SCALABLE STATISTICAL METHODS FOR CELL TYPE DECONVOLUTION AND MIXED MODELS APPLIED TO HIGH DIMENSIONAL GENOMIC DATA

Hillary M. Heiling

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Biostatistics in the Gillings School of Global Public Health.

Chapel Hill
2023

Approved by:

Naim U. Rashid

Joseph G. Ibrahim

Quefeng Li

Donglin Zeng

Jen Jen Yeh

ABSTRACT

Hillary M. Heiling: Scalable Statistical Methods for Cell Type Deconvolution and Mixed Models
Applied to High Dimensional Genomic Data
(Under the direction of Naim U. Rashid and Joseph G. Ibrahim)

Utilizing genomic data in the clinical setting provides new opportunities for biomarker discovery, disease characterization, and personalizing treatment, but also poses new statistical challenges. In the first part of the dissertation, we propose a new computational method, IsoDeconvMM, which estimates cell type fractions using isoform-level RNA-seq gene expression data one gene at a time. The cell type composition of a tissue sample may itself be of interest and is needed for proper analysis of differential gene expression of heterogeneous tissues. Although a variety of existing computational methods estimate cell type proportions using gene-level expression data, isoform-level expression could be equally or more informative for determining cell type origin.

In genomics datasets as well as many other modern biomedical datasets, the data are increasingly high dimensional and exhibit complex correlation structures. Generalized linear mixed models (GLMMs) have long been employed to account for such dependencies. In the second part of this dissertation, we implement several statistical and computational innovations to improve the speed of a high dimensional penalized GLMM framework for simultaneously selecting fixed and random effects, resulting in the efficient R package glmmPen.

Although this framework extends the feasible dimensionality of GLMMs relative to existing methods, new methodology is needed to alleviate computational burden as the dimension increases and allow scalability to hundreds of predictors. We present a novel reformulation of the GLMM using a factor model decomposition of the random effects, enabling scalable computation of GLMMs in higher dimensions by reducing the latent space from a large number of random effects to a smaller set of common factors. We extend our prior work to estimate model parameters and perform simultaneous selection of fixed and random effects using a modified version of the Monte Carlo Expectation Conditional Minimization (MCECM) algorithm. We show that through this factor model decomposition, we can improve the speed and scalability of fitting high dimensional penalized GLMMs.

Finally, we extend our framework on performing high dimensional penalized generalized linear mixed models to survival outcome data. We approximate proportional hazards mixed effects models using piecewise constant hazards mixed effects survival models.

This dissertation is dedicated to my parents, Michael and Valarie Heiling, for the encouragement and emotional support they have provided me throughout my entire life.

ACKNOWLEDGEMENTS

I would like to first thank my advisors, Dr. Naim Rashid and Dr. Joseph Ibrahim. They have taught me how to think critically about statistical research problems and how to implement theoretical methods into computational procedures. They have also helped steer me towards interesting and meaningful research projects.

I next want to thank Dr. Quefeng Li, who has acted as my unofficial third advisor in many ways. His knowledge and advice were indispensable to our work and greatly appreciated.

I want to thank both Allison Deal and Dr. Rashid for helping me figure out what I want to do with my life after graduation. While supervising me in my role as a graduate research assistant at the Lineberger Comprehensive Cancer Center, Allison taught me how to be a better statistician and collaborator; through this experience, I learned that I want to be just like Allison when I grow up. Dr. Rashid has encouraged my interests, and has helped me improve my statistical and communication skills.

I want to thank my parents for always encouraging me in my educational pursuits and for helping me in countless other ways.

Finally, I want to thank my friends who have provided emotional and educational support to me throughout my time at the University of North Carolina at Chapel Hill. These friends include, but are not limited to, Bonnie Shook-Sa, Ethan Alt, Nate Bean, Kimberly Enders, and Ann Marie Weideman.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BIC	Bayesian information criterion
BIC-ICQ	Bayesian information criterion incorporating the ICQ-based selection procedure as described in Ibrahim *et al.* (2011)
BICh	Hybrid Bayesian information criterion defined by Delattre *et al.* (2014)
BN	Bai and Ng (2002) method for estimation of number common factors
E-step	Expectation step of an Expectation-Minimization algorithm
ER	Eigenvalue Ratio: method for estimation of number common factors
FP	False positives
GR	Growth Ratio: method for estimation of number common factors
GLMM	Generalized linear mixed model
M-step	Minimization step of an Expectation-Minimization algorithm
MCECM	Monte Carlo expectation conditional minimization
MCMC	Markov chain Monte Carlo
pGLMM	Penalized generalized linear mixed model
PHMM	Proportional hazard mixed model
POET	Principal Orthogonal complEment Thresholding
TP	True positives

CHAPTER 1: LITERATURE REVIEW

1.1 Cell type composition

1.1.1 Introduction

RNA sequencing data derived from human tissue samples are often mixtures of several different cell types. It is often of interest to quantify the relative abundance of each constituent cell type found within a tissue sample. In some cases, the relative abundances themselves contain relevant information for the main goal of a study. For example, the relative abundance of different types of immune cells within tumor samples can be used to predict patients' response to cancer immunotherapy (Becht *et al.*, 2016b). In other cases, abundance profiles are crucial for proper cell type-specific differential expression analyses (Li and Wu (2019); Jin *et al.* (2020)). Cell-sorting and other physical separation techniques exist to partition tissue samples into purified samples of their constituent cell populations, but such methods can be costly and may even induce changes to the cellular environment which can impact expression profiles (Shen-Orr *et al.* (2010)). As an alternative to physical separation methods, the development of statistical models for the deconvolution of expression profiles from tissue samples has become an active area of research.

1.1.2 Existing methods for cell type composition

Deconvolution methods can generally be categorized as reference-based or reference-free methods. Reference-based deconvolution methods, such as CIBERSORTx (Newman *et al.*, 2015) and Houseman's CP/QP (Houseman *et al.*, 2012), require purified samples for each cell type. In contrast, reference-free methods such as RefFreeEWAS (Houseman *et al.*, 2014) and surrogate variable analysis (SVA) (Leek and Storey, 2007) do not require such purified samples. Here, we focus on referenced-based methods because they have the benefit of allowing for the estimation of cell type proportions at an individual sample level (Teschendorff and Zheng, 2017).

In silico expression deconvolution models can largely be separated into three main developments: ratio-based models, linear models, and infiltration scores. Ratio-based models rely upon computing expression ratios between a mixed expression profile and a "gold standard" reference for a single cell type. The minimum of these ratios across genes roughly approximates the proportion of the referent cell type (Gosink *et al.* (2007); Clarke *et al.* (2010); Wang *et al.* (2014)). These methods are often limited to study two cell types (e.g., tumor vs normal). The linear model and infiltration score approaches can handle more than two cell types. The linear model framework assumes that appropriately normalized mixture expressions can be modeled as a weighted summation of cell-type specific gene expression in two or more cell types (Lu *et al.* (2003); Gong and Szustakowski (2013); Newman *et al.* (2015); Zhong *et al.* (2013)). The infiltration scores approach aim to estimate unitless quantities designed to reflect the abundance each constituent cell type (Becht *et al.* (2016a); Li *et al.* (2016)).

1.1.3 Alternative splicing

Existing methods have been designed to utilize gene-level expression only. Thus, appropriate deconvolution requires that cell types express differently at the gene level. In the case of highly similar cell types, however, it may be the case that gene-level expression differences are minimal. An alternative is to quantify gene expression at a more granular level: isoform expression. Each gene in human genome is often composed by multiple exons separated by introns, and one gene may produce multiple distinct transcripts by taking different combinations of exons. This process, known as alternative splicing, allow a single gene to encode multiple proteins and thus greatly increases the biodiversity of proteins that can be encoded by the genome. More than 90% of human genes could undergo alternative splicing (Wang *et al.* (2008)). Because cell types are often defined through the expression of proteins, the isoform level expression could be more sensitive to cell type identity than higher-level gene expression that is often the summation of gene expression across multiple isoforms.

1.1.4 Exon sets

Later in this document, we outline the development of a statistical model named IsoDeconvMM for expression deconvolution in mixture tissues by exploiting isoform-level expression differences between cell types. This method requires us to summarize read counts at an exon set level. In this section, we define an exon set in the same way as Sun *et al.* (2015), and we provide an example for illustration.

Consider a hypothetical gene composed of m non-overlapping exons that are utilized by I isoforms, or distinct mRNA transcripts formed by unique combinations of these exons. As specified in the gene models, the locations of these exons within the gene are known as are the identities and compositions of all isoforms used by this gene. We define the read count at any exon set A as the number of reads which overlap each of the exons in A and only these exons.

To visualize the setup, consider the hypothetical gene displayed in Figure 1.1. This gene is composed of $E = 4$ exons. An exon set is defined as some subset of the exons, which for this hypothetical gene could include sets containing only a single exon, sets containing two of the four exons, sets containing three of the four exons, or the set with all four exons combined. Each RNA-Seq read from the gene maps to one and only one of the possible exon sets. If an RNA-Seq read maps to each exon in some exon set A and no other exons, we say it belongs to exon set A.

The gene in Figure 1.1 is composed if $I = 3$ isoforms. Suppose that isoforms 1, 2, and 3 compose the set of all isoforms used by the gene and that their structure with respect to the exons is as given in the figure. Consider the exon set $A := \{1, 2, 3\}$. The read count at A is defined as the number of RNA-Seq reads which, when mapped, overlap exons 1, 2, and 3 but do not overlap exon 4.

Identifying the exon set to which an RNA-Seq read belongs gives us insight into the isoform to which the read belongs. Although a gene is composed of $(2^E - 1)$ possible exon sets, the exon

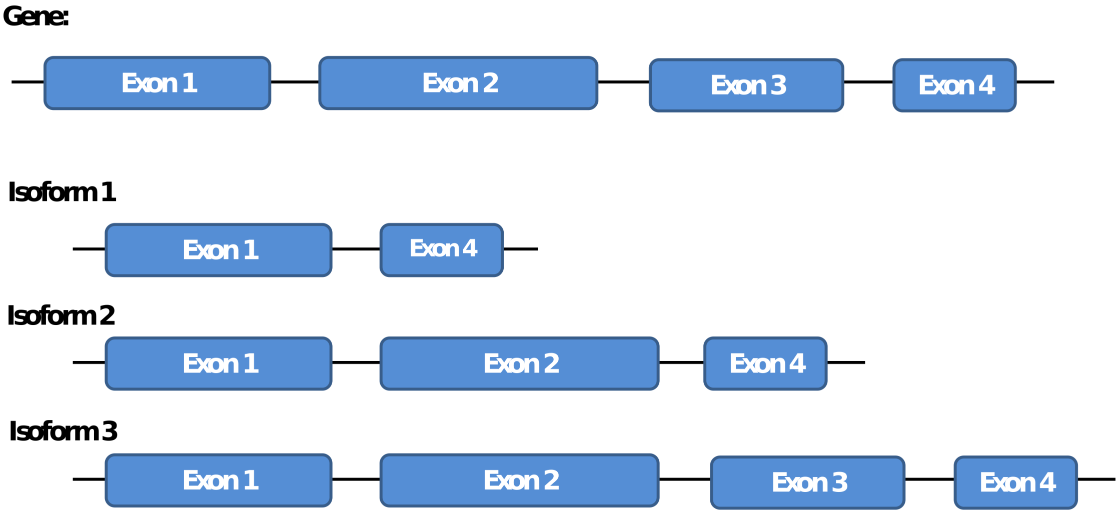


Figure 1.1: Hypothetical gene and isoform construction model.

sets possible for each of the isoforms can be restricted. In this hypothetical example, isoforms 1 and 2 do not contain exon 3, so none of the exon sets containing exon 3 are possible for isoforms 1 and 2. Which exon sets are theoretically possible for each of the three isoforms of this gene is provided in Table 1.1.

In some cases, two exons of a gene overlap partially. When this happens, we handle the situation similar to Sun *et al.* (2015). We split the two exons into three exons: the two non-overlapping sections unique to a particular exon and the overlapping section belonging to both exons. It is also possible for multiple genes to overlap one or more exons, and we consider these overlapping genes as a transcript cluster.

