3.3, 4.6]**	1 [1, 1]	1 [1, 1]	1 [1, 1]
SUGSVarSel	100	19.9 [19.7, 20.5]	1 [1, 1]	1 [1, 1]	1 [1, 1]
PReMiuM	100	14.0 [14.0, 14.1]	0.01 [0, 0.02]	0.99 [0.99, 1]	0 [0, 0]
VBVarSel	1000	**7.2 [7.1, 9.6]**	1 [1, 1]	1 [1, 1]	0.77 [0.72, 0.90]
SUGSVarSel	1000	60.8 [59.8, 64.2]	1 [1, 1]	1 [0.99, 1]	0.78 [0.54, 0.92]

Table 4.1 Performance on Crook et al. (2019b) simulation data where **5%** of the variables are relevant.

Together with our VBVarSel algorithm, we report the performance of SUGSVarSel as in Crook et al. (2019b), and the PReMiuM algorithm on the same data. We ran both VBVarSel and PReMiuM until convergence, while the authors ran SUGSVarSel for only 2 iterations. We investigated PReMiuM convergence[1] of the chains for the number of clusters K and the Dirichlet parameter α calculating the Gelman diagnostic using the *coda* package (Brooks and Gelman, 1998; Gelman and Rubin, 1992; Roberts and Smith, 1994), and looking at the chains trace plots, following the guidance from Crook et al. (2019a). However, we underscore that convergence is only "diagnosed" and not guaranteed. For a more detailed discussion,

[1] In Appendix 2 we include visualisations of convergence diagnostic in Figure B.5, B.6, and B.7

we address the readers to the references. Furthermore, PReMiuM variable selection is continuous, i.e. instead of a binary selection, 0 (discarded) or 1 (retained), we can think of it as a probability of selection between 0 and 1. Hence, we take each covariate's selection probability at every iteration, threshold against 0.5 to discriminate between selected and deselected, and take the Monte Carlo average across all iterations. The cluster allocations are instead estimated using dissimilarity matrices (Fritsch and Ickstadt, 2009).

Method	n	Time, secs	Relevant	Irrelevant	ARI
VBVarSel	100	**3.2 [3.1, 3.9]**	1 [1, 1]	1 [1, 1]	1 [1, 1]
SUGSVarSel	100	19.7 [19.5, 19.9]	1 [1, 1]	1 [1, 1]	1 [1, 1]
PReMiuM	100	14.2 [14.0, 14.3]	0.01 [0, 0.04]	0.99 [0.99, 1]	0.01 [0.01, 0.01]
VBVarSel	1000	**6.5 [6.2, 7.9]**	1 [1, 1]	1 [1, 1]	0.83 [0.81, 0.93]
SUGSVarSel	1000	33.3 [33.0, 33.8]	1 [1, 1]	1 [1, 1]	0.90 [0.80, 0.97]

Table 4.2 Performance on Crook et al. (2019b) simulation data where **10%** of the variables are relevant.

Method	n	Time, secs	Relevant	Irrelevant	ARI
VBVarSel	100	**2.9 [2.0, 3.1]**	1 [1, 1]	1 [1, 1]	1 [1, 1]
SUGSVarSel	100	21.9 [21.9, 22.1]	1 [1, 1]	1 [1, 1]	1 [1, 1]
PReMiuM	100	14.1 [13.2, 14.5]	0.05 [0.02, 0.1]	1 [0.99, 1]	0.26 [0.01, 0.53]
VBVarSel	1000	**5.8 [5.1, 7.2]**	1 [1, 1]	1 [1, 1]	0.97 [0.92, 0.98]
SUGSVarSel	1000	31.2 [30.7, 31.8]	1 [1, 1]	1 [1, 1]	0.78 [0.54, 0.92]

Table 4.3 Performance on Crook et al. (2019b) simulation data where **25%** of the variables are relevant.

When considering moderately small datasets (n = 100), VBVarSel and SUGSVarSel are very competitive in both variable selection and clustering accuracy. On the contrary, PReMium struggles with True Positive Rate, i.e. identifying relevant variables, which compromises its clustering capabilities. By plotting the probabilities of selection for each covariate we observed that while there is an increased probability for the relevant ones compared to irrelevant ones, it rarely goes above 0.3. We evaluated the area under the ROC curve (AUC) metric (Bradley, 1997) for PreMiuM experiments to summarise its performance across all possible thresholds for the covariate selection probabilities scores. The median AUC metrics across experiments were 0.78, 0.60, 0.51, and 0.50 for respectively 50%, 25%, 10%, and 5% relevant variables. Hence, PReMiuM variable selection can be better than random (AUC $= 0.5$) at least for higher percentages of relevant variables. A significant advantage of our approach compared to PReMiuM and MCMC methods in general is the ability to use the ELBO for model selection, which allows us to refine prior initialisation and enhance performance.

Method	n	Time, secs	Relevant	Irrelevant	ARI
VBVarSel	100	**1.9 [1.8, 2.5]**	1 [1, 1]	1 [1, 1]	1 [1, 1]
SUGSVarSel	100	24.6 [23.8, 24.9]	1 [1, 1]	1 [1, 1]	1 [1, 1]
PReMiuM	100	19.2 [19.1, 19.7]	0.12 [0.06, 0.16]	1 [0.99, 1]	0.52 [0.52, 0.54]

Table 4.4 Performance on Crook et al. (2019b) simulation data where **50%** of the variables are relevant.

As for runtime, VBVarSel is consistently faster than the two compared methods. Moreover, when increasing the percentage of relevant variables our algorithm converges even faster. While the number of iterations per second remains stable, fewer are needed.

We then proceed to assess the performance on larger simulated datasets ($n = 1000$) of only the scalable methods. VBVarSel always identifies relevant and irrelevant variables perfectly, and the clustering is generally very accurate. Compared to SUGSVarSel, the stratification is more stable and similar in accuracy. Moreover, VBVarSel is more than one order of magnitude faster and, despite the increase in dataset size, the speed is almost not affected.

Overall, we conclude that VBVarSel is generally faster and more scalable than compared methods, reducing the runtime by up to an impressive tenfold despite the inherent computational disadvantage of the programming language used. This is achieved while maintaining high, if not perfect, accuracy in both stratification and variable selection.

4.3 Investigating parameter sensitivity

In Section 3.4.1 we discussed how parameter initialisation significantly influences the performance and accuracy of our model. In this concise section, we aim to highlight the parameters to which VBVarSel is most sensitive and therefore require careful tuning. Our experiments showed that the model is quite robust to the initialisation of β_0, m_{0j}, a_0 and K. On the contrary, the model performance was influenced by the concentration parameter α_0, the shape parameter d_0, and most significantly the scale b_{0j}. Starting with α_0, this parameter strongly affected the ability to "empty" extra clusters. Nonetheless, any value < 0.5 consistently allowed convergence to the true K in this simulated environment. As for d_0, values < 0.5 led to higher deselection rate, and the opposite is true for $d_0 > 5$; in between the performance was stable on perfect selection. Most importantly, VBVarSel requires very careful tuning of b_{0j}. Even slight deviations from optimal would significantly and detrimentally impact the quality of the stratification.

4.4 Evaluating the benefits of annealing

In Section 2.4.2 we presented the theoretical benefits of introducing a temperature parameter in the VI machinery, particularly when dealing with a multi-modal posterior distribution. In this section, we aim to demonstrate the benefits of annealing in overcoming common challenges faced in real-world data scenarios, such as sub-optimal parameter initialisation, correlated data, and noise. Therefore, we simulate these three different scenarios using synthetic data from Crook et al. (2019b). We always generate $n = 100$ observations with 200 variables, of which 20 (10%) are relevant.

We explore different temperature schedules. We begin with $T = 2$, 3 or 4, which either remain constant throughout inference or follow the geometric or harmonic schedule described in Section 3.3. For time-varying schedules, we set 5 or 10 maximum annealed iterations, given we normally converge in less than 15 iterations. We report the performance of the annealing approaches that allowed more significant advantages, and also the non-annealed model ($T = 1$) for reference.

4.4.1 Sub-optimal parameter initialisation

We begin our evaluation of the benefits of annealing on the original synthetic dataset from Crook et al. (2019b), varying parameter initialisations. Table 4.5 shows the results of our experiments. We refer to *optimal* parameter initialisation as the one used in previous simulations, reported in Table B.1, experiment *synthetic*. As for the *sub-optimal* initialisation, we vary the scale b_{0j} since it is the parameter to which VBVarSel is more sensitive. We randomly choose a value for b_{0j} between 0.01 and 1 in each of the 10 randomisations of the data we ran, and we report the median scores with upper and lower quartiles.

From Table 4.5 we observe how even this little change in b_{0j} initialisation affects the performance of the non-annealed VBVarSel. However, introducing annealing generally allowed the optimiser to better explore the posterior space and ultimately reach the global optimum, i.e. perfect performance. Moreover, the annealed optimiser explored different, sensible, clustering configurations, such as a 2-cluster model grouping together 2 of the 3 data clusters. Although this deteriorates the average performance (worse lower quartiles), it can be desirable behaviour when dealing with real-data. In contrast, the deteriorated performance of the non-annealed model is due to "illogical" stratification, where observations are wrongly allocated or additional spurious clusters are generated.

Initialisation	Temperature	Relevant	Irrelevant	ARI
Optimal	$T = 1$	1 [1, 1]	1 [1, 1]	1 [1, 1]
	$T = 3G$	1 [1, 1]	1 [1, 1]	1 [1, 1]
	$T = 2H$	1 [1, 1]	1 [1, 1]	1 [1, 1]
Sub-optimal	$T = 1$	1 [1, 1]	0.98 [0.97, 0.99]	0.84 [0.75, 0.88]
	$T = 3G$	1 [1, 1]	1 [1, 1]	1 [0.70, 1]
	$T = 2H$	1 [1, 1]	1 [1, 1]	1 [0.94, 1]
Optimal	$T = 2$	1 [1, 1]	1 [1, 1]	1 [1, 1]
Sub-optimal	$T = 2$	1 [1, 1]	1 [1, 1]	1 [0.84, 1]

Table 4.5 Annealed VBVarSel performance on Crook et al. (2019b) synthetic data using *optimal* and *sub-optimal* parameter initialisations. G: Geometric, H: Harmonic schedule and the initial temperature is given.

4.4.2 Adding correlation

The data simulated so far is generally easy to work with, there is always a good separation between clusters, the number of groups is relatively small, observations and covariates are uncorrelated, and there is a clear difference between relevant and irrelevant variables. In order to show the benefits of annealing, we make this simulated data more realistic by first introducing correlation. Instead of using identity variance-covariance matrices to generate relevant variables, we introduce different degrees of correlation, i.e. off-diagonal non-zero entries, both fixed and varying among covariates and components. Table 4.6 reports the performance of VBVarSel with fixed and equal covariance across all dimensions in all components. Table 4.7 reports instead the performance when randomly sampling the correlation for each cluster between 0 and 0.5, but fixed across all covariates; Table 4.8 when randomly sampling all covariances. All experiments were run with *optimal* parameter initialisations and results are averaged across 10 independent runs.

Covariance	Temperature	Relevant	Irrelevant	ARI
0.1	$T = 1$	1 [1, 1]	1 [0.99, 1]	0.97 [0.97, 0.97]
	$T = 2H$	1 [1, 1]	1 [1, 1]	1 [0.97, 1]
0.5	$T = 1$	1 [1, 1]	1 [0.99, 1]	0.68 [0.50, 0.71]
	$T = 3G$	1 [1, 1]	1 [1, 1]	0.76 [0.76, 1]
0.1	$T = 2$	1 [1, 1]	1 [1, 1]	1 [1, 1]
0.5	$T = 2$	1 [1, 1]	1 [1, 1]	0.70 [0.70, 0.73]

Table 4.6 Annealed VBVarSel performance on synthetic data modified to include fixed covariance. G: Geometric, H: Harmonic schedule and the initial temperature is given.

Across all varying degrees of introduced covariance, we observe a general improvement with annealing. This enhancement manifests in several aspects, whether it is an improved stratification or variable selection accuracy, or increased stability across experiments. This is even more pronounced when we amplify the randomness and variability in the correlation structure (Table 4.8). Indeed, as the stochasticity in the correlation structure increases, we observe that implementing an effective exploration-exploitation balance with a geometric schedule becomes more beneficial. Notably, the geometric schedule we used is relatively straightforward, thus demonstrating that annealing does not require intensive fine-tuning efforts to show its benefits in a simulated environment.

Temperature	Relevant	Irrelevant	ARI
$T = 1$	1 [1, 1]	1 [0.99, 1]	0.65 [0.65, 0.70]
$T = 2$G	1 [1, 1]	1 [1, 1]	0.74 [0.74, 1]
$T = 2$	1 [1, 1]	1 [1, 1]	0.71 [0.67, 1]

Table 4.7 Annealed VBVarSel performance on synthetic data modified to include randomly sampled covariances for each cluster. G: Geometric schedule and the initial temperature is given.

Temperature	Relevant	Irrelevant	ARI
$T = 1$	1 [1, 1]	0.99 [0.99, 0.99]	0.48 [0.41, 0.54]
$T = 2$G	1 [1, 1]	1 [1, 1]	0.69 [0.69, 0.71]
$T = 2$	1 [1, 1]	1 [1, 1]	0.59 [0.40, 0.71]

Table 4.8 Annealed VBVarSel performance on synthetic data modified to include randomly sampled covariances across all clusters and relevant covariates. G: Geometric schedule and the initial temperature.

Finally, while the ARI may sometime appear similar across annealed and non-annealed simulations in some experiments, the stratifications differ significantly. An example is shown in Figure 4.1. As previously mentioned, annealed models often group together exactly the observations from two clusters or yield reasonable separation given the PCA feature space. In contrast, non-annealed models often misclassify observations or create additional spurious clusters. The latter is a known problem when dealing with correlated data while assuming conditional independence, as we do in our model.

4.4.3 Adding Gaussian noise

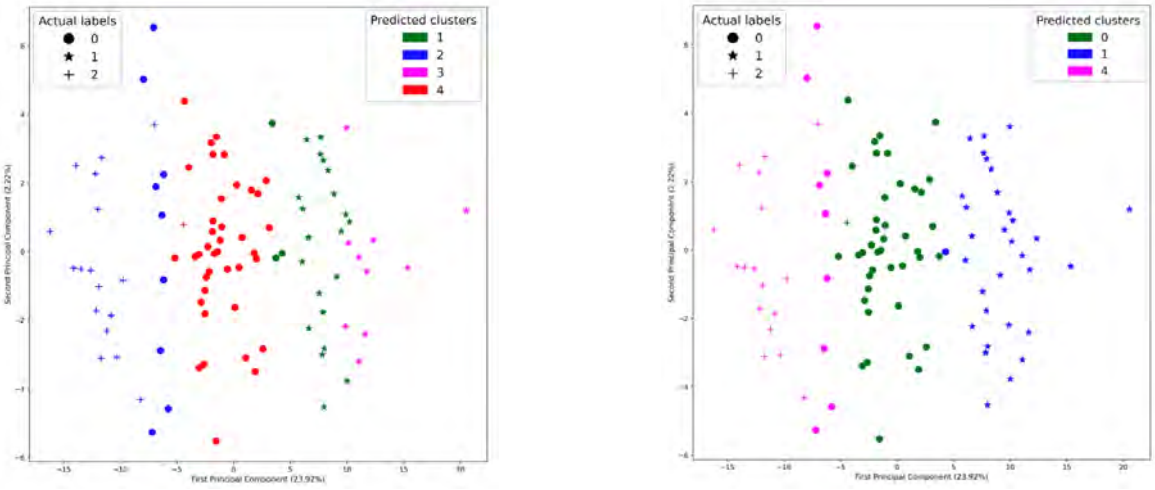
We conclude our investigation by simulating noise contamination, which is very common in real data. Starting from the original Crook et al. (2019b) synthetic dataset, we add Gaussian

noise with zero mean and a varying standard deviation (noise level). Even though in real-world scenarios the noise might not always follow a Gaussian distribution, it is a sensible approximation, providing a good balance between simplicity and realism. All experiments were run with *optimal* parameter initialisation. Results are shown in Table 4.9.

Noise Level	Temperature	Relevant	Irrelevant	ARI
0.1	$T = 1$	1 [1, 1]	0.98 [0.98, 1]	0.89 [0.86, 0.95]
	$T = 3\text{G}$	1 [1, 1]	1 [1, 1]	1 [0.93, 1]
	$T = 3\text{H}$	1 [1, 1]	1 [1, 1]	1 [1, 1]
0.5	$T = 1$	1 [1, 1]	0.98 [0.97, 0.98]	0.90 [0.65, 0.92]
	$T = 2\text{G}$	1 [1, 1]	1 [1, 1]	1 [0.77, 1]
	$T = 2\text{H}$	1 [1, 1]	1 [1, 1]	1 [1, 1]
0.1	$T = 2$	1 [1, 1]	1 [1, 1]	1 [1, 1]
0.5	$T = 4$	1 [1, 1]	1 [1, 1]	1 [0.70, 1]

Table 4.9 Annealed VBVarSel performance on synthetic data modified to include Gaussian noise. We averaged across 10 independent runs. G: Geometric, H: Harmonic schedule and the initial temperature is given.

Across all varying noise levels, although the VBVarSel algorithm is already reasonably robust to noise, introducing even a straightforward temperature schedule yields improved performance and stability, without increasing the computational complexity of the model. Moreover, in the presence of noise, a slower decaying harmonic schedule seems to be preferable over a geometric one to retrieve perfect performance, even if less explorative.



(a) Without annealing

(b) With geometric annealing

Fig. 4.1 PCA scatter plots showcasing VBVarSel stratification on synthetic data modified to include correlations among relevant covariates. We notice the annealed VBVarSel produces a more sensible stratification, even if not perfect.

Chapter 5

Application to TCGA breast cancer data

In Chapter 4 we thoroughly evaluated and benchmarked our VBVarSel algorithm in a simulated environment. As we move into this new chapter, we transition from synthetic data to a real-world medical application, applying the algorithm on breast cancer transcriptomic data from The Cancer Genome Atlas (TCGA). Over the course of our evaluation, we implemented an extensive number of experiments, pushing the boundaries of our model to yield a robust and realistic analysis. Before diving into the results, we introduce established breast cancer subtypes and molecular traits, which is the prior knowledge grounding our expectations on the algorithm's performance.

5.1 Primer on breast cancer subtypes

Breast cancer is the most common type of cancer in women worldwide, with about 1 in 7 being diagnosed with it in their lifetime (NHS, 2022). Given the complexity and heterogeneity of breast cancer molecular traits and disease manifestations, a conventional medicine approach to treatment is suboptimal. Hence, extensive research has been dedicated towards characterising cancer subtypes, each with its own molecular and pathological characteristics. There is evidence for clustering models ranging from two subtypes (Duan et al., 2013) to ten subtypes (Curtis et al., 2012), and anything in between (Akbani et al., 2014; Lock and Dunson, 2013; Prat et al., 2010; Sørlie et al., 2003; Weinstein et al., 2013).

The most widely established stratification, but still continuously refined, is composed of five subtypes, which have traditionally been classified based on the expression of three receptor proteins: estrogen receptor (ER), progesterone receptor (PR), and human epidermal growth factor receptor 2 (HER2) (School et al., 2012). We can stratify patients into Luminal A, Luminal B, HER2-enriched, Basal-like, and Normal-like subtypes. For more information

on the characteristics of each of these groups, we address the reader to Orrantia-Borunda et al. (2022), Prat et al. (2010), and the literature cited before. Most of the existing stratification approaches leverage gene expression profiles from defined biomarker gene sets, of which we focus on the PAM50 set (Parker et al., 2009; Sørlie et al., 2003), where PAM stands for "Prediction Analysis of Microarray". This gene set is a 50-genes signature that has been validated in numerous studies for both subtype identification and risk prediction.

5.2 Introduction to The Cancer Genome Atlas (TCGA)

We extract the breast cancer transcriptomic data from The Cancer Genome Atlas (TCGA). TCGA is a landmark cancer genomics program, which molecularly characterised over 20,000 primary cancers to yield over 2.5 petabytes of genomic, epigenomic, transcriptomic, and proteomic publicly available data (Weinstein et al., 2013).

As it is, the extracted transcriptomic dataset contains 348 breast cancer tumour samples, each with 17814 gene expressions, i.e. covariates. After removing the 441 genes with NaN entries, we were left with a 348 x 17373 dataset. Only 48 of the 50 PAM50 genes are in the dataset. We retrieve demographic and clinical characteristics for each observation from the Supplementary Table 1 of School et al. (2012), including *ground-truth* labels for cluster assignments. These are the 5 breast cancer subtypes, distributed as shown in Figure B.3.

5.3 Unsupervised model-based clustering on PAM50 genes

We begin the appraisal of our model performance on real data focusing on the clustering capabilities, while temporarily neglecting variable selection. To do so, we extract only the PAM50 genes from the full dataset, which should constitute only relevant information, we fix the covariate selection indicators at 1, and disable inference on those.

Experimental set-up

After the thorough investigation described in Section 3.4.1, we report the parameter initialisations maximising the ELBO in Table B.1, experiment *C-TCGA*. We normalised the data to unit variance and zero mean, which was found to improve performance and speed up convergence. We report results as described in Section 3.4.3, and we evaluate the ARI between the inferred stratification and the *ground-truth* cancer subtype of each observation.

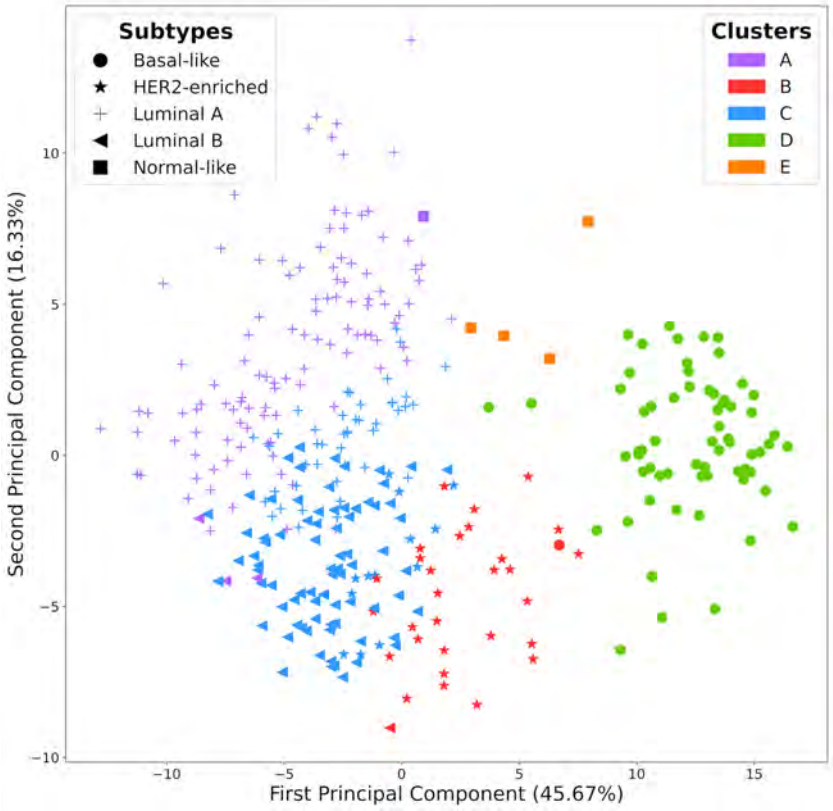


Fig. 5.1 PCA scatter plot of the PAM50 genetic expression of the 348 TCGA samples. The different colours indicate VBVarSel stratification, obtained without variable selection.

Results

When using a scale $b_{0j} = 1$, we obtained 5 clusters that reasonably resemble the breast cancer subtypes (ARI ≈ 0.54), shown in Figure 5.1. Cluster A is associated with Luminal A samples, while Cluster C is mostly associated with Luminal B samples. Cluster B contains only HER2-enriched samples, but it gathers those that are most "distant" in feature space from Luminal B. Cluster D perfectly represents Basal-like samples, and Cluster E seems to have identified the Normal-like samples despite very few occurrences. The overlap between clusters in feature space, which is due to an existing similarity in some genetic expressions, presents the most significant challenge to our model's accuracy. However, the results obtained are aligned with established literature (Crook et al., 2019b; School et al., 2012).

When instead initialising lower b_{0j} and α_0, the algorithm converged to a 2-clusters model, one containing only basal-like samples, and the other grouping together the remaining types. This stratification was indeed maximising the ELBO. Looking at Figure 5.1, this is a very sensible separation, also backed-up by the understanding we have of breast-cancer and relevant literature (Crook et al., 2019b; Dinalankara et al., 2018).

It is worth mentioning that when setting $K > 5$, the algorithm would automatically "empty" additional clusters, or assign very few observations.

5.4 Simultaneous stratification and biomarker selection

Having established that our model's clustering performance aligns with the current literature and expected outcomes, we proceed to enable variable selection. We maintain the PAM50 genes in our dataset, but also progressively add covariates randomly selected from the full dataset. Throughout this section, we detail the simulations conducted to rigorously investigate our model's performance, verify its underlying assumptions, and identify potential limitations.

Experimental set-up

The dataset used always includes the 48 PAM50 genes, to which we add progressively 50, 100, 500, and 1000 genes to simulate scenarios where the number of covariates is either lower, comparable or larger than the number of observations. We then evaluate the performance of the VBVarSel on the full TCGA Breast cancer dataset, and on a pre-processed version of it, as in Crook et al. (2019b) and Lock and Dunson (2013), to allow for comparison. In all the simulations, we simultaneously infer cluster assignment and variable selection. The parameters are initialised as reported in Table B.1, experiment *TCGA*, and data is normalised to zero mean and unit variance.

5.4.1 Evaluating variable selection on the PAM50 genes

The results obtained applying the VBVarSel algorithm to only the set of 48 PAM50 genes are aligned with expectations (first row of Table 5.1). Indeed we always retain at least 39 covariates (lower quartile) (*Binomial test*, $p < 0.0001$). We examined deselected genes to look for recurring patterns and we found that the gene "MMP11" was deselected in 9 out of 10 runs (*Binomial test*, $p \approx 0.01$), and genes "MDM2" and "FGFR4" were discarded in 6 and 5 out of 10 runs respectively. To further investigate this, we analysed the correlation between the frequently deselected genes and the remaining ones using the correlation matrix in Figure 5.2. As expected, we observe a difference in the correlation levels of these frequently deselected genes as these are neither positively nor negatively correlated with the other PAM50 genes, especially gene "MMP11".

As for the clustering task, the inferred stratification achieved a median ARI ≈ 0.47, which is comparable to what was obtained without variable selection. This confirms that we effectively removed covariates that were only introducing more noise rather than signal.

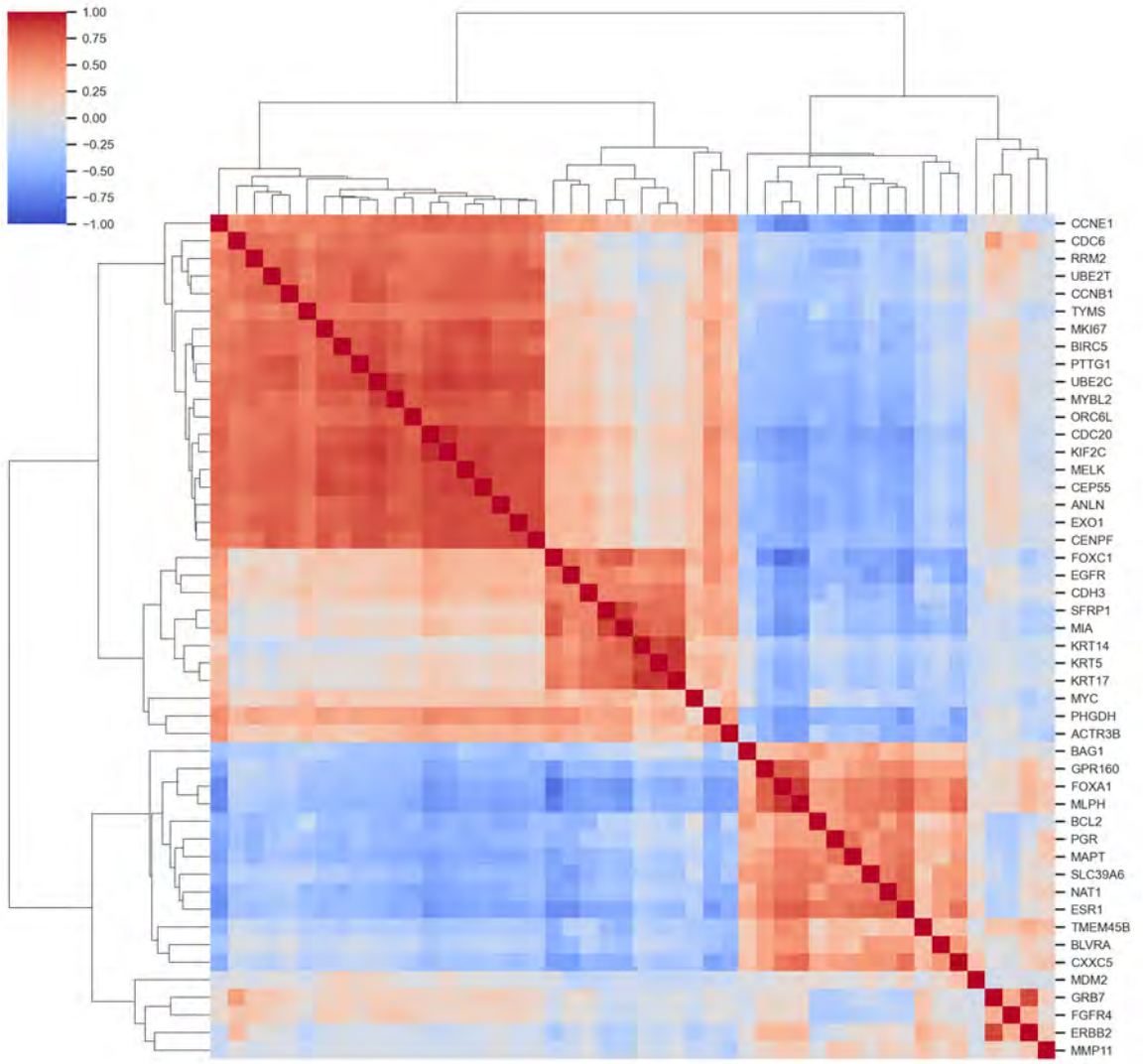


Fig. 5.2 A correlation heatmap between PAM50 genes with hierarchical clustering.

In Figure 5.3 we show a heatmap of the data with the clustering produced by VBVarSel using only the PAM50 set, with variable selection. While it is uncommon for a cancer subtype to be linked to just one cluster, there are clear correlations between gene expressions, subtypes, and clusters. Moreover, the clustering structure found by VBVarSel seems more aligned with the genetic expression than the actual subtypes.

We tested the covariate selection's reliability by permuting the rows of either the last 5, 10, or 25 columns (i.e. covariates). The aim is to disrupt or "break" the existing clustering information in the permuted genes, making them irrelevant for stratification, and observe if the model discards them. Indeed, in all three sets of experiments, the permuted covariates were entirely deselected while at least 87% of the "unperturbed" PAM50 genes were retained.

We then tested whether the order of relevant/irrelevant covariates influences the selection. Theoretically, this should not happen as there is no dependence between covariates in the inference process, i.e. the VBVarSel update equations never show dependency of covariate j on $j-1$, $j+1$, or any other. Hence, we permute the rows of 5, 10, or 25 columns, randomly chosen in the dataset. As expected, the permuted genes were discarded in all experiments, no matter their location in the dataset, and at least 90% of the "unperturbed" genes were retained.