Table 1.1: The exon sets available for each of the three isoforms from the hypothetical gene in Figure 1.1. Value of 1 indicates that a paired-end read could theoretically maps to that exon set given that the read comes from the isoform specified; value of 0 otherwise.

Exon Set	Isoform 1	Isoform 2	Isoform 3
$\{E_1\}$	1	1	1
$\{E_2\}$	0	1	1
$\{E_3\}$	0	0	1
$\{E_4\}$	1	1	1
$\{E_1, E_2\}$	0	1	1
$\{E_1, E_3\}$	0	0	1
$\{E_1, E_4\}$	1	1	1
$\{E_2, E_3\}$	0	0	1
$\{E_2, E_4\}$	0	1	1
$\{E_3, E_4\}$	0	0	1
$\{E_1, E_2, E_3\}$	0	0	1
$\{E_1, E_2, E_4\}$	0	1	1
$\{E_1, E_3, E_4\}$	0	0	1
$\{E_2, E_3, E_4\}$	0	0	1
$\{E_1, E_2, E_3, E_4\}$	0	0	1

1.2 Variable selection in generalized linear mixed models

1.2.1 Introduction to generalized linear mixed models

Generalized Linear Mixed Models (GLMMs) are widely used in scientific research, with applications spanning the social sciences (Schmidt-Catran and Fairbrother, 2016), biomedical sciences (Fitzmaurice *et al.*, 2012), and public health (Szyszkowicz, 2006; Kleinman *et al.*, 2004). GLMMs are generalized linear models where the linear predictor contains both “fixed” and “random” effects; the latter portion pertains to variables whose effects are presumed to vary randomly across “groups” of observations within the data, leading to group-specific effect estimates (Fitzmaurice *et al.*, 2012). In practical applications, these “groups” may pertain to clusters of samples, repeated measures within the same individual, or observations resulting from nested designs. Multiple studies have shown that omitting important random effects leads to bias in the estimated variance of the fixed effects, and including unnecessary random effects could lead to computational difficulties (Thompson *et al.*, 2017; Gurka *et al.*, 2011; Bondell *et al.*, 2010). As a result, proper specification of fixed and random effects is an important and critical step in the application of GLMMs.

1.2.2 Existing methods for variable selection in GLMMs

Although proper specification of fixed and random effects is important, it is often unknown *a priori* which variables should be specified as fixed or random in the model. In higher dimensional settings, the feature space may also be sparse with many variables unrelated to the outcome. Therefore, variable selection approaches are employed to evaluate candidate models. **R** packages such as **lme4** (Bates *et al.*, 2015), **mcemGLM** (Archila, 2020), and **MCMCglmm** (Hadfield, 2010) allow users to fit a set of pre-specified models, which can then be compared using model selection criteria such as the profile conditional AIC (Donohue *et al.*, 2011), the BIC-ICQ criterion (Ibrahim *et al.*, 2011), the hybrid BICh criterion (Delattre *et al.*, 2014), or other similar criteria (see additional details in Section 1.2.3). However, all-subsets selection or direct model comparison strategies are not feasible even for small dimensions, as with p predictors there are 2^{2p} possible combinations of fixed and random effects to be evaluated. Packages such as **glmnet** (Friedman *et al.*, 2010), **ncvreg** (Breheny and Huang, 2011), and **grpreg** (Breheny and Huang, 2015) avoid this limitation for GLMs via coordinate-descent based penalized likelihood methods for variable selection, and are therefore much more scalable. Unfortuntely, none of these methods can account for random effects during their variable selection procedure. Other packages such as **glmmLASSO** (Groll, 2017) and **glmmixedLASSO** (Schelldorfer *et al.*, 2014) alternatively allow the inclusion of random effects in the model while performing variable selection, but only allow for variable selection on the fixed effects. Prior work has shown that simultaneous selection of fixed and random effects is advantageous because improper specification of the random effects can significantly affect the selection of the fixed effects, and vice versa (Bondell *et al.*, 2010). In addition, there may not be *a priori* knowledge of which variables may vary randomly across groups in their effects. Therefore specification of random effects may be difficult in practical applications, particularly as the dimension of the data grows.

1.2.3 BIC-type model selection criteria for mixed models

As mentioned above, there are several existing model selection criteria that have been used to select the best mixed model from candidate models. In this section, we discuss in more depth the calculation of BIC-type model selection criteria for mixed models.

We first introduce some relevant notation. We consider the case where we want to analyze data from K independent groups of any kind. For instance, we could be interested in analyzing data from K different studies, or we could be interested in analyzing longitudinal data from K individuals. For each group $k = 1, ..., K$, there are n_k observations for a total sample size of $N = \sum_{k=1}^{K} n_k$. For the k^{th} group, let $\boldsymbol{y}_k = (y_{k1}, ..., y_{kn_k})^T$ be the vector of n_k independent responses, let $\boldsymbol{x}_{ki} = (x_{ki,1}, ..., x_{ki,p})^T$ be the p-dimensional vector of predictors, and let $\boldsymbol{X}_k = (\boldsymbol{x}_{k1}, ..., \boldsymbol{x}_{kn_k})^T$. In GLMMs, we assume that the conditional distribution of $\boldsymbol{y}_k$ given $\boldsymbol{X}_k$ belongs to the exponential family and has the following density:

$$f(\boldsymbol{y}_k|\boldsymbol{X}_k, \boldsymbol{\alpha}_k; \theta) = \prod_{i=1}^{n} c(y_{ki}) \exp\big[\tau^{-1}\{y_{ki}\eta_{ki} - b(\eta_{ki})\}\big], \tag{1.1}$$

where $c(y_{ki})$ is a constant that only depends on y_{ki}, τ is the dispersion parameter, $b(\cdot)$ is a known link function, η_{ki} is the linear predictor, and $\boldsymbol{\alpha}_k$ is a q-dimensional vector of unobservable random effects.

The traditional BIC criterion is specified below:

$$\text{BIC}(\boldsymbol{\theta}_\lambda) = -2 * \ell(\boldsymbol{\theta}_\lambda) + d_\lambda * \log(N),$$

where $\boldsymbol{\theta}_\lambda$ are the coefficients of the penalization model, $\ell(\boldsymbol{\theta}_\lambda)$ is the marginal log-likelihood for the model, d_λ is the number of nonzero coefficients for the model, and N can be either the total number of observations in the data (N_{obs}) or the total number of independent observations (i.e. number of levels within the grouping factor, N_{grps}) in the data. The marginal log-likelihood is as follows:

$$\ell(\boldsymbol{\theta}) = \sum_{k=1}^{K} \ell_k(\boldsymbol{\theta}) = \frac{1}{n_k} \log \int f(\boldsymbol{y}_k | \boldsymbol{X}_k, \boldsymbol{\alpha}_k; \boldsymbol{\theta}) \phi(\boldsymbol{\alpha}_k) d\boldsymbol{\alpha}_k, \tag{1.2}$$

where $\phi(\boldsymbol{\alpha}_k)$ is the prior for the random effects.

The use of $\log(N_{obs})$ or $\log(N_{grps})$ in the BIC penalty term is related to the debate on the definition of the sample size in mixed models: the number of total observations in the data, or the total number of independent units (i.e. number of groups in the data). There has not traditionally been a consensus on what version to use, with different software using different versions. For instance, the $\log(N_{obs})$ penalty is used in the R package **nlme** (Pinheiro *et al.*, 2017), and the $\log(N_{grp})$ penalty is used in SAS proc NLMIXED (SAS Institute Inc., 2008). In practice, the performance of the different versions of the BIC penalty term may depend on the true underlying model (Lorah and Womack, 2019; Delattre *et al.*, 2014), with Delattre et al. (2014) observing that the $\log(N_{obs})$ penalty performed better when the true model had very few random components, and the $\log(N_{grp})$ penalty performed better when the true model had a large number of random components. Both Delattre et al. (2014) and Lorah and Womack (2019) suggest using some combination of these sample size definitions.

Specifically, Delattre *et al.* (2014) suggested using a 'hybrid' BICh selection criteria to select the best model, defined below:

$$\text{BICh}(\boldsymbol{\theta}_\lambda) = -2 * \ell(\theta_\lambda) + d_{\lambda,f} * \log(N_{obs}) + d_{\lambda,r} * \log(N_{grps}), \tag{1.3}$$

where $d_{\lambda,f}$ and $d_{\lambda,r}$ are the number of nonzero fixed and random effect coefficients, respectively.

The calculation of the BIC and BICh criteria require a calculation of the marginal log-likelihood $\ell(\boldsymbol{\theta})$ for each model. In mixed models, the calculation of the marginal likelihood is complicated by the fact that the integrals within $\ell(\boldsymbol{\theta})$ are generally intractable. Some methods for the estimation of this marginal log-likelihood are only practical in lower dimensions, such as the Laplace estimation used in the **lme4** package (Bates *et al.*, 2015). There have been several other proposed marginal likelihood estimates that utilize output from Markov Chain Monte Carlo (MCMC) posterior samples and are more appropriate for higher dimensions. One general class of such methods includes importance sampling methods that use MCMC samples to inform appropriate importance sample distributions, such as the Corrected Arithmetic Mean Estimator (CAME) from Pajor (2017) and the methods by Crooks *et al.* (2007) and Heavens *et al.* (2017). Other methods directly use the MCMC chain results in their estimation procedure, such as the modified harmonic mean estimator (Chan and Grant, 2015), and the Chib method (Chib and Jeliazkov, 2001). As discussed in Fourment *et al.* (2020), there is a wide variation in the required calculation time and the accuracy of the many possible marginal likelihood estimates. Here, we discuss the estimation of the marginal log-likelihood using the CAME estimator described by Pajor (2017), which is relatively fast and easy to compute for our mixed model application and was shown to be accurate when compared against the **lme4** log-likelihood estimate in low dimensions (we considered the **lme4** method to be the gold standard for estimation of low-dimensional GLMMs; log-likelihood comparison results not presented in this paper).

To calculate the CAME, we focus on a single group k and define a set $A_k \subseteq \Theta$ as a subset of the parameter space of the random effects for group k, where $P(A_k)$ and $P(A_k|\boldsymbol{y}_k, \boldsymbol{X}_k; \boldsymbol{\theta})$ are nonzero probabilities. We first start with the knowledge

$$\begin{aligned} P(A_k|\boldsymbol{y}_k, \boldsymbol{X}_k; \boldsymbol{\theta}) &= \int_{A_k} \phi(\boldsymbol{\alpha}_k|\boldsymbol{y}_k, \boldsymbol{X}_k; \boldsymbol{\theta}) d\boldsymbol{\alpha}_k \\ &= \int_{\Theta} \frac{1}{f(\boldsymbol{y}_k|\boldsymbol{X}_k; \boldsymbol{\theta})} f(\boldsymbol{y}_k|\boldsymbol{X}_k, \boldsymbol{\alpha}_k; \boldsymbol{\theta}) \phi(\boldsymbol{\alpha}_k) I(\boldsymbol{\alpha}_k \in A_k) d\boldsymbol{\alpha}_k, \end{aligned} \tag{1.4}$$

where I(.) is an indicator function, $f(\boldsymbol{y}_k|\boldsymbol{X}_k; \boldsymbol{\theta}) = \int f(\boldsymbol{y}_k|\boldsymbol{X}_k, \boldsymbol{\alpha}_k; \boldsymbol{\theta}) \phi(\boldsymbol{\alpha}_k) d\boldsymbol{\alpha}_k$ is the marginal likelihood for group k, and all other terms are described earlier in this section. The above relationship allows us to obtain the result:

$$\begin{aligned} f(\boldsymbol{y}_k|\boldsymbol{X}_k; \boldsymbol{\theta}) &= \frac{1}{P(A_k|\boldsymbol{y}_k, \boldsymbol{X}_k; \boldsymbol{\theta})} \int_{\Theta} f(\boldsymbol{y}_k|\boldsymbol{X}_k, \boldsymbol{\alpha}_k; \boldsymbol{\theta}) \phi(\boldsymbol{\alpha}_k) I(\boldsymbol{\alpha}_k \in A_k) d\boldsymbol{\alpha}_k \\ &= \frac{1}{P(A_k|\boldsymbol{y}_k, \boldsymbol{X}_k; \boldsymbol{\theta})} \int_{\Theta} \frac{f(\boldsymbol{y}_k|\boldsymbol{X}_k, \boldsymbol{\alpha}_k; \boldsymbol{\theta}) \phi(\boldsymbol{\alpha}_k) I(\boldsymbol{\alpha}_k \in A_k) s(\boldsymbol{\alpha}_k) d\boldsymbol{\alpha}_k}{s(\boldsymbol{\alpha}_k)}, \end{aligned} \tag{1.5}$$

where $s(.)$ is an importance sampling function.