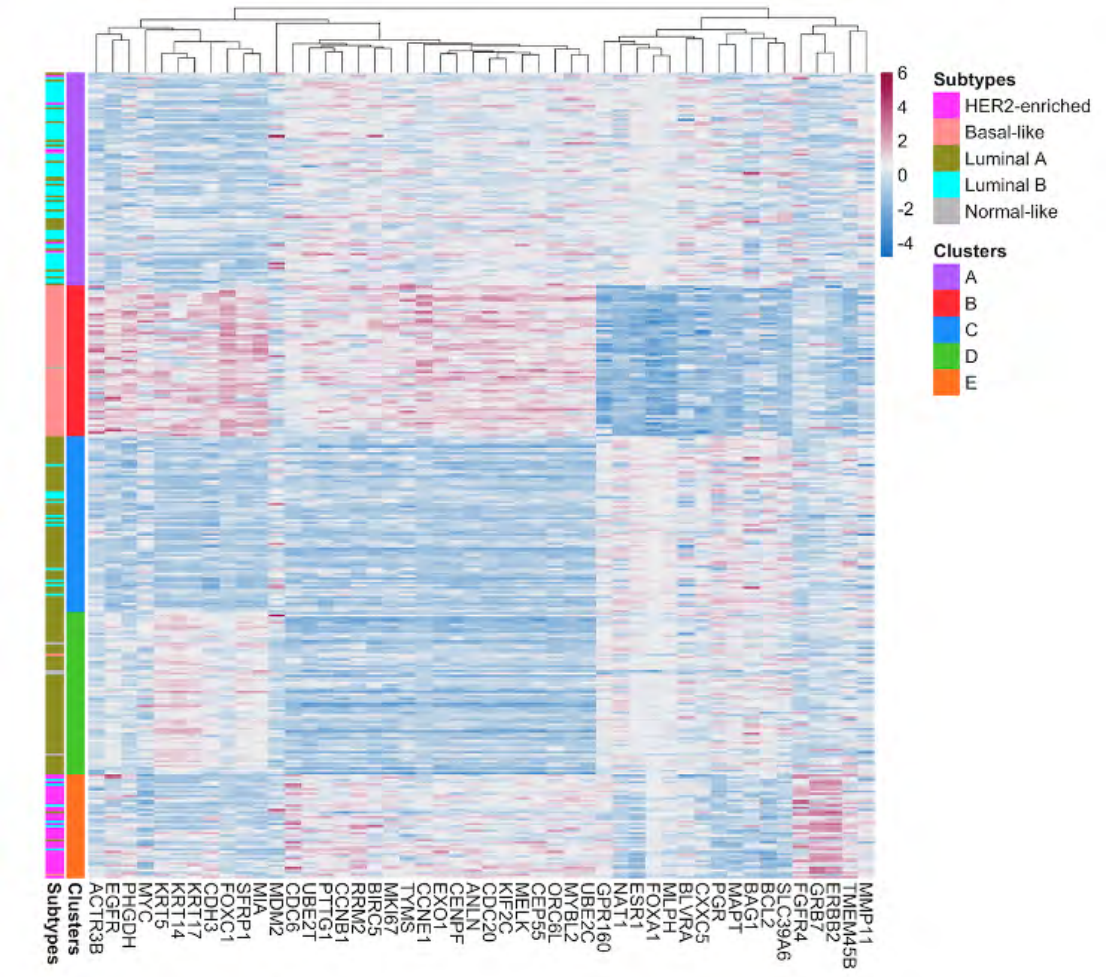


Fig. 5.3 A heatmap of the normalised PAM50 genetic expressions of each observation. The annotation bars indicate the different cancer subtypes and VBVarSel clusters.

In summing up this section, we mention a few key insights from our experiments. Primarily, the variable selection machinery is correctly discriminating informative covariates from uninformative ones, independent of their locations and proximity, to retrieve a sensible stratification. Additionally, we found a significant amount of correlation exists between PAM50 genes, which seems tied to the amount of clustering information they convey.

5.4.2 Adding randomly sampled genes

So far, we worked with only relevant variables and used permutations to artificially generate irrelevant ones. We now progressively integrate an increasing number of additional covariates which are randomly sampled from the complete TCGA breast cancer dataset. We do not expect to discard all the randomly sampled genes, as there is no theoretical basis to conclude they are irrelevant to the clustering task. However, it is reasonable to assume that the majority of these genes might not contribute to the stratification. Conversely, we expect to retain the majority of the PAM50 genes.

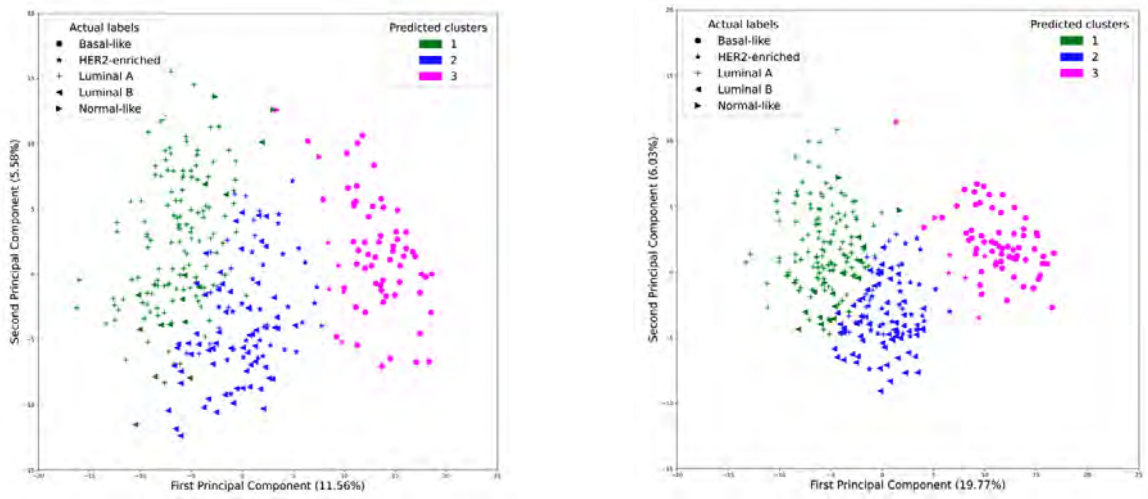
Table 5.1 shows the resulting averaged performance of the VBVarSel algorithm for varying numbers of p additional randomly sampled genes. The first row reports the results obtained with PAM50 genes only. In the subsequent rows, we increased the dataset size by either $p = 50$, 100, 500, or 1000 randomly sampled genes. We report the time, in seconds per iteration, the number of retained PAM50 genes, together with the total number of relevant and irrelevant variables, and the ARI between the inferred stratification and the *ground-truth* cancer subtype of each observation.

p	sec/itr	Relevant PAM50	Relevant	Irrelevant	ARI
0	0.12	42 [39, 44]	44 [43, 45]	4 [3, 5]	0.47 [0.44, 0.51]
50	0.20	41 [39, 45]	58 [54, 63]	40 [36, 44]	0.48 [0.45, 0.50]
100	0.29	42 [40, 44]	78 [72, 83]	70 [66, 78]	0.49 [0.43, 0.51]
500	0.91	42 [41, 45]	163 [158, 168]	385 [380, 390]	0.48 [0.42, 0.48]
1000	2.04	41 [40, 44]	371 [360, 377]	629 [623, 640]	0.47 [0.43, 0.49]

Table 5.1 VBVarSel performance on varying subsets of TCGA data. We present the median scores with the upper and lower quartiles across 10 independent runs on different data randomisation.

The results are promising. VBVarSel is able to retain at least 39 PAM50 genes across all the different experiments (lower quartile), independent of the number of randomly sampled genes. Statistically, the retention of PAM50 genes is significantly better than random across all experiments (*Fisher Test*, $p < 0.00001$). Moreover, the percentage of randomly sampled genes that are retained by the model is fairly constant. As for the stratification quality, we do not observe significant variability in ARI between the different dataset sizes which implies the model is successfully discarding noisy variables to retrieve sensible cluster allocations. However, we observed more variability in the number of clusters of the obtained stratification as the model would often group together two or more subtypes. The performance maximising the ELBO was achieved by a two-clusters model, separating basal subtype samples from non-basal samples, as in Crook et al. (2019b).

In Figure 5.4 we visualise the scatter plots of the stratification obtained with 500 additional genes. We observe how the reduced variable set produces smaller and tighter clusters and increases the separation between them. Indeed, cluster 3 which is strongly associated with basal samples is more separated and there is a neater boundary between clusters 1 and 2.



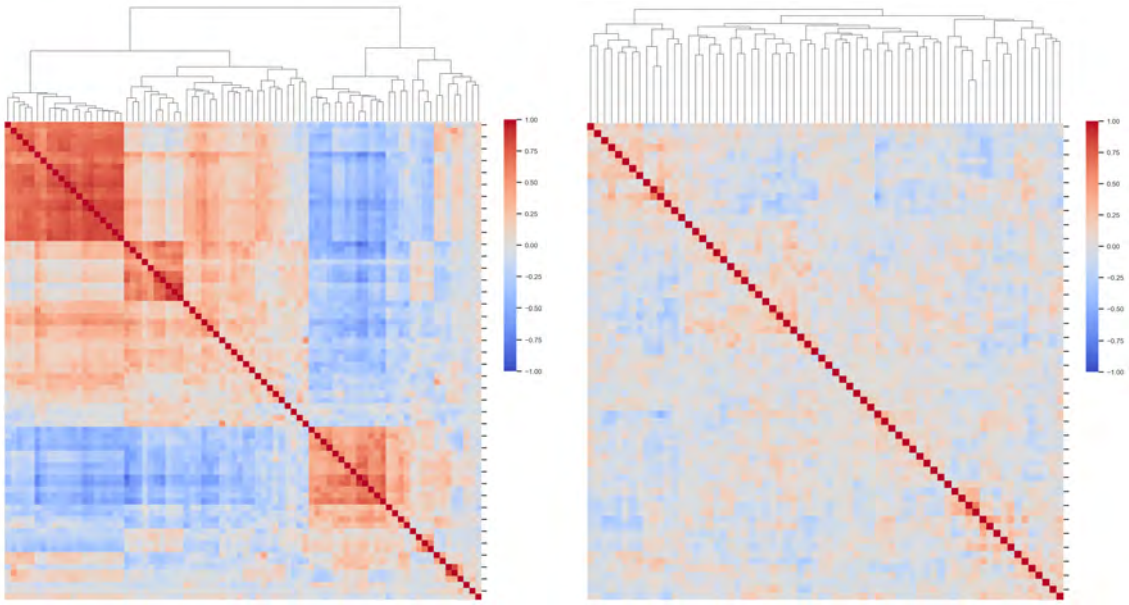
(a) On all covariates (b) On only selected covariates

Fig. 5.4 Scatter plots of VBVarSel stratification on the 348 TCGA samples when PCA is applied to either all PAM50 plus 500 genes, or only the selected ones.

As before, we investigate the differences in the correlation structure of selected/deselected covariates. To allow for better visualisations, we focus on the dataset with 100 additional covariates. Figure 5.5 shows the correlation matrices between the selected genes and the deselected genes. The discarded genes are almost uncorrelated while we observe significant positive and negative correlations amongst the retained ones, as expected. Moreover, while there is a clustering structure within the selected covariates, which is also reflected in the dendrogram, there seems to be only noise in the deselected covariates. This confirms our claim that the correlation structure in the genes strongly drives stratification, and most importantly that VBVarSel is correctly using it to discard uninformative covariates.

If we permute the rows of the "random" genes retained by the model, those are discarded as expected, independent of the dataset size. This implies that there was indeed some clustering information encoded in those covariates. Moreover, we also observed a significant overlap in the PAM50 genes discarded by the model across all experiments.

In summary, our experiments revealed that VBVarSel scales well with increasing number of covariates and the performance stays approximately constant amidst noisy variables.



(a) Selected covariates (b) Deselected covariates

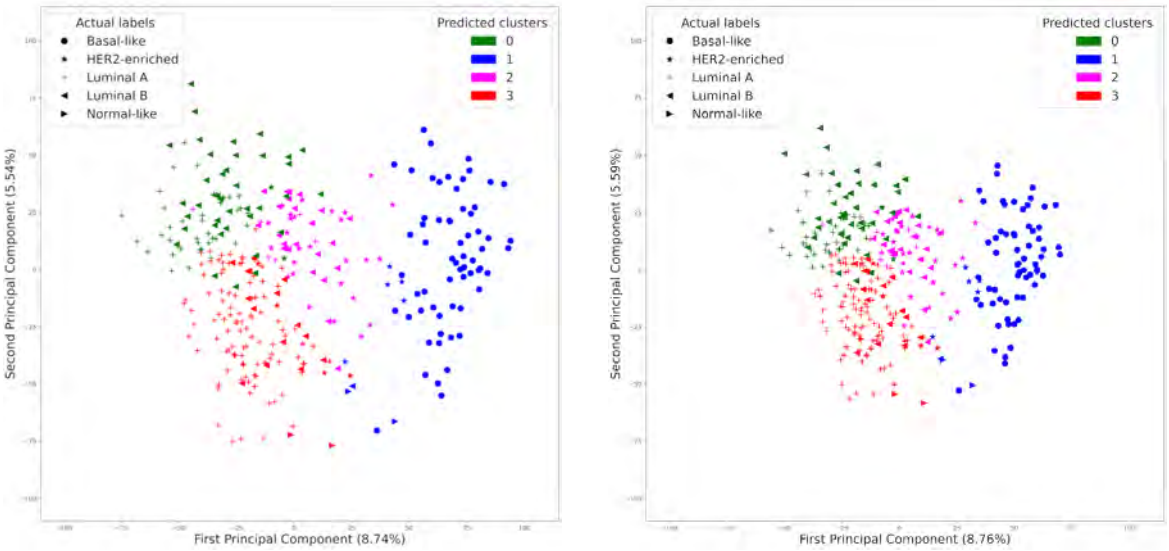
Fig. 5.5 Correlation heatmaps and hierarchical clustering on the covariates VBVarSel selects vs. deselects when applied to the PAM50 plus 100 genes. The stratification obtained is shown in Figure B.1.

We confirmed the link between correlation structure and clustering information conveyed. VBVarSel also kept a very manageable runtime across all dataset sizes.

5.4.3 Experimental validation on complete dataset

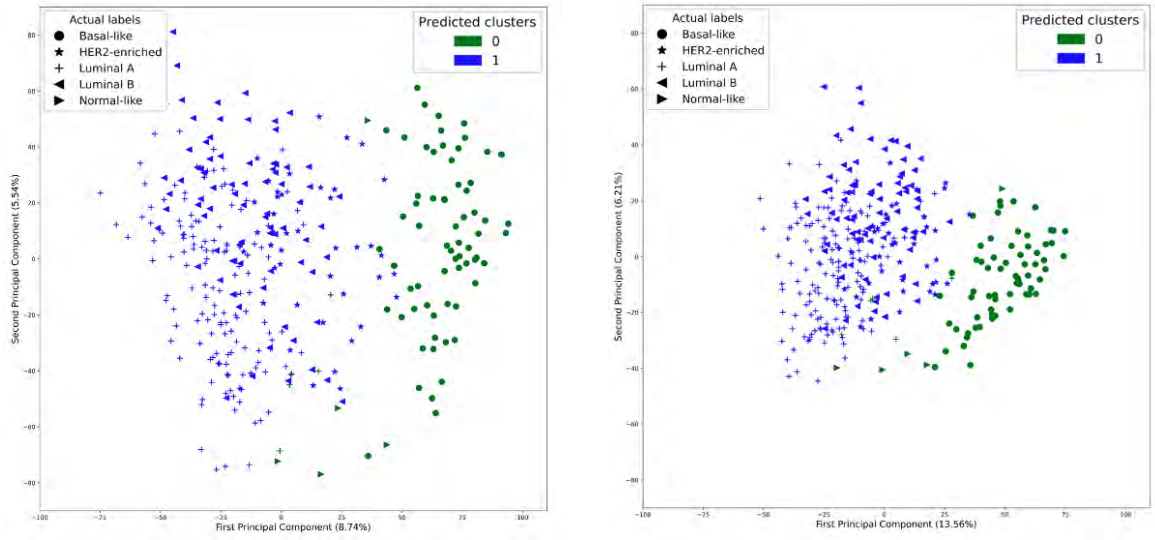
Finally, we evaluate the VBVarSel algorithm on the full 348 x 17373 transcriptomic breast cancer dataset. The magnitude of this dataset underscores one of the key and unique strengths of our algorithm which is its ability to effectively handle high-dimensional datasets. Existing methodologies most often require preprocessing of different extents due to scalability issues, of which we will provide an example in the next section. In terms of runtime, while exact time is influenced by the specific hardware, the algorithm was always able to converge in less than 1 hour, which is a very feasible runtime.

Remarkably, we do not observe any considerable difference in VBVarSel performance and variable selection rates as we significantly scale up the dataset size. Using the same prior initialisation as before, the model settled on a 4-5 clusters model where a median of 6723 covariates were selected, approximately 39% of the full set. Among the PAM50 genes, the median rate of selection was 75%, which is similar to what was achieved with smaller dataset sizes, and significantly better than random (*Fisher Test*, $p \ll 0.00001$). The stratification is sensible and comparable to previous experiments, as can be observed in Figure 5.6.



(a) On all covariates (b) On only selected covariates

Fig. 5.6 Scatter plots of VBVarSel 4-cluster model on the complete TCGA dataset, when PCA is applied to either all covariates, or only the selected ones.