Suppose we obtain M samples from the posterior distribution of the random effects for group k, $\tilde{\boldsymbol{\alpha}}_k = ((\boldsymbol{\alpha}_k^{(1)})^T, ..., (\boldsymbol{\alpha}_k^{(M)})^T)^T$. Let us set $A_k = \tilde{\boldsymbol{\alpha}}_k$; this reduces $P(A_k|\boldsymbol{y}_k, \boldsymbol{X}_k; \boldsymbol{\theta})$ to 1. For practical purposes, we can define the importance sampling function $s(.)$ using the posterior samples $\boldsymbol{\alpha}_k^{(m)}$ for $m = \{1, ..., M\}$. For instance, we could set $s(.)$ to a multivariate normal distribution with a mean vector equal to the mean of the posterior samples $\frac{1}{M}\sum_{m=1}^{M} \boldsymbol{\alpha}_k^{(m)}$ and a covariance matrix equal to the covariance matrix of a thinned subset of the posterior samples. If we draw $M^\star$ samples $\boldsymbol{\alpha}_k^\star = ((\boldsymbol{\alpha}_k^{\star(1)})^T, ..., (\boldsymbol{\alpha}_k^{\star(M^\star)})^T)^T$ from this importance sampling function, then equation (1.5) indicates that we can estimate the marginal likelihood for group k as

$$f(\boldsymbol{y}_k|\boldsymbol{X}_k; \boldsymbol{\theta}) \approx \frac{1}{M^\star} \sum_{m=1}^{M^\star} \frac{f(\boldsymbol{y}_k|\boldsymbol{X}_k, \boldsymbol{\alpha}_k^{\star m}; \boldsymbol{\theta})\phi(\boldsymbol{\alpha}_k^{\star m}) I(\boldsymbol{\alpha}_k^{\star m} \in A_k)}{s(\boldsymbol{\alpha}_k^{\star m})}. \tag{1.6}$$

We can then repeat the estimation in (1.6) for all K groups in order to calculate the full desired marginal log-likelihood $\ell(\boldsymbol{\theta})$. This final marginal log-likelihood can then be used in the previously mentioned BIC and BICh calculations for each candidate model of interest. We refer to this marginal log-likelihood as the Pajor log-likelihood throughout the remainder of the paper.

As an alternative to the BIC or BICh selection criteria, we could instead use the BIC-ICQ criterion (Ibrahim *et al.*, 2011) for model selection. This BIC-ICQ criterion is calculated by first fitting a 'full' model with either no penalty or a small penalty on the coefficients. The BIC-ICQ is expressed below:

$$\begin{aligned}\text{BICq}(\boldsymbol{\theta}_\lambda) &= 2\{Q_\lambda(\boldsymbol{\theta}_\lambda|\boldsymbol{\alpha}_0)\} + d_\lambda * \log(N) \\ &\approx \left\{ -\frac{2}{M} \sum_{m=1}^{M} \sum_{k=1}^{K} \left[\log f(\boldsymbol{y}_k|\boldsymbol{X}_k, \boldsymbol{\alpha}_{0,k}^{(m)}; \boldsymbol{\theta}_\lambda) + \log \phi(\alpha_{0,k}^{(m)}) \right] \right\} + d_\lambda * \log(N),\end{aligned} \tag{1.7}$$

where $\boldsymbol{\theta}_\lambda$ are the coefficients of the penalized model, $\boldsymbol{\alpha}_0$ are the posterior samples from a 'full' model with either no penalty or a minimum penalty used on the fixed and random effects, and $\alpha_{0,k}^{(m)}$ is the m^{th} posterior sample for group k from such a full model, d_λ is the number of nonzero coefficients for the model (all nonzero fixed effects parameters $\boldsymbol{\beta}$ plus all nonzero random effects parameters $\boldsymbol{\gamma}$), N is the total number of observations in the data (N_{obs}), and Q_λ, the Q-function, is defined below and evaluated using the posterior samples from the full model:

$$Q_\lambda(\boldsymbol{\theta}|\boldsymbol{\alpha}_0) = \sum_{k=1}^{K} E\left\{-\log[f(\boldsymbol{y}_k, \boldsymbol{X}_k, \boldsymbol{\alpha}_{0,k}; \boldsymbol{\theta}|\boldsymbol{d}_o; \boldsymbol{\theta}^{(s)})]\right\} \tag{1.8}$$

1.3 Factor analysis as a tool in high dimensions

1.3.1 General factor analysis models

Factor analysis and the closely-related component analysis have long been employed as dimension reduction tools in a variety of areas. Such areas of application range from the behavioral sciences such as psychology and education (Jacobson *et al.*, 1996; Baynton, 1992) to economics and finance (Chamberlain and Rothschild, 1982; Fama and French, 1992; Bai and Ng, 2002) to genomics (Nazarov *et al.*, 2019). In this section, we review the common factor model and the full component model (Gorsuch, 2014) and how they can be used in dimension reduction.

To illustrate these models, we consider the case where we have observed outcomes y_{it} for features $i = \{1, ..., N\}$ over situations $t = \{1, ..., T\}$. These situations could apply to time points in time series analysis (Fan *et al.*, 2013; Bai and Ng, 2002) or to human subjects (Nazarov *et al.*, 2019). The common factor model is then defined as

$$\boldsymbol{y}_t = \boldsymbol{B}\boldsymbol{f}_t + \boldsymbol{\epsilon}_t, \tag{1.9}$$

where $\boldsymbol{y}_t = (y_{1t}, ..., y_{Nt})^T$ is the vector of observed features at situation t, $\boldsymbol{f}_t$ is an r-dimensional vector of common factors ($r < N$), $\boldsymbol{B}$ is the factor loadings matrix of dimension $N \times r$, and $\boldsymbol{\epsilon}_t$ are idiosyncratic components (i.e. error terms) that have the distribution $\boldsymbol{\epsilon}_t \sim N_N(\boldsymbol{0}, \boldsymbol{\Sigma}_\epsilon)$. In general, f_t is not required to have any particular distributional assumptions. However, we assume in this paper that f_t is normally distributed such that $f_t \sim N_r(\boldsymbol{0}, \boldsymbol{\Sigma}_f)$. The component model is similar to the factor model, but without the error terms:

$$\boldsymbol{y}_t = \boldsymbol{B}\boldsymbol{f}_t. \tag{1.10}$$

The component model assumes that the N outcomes $\boldsymbol{y}_t$ can be completely recreated from linear combinations of the r common factors of $\boldsymbol{f}_t$ such that $y_{it} = \boldsymbol{b}_i^T \boldsymbol{f}_t$ where $\boldsymbol{b}_i$ is an r-length vector of factor loadings (the i-th row of the loading matrix $\boldsymbol{B}$). In comparison, the common factor model incorporates additional sources of variation that are not attributable to the common factors (Gorsuch, 2014). In either case, we are effectively representing the N total features of interest with a lower dimensional set of common factors r with very little loss of total information.

There have been several publications that have compared and contrasted the use of the component analysis and factor analysis methods. Snook and Gorsuch (1989) examined the impact of assuming a component model when the true underlying model was the factor model. When comparing the numeric values of the loadings, they found that the loadings estimates from the component model were most biased compared with the factor model results when the number of features N was relatively small and/or when the proportion of variation that can be explained by the common factors is relatively small (i.e. the idiosyncratic error is high). Velicer and Jackson (1990) compared the correlations of several possible loadings estimates derived from the two methods using results from 9 different studies, and they found that the correlations were generally very high ($\geqslant 0.99$). They concluded that when the same number of r common factors/-components are used in the analysis, the component analysis and factor analysis give very similar results.

In both factor analysis and component analysis, the stability of the loadings estimates improves when there is a large ratio of the number of features N to the number of factors or components r (Fava and Velicer, 1992; Guadagnoli, 1984).

1.3.2 Factor analysis to estimate high dimensional covariance matrices

Several publications (Fan *et al.*, 2008, 2013; Tran *et al.*, 2020) have utilized factor analysis model assumptions to estimate high-dimensional covariance matrices. In this section, we first discuss some of the underlying assumptions of the method proposed by Fan *et al.* (2013), a method that allows for the factors to be unobserved (i.e. latent). Fan *et al.* (2013) assumes the following approximate factor model (equivalent to the previously mentioned common factor model in equation 1.9):

$$\boldsymbol{y}_t = \boldsymbol{B}\boldsymbol{f}_t + \boldsymbol{\epsilon}_t, \tag{1.11}$$

where $\boldsymbol{y}_t = (y_{1t}, ..., y_{Nt})^T$ is the N-dimensional vector of observed features for time point $t = \{1, ..., T\}$, $\boldsymbol{f}_t$ is an r-dimensional vector of unobserved latent factors, $\boldsymbol{B}$ is the factor loadings matrix, and $\boldsymbol{\epsilon}_t$ are idiosyncratic components (i.e. error terms) that are uncorrelated with $\boldsymbol{f}_t$.

Fan *et al.* (2013) make a conditional sparsity assumption where given the common factors, the outcomes are weakly correlated. The covariance matrix of the outcomes $\boldsymbol{y}_t$ can then be expressed as $\boldsymbol{\Sigma} = \boldsymbol{B}\text{cov}(\boldsymbol{f}_t)\boldsymbol{B}^T + \Sigma_\epsilon$, a combination of a low-rank matrix ($\boldsymbol{B}\text{cov}(\boldsymbol{f}_t)\boldsymbol{B}^T$, where $\text{cov}(\boldsymbol{f}_t) = \boldsymbol{\Sigma}_f$) and a sparse matrix ($\boldsymbol{\Sigma}_\epsilon$).

Some assumptions of this method are that both N and T diverge to infinity, the number of common factors r is fixed, and the first r eigenvalues of $\boldsymbol{\Sigma}$ are spiked and grow at rate $O(N)$. By assuming that the first r eigenvalues are spiked, the common components and the idiosyncratic components can be identified. They also assume without loss of generality that cov($\boldsymbol{f}_t$) = $\boldsymbol{I}_r$, the identity matrix. Consequently, $\Sigma = \boldsymbol{B}\boldsymbol{B}^T + \Sigma_\epsilon$.

Fan *et al.* (2013) propose that their method has wide applicability in statistical genetics. They cite Carvalho *et al.* (2008), who studied breast cancer hormonal pathways using a Bayesian sparse factor model. In their real-data results, Carvalho *et al.* (2008) identified two common factors that have highly loaded genes. This would translate to the gene expression's covariance matrix having one or two very spiked eigenvalues. Consequently, Fan *et al.* (2013) argue that their method could be applied to estimate such a covariance matrix.

Tran *et al.* (2020) utilize a similar factor model assumption for the parameterization of the covariance matrix of their model's posterior distribution, assuming that $\boldsymbol{\Sigma} = \boldsymbol{B}\boldsymbol{B}^T + \boldsymbol{D}^2$, where $\boldsymbol{B}$ is the same factor loading matrix described above and $\boldsymbol{D}$ is a diagonal $N \times N$ matrix that represents the standard deviations of the idiosyncratic noise.

1.3.3 Estimation of number of common factors

Performing factor model analyses requires specifying the number of common factors r. Since r is typically unknown *a priori*, this value needs to be estimated. There have been several proposed methods of estimating r for the approximate factor model given in equation 1.11 assuming high dimensions of features N and time series observations T. These methods include those developed by Bai and Ng (2002), Ahn and Horenstein (2013), Onatski (2010), and Kapetanios (2010). In this section, we will describe the Bai and Ng (2002) method, henceforth denoted BN, and the Eigenvalue Ratio (ER) and Growth Ratio (GR) methods developed by Ahn and Horenstein (2013).