(a) On all covariates (b) On only selected covariates

Fig. 5.7 Scatter plots of VBVarSel 2-cluster model on the complete TCGA dataset, when PCA is applied to either all covariates, or only the selected ones.

Reducing α_0 and b_{0j}, we obtained a 2-clusters model, shown in Figure 5.7, in which similarly a median of 6203 covariates were selected. Among the PAM50, the median rate of selection was slightly higher, approximately 82%. Cluster 0 is significantly associated with Basal-like tumours (*Fisher Test*, $p \ll 0.00001$). In both models, variable selection produced smaller, tighter clusters and increased the separation between them.

5.4.4 Benchmarking on pre-processed TCGA expression dataset

To allow comparison with existing literature, we pre-process the complete TCGA dataset as in Lock and Dunson (2013) and Crook et al. (2019b). We keep 645 genes for each of the 348 tumour samples, of which 14 are from the PAM50 subset. Initialising lower α_0 and b_{0j}, VBVarSel converges to two clusters, and 318 variables are selected to discriminate between the two groups, which includes all the 14 PAM50 genes (*Fisher Test*, $p < 0.00005$). These results are comparable to what is reported in Crook et al. (2019b), although VBVarSel tends to select more variables overall, and they are also in agreement with stratifications and selection rates obtained in previous experiments. Indeed, we again observe smaller, tighter, and more separable clusters when focusing on the retained variables.

5.5 Motivation for annealing

Over the course of our evaluation across the extensive range of experiments implemented, in addition to assessing performance and strengths of our algorithm, we gathered potential concerns and encountered challenges. First, we underscore again the high sensitivity of the algorithm to the initialisation of some parameters, namely the concentration parameter α_0, the shape parameter d_0, and most significantly the scale b_{0j}. We also identify two possible areas of caution: extremely quick convergence and the conditional independence assumptions.

5.5.1 Investigating convergence

While rapid convergence could be a desirable behaviour of the optimiser, we noticed a surprisingly and worryingly quick convergence in most experiments, even in less than 5 iterations for some. Such a swift convergence might indicate that the algorithm is settling on a local optimum, which compromises its ability to capture the underlying complexity. A typical ELBO pattern would look like Figure 5.8. After a very sharp increase in the first few iterations, the ELBO plateaus and only changes by very small amounts. This may indicate that the VBVarSel algorithm is finding a local optimum almost immediately, which may not be the global optimum, but it gets trapped there and fine-tunes within that region.

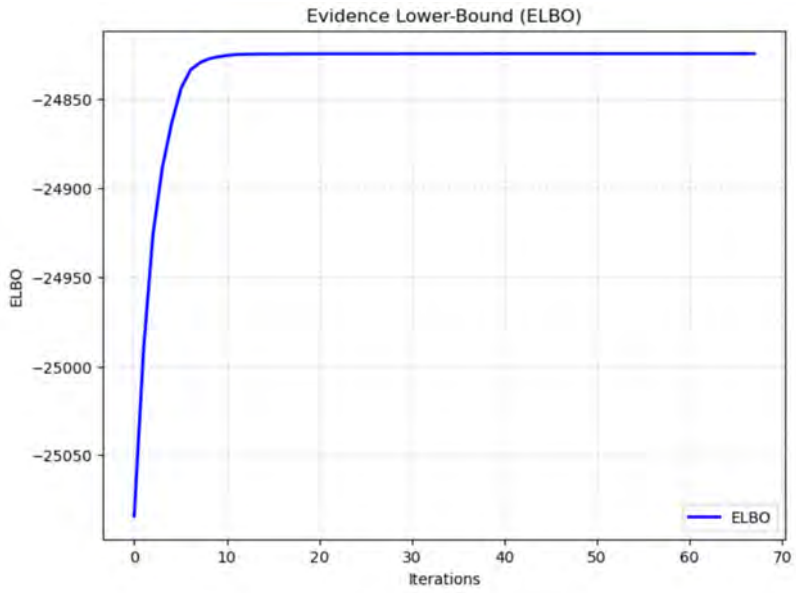


Fig. 5.8 Typical ELBO shape until convergence of VBVarSel on the PAM50 set only.

In our context, this can be partly explained by the data we used. From the scatter plots and heatmaps we have previously shown, the most evident separation is the one between basal and non-basal samples. Moreover, Figure 5.9 shows hierarchical clustering on the normalised PAM50 genes expression clearly identifies only two major subgroups, partly explaining why we only maximise the ELBO when we retrieve a 2-clusters model. Conceivably, the VBVarSel algorithm quickly, and relatively accurately finds a separation of the data into basal and non-basal samples, and then tries to fine-tune from there. This becomes a *sub-clustering* problem.

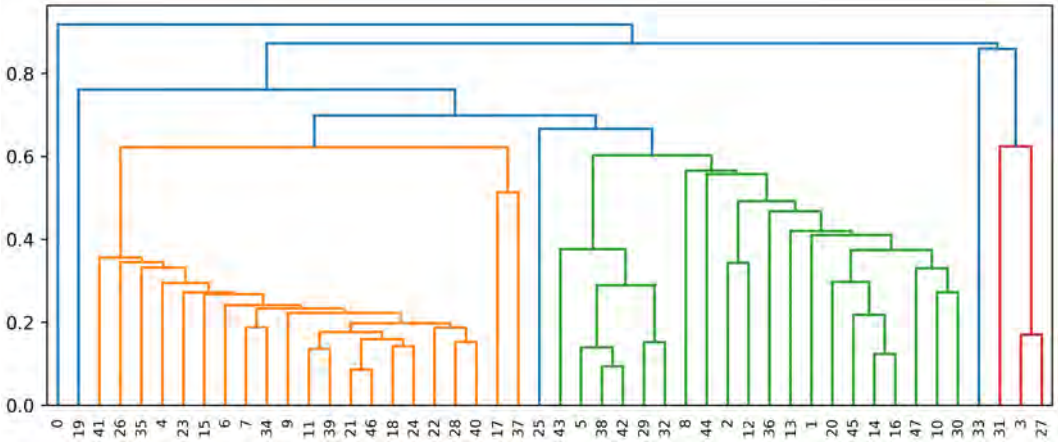


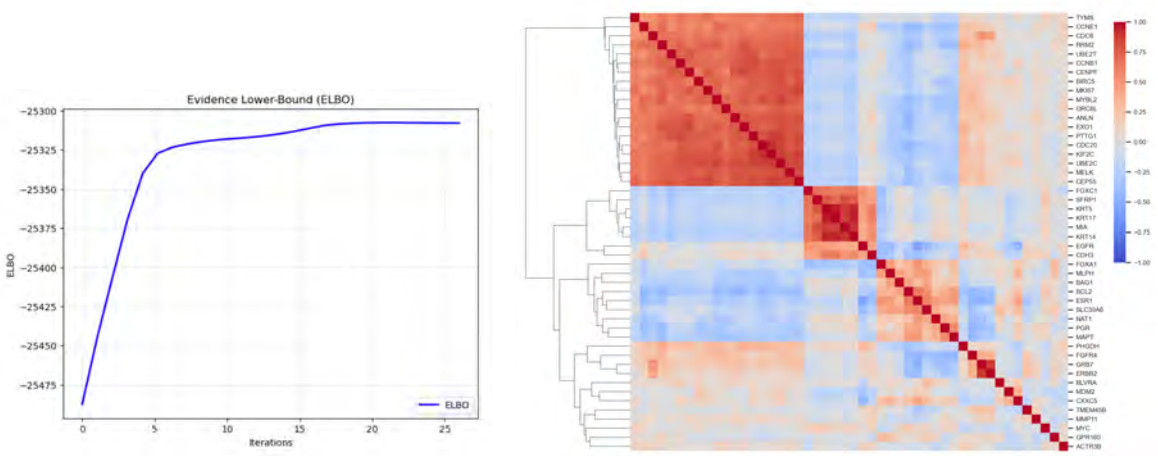
Fig. 5.9 Hierarchical clustering dendrogram on the normalised PAM50 gene expressions.

Sub-clustering

Sub-clustering, as the name suggests, is concerned with the task of clustering within an existing cluster which can also be defined as a nested clustering problem. With our data, after identifying a broad split between basal and non-basal samples, the VBVarSel algorithm tries to further divide the non-basal samples into more specific clusters. This is a very complex task, which requires a greater degree of precision and sophistication as it involves finding structure within a group that already seems homogeneous. There exist algorithms specifically built for this (Li et al., 2010; Liao et al., 2004; Murtagh, 1983; Patel et al., 2015).

Evaluation of VBVarSel on non-basal samples

To confirm our hypothesis around the rapid convergence observed, we remove the basal samples from the dataset and apply VBVarSel on the remaining observations, using only the PAM50 genes. We expect that by removing the evident parent clustering structure, the model would take longer to converge and potentially better stratify the non-basal samples. Indeed, Figure 5.10 shows that convergence is slower than Figure 5.8. Moreover, Figure 5.11 shows a better stratification of non-basal samples, particularly Luminal B and HER2-enriched, despite the four subtypes still looking like a homogeneous cluster.



(a) (b)

Fig. 5.10 (a) Typical ELBO shape until convergence of VBVarSel on non-basal samples. (b) A correlation heatmap between PAM50 genes for non-basal samples.

Interestingly, from a biological perspective, variable selection discarded considerably more PAM50 genes. Across 10 runs on different data randomisations, the median number of discarded genes was 15, with upper and lower quartiles of 18 and 12 respectively. Examining the correlation plot in Figure 5.10, we notice fewer correlated variables compared to Figure

5.2. Moreover, the PAM50 genes discarded by the algorithm correspond to the uncorrelated ones for non-basal samples (*Fisher Test*, $p < 0.0001$) and align with the green subgroup in Figure 5.9. This suggests that a large part of the PAM50 gene set characterises predominantly Basal-like tumour samples.

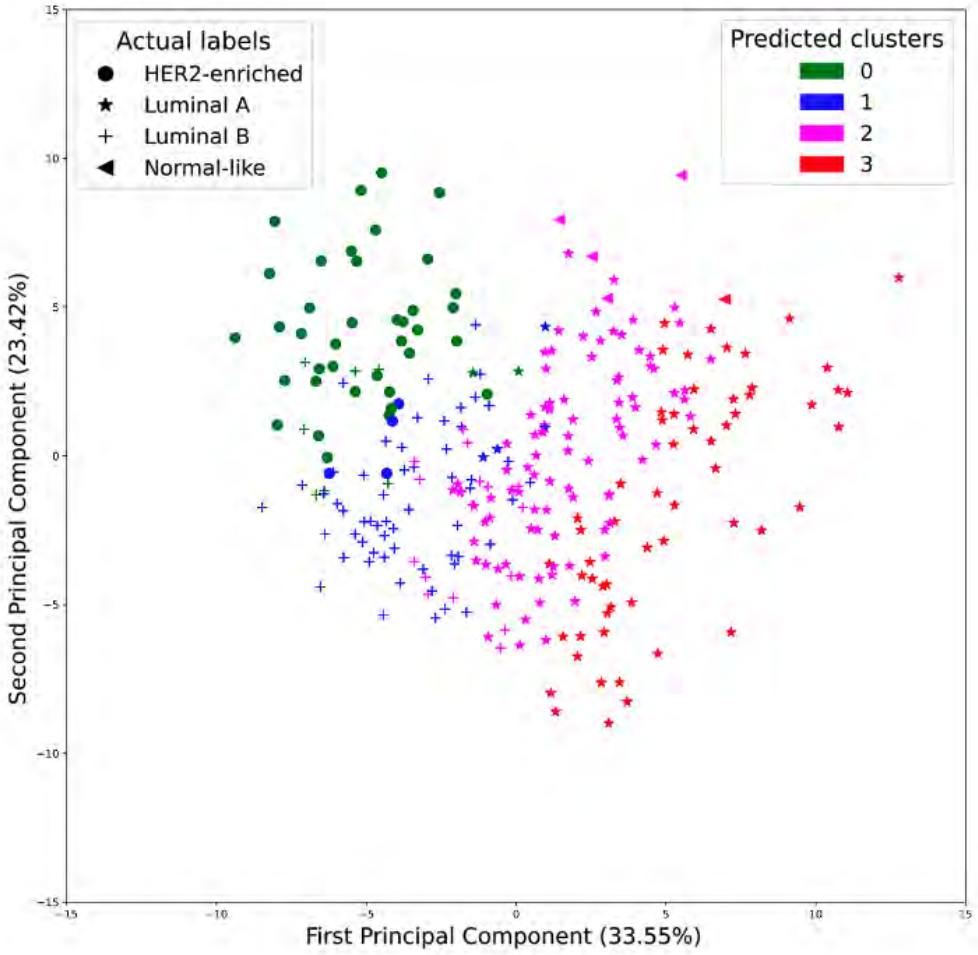
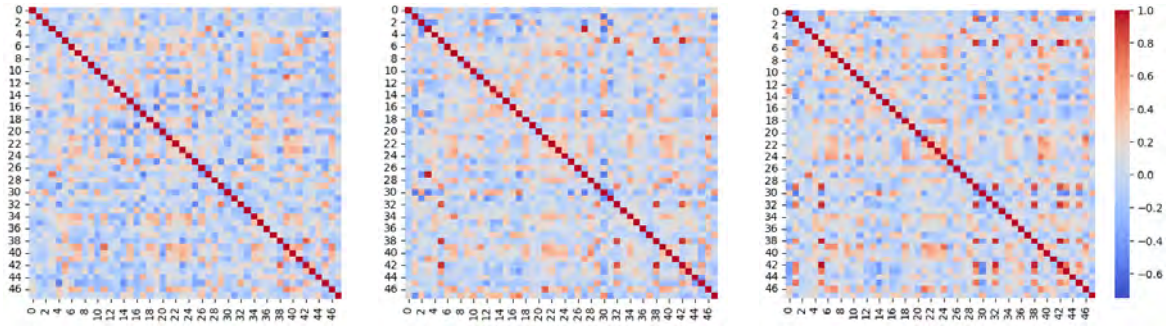


Fig. 5.11 PCA scatter plot of VBVarSel stratification on TCGA data using only the PAM50 set and non-basal observations.

In conclusion, although we partly explained the rapid ELBO convergence in this context as a sub-clustering problem, the challenge still remains. Ideally, we would want the model to better explore possible stratifications and variable selection configurations, instead of quickly exploiting one.

5.5.2 Investigating conditional independence assumption

The second potential limitation identified involves our model's core assumption: conditional independence between covariates, given the cluster assignment. While this is a common assumption in many models to simplify calculations, it may not hold true in real-world data. Indeed, we observed significant correlations, especially among PAM50 genes, which influence variable selection (Figure 5.2). Therefore, we wish to investigate whether our assumption is sensible, i.e. the correlation diminishes upon conditioning on cluster assignment, or if it does not hold in practice. If the latter is true, our model might be oversimplifying the complexity of the data, compromising accuracy and performance more broadly.



(a) Cluster A of Figure 5.3 (b) Cluster B of Figure 5.3 (c) Cluster E of Figure 5.3

Fig. 5.12 Correlation heatmaps of the normalised PAM50 gene expression, conditioned on cluster assignment.

We examined the data after clustering and in Figure 5.12 we show the correlation among the PAM50 genes after conditioning on cluster assignment, for all the inferred clusters that align well with the *ground-truth* subtypes. The model we used is the one presented in Section 5.4.1. Compared to Figure 5.2, the correlation structure is notably reduced. However, while our conditional independence assumption reasonably stands validated, we cannot fully ignore the potential impact of neglecting the correlation structure on our model's performance.

5.5.3 How could annealing enhance inference?

The two potential concerns identified could be mitigated by annealing. By introducing a temperature schedule we could encourage the algorithm to explore a broader range of solutions and uncover better stratifications. This would in turn result in a slower convergence. Furthermore, Ruffieux et al. (2020) has shown annealing is particularly beneficial when dealing with correlated data. Therefore, if and when the conditional independence assumption does not hold strictly, the annealed VBVarSel could still be capable of exploring this complexity and not rapidly committing to a specific cluster assignment. In the subsequent section, we present the performance of the annealed VBVarSel for this specific application.

5.6 Introducing annealing

We replicated three of the experiments previously performed. First, we applied the annealed VBVarSel to PAM50 genes; then we used the PAM50 set plus 100 randomly chosen covariates; lastly, we worked with the pre-processed TCGA expression dataset, as in Section 5.4.4. For all experiments, we normalise the data to zero mean and unit variance, and the parameters were initialised according to Table B.1, experiment *A-TCGA*. We explored a wide variety of annealing approaches and report the temperature configurations allowing more evident advantages over the non-annealed VBVarSel.