1.3.3.1 Bai and Ng (BN) method

The BN method estimates r by

$$\widehat{r} = \text{arg min}_{0 \leqslant r_1 \leqslant M} \log \left\{ \frac{1}{NT} ||\boldsymbol{Y} - T^{-1} \boldsymbol{Y} \widehat{\boldsymbol{F}}_{r_1} \widehat{\boldsymbol{F}}'_{r_1} ||_F^2 \right\} + r_1 g(T, N), \tag{1.12}$$

where $\boldsymbol{Y}$ is the $N \times T$ matrix of the observed N features for all T time points, M is some prescribed upper bound, $\widehat{\boldsymbol{F}}_{r_1}$ is a $T \times r_1$ matrix whose columns are $\sqrt{T}$ times the eigenvectors corresponding to the r_1 largest eigenvalues of the $T \times T$ matrix $\boldsymbol{Y'Y}$, $||A||_F^2 = \text{tr}(A^T A)$, and $g(T, N)$ is a penalty function. Two penalty functions suggested by Bai and Ng (2002) are:

$$g_1(T, N) = \frac{N+T}{NT} \log \left(\frac{NT}{N+T} \right), \tag{1.13}$$

$$g_2(T, N) = \frac{N+T}{NT} \log(\min(N, T)). \tag{1.14}$$

Bai and Ng (2002) suggestion for the prescribed upper bound M was $M = 8{*}\text{int}[(\min(N, T)/100)^{1/4}]$.

1.3.3.2 Eigenvalue Ratio and Growth Ratio methods

Before we discuss the Eigenvalue Ratio (ER) and Growth Ratio (GR) methods proposed by Ahn and Horenstein (2013), we first introduce some additional notation. Let $\psi_k(A)$ be the k-th largest eigenvalue of the positive semidefinite matrix A, and let $\tilde{\mu}_{NT,k} \equiv \psi_k(\boldsymbol{Y}\boldsymbol{Y}^T/(NT)) = \psi_k(\boldsymbol{Y}^T\boldsymbol{Y}/(NT))$ where $\boldsymbol{Y}$ is again the $N \times T$ matrix of the observed N features for all T time points.

To find the ER and GR estimators, we first order the eigenvalues of $\boldsymbol{Y}\boldsymbol{Y}^T/(NT)$ from largest to smallest. The ER estimator estimates the number of factors r by taking the ratio of each pair of two adjacent eigenvalues up to the prescribed upper bound M,

$$ER(k) \equiv \frac{\tilde{\mu}_{NT,k}}{\tilde{\mu}_{NT,k+1}}, \quad k = 1, 2, ..., M \tag{1.15}$$

and the estimate of r is simply the maximizer of these ratios,

$$\widehat{r}_{ER} = \text{max}_{1 \leqslant k \leqslant M} ER(k). \tag{1.16}$$

The GR estimator estimates r by taking a ratio of the growth rates of residual variances as one fewer principal component is used in the time series regressions. The GR estimator calculates the following ratios:

$$GR(k) \equiv \frac{\log[V(k-1)/V(k)]}{\log[V(k)/V(k+1)]} = \frac{\log\big(1 + \tilde{\mu}^*_{NT,k}\big)}{\log\big(1 + \tilde{\mu}^*_{NT,k+1}\big)}, \quad k = 1, 2, ..., M \tag{1.17}$$

where $V(k) = \sum_{j=k+1}^{\min(N,T)} \tilde{\mu}_{NT,j}$ and $\tilde{\mu}^*_{NT,k} = \tilde{\mu}_{NT,k}/V(k)$. The estimate of r is then the maximizer of these ratios,

$$\widehat{r}_{GR} = \text{max}_{1 \leqslant k \leqslant M} GR(k) \tag{1.18}$$

If there exists *a priori* knowledge about the maximum value of r, r_{max}, then Ahn and Horenstein (2013) recommend setting the upper bound M to $M = 2r_{max}$. If there is no prior knowledge of r, then they recommend using $M = \min(M^{*}, 0.1 \min(N, T))$, where $M^{*} = \#\{k | \tilde{\mu}_{NT,k} \geqslant V(0)/\min(N, T), k \geqslant 1\}$.

1.4 Variable selection in models of survival data

In this section, we review existing models and methods of fitting survival data as well as existing methods used to performing variable selection in mixed effects survival data.

1.4.1 Piecewise constant hazard approximation to Cox proportional hazards model

Modeling survival outcomes has great clinical significance in medical and public health research. In particular, the Cox proportional hazards model has been widely utilized in order to characterize the relationship between treatments, exposures, or other covariates and patients' survival. The proportional hazards model can be approximated using the piecewise constant hazard survival model. In the piecewise constant hazard model, the follow-up time of the study is split into time intervals where the baseline hazard is assumed to be constant within these intervals (Friedman, 1982; Laird and Olivier, 1981; Holford, 1980; Rodriguez, 2010). This piecewise constant hazard model can be fit using a log-linear model which incorporates the duration of exposure within each interval. We outline the piecewise constant hazard model and its approximation to the proportional hazards model here, and we illustrate this model assuming fixed effects only (no random effects).

Suppose each of N subjects have the following observed data: observed times y_i for $i = \{1, ..., N\}$, where $y_i = \min(T_i, C_i)$, T_i represents the subject's event time, and C_i represents their censoring time; observed event indicators $\delta_i = I(T_i < C_i)$; and observed covariates $\boldsymbol{x}_i = (x_{i,1}, ..., x_{i,p})$.

Suppose we wish to fit the proportional hazards model

$$h_i(t|\boldsymbol{x}_i) = h_0(t)\exp\bigl(\boldsymbol{x}_i^T\beta\bigr), \tag{1.19}$$

where $h_i(t)$ represents the i-th individual's hazard at time t, $h_0(t)$ represents the baseline hazard at time t, and $\boldsymbol{\beta}$ are the p-dimensional set of hazard ratios corresponding to the p predictors in the model.

We can approximate (1.19) using a piecewise constant hazard model. We partition the follow-up time of the study into J intervals with cut points $0 = \tau_0 < \tau_1 < ... < \tau_J = \infty$, where the j-th interval is $[\tau_{j-1}, \tau_j)$. For each of these j intervals, we assume that the baseline hazard is constant within each interval such that

$$h_{ij} = h_j\exp\bigl(\boldsymbol{x}_i^T\beta\bigr), \tag{1.20}$$

where h_{ij} is the hazard corresponding to individual i in interval j, h_j is the baseline hazard for interval j, and $\exp\bigl(\boldsymbol{x}_i^T\beta\bigr)$ represents the relative risk for an individual compared to baseline at any given point.

For each interval, let us define analogous interval-specific measures of the observed times y_i and the event indicators δ_i, where t_{ij} is the time lived by subject i in the j-th interval $[\tau_{j-1}, \tau_j)$ and $d_{ij} = I(\tau_{j-1} \leqslant y_i < \tau_j, \delta_i = 1)$ is the indicator of whether the subject died in interval j (1 if true, 0 otherwise). Then, we can treat the death indicators d_{ij} as if they were independent Poisson observations with means

$$\mu_{ij} = t_{ij}h_{ij}, \tag{1.21}$$

allowing us to fit the data using the log-linear model

$$\log\mu_{ij} = \log t_{ij} + \psi_j + \boldsymbol{x}_i^T\beta, \tag{1.22}$$

where $\psi_j = \log h_j$ is the log of the baseline hazard for time interval j and the log of the time lived by the subject within interval j (defined as $\log t_{ij}$) is treated as an offset in the model.

Both the proportional hazards model and the piecewise constant hazard model can be extended to include mixed effects (Cortiñas Abrahantes and Burzykowski, 2005; Austin, 2017). In such models, the linear predictor contains both "fixed" and "random" effects, just as in generalized linear mixed models. See more details in Chapter 5.

1.4.2 Existing methods for variable selection in proportional hazards mixed models

In high dimensional settings, in which the feature space is generally assumed to be sparse, it is often unknown *a priori* which covariates should be specified as fixed or random in the model. Variable selection methods such as LASSO and SCAD exist for high dimensional proportional hazards models or frailty models (Tibshirani, 1997; Bradic *et al.*, 2011; Simon *et al.*, 2011; Fan and Li, 2002), but they do not allow for the selection of random effects. Several mixed effects model selection methods that rely on the specification of candidate models have been proposed, including likelihood ratios, profile Akaike information criterion (AIC) (Xu *et al.*, 2009), and conditional AIC (Donohue *et al.*, 2011). However, specifying all 2^p possible candidate models in high dimensions is impractical. Lee *et al.* (2014) developed a stochastic search variable selection (SSVS) method that selects both fixed and random effects in proportional hazards mixed effects models in a Bayesian framework, but their method is only computationally feasible for small or moderate dimensions.

1.5 Proximal gradient and Majorization-Minimization algorithms

As will be discussed in Chapters 3, 4, and 5, we use a Majorization-Minimization algorithm to fit the minimization step (M-step) of the Monte Carlo Expectation Conditional Minimizaiton (MCECM) algorithm (Rashid *et al.*, 2020) utilized in these chapters. The Majorization-Minimization algorithm requires the specification of a constant that provides an upper bound on the Hessian of the loss function (Breheny and Huang, 2015). For the Binomial and Gaussian families, this maximum value is well recognized as 0.25 or 1, respectively (Breheny and Huang, 2015). For the Poisson family or a related family that can be fit using a log-linear model, which we use in Chapters 4 and 5, this maximum value cannot be easily determined. Consequently, we utilize the proximal gradient line search algorithm described in Parikh *et al.* (2014) to estimate a step size value; the inverse of this step size value approximates the upper bound constant discussed above. The proximal gradient approach and the line search algorithm described in Parikh *et al.* (2014) is outlined below.

Suppose we consider optimization problems with only fixed effects. Consider the general problem

$$\text{minimize} \quad f(\beta) + g(\beta), \tag{1.23}$$

where $f(\beta)$ is the differentiable loss function, $g(\beta)$ represents the penalty function of interest (e.g. the L_1 penalty LASSO, MCP, or SCAD (Breheny and Huang, 2011; Friedman *et al.*, 2010)), and β are the coefficients of interest. In problems with only fixed effects, this loss function is the negative of the log-likelihood:

$$L(\beta) = -\frac{1}{N}\sum_{i=1}^{N} \ell_i(\beta) = -\frac{1}{N}\sum_{i=1}^{N} \log f(y|x_i, \beta). \tag{1.24}$$

As will be explained later in Chapters 3, 4, and 5, the loss function we use for our mixed effects problems will be the Q-function.

The proximal gradient method is defined as:

$$\beta^{(s+1)} := \textbf{prox}_{\delta g}(\beta^{(s)} - \delta \Delta f(\beta^{(s)})) \tag{1.25}$$

where $\delta > 0$ is the step size, $\beta^{(s)}$ is the value of the coefficients of interest evaluated at a previous iteration of the algorithm, and $\beta^{(s)+1}$ is the updated set of coefficients.

A majorization-minimization algortihm for minimizing a function $\psi : \boldsymbol{R}^n \longrightarrow \boldsymbol{R}$ consists of the iteration

$$\beta^{(s+1)} := \text{argmin}_{\beta} \quad \hat{\psi}(\beta, \beta^{(s)}) \tag{1.26}$$

where $\hat{\psi}(\beta, \beta^{(s)})$ is a convex upper bound to ψ that is tight at $\beta^{(s)}$ such that $\hat{\psi}(\beta, \beta^{(s)}) \geqslant \psi(\beta)$ and $\hat{\psi}(\beta, \beta) = \psi(\beta)$ for all β.

For an upper bound of f, consider $\hat{f}_\delta$, which is derived from taking the Taylor series expansion of f about the value of $\beta^{(s)}$:

$$\hat{f}_\delta(\beta, \beta^{(s)}) = f(\beta^{(s)}) + \Delta f(\beta^{(s)})^\top (\beta - \beta^{(s)}) + \frac{1}{2\delta} ||\beta - \beta^{(s)}||_2^2. \tag{1.27}$$

The algorithm

$$\beta^{(s+1)} := \text{argmin}_\beta \quad \hat{f}_\delta(\beta, \beta^{(s)}) \tag{1.28}$$

is thus a majorization-minimzation algorithm. It then follows that the function q_δ given by

$$q_\delta(\beta, \beta^{(s)}) = \hat{f}_\delta(\beta, \beta^{(s)}) + g(\beta) \tag{1.29}$$

is a surrogate for the $f + g$ function of interest (with fixed y). The majorization-minimization algorithm

$$\beta^{(s+1)} := \text{argmin}_\beta \quad q_\delta(\beta, \beta^{(s)}) \tag{1.30}$$

can be shown to be equivalent to the proximal gradient iteration in equation 1.25.