5.6.1 Evaluation

Table 5.2 shows the achieved performance with different types of temperature schedules.

Data	T	PAM50	Relevant	Irrelevant	ARI
PAM50	2 F	45 [45, 46]	45 [45, 46]	3 [2, 3]	0.52 [0.50, 0.57]
PAM50	3 G	35 [35, 40]	35 [35, 40]	13 [8, 13]	0.51 [0.48, 0.56]
PAM50	3 H	46 [44, 47]	46 [44, 47]	2 [1, 4]	0.48 [0.45, 0.53]
PAM50 + 100	3 G	25 [25, 26]	34 [33, 36]	114 [112, 115]	0.48 [0.48, 0.55]
Preprocessed	2 F	13 [13, 14]	244 [230, 260]	401 [385, 415]	0.42 [0.37, 0.48]
Preprocessed	4 G	12 [10, 14]	160 [160, 200]	485 [445, 485]	0.35 [0.35, 0.36]

Table 5.2 Annealed VBVarSel performance across different experiments on TCGA data. F: Fixed Temperature, G: Geometric H: Harmonic schedule and the initial temperature is given. We present the median scores with the upper and lower quartiles across 10 independent runs on different data randomisation.

The first three rows describe the application of VBVarSel on the PAM50 genes exclusively. With fixed temperature, the model stably retains a higher number of PAM50 covariates. The stratification obtained, which is shown in Figure 5.13, is very accurate, particularly in sub-clustering non-basal samples, and is significantly better than random (*chi-squared* $p < 0.00001$). As expected, we observed a slower ELBO convergence (Figure B.2). When using a geometric or harmonic schedule we can more easily compare the performance to the non-annealed case as we are targeting the same (approximate) posterior, as discussed in Section 3.3. We annealed the first 5 or 20 iterations respectively, starting from an initial temperature of 3. The model stably selected fewer (geometric) or more (harmonic) PAM50 genes on average but both stratifications were better and more stable than the non-annealed model.

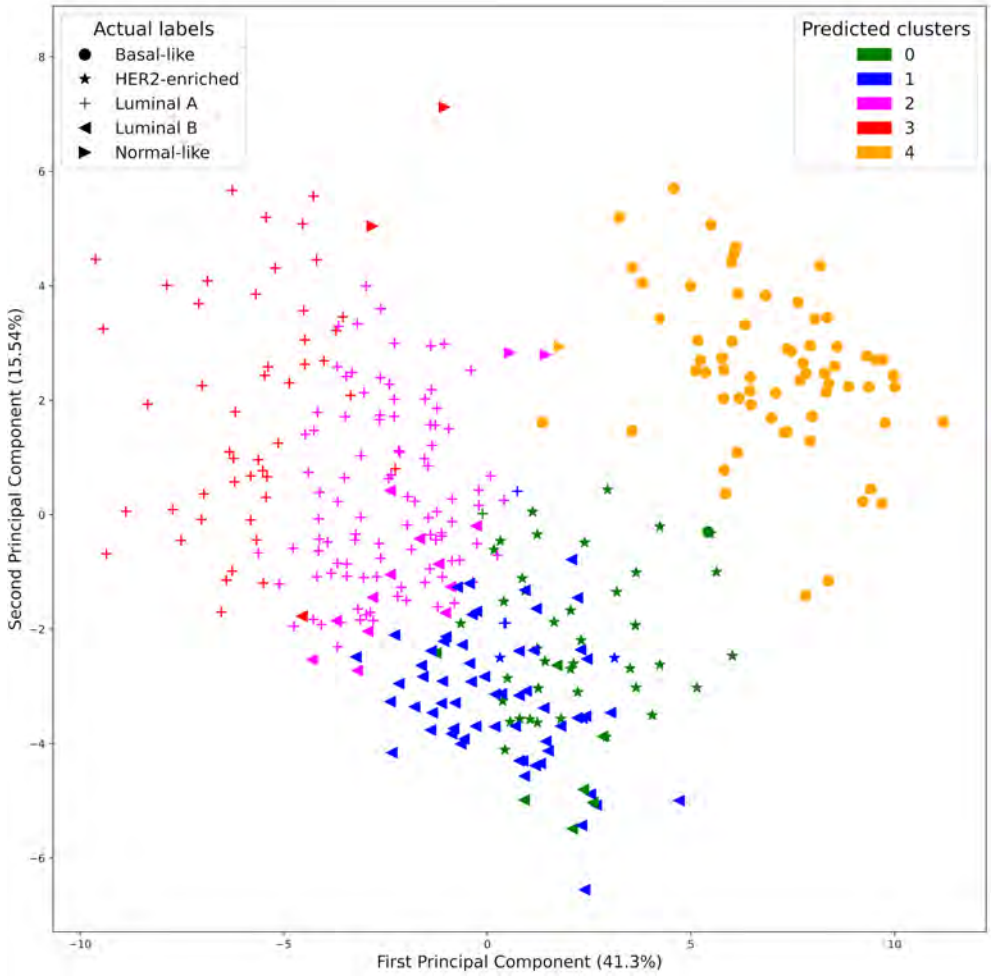
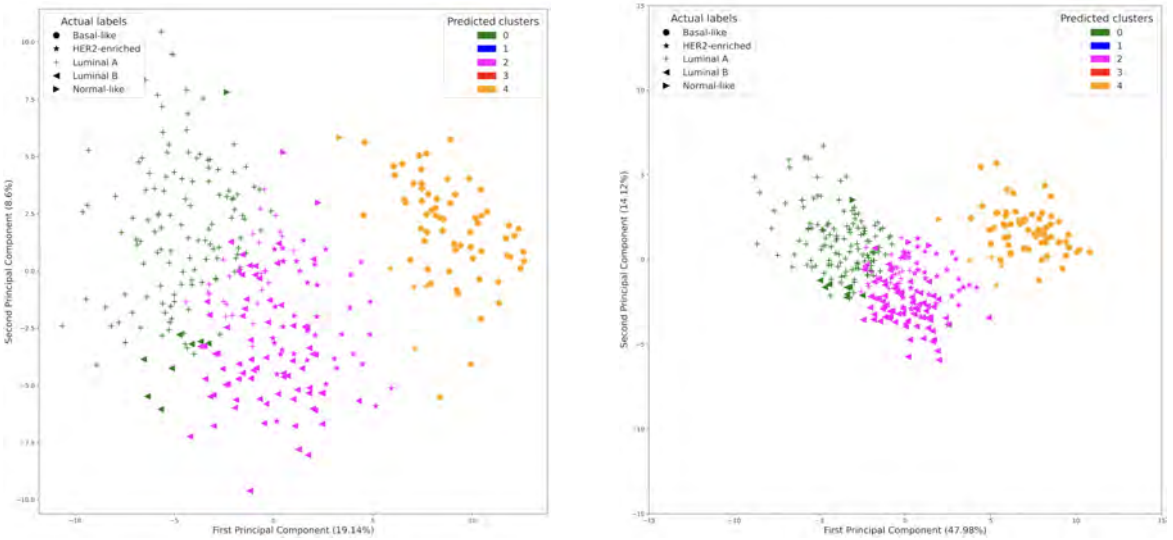


Fig. 5.13 PCA scatter plot of VBVarSel stratification on TCGA data using only the PAM50 set and annealing with fixed temperature.

Then, we extracted the PAM50 covariates and 100 randomly sampled genes to evaluate annealing advantages with more irrelevant variables. This dataset was very sensitive to the temperature schedule. We found optimal performance was achieved with a geometric schedule, starting from a temperature of 3 and gradually reducing it to 1 in the first 5 iterations. The annealed VBVarSel algorithm converged to a much-reduced covariate space compared to the non-annealed model and around half of the PAM50 variables were deselected. Nonetheless, the stratification obtained is sensible and more stable. The algorithm clustered observations into either 3 or 4 groups, which were tighter and more separated in PCA space, proving the increased deselection rate was indeed beneficial (Figure 5.14). Perhaps, given the model is encouraged to explore, it can find local optima in the posterior space that the non-annealed optimiser is very unlikely to reach, such as a much-reduced covariate space.

Finally, we applied the annealed VBVarSel on the pre-processed data as in Section 5.4.4. We found both fixed temperature and geometric schedules to work well. When using fixed temperature we effectively encourage exploration throughout all iterations. This results in more variability in the stratification, with different numbers of clusters in the obtained models, and in the variable selection. When instead using a geometric schedule, we started from a temperature of 4 and annealed the first 10 iterations. The model gradually shifts the focus from exploration to exploitation, focusing on refining the best configuration it found. Indeed, we observed VBVarSel would more stably converge to a 2-cluster model, with less variability in the selected covariates.



(a) On all covariates (b) On only selected covariates

Fig. 5.14 Scatter plots of geometrically annealed VBVarSel stratification on the 348 TCGA samples when PCA is applied to either all PAM50 plus 100 genes, or only the selected ones.

It is difficult to set concrete expectations on which advantages we should observe with the introduction of annealing, especially in the context of real data. Nonetheless, we generally observed an improvement in performance, particularly for what concerns sub-clustering the non-basal patients, as well as more stability in the inference. Moreover, annealing enhanced the algorithm's robustness to parameter initialisation and its ability to completely "empty" superfluous clusters. The only parameter that required very careful tuning was the temperature T and its schedule. Both fixed and time-dependent (geometric or harmonic) temperatures yield some advantages over the non-annealed VBVarSel, even though they target different (approximate) posterior distributions. Nonetheless, the time-dependent schedule offers a more balanced exploration, which is generally preferable.

Chapter 6

Application to TCPA proteomic pan-cancer data

We conclude our exploration of the VBVarSel algorithm with its application on proteomic pan-cancer data from The Cancer Proteome Atlas (TCPA). Though our experiments on TCGA were more extensive, our aim with TCPA is to further examine the algorithm capabilities and robustness on a dataset that is inherently different, as we will now explain.

6.1 Introduction to The Cancer Proteome Atlas (TCPA)

The Cancer Proteome Atlas (TCPA) (Akbani et al., 2014; Li et al., 2013) contains protein expression data over a large number of tumor and cell line samples, obtained using reverse-phase protein arrays (RPPAs) (Sheehan et al., 2005). We extract a total of 5157 tumour samples and we only keep the 217 proteins measured for all of those. The labels of these samples identify a total of 19 cancer types, distributed as shown in Figure B.4.

Compared to TCGA, the data we extract from TCPA is relatively low-dimensional as the number of observations greatly exceeds the number of variables, and we find less correlation between covariates, as shown in Figure 6.1. Furthermore, there is little prior knowledge about how the model should perform. There are 19 cancer types but there could be subtypes within those as well as inter-cancer relationships (School et al., 2012; Uhlen et al., 2017; Weinstein et al., 2013). As for variable selection, the proteins profiled in TCPA are already pre-selected based on their relevance to cancer biology and therapy (Akbani et al., 2014).

6.2 Evaluation

For all experiments, we normalise the data to zero mean and unit variance, and the parameters are initialised according to Table B.1, experiment *TCPA*. We evaluate performance as explained in Section 3.4.3.

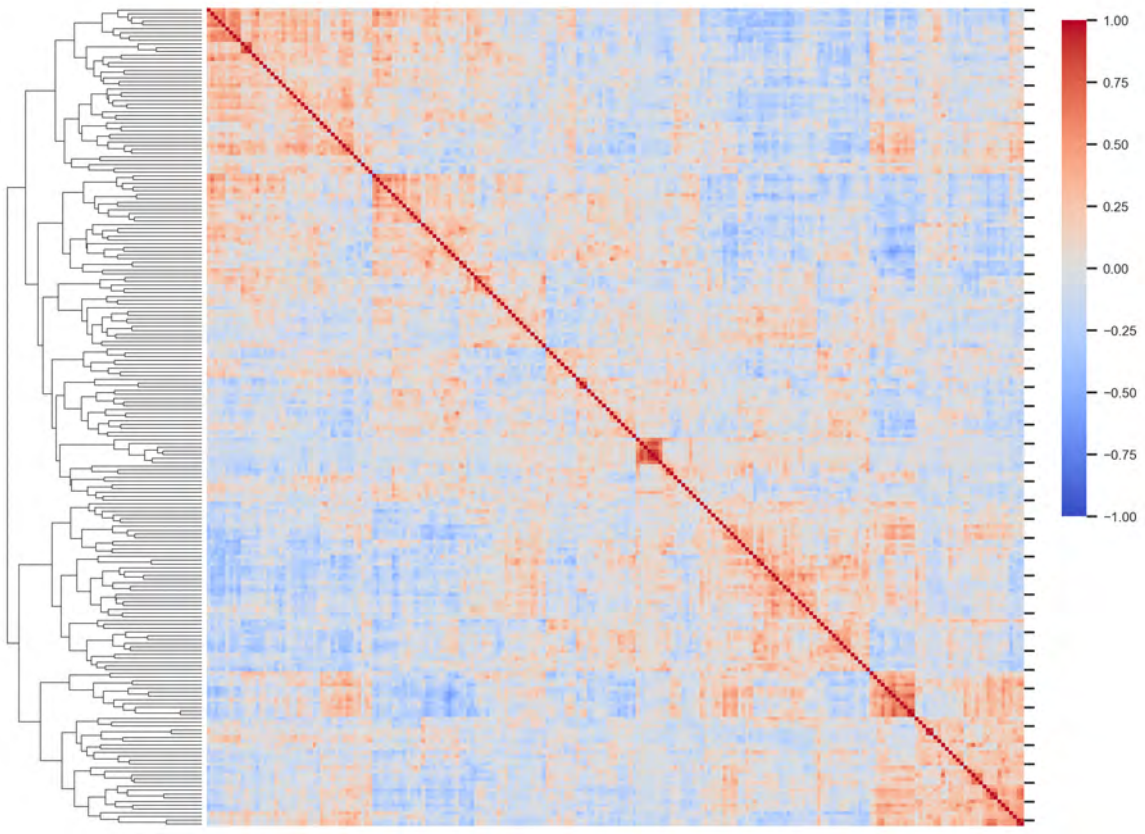


Fig. 6.1 A correlation heatmap of the normalised protein expressions in TCPA data.

Results

Remarkably, despite very little knowledge about the data and expected performance to inform our parameter initialisation, VBVarSel was able to converge to sensible results with "standard" configurations obtained using the ELBO for model selection. The algorithm assigns more than 20 observations to 25 clusters on average. Figure 6.2 shows the correspondence between the inferred clusters and the cancer subtypes. Most cancers are strongly associated with a unique cluster. When there is some overlap, this is in agreement with other relevant literature (Akbani et al., 2014; Crook et al., 2019b; Hoadley et al., 2014). For instance, the cancers

HNSC, LUAD, and LUSC which are all aero-digestive cancers are most often grouped together. Same applies to STAD, COAD, and READ which are cancers of the digestive tract. In contrast, breast cancer (BRCA) and endometrial cancer (UCEC) are split into subgroups (Akbani et al., 2014).