The line search algorithm to identify the step size δ is:

Given β^k, δ^{k-1}, and parameter $c \in (0, 1)$ (e.g. $c = 0.95$),

Let $\delta := \delta^{k-1}$

Repeat:

1. Let z be the β update from the Majorization-Minimization solution given the above values

2. Break if $f(z) \leqslant \hat{f}_\delta(z, \beta^k)$

3. Update $\delta = c * \delta$

Return $\delta^k := \delta$, $\beta^{k+1} = z$.

In other words, suppose for a Majorization-Minimization algorithm iteration $k+1$ that we have solved for a particular solution to β, which we label as z, and we have saved the previous solution β^k. We evaluate whether the loss function evaluated at z is less than or equal to the upper bound of the loss function specified in (1.25) evaluated using the step size δ and the previous solution β^k. If this inequality does not hold, then we reduce the step size by some constant between 0 and 1. Additional details about the $\hat{f}$ function used in the log-linear mixed effects models in Chapters 4 and 5 are given in the Chapter 4 Appendix C.3.

CHAPTER 2: ESTIMATING CELL TYPE COMPOSITION USING ISOFORM EXPRESSION ONE GENE AT A TIME

2.1 Introduction

In this chapter, we outline the development of a statistical model named IsoDeconvMM for expression deconvolution in mixture tissues by exploiting isoform-level expression differences between cell types. A crucial factor for the success of expression deconvolution is to identify a good set of signature genes/isoforms whose expression has much higher variation across cell types than within cell types. However, even for such carefully selected genes/isoforms, there are still biological variation of cell type-specific gene/isoform expression across individuals. IsoDeconvMM is designed to explicitly model biological variability to achieve robust performance. We demonstrate the utility of our method using the Blueprint dataset (Chen *et al.* (2016)). This dataset contains human bulk RNA-seq samples for three sorted immune cell populations (CD4-positive, alpha-beta T cell; CD14-positive, CD16-negative classical monocyte; and mature neutrophil) from up to 197 individuals. In an *in silico* data analysis, we used this data to model the variability across individuals and test the performance of our method given this biological variability.

The rest of the chapter is organized as follows. In Section 2.2, we present the statistical models and algorithm used to estimate cell type proportions in mixture tissues, and describe the data and materials needed for the method. In Section 2.3, we present *in silico* data analyses. In these analyses, we compare the performance of our method with the performance of CIBERSORTx (Newman *et al.* (2015)). In Section 2.4, simulations are conducted to assess the performance of the IsoDeconvMM method when different underlying data distributions are assumed. Concluding remarks are given in Section 2.5. Technical proofs are given in the Appendix A.2. Additional details regarding the procedures and materials required for the analyses in Section 2.3 and 2.4 are given in the Appendix A.1 and the Appendix A.3.

2.2 Methods

2.2.1 Required data and resources

Consider a biological tissue sample composed of K different cell types. We seek to estimate the unknown relative abundance of each cell type k—or the proportion of cells of type k—in the heterogeneous sample. In order to estimate these proportions, IsoDeconvMM requires a single RNA-seq experiment performed on the mixture sample. In addition, it is assumed that there exist RNA-seq data for N_k purified samples for cell type k. For each sample, RNA-seq read counts are summarized at the exon level by counting the number of reads (or RNA-seq fragments for paired-end reads) overlapping various sets of exons. The definition of an exon set and an illustrative example were given in Chapter 1 Section 1.1.4. It is assumed that a detailed gene model on the location of each exon and the structure of each isoform is available for each gene. Consider a hypothetical gene composed of m non-overlapping exons that are utilized by I isoforms, or distinct mRNA transcripts formed by unique combinations of these exons. As specified in the gene models, the locations of these exons within the gene are known as are the identities and compositions of all isoforms used by this gene. We define the read count at any exon set A as the number of reads which overlap each of the exons in A and only these exons.

IsoDeconvMM also assumes that there exists a list of cell-type specific genes wherein there are gene- and/or isoform- expression differences across the K cell types. Such a list of genes can be found using one of a variety of expression testing methods for RNA-seq data. Furthermore, IsoDeconvMM requires empirical knowledge of the fragment length distribution for the bulk RNA-seq samples.

2.2.2 The IsoDeconvMM model and algorithm

2.2.2.1 Model parameters

Within the IsoDeconvMM model, cell type proportions are estimated independently within each gene, and these gene-specific proportion estimates are then aggregated to produce a sample level cell-type relative abundance estimate. To simplify discussion, we outline the IsoDeconvMM model for a single gene.

RNA-seq expression is commonly corrected for feature length. Previously, however, the notion of feature length pertained to the length of the genes or isoforms being measured and not to the lengths of exon sets. Sun *et al.* (2015) extended the definition of feature length to exon-sets and referred to it as the effective length for exon sets. Briefly, the effective length of an exon set is the expected number of starting locations where an RNA-seq fragment that overlaps with all the exons of this exon set can be sampled. Such expectation is taken over the distribution of RNA-seq fragment length. Note that the effective length of an exon set varies across isoforms. For example, isoforms that do not contain all the exons within the set cannot produce reads in that exon set, thus the effective length of the exon set for such isoforms will be zero. See the supplementary materials of Sun *et al.* (2015) for more details.

We first consider the models and parameters used to describe the gene expression in cell type-specific samples. In all the notation, we utilize the subscripts kj to denote the parameters for sample j of cell type k. Let $\boldsymbol{Y}_{kj} = \{Y_{kjA}\}$ denote the vector of read counts across all E exon sets in the given gene/transcript cluster for sample j of cell type k. Also denote $Y_{kj(O)}$ as the total read count outside the gene of interest in this sample. We assume that the vector $\left(Y_{kj(O)}, \boldsymbol{Y}_{kj}^T\right)^T$ follows a multinomial distribution

$$\begin{bmatrix} Y_{kj(O)} \\ \boldsymbol{Y}_{kj} \end{bmatrix} \Bigg| \tau_{kj}, \boldsymbol{\gamma}_{kj} \sim \text{Multinomial}\left(t_{kj}, \begin{bmatrix} 1 - \tau_{kj} \\ \tau_{kj} \boldsymbol{X} \boldsymbol{\gamma}_{kj} \end{bmatrix} \right), \tag{2.1}$$

where τ_{kj} is the probability that a randomly selected read maps to the gene of interest, $\boldsymbol{\gamma}_{kj} = (\gamma_{kj1}, ..., \gamma_{kjI})^T$ is the vector of I isoform expression parameters, t_{kj} is the total read count in the sample, and $\boldsymbol{X}$ is a matrix of effective lengths such that column i of $\boldsymbol{X}$ is the vector of effective lengths for all the exon sets of isoform i.

We further describe the probability τ_{kj} and the isoform parameters $\boldsymbol{\gamma}_{kj}$ with the following beta and Dirichlet distributions:

$$\begin{aligned} \tau_{kj} &\sim \text{Beta}(\boldsymbol{\beta}_k), \\ \tilde{\boldsymbol{l}} \circ \boldsymbol{\gamma}_{kj} &\sim \text{Dirichlet}(\boldsymbol{\alpha}_k), \end{aligned} \tag{2.2}$$

where $\tilde{\boldsymbol{l}} = (\tilde{l}_1, ..., \tilde{l}_I)$, $\tilde{l}_i = \sum_{A \in \text{isoform } i} \boldsymbol{X}_A$ represents the total effective lengths of isoform i for $1 \leqslant i \leqslant I$, and $\circ$ represents element-wise multiplication of two vectors. It should be noted that the $\boldsymbol{\gamma}_{kj}$ parameters can be interpreted as per-unit-of-effective-length conditional probabilities that a read maps to isoform i given that it maps to the gene which utilizes isoform i. The fact that we model gene expression for each sample j of cell type k separately in the above models allows us to capture the biological variation across samples. The similarity of all the samples from cell type k is modeled by the shared beta or Dirichlet distribution in equation (2.2). We next consider the models and parameters used to describe the exon set counts in the mixture sample. Let $\boldsymbol{Z} = \{Z_A\}$ denote the vector of read counts across all E exon sets in the given gene for the mixture sample, and let $Z_T = \sum_A Z_A$ denote the sum of the read counts for the given gene. We assume that the vector of counts $\boldsymbol{Z}$ follows a multinomial distribution such that

$$\left[\boldsymbol{Z}\right] \Big| \tau_k^*, \boldsymbol{\gamma}_k^* \sim \text{Multinomial}\left(Z_T, \left[\frac{\sum_{k=1}^K \rho_k \tau_k^* \boldsymbol{X} \boldsymbol{\gamma}_k^*}{\sum_{k=1}^K \rho_k \tau_k^*}\right]\right), \tag{2.3}$$

where τ_k^* represents the probability that a randomly selected read from cell type k maps to the gene of interest in the mixture sample, $\boldsymbol{\gamma}_k^* = (\gamma_{k1}^*, ..., \gamma_{kI}^*)$ is the vector of I isoform expression parameters unique to cells of type k found within the mixture sample, and ρ_k is the proportion of cell type k in the mixture sample.

Using the same cell type k gene expression hyperparameters $\boldsymbol{\beta}_k$ and isoform expression hyperparameters $\boldsymbol{\alpha}_k$ from the pure sample models in equation (2.2), we further describe the probabilities τ_k^* and the mixture isoform parameters $\boldsymbol{\gamma}_k^*$ as follows:

$$\begin{aligned} \tau_k^* &\sim \text{Beta}(\boldsymbol{\beta}_k), \\ \tilde{\boldsymbol{l}} \circ \boldsymbol{\gamma}_k^* &\sim \text{Dirichlet}(\boldsymbol{\alpha}_k). \end{aligned} \tag{2.4}$$

Given those shared parameters $\boldsymbol{\beta}_k$ and $\boldsymbol{\alpha}_k$, we assume independence across samples.

2.2.2.2 Model estimation

Within each gene, the model is fit using a staged estimation approach with three stages. In stage one, the gene and isoform expression parameters are estimated separately for each purified reference sample by maximum likelihood estimation. The likelihood used for stage one involves only equation (2.1). Under such a framework, closed form estimates of τ_{kj} are obvious and a logarithmic adaptive barrier algorithm can be used to obtain estimates of the $\boldsymbol{\gamma}_{kj}$ subject to boundary constraints. Once obtained for each cell type and sample, these estimates are held fixed for all further stages.

Within stage two, the values of τ_{kj} and $\boldsymbol{\gamma}_{kj}$ estimated during stage one are treated as observations from equation (2.2). Estimates of $\boldsymbol{\alpha}_k$ and $\boldsymbol{\beta}_k$ are obtained via maximum likelihood estimation within separate Dirichlet models. Once obtained, these estimates of $\boldsymbol{\alpha}_k$ and $\boldsymbol{\beta}_k$ are fixed for stage three.

Finally, in stage three, the $\boldsymbol{\alpha}_k$ and $\boldsymbol{\beta}_k$ estimates are used in Dirichlet distributions as penalty functions in the estimation of the $\boldsymbol{\gamma}_k^*$ and τ_k^*. In this way, we regularize estimates of $\boldsymbol{\gamma}_k^*$ and τ_k^* to be similar to those estimates obtained in the pure cell type samples. The use of an EM algorithm allows separation of the full likelihood into $K + 1$ independent components in the M step. The first K components pertain to the isoform expression parameters from each of the K cell types. Each of these components is optimized using a Newton-Raphson algorithm on the $\log(\boldsymbol{\gamma}_k^*)$ until convergence of isoform parameters. The last component contains information regarding the ρ_k and $\log(\tau_k^*)$ values, which are optimized using a quasi-Newton's method optimization procedure (BFGS). Estimation is seeded at various start points to identify global maxima. The E-step updates the posterior means of the exon set counts in the mixture sample ($\boldsymbol{Z}$) attributable to cell type k. The expectation has a closed form solution, provided in Appendix A.2. The EM algorithm is iterated until convergence in the proportion estimates. Proportion estimates across multiple genes are then aggregated using the spatial median to obtain final estimates of cell type proportions.