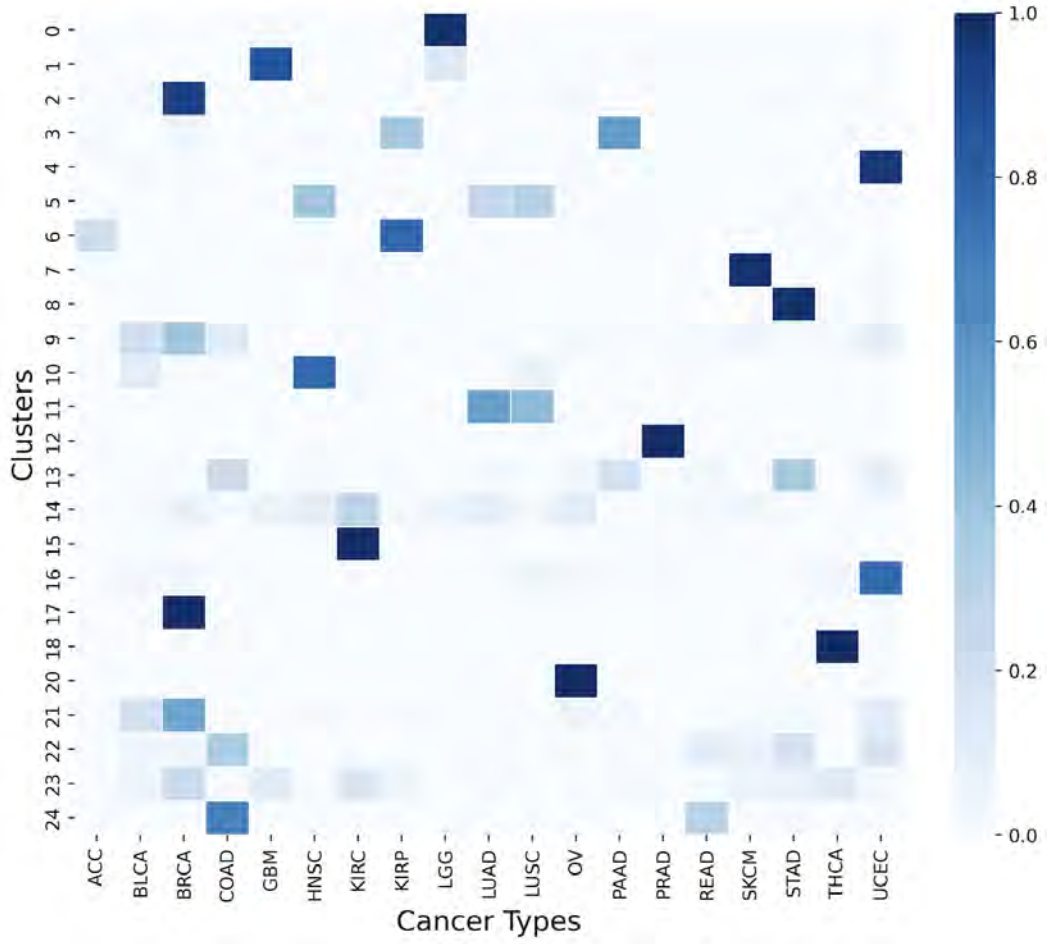


Fig. 6.2 A heatmap of the correspondence between VBVarSel clusters and the cancer subtypes. We filtered out clusters with less than 20 observations.

As for variable selection, VBVarSel tends to retain most of the variables, with a median rate of retention of 90%. Given the profiled proteins in TCPA are already pre-selected (Akbani et al., 2014), and we obtain sensible stratification, there is no indication that the rate is inadequate. Moreover, with different parameter initialisation, such as lower d_0 or c_j, we were able to obtain a lower retention rate but the stratification obtained was considerably worse. To better assess variable selection, we permuted the rows of varying numbers of randomly selected covariates. As expected, at least 91% of the "perturbed" variables were deselected.

Figure 6.3 shows a heatmap of the stratification and the retained genetic expressions. We observe nice agreement between clusters, cancer subtypes, and genetic expressions.

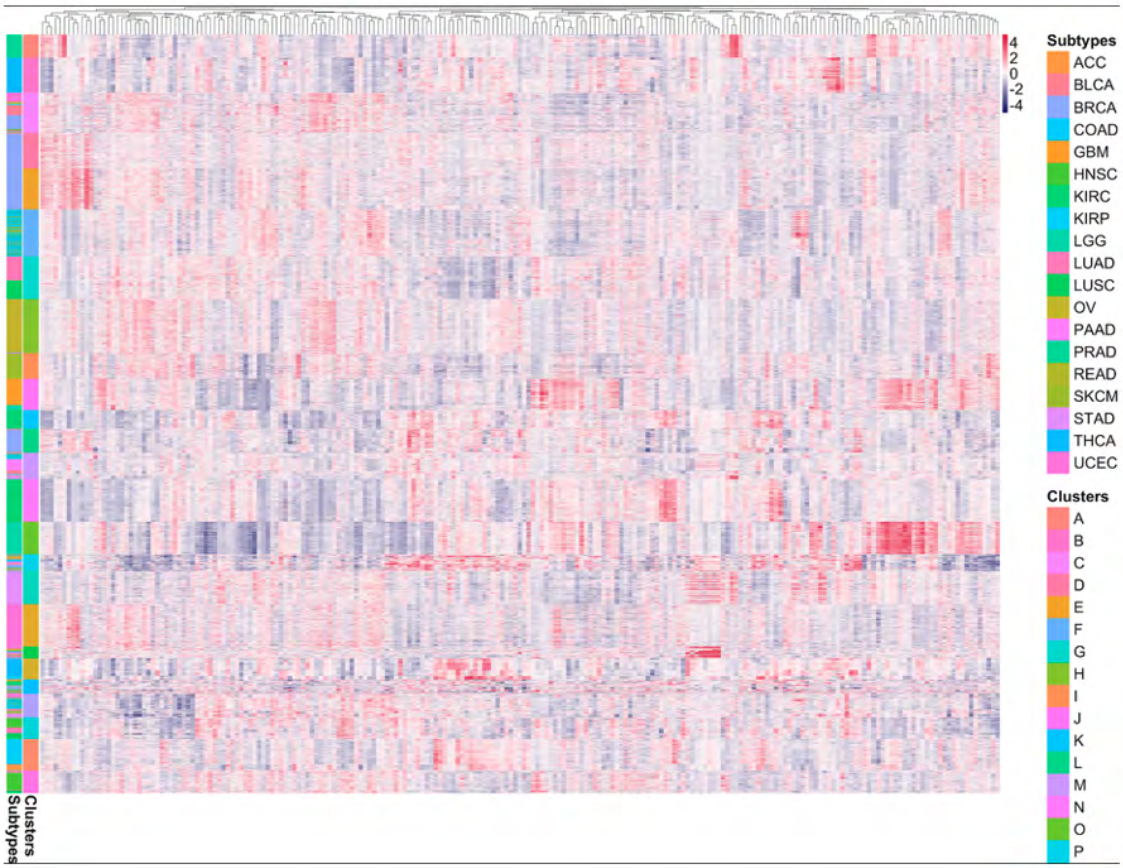


Fig. 6.3 A heatmap of the TCPA expression data using VBVarSel stratification.

In summary, our experiments revealed that VBVarSel scales well not only with increasing number of covariates, but also with increasing number of observations. Remarkably, the algorithm can adeptly adjust variable selection based on the data and eventually retain a large number of covariates with clustering information. Finally, we underscore the pivotal role of the ELBO in model selection which allows us to fine-tune our parameters initialisation even in the absence of prior knowledge of the data.

Chapter 7

Conclusion

In this thesis we unveiled VBVarSel, a novel algorithm for simultaneous model-based clustering and variable selection. Combining finite Gaussian mixture models, with a latent binary covariate selection indicator within a variational framework, our model achieved exceptional speed, computational efficiency, and scalability - all without compromising performance and accuracy.

After establishing a precise formulation of the problem setting and describing the core theoretical concepts of our model in Chapter 2, we delved into the mathematical and technical intricacies of the proposed algorithm in Chapter 3, providing clarity on the inference process and practical implementation. In Chapter 4 we began the evaluation of our model with synthetic data to allow comparison with established methods in a controlled scenario. Remarkably, our model reduced the runtime by up to an impressive tenfold, while achieving near-perfect performance. This allowed us to empirically demonstrate the benefits of Variational Inference as a computationally efficient alternative to other more popular inference methods in the field.

We believe a distinctive feature of VBVarSel is its capability to handle complete datasets in a remarkably efficient runtime, without the necessity for pre-processing. This is of particular relevance in the context of precision medicine, where the algorithm can be deployed to find complex relationships and patterns in high-dimensional biomedical data, facilitating the identification of disease subtypes and relevant biomarkers. We demonstrated this throughout Chapters 5 and 6, with an extensive number of experiments, to showcase VBVarSel's unparalleled speed and scalability, as well as good and sensible performance, in line with established research.

Notably, VBVarSel uniquely integrates annealing to tackle the *local-optima trap*. Our empirical findings supported the theoretical claims that establishing an effective balance between exploration and exploitation with a time-dependent temperature schedule would

enhance inference in multi-modal posterior landscapes. Indeed, we observed increased robustness and adaptability to sub-optimal parameter initialisations, correlated data, and noise and a stabilised inference with both synthetic and real data.

We began this project with an ambitious, yet clear objective: to provide a compact algorithm, capable of accurately and reliably performing simultaneous clustering and feature selection, with superior speed and efficiency, especially when scaling to large, high-dimensional datasets. We now conclude this thesis with VBVarSel - an embodiment of our vision, which ticks all the desired boxes and also uniquely introduces annealing. While the algorithm is not perfect, limitations only pave the way for future refinements.

7.1 Limitations and Future Directions

A critical and concrete next step is the conversion of our Python-based codebase into an R package, given the majority of the existing methods are available as such. We believe this will also further increase its speed and efficiency.

More broadly, the methodologies and findings presented in this study, as well as limitations identified in the model, open a wide range of opportunities for future research. Starting from the clustering task, while the model showed promising and sensible results on real data, pushing its stratification accuracy beyond a certain threshold was challenging. A future direction could be a semi-supervised approach such as outcome-guided clustering. The idea is to introduce a measurable response variable, such as survival time, to guide clustering and find patterns associated with differences in outcomes. Another area for enhancement is our choice of the covariate selection indicator. We believe allowing both a continuous or binary indicator could offer a more nuanced understanding of each covariate's importance, particularly in datasets like TCPA, where the differences in variable salience are subtle. Finally, due to time constraints, we were unable to thoroughly explore and implement different annealing temperature schedules, which we leave as future work.

Looking ahead, broadening VBVarSel application across medical and biological domains is promising. Moreover, given VBVarSel strength in handling high-dimensional data, it could be valuable in other research areas such as Finance, for tasks like portfolio optimisation, or Digital Marketing, for customer segmentation and content personalisation.

References

Akbani, R., Ng, P. K. S., Werner, H. M., Shahmoradgoli, M., Zhang, F., Ju, Z., Liu, W., Yang, J.-Y., Yoshihara, K., Li, J., et al. (2014). A pan-cancer proteomic perspective on the cancer genome atlas. *Nature communications*, 5(1):3887.

Bellman, R. (1957). *Dynamic Programming*. Princeton University Press.

Bensmail, H. and Meulman, J. J. (1998). MCMC inference for model-based cluster analysis. In Rizzi, A., Vichi, M., and Bock, H.-H., editors, *Advances in Data Science and Classification*, pages 191–196. Springer Berlin Heidelberg.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

Bouveyron, C. and Brunet-Saumard, C. (2014). Model-based clustering of high-dimensional data: A review. *Computational Statistics Data Analysis*, 71:52–78.

Boyd, S. P. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Bradley, A. P. (1997). The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159.

Brooks, S. P. and Gelman, A. (1998). General methods for monitoring convergence of iterative simulations. *Journal of computational and graphical statistics*, 7(4):434–455.

Celeux, G., Martin-Magniette, M.-L., Maugis-Rabusseau, C., and Raftery, A. (2013). Comparing model selection and regularization approaches to variable selection in model-based clustering. *Journal de la Societe francaise de statistique (2009)*, 155.

Claeskens, G. and Hjort, N. L. (2008). *Model Selection and Model Averaging*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

Constantinopoulos, C., Titsias, M. K., and Likas, A. (2006). Bayesian feature and model selection for gaussian mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(6):1013–1018.

Cremin, C. J., Dash, S., and Huang, X. (2022). Big data: Historic advances and emerging trends in biomedical research. *Current Research in Biotechnology*, 4:138–151.

Crook, O. M., Breckels, L. M., Lilley, K. S., Kirk, P. D., and Gatto, L. (2019a). A bioconductor workflow for the bayesian analysis of spatial proteomics. *F1000Research*, 8.

Crook, O. M., Gatto, L., and Kirk, P. D. W. (2019b). Fast approximate inference for variable selection in dirichlet process mixtures, with an application to pan-cancer proteomics. *Statistical Applications in Genetics and Molecular Biology*, 18(6):20180065.

Curtis, C., Shah, S. P., Chin, S.-F., Turashvili, G., Rueda, O. M., Dunning, M. J., Speed, D., Lynch, A. G., Samarajiwa, S., Yuan, Y., et al. (2012). The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature*, 486(7403):346–352.

Dinalankara, W., Ke, Q., Xu, Y., Ji, L., Pagane, N., Lien, A., Matam, T., Fertig, E., Price, N., Younes, L., Marchionni, L., and Geman, D. (2018). Digitizing omics profiles by divergence from a baseline. *Proceedings of the National Academy of Sciences*, 115:201721628.

Duan, Q., Kou, Y., Clark, N., Gordonov, S., and Ma'ayan, A. (2013). Metasignatures identify two major subtypes of breast cancer. *CPT: pharmacometrics & systems pharmacology*, 2(3):1–10.

Earl, D. J. and Deem, M. W. (2005). Parallel tempering: Theory, applications, and new perspectives. *Physical Chemistry Chemical Physics*, 7(23):3910–3916.

Ferguson, T. S. (1973). A Bayesian Analysis of Some Nonparametric Problems. *The Annals of Statistics*, 1(2):209 – 230.

Fop, M. and Murphy, T. B. (2018). Variable selection methods for model-based clustering. *Statistics Surveys*, 12(none):18 – 65.

Fraley, C. and Raftery, A. (2007). Bayesian regularization for normal mixture estimation and model-based clustering. *Journal of Classification*, 24:155–181.

Fraley, C. and Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631.

Fritsch, A. and Ickstadt, K. (2009). Improved criteria for clustering based on the posterior similarity matrix. *Bayesian Analysis*, 4(2):367 – 391.

Galili, T. (2015). dendextend: an R package for visualizing, adjusting, and comparing trees of hierarchical clustering. *Bioinformatics*.

Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical science*, 7(4):457–472.

Gnanadesikan, R., Kettenring, J., and Tsao, S. (1995). Weighting and selection of variables for cluster analysis. *Journal of Classification*, 12:113–136.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., et al. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *science*, 286(5439):531–537.

Hancer, E., Zhang, M., and Xue, B. (2020). A survey on feature selection approaches for clustering. *Artificial Intelligence Review*, 53:4519–4545.

Hastie, T., Tibshirani, R., Friedman, J., and Franklin, J. (2004). The elements of statistical learning: Data mining, inference, and prediction. *Math. Intell.*, 27:83–85.

Hoadley, K. A., Yau, C., Wolf, D. M., Cherniack, A. D., Tamborero, D., Ng, S., Leiserson, M. D., Niu, B., McLellan, M. D., Uzunangelov, V., Zhang, J., Kandoth, C., Akbani, R., Shen, H., Omberg, L., Chu, A., Margolin, A. A., van't Veer, L. J., Lopez-Bigas, N., Laird, P. W., Raphael, B. J., Ding, L., Robertson, A. G., Byers, L. A., Mills, G. B., Weinstein, J. N., Van Waes, C., Chen, Z., Collisson, E. A., Benz, C. C., Perou, C. M., and Stuart, J. M. (2014). Multiplatform analysis of 12 cancer types reveals molecular classification within and across tissues of origin. *Cell*, 158(4):929–944.

Hubert, L. J. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2:193–218.

Katahira, K., Watanabe, K., and Okada, M. (2008). Deterministic annealing variant of variational bayes method.

Kim, S., Tadesse, M. G., and Vannucci, M. (2006). Variable selection in clustering via dirichlet process mixture models. *Biometrika*, 93(4):877–893.

Kirk, P. D. W., Pagani, F., and Richardson, S. (2023). Bayesian outcome-guided multi-view mixture models with applications in molecular precision medicine.

Kirkpatrick, S., Gelatt Jr, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, 220(4598):671–680.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Lau, J. and Green, P. (2007). Bayesian model-based clustering procedures. *Journal of Computational and Graphical Statistics - J COMPUT GRAPH STAT*, 16.

Law, M. H., Figueiredo, M. A., and Jain, A. K. (2004). Simultaneous feature selection and clustering using mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1154–1166.

Li, J., Lu, Y., Akbani, R., Ju, Z., Roebuck, P. L., Liu, W., Yang, J.-Y., Broom, B. M., Verhaak, R. G., Kane, D. W., Wakefield, C., Weinstein, J. N., Mills, G. B., and Liang, H. (2013). TCPA: a resource for cancer functional proteomics data. *Nature Methods*, 10:1046–1047.

Li, X., Ye, Y., Li, M. J., and Ng, M. K. (2010). On cluster tree for nested and multi-density data clustering. *Pattern Recognition*, 43(9):3130–3143.

Liao, W.-k., Liu, Y., and Choudhary, A. (2004). A grid-based clustering algorithm using adaptive mesh refinement. In *7th workshop on mining scientific and engineering datasets of SIAM international conference on data mining*, volume 22, pages 61–69.

Liu, J., Zhang, J., Palumbo, M., and Lawrence, C. (2003). Bayesian clustering with variable and transformation selections. *Bayesian Statistics*, 7:249–275.

Liverani, S., Hastie, D. I., Azizi, L., Papathomas, M., and Richardson, S. (2014). PReMiuM: An R package for profile regression mixture models using dirichlet processes.

Lock, E. F. and Dunson, D. B. (2013). Bayesian consensus clustering. *Bioinformatics*, 29(20):2610–2616.

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

Mandt, S., McInerney, J., Abrol, F., Ranganath, R., and Blei, D. (2016). Variational tempering.

Martin, A. D., Quinn, K. M., and Park, J. H. (2011). MCMCpack: Markov chain monte carlo in R. *Journal of Statistical Software*, 42(9):22.

McLachlan, G. J. and Rathnayake, S. (2014). On the number of components in a gaussian mixture model. *WIREs Data Mining and Knowledge Discovery*, 4(5):341–355.

McNicholas, P. D. (2016). Model-based clusterig. 33:331–373.

Miao, J. and Niu, L. (2016). A survey on feature selection. *Procedia Computer Science*, 91:919–926.

Morgan, M. and Ramos, M. (2023). *BiocManager: Access the Bioconductor Project Package Repository*. R package version 1.30.21.

Mounir, M., Lucchetta, M., C, S. T., Olsen, C., Bontempi, G., Chen, X., Noushmehr, H., Colaprico, A., and Papaleo, E. (2019). New functionalities in the TCGAbiolinks package for the study and integration of cancer data from GDC and GTEx. *PLoS computational biology*, 15(3):e1006701.

Murtagh, F. (1983). A survey of recent advances in hierarchical clustering algorithms. *The computer journal*, 26(4):354–359.

NHS (2022). Breast cancer in women. Accessed: 2023-07-06.

Orrantia-Borunda, E., Anchondo-Nuñez, P., Acuña-Aguilar, L. E., Gómez-Valles, F. O., and Ramírez-Valdespino, C. A. (2022). Subtypes of breast cancer. *Breast Cancer [Internet]*.

Papathomas, M., Molitor, J., Hoggart, C., Hastie, D., and Richardson, S. (2012). Exploring data from genetic association studies using bayesian variable selection and the dirichlet process: Application to searching for gene × gene patterns. *Genetic Epidemiology*, 36(6):663–674.

Parisi, G. (1979). Toward a mean field theory for spin glasses. *Physics Letters A*, 73(3):203–205.

Parker, J. S., Mullins, M., Cheang, M. C., Leung, S., Voduc, D., Vickery, T., Davies, S., Fauron, C., He, X., Hu, Z., et al. (2009). Supervised risk predictor of breast cancer based on intrinsic subtypes. *Journal of clinical oncology*, 27(8):1160.

Patel, S., Sihmar, S., and Jatain, A. (2015). A study of hierarchical clustering algorithms. In *2015 2nd international conference on computing for sustainable global development (INDIACom)*, pages 537–541. IEEE.

Plummer, M., Best, N., Cowles, K., and Vines, K. (2006). CODA: Convergence diagnosis and output analysis for MCMC. *R News*, 6(1):7–11.

Prat, A., Parker, J. S., Karginova, O., Fan, C., Livasy, C., Herschkowitz, J. I., He, X., and Perou, C. M. (2010). Phenotypic and molecular characterization of the claudin-low intrinsic subtype of breast cancer. *Breast cancer research*, 12(5):1–18.

Raftery, A. E. and Dean, N. (2006). Variable selection for model-based clustering. *Journal of the American Statistical Association*, 101(473):168–178.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850.

Richardson, S. and Green, P. J. (1997). On bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(4):731–792.

Roberts, G. and Smith, A. (1994). Simple conditions for the convergence of the gibbs sampler and metropolis-hastings algorithms. *Stochastic Processes and their Applications*, 49(2):207–216.

Rose, K., Gurewitz, E., and Fox, G. (1990). A deterministic annealing approach to clustering. *Pattern Recognition Letters*, 11(9):589–594.

Rousseau, J. and Mengersen, K. (2011). Asymptotic behaviour of the posterior distribution in overfitted mixture models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 73(5):689–710.

Ruffieux, H., Davison, A. C., Hager, J., Inshaw, J., Fairfax, B. P., Richardson, S., and Bottolo, L. (2020). A global-local approach for detecting hotspots in multiple-response regression. *The Annals of Applied Statistics*, 14(2):905 – 928.

School, B. . W. H. . H. M., Lynda, C., J., P. P., Raju, K., data analysis: Baylor College of Medicine Creighton Chad J., G., A., D. L., , , for Systems Biology Reynolds Sheila, I., Kreisberg Richard B., Bernard Brady, B. R. E. T. L. J. T. V. Z. W. S. I., et al. (2012). Comprehensive molecular portraits of human breast tumours. *Nature*, 490(7418):61–70.

Sheehan, K. M., Calvert, V. S., Kay, E. W., Lu, Y., Fishman, D., Espina, V., Aquino, J., Speer, R., Araujo, R., Mills, G. B., et al. (2005). Use of reverse phase protein microarrays and reference standard development for molecular network analysis of metastatic ovarian carcinoma. *Molecular & Cellular Proteomics*, 4(4):346–355.

Sørlie, T., Tibshirani, R., Parker, J., Hastie, T., Marron, J. S., Nobel, A., Deng, S., Johnsen, H., Pesich, R., Geisler, S., et al. (2003). Repeated observation of breast tumor subtypes in independent gene expression data sets. *Proceedings of the national academy of sciences*, 100(14):8418–8423.

Steinley, D. and Brusco, M. J. (2008). Selection of variables in cluster analysis: An empirical comparison of eight procedures. *Psychometrika*, 73(1):125–144.

Stephens, M. (2000). Bayesian analysis of mixture models with an unknown number of components—an alternative to reversible jump methods. *The Annals of Statistics*, 28(1):40 – 74.

Swartz, M. D., Mo, Q., Murphy, M. E., Lupton, J. R., Turner, N. D., Hong, M. Y., and Vannucci, M. (2008). Bayesian variable selection in clustering high-dimensional data with substructure. *Journal of Agricultural, Biological, and Environmental Statistics*, 13(4):407–423.

Tadesse, M. G., Sha, N., and Vannucci, M. (2005). Bayesian variable selection in clustering high-dimensional data. *Journal of the American Statistical Association*, 100(470):602–617.

Ueda, N. and Nakano, R. (1998). Deterministic annealing EM algorithm. *Neural networks : the official journal of the International Neural Network Society*, 11(2):271–282.

Uhlen, M., Zhang, C., Lee, S., Sjöstedt, E., Fagerberg, L., Bidkhori, G., Benfeitas, R., Arif, M., Liu, Z., Edfors, F., et al. (2017). A pathology atlas of the human cancer transcriptome. *Science*, 357(6352):eaan2507.

Wang, L. and Dunson, D. B. (2011). Fast bayesian inference in dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 20(1):196–216.

Ward Jr, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244.

Weinstein, J. N., Collisson, E. A., Mills, G. B., Shaw, K. R., Ozenberger, B. A., Ellrott, K., Shmulevich, I., Sander, C., and Stuart, J. M. (2013). The cancer genome atlas pan-cancer analysis project. *Nature genetics*, 45(10):1113–1120.

Witten, D. M. and Tibshirani, R. (2010). A framework for feature selection in clustering. *Journal of the American Statistical Association*, 105(490):713–726.

Yau, C. and Holmes, C. (2011). Hierarchical Bayesian nonparametric mixture models for clustering with variable relevance determination. *Bayesian Analysis*, 6(2):329 – 351.

Zhang, X., Nott, D. J., Yau, C., and Jasra, A. (2014). A sequential algorithm for fast fitting of dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 23(4):1143–1162.

Appendix A

Variational update equations

Given the word count limit, we provide only example derivations for the update of γ and δ in Section 3.2.2. For the mixture model parameters, Section 10.2.1 in (Bishop, 2006) presents similar derivations, but without variable selection. The complete mathematical derivation of our model can be found in the code directory provided in the Declaration and will be made publicly available as part of a publication for future work.

Updating γ_j

Starting from Equation (2.11), we write an expression for $\ln q^*(\gamma_j)$

$$\begin{aligned}
\ln q^*(\gamma_j) &= \mathbb{E}_{Z,\Phi,\pi,\delta}[\ln p(X,Z,\pi,\Phi,\gamma,\delta)] + const \\
&= \mathbb{E}_{Z,\Phi,\pi,\delta}[\ln(p(X|Z,\Phi,\gamma)p(Z|\pi)p(\pi)p(\Phi)p(\gamma|\delta)p(\delta))] + const \\
&= \mathbb{E}_{Z,\Phi,\delta}[\ln(p(X|Z,\Phi,\gamma)p(\gamma|\delta))] + const \\
&= \mathbb{E}_{\delta_j}[\ln p(\gamma_j|\delta_j)] + \mathbb{E}_{Z,\Phi}[\ln p(X|Z,\Phi,\gamma_j)] + const
\end{aligned}$$

where we keep only the terms that depend on γ_j. Note that using Equation (3.19), we write

$$\mathbb{E}_{\delta_j}[\ln p(\gamma_j|\delta_j)] = \gamma_j \mathbb{E}_{\delta_j}[\ln(\delta_j)] + (1-\gamma_j)\mathbb{E}_{\delta_j}[\ln(1-\delta_j)],$$

and using Equation (3.14) we write

$$\begin{aligned}
\mathbb{E}_{\Phi,Z}[\ln p(X|Z,\Phi,\gamma_j)] &= \mathbb{E}_Z[\mathbb{E}_\Phi[\sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}(\gamma_j \ln f_j(x_{nj}|\Phi_{kj}) + (1-\gamma_j)\ln f_j(x_{nj}|\Phi_{0j}))]] \\
&= \gamma_j \left(\sum_{n=1}^{N}\sum_{k=1}^{K} \mathbb{E}_Z[z_{nk}]\mathbb{E}_\Phi[\ln f_j(x_{nj}|\Phi_{kj})]\right) + (1-\gamma_j)\left(\sum_{n=1}^{N}\sum_{k=1}^{K} \mathbb{E}_Z[z_{nk}]\ln f_j(x_{nj}|\Phi_{0j})]\right),
\end{aligned}$$

Gathering terms in γ_j and $(1-\gamma_j)$, we rewrite $\ln q^*(\gamma_j)$ as

$$\ln q^*(\gamma_j) = \gamma_j \ln \eta_{1j} + (1-\gamma_j)\ln \eta_{2j} + const \tag{A.1}$$

where

$$\ln \eta_{1j} = \mathbb{E}_{\delta_j}[\ln(\delta_j)] + \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\mathbb{E}_{\Phi}[\ln f_j(x_{nj}|\Phi_{kj})], \tag{A.2}$$

$$\ln \eta_{2j} = \mathbb{E}_{\delta_j}[\ln(1-\delta_j)] + \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\ln f_j(x_{nj}|\Phi_{0j})]. \tag{A.3}$$

Exponentiating both sides, it follows that $q^*(\gamma_j) \propto \eta_{1j}^{\gamma_j}\eta_{2j}^{1-\gamma_j}$, and hence

$$q^*(\gamma_j) = c_j^{\gamma_j}(1-c_j)^{1-\gamma_j}, \tag{A.4}$$

where $c_j = \frac{\eta_{1j}}{\eta_{1j}+\eta_{2j}} = \mathbb{E}_{\gamma}[\gamma_j]$.

Updating δ_j

Similarly, we consider δ_j, and use Equation (2.11) to write down an expression for $\ln q^*(\delta_j)$:

$$\begin{aligned}
\ln q^*(\delta_j) &= \mathbb{E}_{Z,\Phi,\pi,\gamma}[\ln p(X,Z,\pi,\Phi,\gamma,\delta)] + const \\
&= \mathbb{E}_{Z,\Phi,\pi,\gamma}[\ln(p(X|Z,\Phi,\gamma)p(Z|\pi)p(\pi)p(\Phi)p(\gamma|\delta)p(\delta))] + const \\
&= \mathbb{E}_{\gamma}[\ln(p(\gamma_j|\delta_j)\ln p(\delta_j))] + const \\
&= \mathbb{E}_{\gamma}[\gamma_j \ln\delta_j + (1-\gamma_j)\ln(1-\delta_j)] + (d_0-1)(\ln\delta_j + \ln(1-\delta_j)) + const \\
&= (c_j + d_0 - 1)\ln\delta_j + (1-c_j+d_0-1)\ln(1-\delta_j) + const
\end{aligned}$$

where we keep only the terms that depend on δ_j and used Equations (3.19)-(3.20) for the probability distributions over γ_j and δ_j. We exponentiate $\ln q^*(\delta_j)$ to give

$$q^*(\delta_j) = \text{Beta}(c_j + d_0, 1 - c_j + d_0) \tag{A.5}$$

From properties of Beta distributions, it follows that

$$\mathbb{E}_{\delta_j}[\ln\delta_j] = \psi(c_j + d_0) - \psi(2d_0+1) \tag{A.6}$$

$$\mathbb{E}_{\delta_j}[\ln(1-\delta_j)] = \psi(1 - c_j + d_0) - \psi(2d_0+1) \tag{A.7}$$

which are required to evaluate Equations (A.2) and (A.3).

Appendix B

Supporting material

Parameter initialisation

Experiment	K	α_0	m_{0j}	β_{0j}	a_{0j}	b_{0j}	d_0	c_j
Synthetic	[3,10]	[0.1, 1]	mean(X_j)	10^{-3}	3.	[0.1, 1]	0.9	[0.5, 1]
C-TCGA	[5,10]	0.01	mean(X_j)	10^{-3}	3.	[0.01, 1]	-	-
TCGA	[5, 8]	[0.01, 0.1]	mean(X_j)	10^{-3}	3.	[0.1, 1]	[1, 5]	1
A-TCGA	[5, 7]	$1/K$	mean(X_j)	10^{-3}	[3, 10]	[0.1, 1]	[0.9, 5]	1
TCPA	[25, 40]	10^{-3}	mean(X_j)	10^{-3}	3.	0.1	0.5	[0.8, 1]

Table B.1 Parameter initialisations. For some parameters we provide fixed values, for others a range of values that worked well. We omit z_{nk} and δ_j as we always sample them from the corresponding distributions.

Legend:

- K: maximum number of clusters in the overfitted mixture.
- α_0: concentration of the Dirichlet prior on the mixture weights π (Eq. (3.15))
- m_{0j} and β_{0j}: mean and shrinkage of the Gaussian conditional prior on the components mean μ_{kj} (Eq. (3.16))
- a_{0j} and b_{0j}: degrees of freedom and scale of the Gamma prior on the components precision τ_{kj} (Eq. (3.16))
- d_0: shape of the Beta prior on the covariate selection probabilities δ_j (Eq. (3.20))
- c_j: covariate selection indicator
- z_{nk}: cluster assignment

Additional visualisations

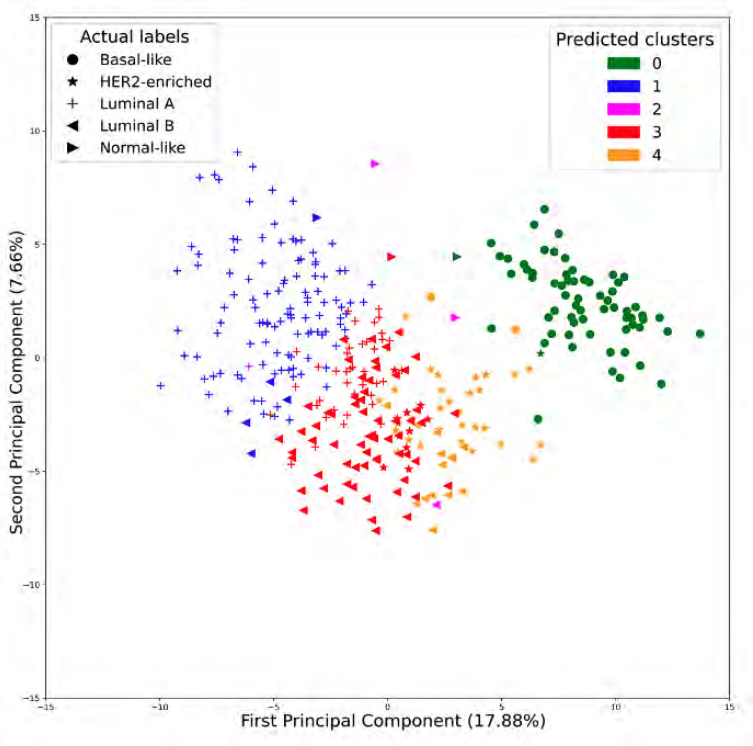


Fig. B.1 PCA plot of VBVarSel stratification on TCGA data using the PAM50 plus 100 genes.