Technical proofs and further details about the models and methods can be found in Appendix A.2. Table A.2 in Appendix A.2 contains a summary of the notation presented in the above Section 2.2. A discussion about why a staged estimation approach was used instead of a joint estimation approach is included in Section A.2.6.

2.3 *In Silico* Blueprint data analysis

To the best of our knowledge, our IsoDeconvMM method is the first method that estimates cell type proportions using isoform expression. Since there are already several methods for cell type composition estimation using gene expression (instead of isoform expression) data, an immediate question is what is the advantage to use isoform expression. In this section, we compare our IsoDeconvMM method with CIBERSORTx (Newman *et al.* (2019)), a representative and popular method for cell type composition estimation using gene expression, and demonstrate that IsoDeconvMM has similar performance with CIBERSORTx when the number of genes is relatively large and it outperforms CIBERSORTx with large margin when the number of genes is small. To compare IsoDeconvMM and CIBERSORTx, we utilize the Blueprint data set (Chen *et al.* (2016)) discussed in Section 2.1. We arbitrarily label the three cell types as follows: CT1 represents CD4-positive, alpha-beta T cell; CT2 represents CD14-positive, CD16-negative classical monocyte; and CT3 represents mature neutrophil.

In order to create mixture files from the Blueprint data, we selected 100 individuals who had pure reference samples collected from all three cell types. For each of these individuals, we used their pure reference samples to create a mixture file. The 100 mixture proportions were randomly selected from the distribution $\rho_{mix} \sim$ Dirichlet$(2, 2, 2)$. Relatively extreme probabilities, defined as probability vectors that assigned one or more cell types to have a probability less than 0.05, were eliminated from consideration.

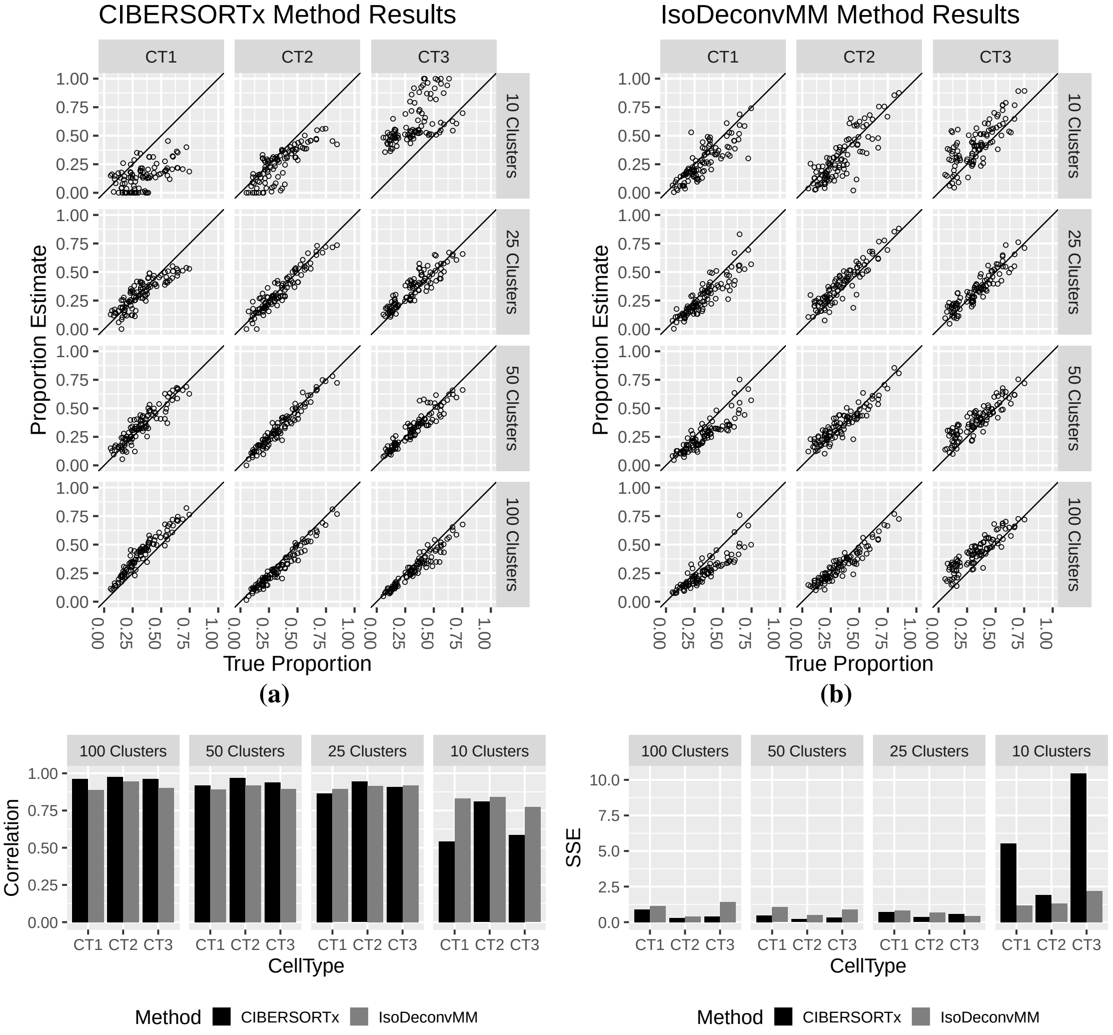
To select genes/transcript clusters to be used by the IsoDeconvMM method, we sought to identify differential isoform usage (DU) transcript clusters that had the largest difference between the isoform distributions in the three cell types. To this end, we identified clusters that had at least one isoform highly expressed in one cell type and either minimally expressed or not expressed at all in the other two cell types, collectively. The selection of transcript clusters proceeded as follows. We selected 10 pure reference samples (not used in the mixture file creation) from each of the three cell types present in the Blueprint data. We then used the `isoDetector` function in the isoform R package (Sun *et al.* (2015)) to acquire isoform abundance information for transcript clusters present on chromosomes one through four for all of the 30 pure reference samples. Using the abundance information output, we examined both fold change magnitudes and Wilcoxon rank sum tests comparing abundance levels for the isoforms in the cluster between a single cell type and the other two cell types combined. Using these results, we identified isoforms of interest. The transcript clusters that these isoforms belonged to were then selected for further analysis. A full description of the procedure to identify DU clusters of interest can be found in the Appendix A.1.

For the CIBERSORTx method, we aimed to select DE transcript clusters in a similar manner to the DU transcript clusters used in the IsoDeconvMM analysis. We first quantified the total expression per transcript cluster, restricting the transcript clusters considered to those present on chromosomes one though four. Then we applied DESeq2 (Love *et al.* (2014)) to identify transcript clusters with differential expression that were relatively overexpressed in one cell type compared to the other two cell types combined.

In the Appendix A.3, we compared the performance of the IsoDeconvMM algorithm across different algorithm settings. Based on results presented in the Appendix A.3 (see Figure A.1), we concluded that using five samples per cell type in the IsoDeconvMM analysis was sufficient. Therefore, all further IsoDeconvMM and CIBERSORTx results utilize five pure reference samples per cell type. Since the IsoDeconvMM algorithm requires multiple initial points in order to optimize the accuracy of the results, we also explored how many initial points were sufficient to use. Figure A.2 in the Appendix A.3 suggests that using the 10 generic initial points specified in Table A.1 in the Appendix A.1 is sufficient for this case of three total cell types. Therefore, all IsoDeconvMM algorithm results presented in this section utilized these 10 initial points in the algorithm. Recommendations of initial points for the generic case of K cell types is given in the Appendix A.1. The IsoDeconvMM package gives automated recommendations for initial points.

In the exploratory analyses presented in the Appendix A.3, we found that the estimates of the cell type specific isoform parameters could be unstable for a small number of transcript clusters. This could be due to extra variance or outliers in these genes. In those clusters, the estimate of the $\boldsymbol{\alpha}_k$ parameters of equation (2.2) (estimated in stage one of the model fit algorithm) tended be much larger than other clusters. Therefore we performed a filtering step such that if two or more cell type specific isoform parameter estimates for a transcript cluster were greater than 500, the cluster was excluded from further analysis. We now compare the the performance of IsoDeconvMM and CIBERSORTx results when different numbers of transcript clusters were used in the analysis (Figure 2.1). In each method, the best N of the available transcript clusters were selected by first choosing the best n_s clusters per cell type comparison (cell type j vs the other two cell types collectively) and then take their union. The number n_s was adjusted such that the union gave $N = \{100, 50, 25, 10\}$ clusters.

When 100 or 50 transcript clusters are used in the analysis, both the CIBERSORTx and IsoDeconvMM methods perform well, with CIBERSORTx performing slightly better than IsoDeconvMM. For the 25 cluster case, both methods perform equally well. For the case when only 10 clusters are used, the CIBERSORTx method is very unstable. In contrast, the IsoDeconvMM method is still reasonably accurate.



(a) (b) (c) (d)

Figure 2.1: Blueprint mixture proportion estimate results calculated using the CIBERSORTx and IsoDeconvMM methods. Results separated by cell types and number of transcript clusters used in the analysis. (a) Proportion estimates vs true proportions for CIBERSORTx method (used DE clusters only). (b) Proportion estimates vs true proportions for IsoDeconvMM method (used DU clusters only). (c) Correlation and (d) sum-of-square (SSE) results compared across methods.

2.4 Simulation studies

Our model assumes an underlying Dirichlet-multinomial distribution, which allows overdispersion beyond the variance of multinomial distribution. However, it is still possible that a Dirichlet-multinomial distribution cannot fit the real data well. In this section, we evaluate the performance of IsoDeconvMM when the observed data are generated from Dirichlet-negative binomial distributions. We simulated bulk RNA-seq counts data from three sorted cell populations given the generic labels of CT1, CT2, and CT3.

For all of the simulations, we first generated the gene-level counts from a Dirichlet-multinomial distribution. In order to make the distribution of gene counts as realistic as possible, we used gene count distribution from a real data set (Parikshak *et al.* (2016)) that contained the number of RNA-seq reads per gene for 89 human bulk RNA-seq samples. The genes present in this data set were filtered such that each transcript cluster was comprised of a single gene (for convenience purposes) and genes with low expression were excluded. The genes were then limited to those present on chromosomes one to nine in order to reduce computational burden, resulting in 5,172 total genes. A full description of the gene selection procedure is provided in the Appendix A.3. We fit a Dirichlet distribution to these data using the R package DirichletReg (Maier (2014)).

For each simulated pure sample, the Dirichlet distribution described above generated a probability vector associated with the genes. The total read count per sample was selected from a normal distribution with mean 7 million and standard deviation 1 million. This normal distribution was based on the distribution of the total read counts of the selected 5,172 genes in the 89 bulk RNA-seq samples (Parikshak *et al.* (2016)). Individual gene counts were then generated using a multinomial distribution.

Of the total 5,172 genes, we selected 1,000 genes with relatively high expression and at least three isoforms as possible genes to be used for the mixture sample proportion estimate in the IsoDeconvMM analysis. For all 1,000 of these genes of interest, we calculated the effective length design matrix $\boldsymbol{X}$ as described in Section 2.2. After additional filtering to exclude genes with over 15 isoforms, we randomly selected 100 genes for differential isoform usage.

The Dirichlet-multinomial and the Dirichlet-negative binomial simulations diverge on the simulation of the exon set counts. For each cell type, we gave each of the 1,000 genes of interest a Dirichlet distribution for their isoforms. These Dirichlet distributions only differed between the three cell types for the 100 genes specified for differential isoform usage. A probability vector

$\boldsymbol{\pi}_g = (\pi_{g1}, ..., \pi_{gI})$ was drawn from these Dirichlet distributions, where π_{gi} is the proportion of read counts in isoform i given that the read comes from gene g.