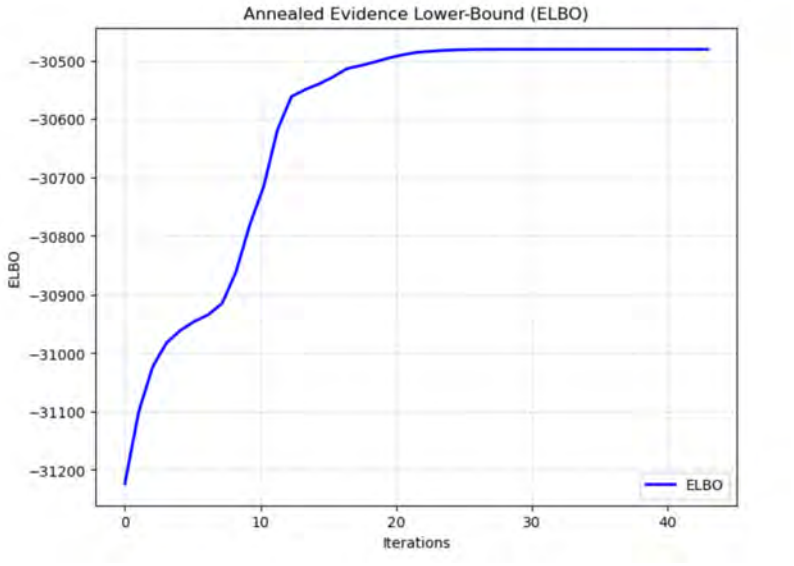


Fig. B.2 Annealed ELBO shape until convergence of VBVarSel on PAM50 set.

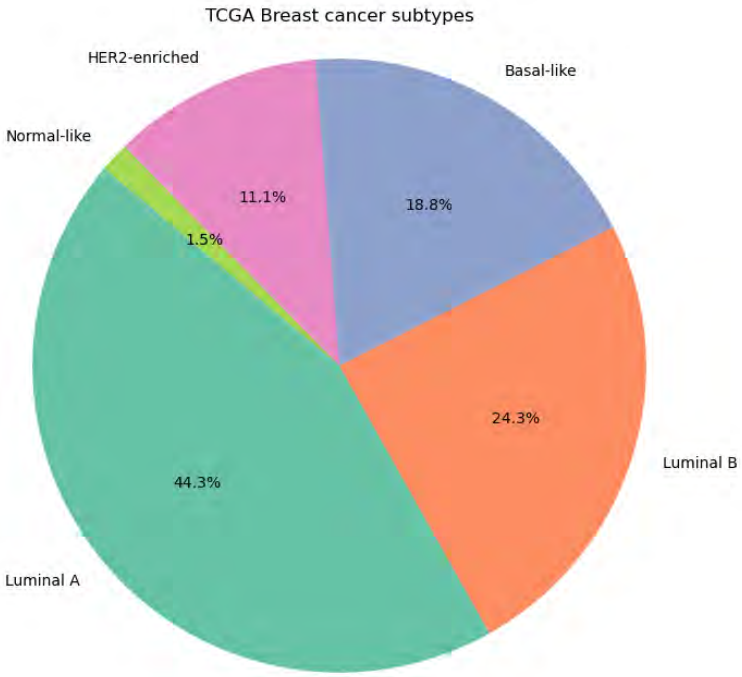


Fig. B.3 Pie chart of the breast cancer subtypes in TCGA.

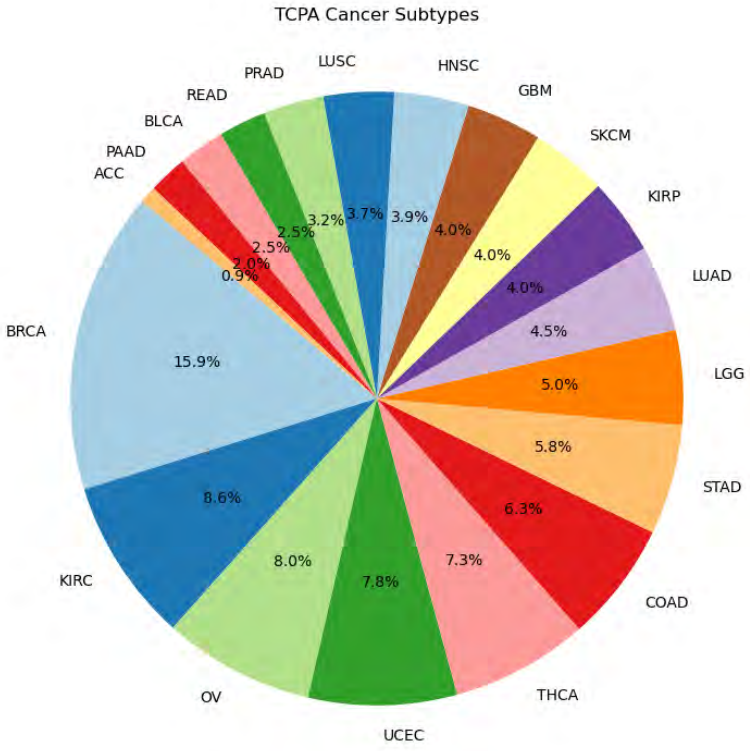


Fig. B.4 Pie chart of the cancer types in TCPA.

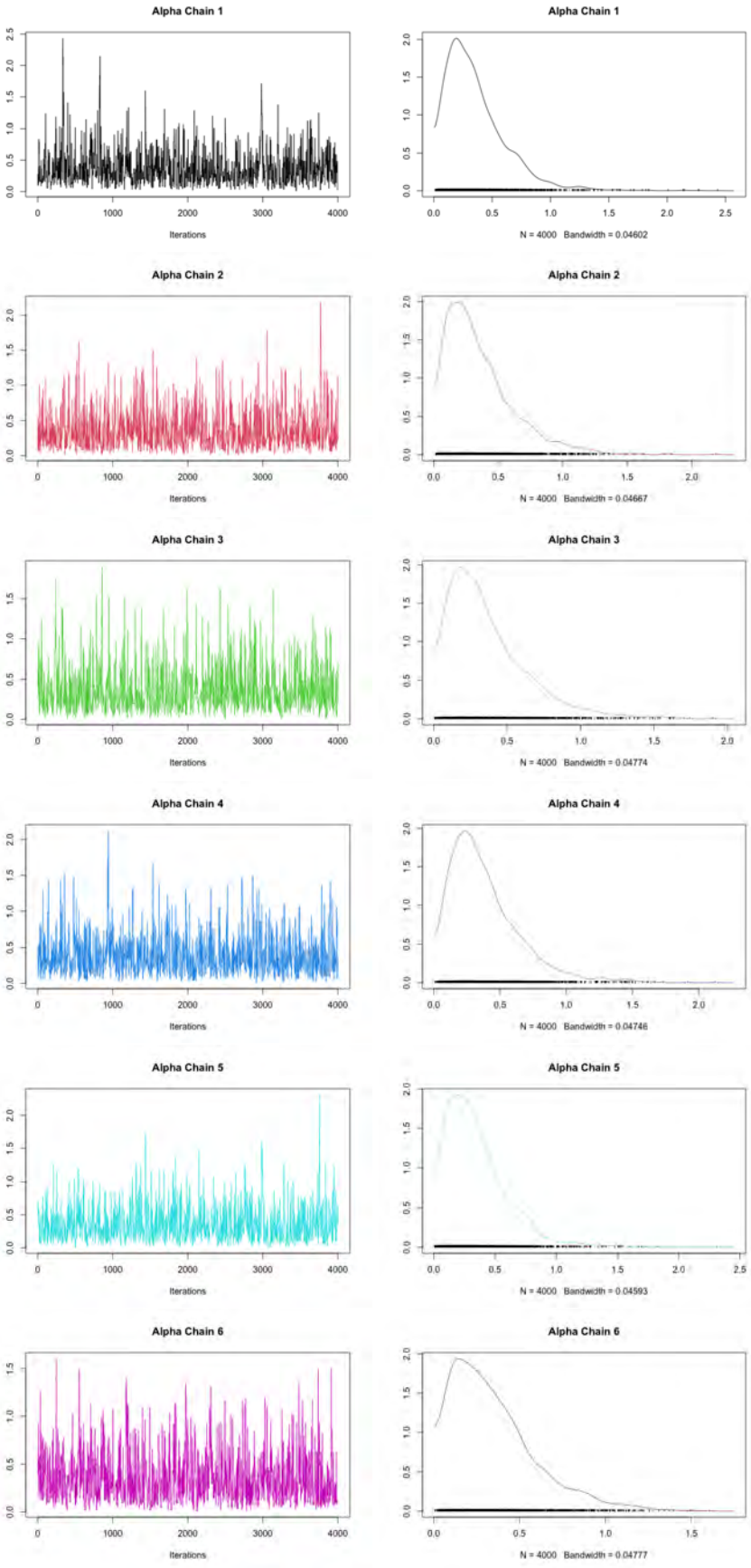


Fig. B.5 Trace (left) and density (right) of 6 PReMiuM (MCMC) chains of the Dirichlet concentration α.

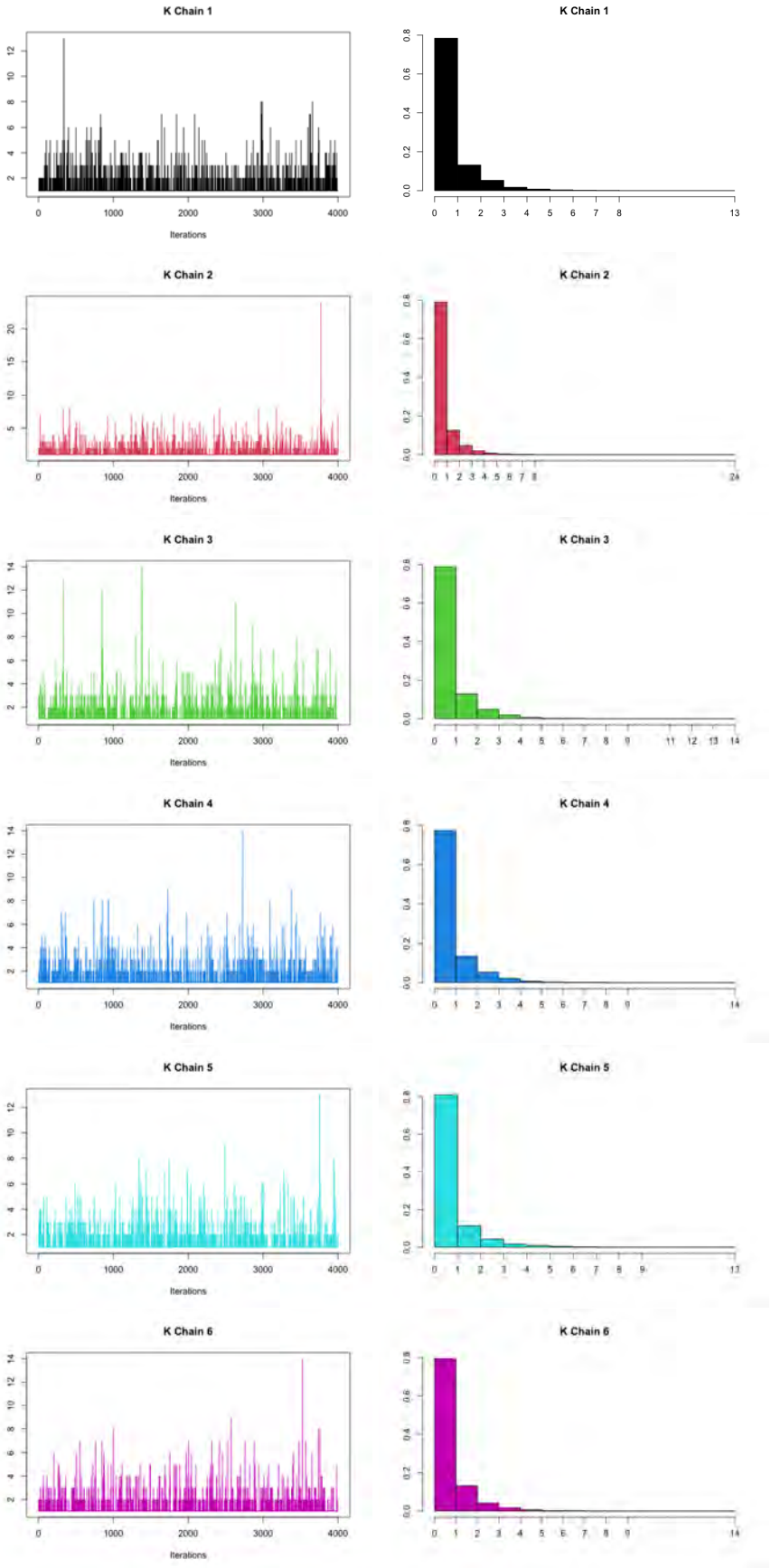
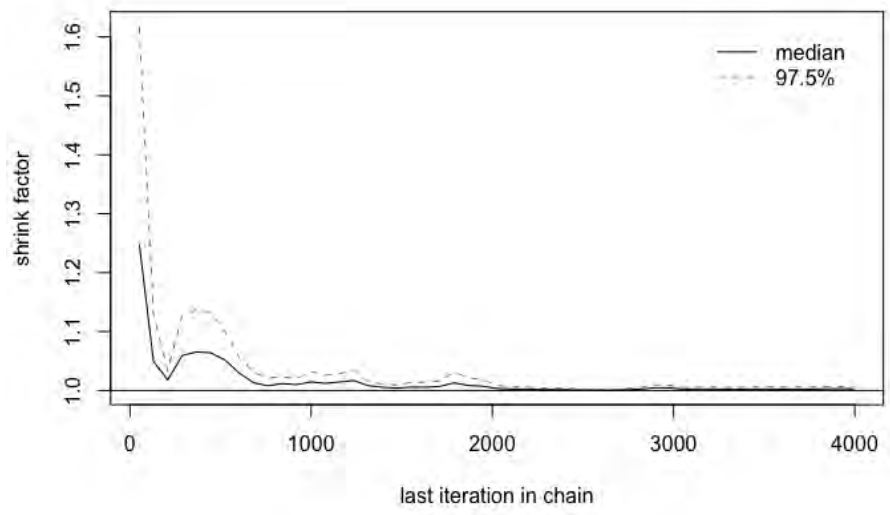
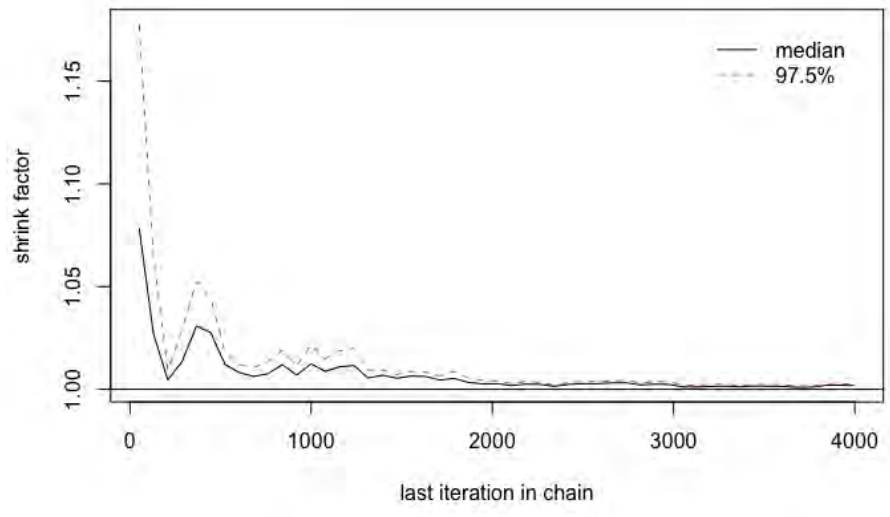


Fig. B.6 Trace (left) and density (right) of 6 PReMiuM (MCMC) chains of the mean number of clusters K.



(a) Alpha



(b) K

Fig. B.7 Gelman diagnostic for 6 PReMiuM (MCMC) chains. Gelman and Rubin (1992) suggests that chains with a factor < 1.2 are likely to have converged.