In the Dirichlet-multinomial simulation, the vector π_g was set equal to the $\tilde{l} \circ \gamma_g$ vector described in Section 2.2 in equation (2.2). We then model the exon set counts for gene g by

$$\boldsymbol{y}_g \sim \text{Multinomial}(T_g, s_g \sum_{i=1}^{I} \boldsymbol{x}_{gi}\gamma_{gi}), \quad \gamma_{gi} \geqslant 0 \tag{2.5}$$

where $\boldsymbol{x}_{gi}$ for $1 \leqslant i \leqslant p$ represents the vector of effective lengths of all of the exon sets for the i^{th} isoform for gene g, T_g is the total read count for gene g, and s_g is the scaling factor such that $s_g \sum_{i=1}^{I} \boldsymbol{x}_{gi}\gamma_{gi} = 1$.

In the Dirichlet-negative binomial simulation, the probability vector π_{gi} was used differently. The vector of counts of the possible exon sets within the gene, $\boldsymbol{y}_g$, was given a negative binomial distribution $\Psi(\boldsymbol{\mu}_g, \phi)$ with mean $\boldsymbol{\mu}_g$ and dispersion parameter ϕ. We model $\boldsymbol{\mu}_g$ by

$$\boldsymbol{\mu}_g = \boldsymbol{X_g \beta_g} = \sum_{i=1}^{p} \boldsymbol{x}_{gi}\beta_{gi} = \sum_{i=1}^{p} \boldsymbol{x}_{gi}\pi_{gi}r_g, \quad \beta_{gi} \geqslant 0 \tag{2.6}$$

where π_{gi} again is the proportion of read counts in isoform i given that the read comes from gene g, $\boldsymbol{X}_g = (\boldsymbol{x}_{g1}, ..., \boldsymbol{x}_{gp})$, $\boldsymbol{x}_{gi}$ for $1 \leqslant i \leqslant p$ represents the effective lengths of all of the exon sets for the i^{th} isoform for gene g, and r_g is a scaling factor equal to the ratio of the total read count of the gene and the sum of the vector $\sum_{i=1}^{p} \boldsymbol{x}_{gi}\pi_{gi}$.

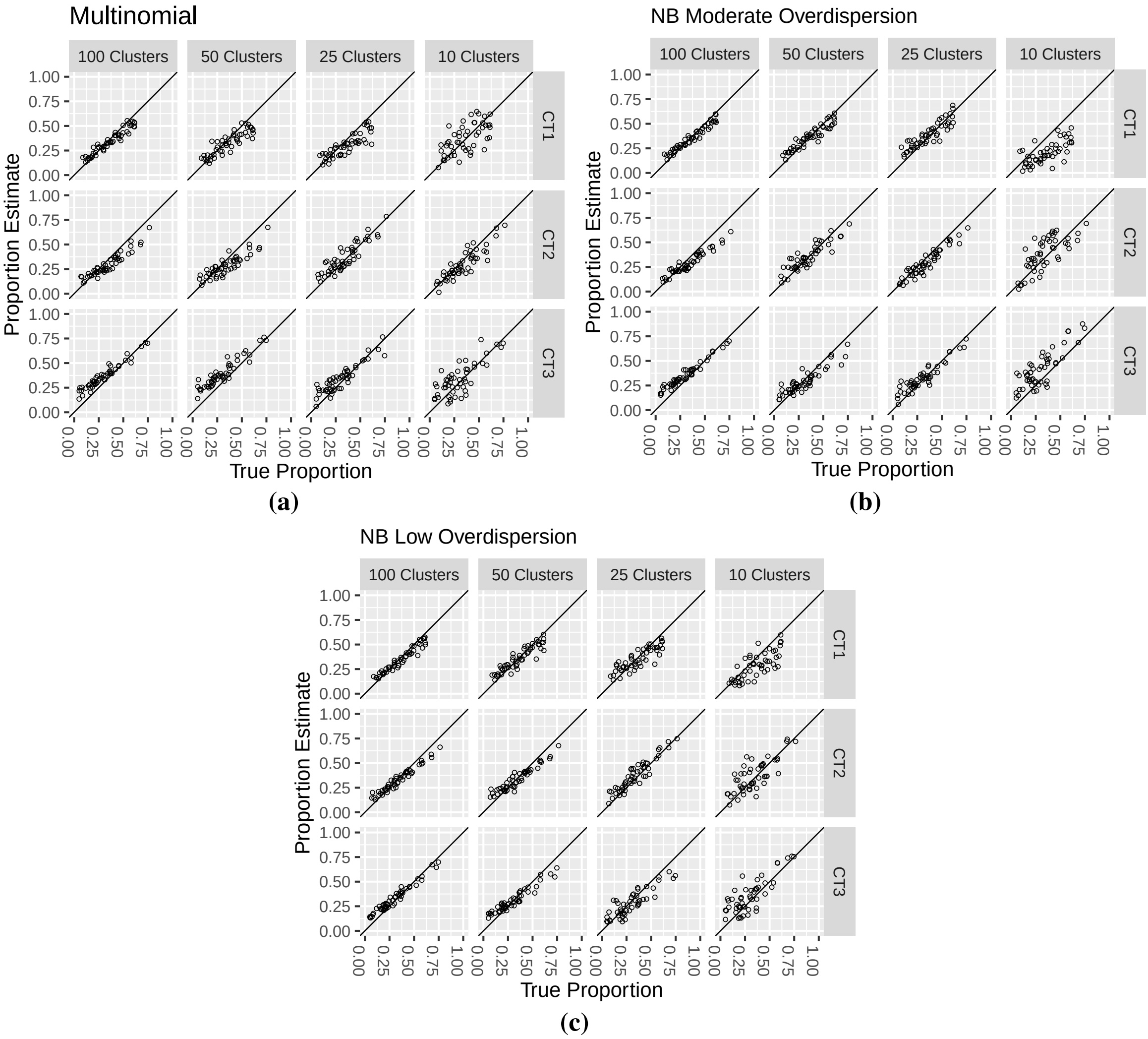
In the Dirichlet-negative binomial simulations, we also compared the algorithm fit results under low and moderate overdispersion assumptions for the Negative Binomial portion of the model. The dispersion parameter ϕ was given the range $1/90$ to $1/120$ for the low dispersion set-up and the range $1/50$ to $1/60$ for the moderate dispersion set-up.

In order to make the isoform Dirichlet distributions for the DU genes as realistic as possible, we modeled these distributions using the results from the Blueprint data set analysis described in Section 2.3. The cell type specific isoform Dirichlet parameter $\boldsymbol{\alpha}_k$ (estimated by Dirichlet-multinomial distribution) were used in the simulations. In the Dirichlet-multinomial simulations, these values were used directly. In the Dirichlet-negative binomial simulations, these values were multiplied by a constant of five so that the overall variance between Dirichlet-multinomial and Dirichlet-negative binomial are similar.

Three data sets were simulated using the three different data modeling assumptions: Dirichlet-multinomial, Dirichlet-negative binomial with moderate overdispersion, and Dirichlet-negative binomial with low overdispersion. We generated 15 pure reference samples per cell type for each simulation set-up. We partitioned the pure samples such that for each cell type, 10 samples were used to generate the mixture samples and the other 5 were used to estimate cell type-specific gene/isoform expression. Fifty mixture proportions were randomly selected from the distribution $\rho_{mix} \sim$ Dirichlet$(2, 2, 2)$. Relatively extreme probabilities, defined as probability vectors that assigned one or more cell types to have a probability less than 0.05, were eliminated from consideration.

For the fragment length distribution file, we chose to simulate paired-end read lengths from a truncated normal distribution with mean 300 bp, standard deviation 50 bp, and truncated to the left at 150 bp. For the initial points, we used the same 10 generic initial points used in Section 2.3, provided in the Appendix in Table A.1.

All three of the simulated data sets were then fit using the IsoDeconvMM algorithm and we examine the performance of IsoDeconvMM when different number of transcript clusters are used to estimate cell type proportions. In each simulation setup, we randomly selected the desired number of transcript clusters from the 100 simulated DU clusters. The results presented in Figures 2.2 and 2.3 suggest that the results of our IsoDeconvMM method is robust to the data generation mechanisms. The only situation where the performance of IsoDeconvMM is slightly worse is when the number of transcript clusters is small (i.e., only 10 clusters) and the Dirichlet-negative binomial has moderate overdispersion.

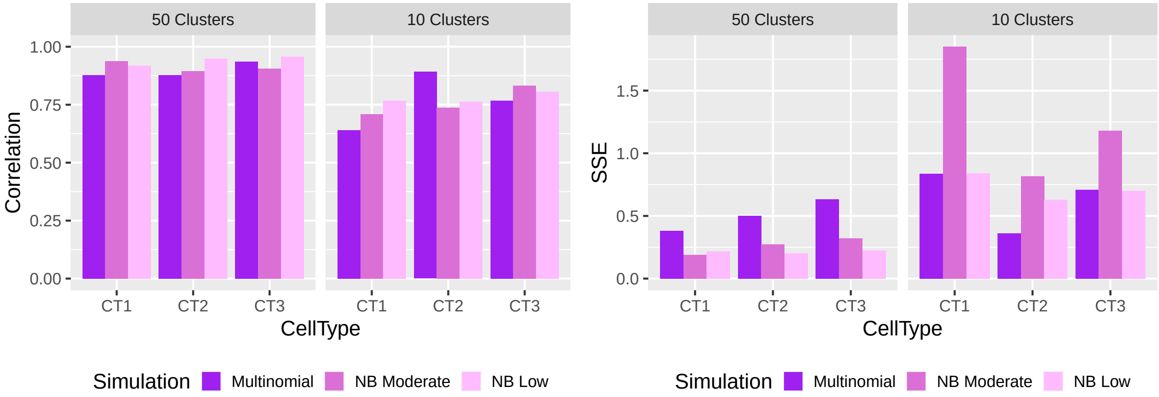


(a)

(b)

(c)

Figure 2.2: IsoDeconvMM proportion estimates for the following underlying data models: (a) Dirichlet-multinomial, (b) Dirichlet-negative binomial with moderate overdispersion, and (c) Dirichlet-negative binomial with low overdispersion. Results separated by cell types (rows) and number of genes used in the analysis (columns).



(a) **(b)**

Figure 2.3: Correlation and sum-of-square error (SSE) results comparing the IsoDeconvMM proportion estimates vs the true proportions for simulations assuming different underlying data models: Dirichlet-multinomial, Dirichlet-negative binomial with moderate overdispersion, and Dirichlet-negative binomial with low overdispersion. Results separated by cell types and number of genes used in the analysis.

2.5 Discussion

We have developed a new statistical method named IsoDeconvMM that estimates cell type abundance of bulk RNA-seq samples that are mixtures of multiple cell types. This method is unique from other deconvolution methods in that it utilizes differential isoform usage information. We anticipate that this method will be of particular relevance in cases where differential isoform usage is more informative than differential gene expression, or when the number of available genes is small. Currently, application of our method is limited by the availability of cell type-specific and isoform-specific gene expression data. Single cell RNA-seq (scRNA-seq) is a popular approach to generate cell type-specific gene expression data across different cell types, though most scRNA-seq pipelines cannot capture the complete information of different isoforms. However, the emerging spatial RNA-seq data show that it is possible to capture isoform level gene expression for each cell or a few cells around a locus (Lebrigand *et al.* (2020); Maynard *et al.* (2020)). We expect that the full advantage of IsoDeconvMM can be demonstrated when combining such cell type-specific and isoform-specific expression derived from these new pipelines.

We did not know of another deconvolution method that utilizes differential isoform usage information with which to compare our method. Instead, we compared IsoDeconvMM with CIBERSORTx (Newman *et al.* (2015)), which utilizes differential gene expression information. We believe a key advantage of our method over existing reference-based deconvolution methods is that we can estimate cell type fractions using the gene expression data from a single gene by exploiting the relative expression of each isoform within a gene. We tested this theory by comparing our method with CIBERSORTx, which uses information across genes. We found that our method performs similarly compared with CIBERSORTx when a moderate number of genes or transcript clusters are used and outperforms CIBERSORTx when a small number of transcript clusters are used. In addition to seeing this pattern in the *in silico* Blueprint analyses presented in Section 2.3, we also found similar results when we performed both IsoDeconvMM and CIBER-

SORTx on simulated Dirichlet-Multinomial data (see Appendix A.3 for details and results). This could be very useful when it is desired to distinguish between highly similar cell types, such as closely related neuron cells, in which case there may not be many transcript clusters that can truly discriminate between the cell types. Additionally, this could be useful in clinical settings that utilize a small panel of genes.

Although we could have compared our method with CIBERSORTx using isoforms instead of genes (or transcript clusters), thereby using information across isoforms, we felt applying CIBERSORTx on isoforms has several limitations. The estimate of isoform expression is generally more noisy and has more measurement error. Furthermore, a major limitation of approaches that use information across genes/isoforms is that a sufficient sample size of genes or isoforms is required. This limitation, which was illustrated in the *in silico* analysis results, is the same limitation for either CIBERSORTx on genes or CIBERSORTx on isoforms. Consequently, using CIBERSORTx on isoforms would not provide a benefit. In contrast to methods that use information across genes/isoforms, IsoDeconvMM utilizes gene expression variation across exon sets. The number of exon sets can increase quickly with the number of exons, and thus there are many genes with enough sample size within a gene itself.

The IsoDeconvMM method has other beneficial properties. In the Appendix A.3, we have demonstrated that our method only requires a small number of pure reference samples per cell type. The simulations also show that the IsoDeconvMM method is robust to some model misspecification.

The IsoDeconvMM method has some limitations related to its computation time. Part of the reason for this time limitation is due to the fact that it requires an input of multiple initial points. However, this could be remedied using parallel computation techniques. Parallel computation techniques can be easily used in conjunction with the IsoDeconvMM method because a separate proportion estimate is calculated for each transcript cluster, and these individual estimates are later aggregated to get the overall proportion estimate. The IsoDeconvMM package, available for download in GitHub, allows for either serial or parallel computation. When the algorithm was run in serial using the UNC Longleaf computing cluster (CPU Intel processors between 2.3Ghz and 2.5GHz), it took an average of 16.55 minutes to estimate the mixture proportion for a transcript cluster using 10 initial points.

More generally, the IsoDeconvMM procedure has the same limitations that apply to all reference-based deconvolution methods. These methods require assumptions about the true number and identity of cell types in the mixture samples. In many applications, this cell type information is unknown.

We looked further into the cell type 1 bias seen in the *in silico* Blueprint analyses. We performed 10 replicates of the *in silico* analyses, picking different sets of 100 individuals to create the mixture samples, picking different sets of pure reference samples, but using the same transcript clusters used in Section 2.3. We found that 2 of the 10 analysis replicates resulted in similar V shapes in the cell type 1 scatter plots seen in the paper results, but the other 8 replicates did not. This led us to believe that this concerning V shape in the cell type 1 scatter plot results were likely a result of unlucky randomness in the simulation set-up. See Appendix A.3 for further details and results.

It should be noted that the IsoDeconvMM method is sensitive to the isoform distribution effect size across the different cell types. We recommend users to be conscientious about selecting isoforms with the greatest effect sizes between the different cell types, regardless of what method they choose to identify isoforms with differential usage across the cell types.

CHAPTER 3: HIGH DIMENSIONAL PENALIZED GENERALIZED LINEAR MIXED MODELS: THE GLMMPEN R PACKAGE

3.1 Introduction

In Section 1.2, we discussed the limitations of existing methods for performing variable selection in generalized linear mixed models. In existing methods for fitting generalized linear mixed models (GLMMs), one must specify individual candidate models in order to perform variable selection, which can be prohibitive in high dimensions. In contrast, available variable selection methods either do not allow for the inclusion of random effects or they only allow a fixed pre-specification of random effects.

To address these limitations, we present the **glmmPen R** package, one of the first to allow for simultaneous selection of fixed and random effects in high dimension through the use of penalized Generalized Linear Mixed Models (pGLMMs). The package leverages a Monte Carlo Expectation Conditional Minimization (MCECM) algorithm with several techniques to improve the computational efficiency of the algorithm. In the MCECM E-step, **glmmPen** utilizes the **Stan** software to acquire efficient samples of the posterior, and a Majorization-Minimization coordinate descent algorithm in the M-step. Within the M-step, the **glmmPen** package reduces the required memory usage and improves scalability by utilizing the fast looping capabilities within **Rcpp** and **RcppArmadillo** in order to recalculate rows of large matrices pertaining to intermediate quantities without necessitating their storage. The **glmmPen** package is also able to improve the speed of the overall variable selection procedure by strategic coefficient initialization (see Section B.1) and strategic restriction of random effects (see Section 3.4.2).

The remainder of the chapter is organized as follows. Section 3.2 reviews the models assumed for the fitting of pGLMMs, first described in Rashid *et al.* (2020). Section 3.3 describes the MCECM algorithm used by **glmmPen** to fit pGLMM models. Section 3.4 describes the model selection procedure of the package and the BIC-type selection criteria available for use. Section 3.6 illustrates the use of the software through an application to a cancer subtype classification dataset. Section 3.5 provides some simulation results. Section 3.7 provides concluding comments. The package is available from the Comprehensive **R** Archive Network (CRAN) at `https://cran.r-project.org/package=glmmPen`.

3.2 Generalized linear mixed models

We review the notation and model formulation of our approach, first introduced in Rashid *et al.* (2020). We consider the case where we want to analyze data from K independent groups of any kind. For instance, we could be interested in analyzing data from K different studies, or we could be interested in analyzing longitudinal data from K individuals. For each group $k = 1, ..., K$, there are n_k observations for a total sample size of $N = \sum_{k=1}^{K} n_k$. For the k^{th} group, let $\boldsymbol{y}_k = (y_{k1}, ..., y_{kn_k})^T$ be the vector of n_k independent responses, let $\boldsymbol{x}_{ki} = (x_{ki,1}, ..., x_{ki,p})^T$ be the p-dimensional vector of predictors, and let $\boldsymbol{X}_k = (\boldsymbol{x}_{k1}, ..., \boldsymbol{x}_{kn_k})^T$. Although the **glmmPen** package allows for different n_k for the K groups, we will set $\{n_k\}_{k=1}^{K} = n$ for simplicity of notation in future equations. In GLMMs, we assume that the conditional distribution of $\boldsymbol{y}_k$ given $\boldsymbol{X}_k$ belongs to the exponential family and has the following density:

$$f(\boldsymbol{y}_k|\boldsymbol{X}_k, \boldsymbol{\alpha}_k; \theta) = \prod_{i=1}^{n} c(y_{ki}) \exp\big[\tau^{-1}\{y_{ki}\eta_{ki} - b(\eta_{ki})\}\big], \tag{3.1}$$

where $c(y_{ki})$ is a constant that only depends on y_{ki}, τ is the dispersion parameter, $b(\cdot)$ is a known link function, and η_{ki} is the linear predictor. The **glmmPen** algorithm currently allows for the Gaussian, Binomial, and Poisson families with canonical links.

In the GLMM, the linear predictor has the form

$$\eta_{ki} = \boldsymbol{x}_{ki}^T\boldsymbol{\beta} + \boldsymbol{z}_{ki}^T\boldsymbol{\Gamma}\boldsymbol{\alpha}_k, \tag{3.2}$$

where $\boldsymbol{\beta} = (\beta_1, ..., \beta_p)^T$ is a p-dimensional vector for the fixed effects coefficients (including the intercept), $\boldsymbol{\alpha}_k$ is a q-dimensional vector of unobservable random effects (including the random intercept), $\boldsymbol{z}_{ki}$ is a q-dimensional subvector of $\boldsymbol{x}_{ki}$, and $\boldsymbol{\Gamma}$ is a lower triangular matrix.

In Rashid *et al.* (2020), the random effects vector $\boldsymbol{\alpha}_k$ is assumed to follow $N_q(\boldsymbol{0}, \boldsymbol{I})$ so that $\boldsymbol{\Gamma}\boldsymbol{\alpha}_k$ follows $N(\boldsymbol{0}, \boldsymbol{\Gamma}\boldsymbol{\Gamma}^T)$. In this way, the random component of the linear predictor has variance Var($\boldsymbol{\Gamma}\boldsymbol{\alpha}_k$) = $\boldsymbol{\Gamma}\boldsymbol{\Gamma}^T$.

To simplify the procedure of estimating $\boldsymbol{\Gamma}$, we consider a vector $\boldsymbol{\gamma}$ containing all of the nonzero elements of $\boldsymbol{\Gamma}$ such that $\boldsymbol{\gamma}_t$ is a t x 1 vector consisting of nonzero elements of the t^{th} row of $\boldsymbol{\Gamma}$ and $\boldsymbol{\gamma} = (\boldsymbol{\gamma}_1^T, ..., \boldsymbol{\gamma}_q^T)^T$. We can then reparameterize the linear predictor (Chen and Dunson, 2003; Ibrahim *et al.*, 2011) to

$$\eta_{ki} = \boldsymbol{x}_{ki}^T\boldsymbol{\beta} + \boldsymbol{z}_{ki}^T\boldsymbol{\Gamma}\boldsymbol{\alpha}_k = \begin{pmatrix}\boldsymbol{x}_{ki}^T & (\boldsymbol{\alpha}_k \otimes \boldsymbol{z}_{ki})^T\boldsymbol{J}_q\end{pmatrix}\begin{pmatrix}\boldsymbol{\beta} \\ \boldsymbol{\gamma}\end{pmatrix} \tag{3.3}$$

where $\boldsymbol{J}_q$ is a matrix that transforms $\boldsymbol{\gamma}$ to vec($\boldsymbol{\Gamma}$) such that vec($\boldsymbol{\Gamma}$) = $\boldsymbol{J}_q\boldsymbol{\gamma}$. $\boldsymbol{J}_q$ is of dimension $q^2 \times q(q+1)/2$ when the random effects covariance matrix $\boldsymbol{\Gamma}\boldsymbol{\Gamma}^T$ is unstructured; alternatively, $\boldsymbol{J}_q$ is of dimension $q^2 \times q$ when the random effects covariance matrix has an independence structure (i.e. diagonal). The vector of parameters $\boldsymbol{\theta} = (\boldsymbol{\beta}^T, \boldsymbol{\gamma}^T, \tau)^T$ are the main parameters of interest. We denote the true value of $\boldsymbol{\theta}$ as $\boldsymbol{\theta}^* = (\boldsymbol{\beta}^{*T}, \boldsymbol{\gamma}^{*T}, \tau^*)^T =$ argmin$_{\boldsymbol{\theta}}$E$_{\boldsymbol{\theta}}[-\ell(\boldsymbol{\theta})]$ where $\ell(\boldsymbol{\theta})$ is the observed marginal log-likelihood across all K groups such that $\ell(\boldsymbol{\theta}) = \sum_{k=1}^{K} \ell_k(\boldsymbol{\theta})$, $\ell_k(\boldsymbol{\theta}) = (1/n) \log \int f(\boldsymbol{y}_k | \boldsymbol{X}_k, \boldsymbol{\alpha}_k; \boldsymbol{\theta}) \phi(\boldsymbol{\alpha}_k) d\boldsymbol{\alpha}_k$.

Let us consider the high dimensional case where we want to select the true nonzero fixed and true nonzero random effects. In other words, we aim to identify the set

$$S = S_1 \cup S_2 = \{j : \beta_j^* \neq 0\} \cup \{t : ||\boldsymbol{\gamma}_t^*||_2 \neq 0\},$$

where the set S_1 represents the selection of true nonzero fixed effects and the set S_2 represents the selection of true nonzero random effects. When $\boldsymbol{\gamma}_t = \mathbf{0}$, this sets row t of $\boldsymbol{\Gamma}$ entirely equal to 0, indicating that effect of covariate t is fixed across the K groups.

We aim to solve the following penalized likelihood:

$$\widehat{\boldsymbol{\theta}} = \text{argmin}_{\boldsymbol{\theta}} - \ell(\boldsymbol{\theta}) + \lambda_0 \sum_{j=1}^{p} \rho_0 \left(\beta_j\right) + \lambda_1 \sum_{t=1}^{q} \rho_1 \left(||\boldsymbol{\gamma}_t||_2\right), \tag{3.4}$$

where $\ell(\boldsymbol{\theta})$ is the observed marginal log-likelihood for all K groups defined earlier, $\rho_0(t)$ and $\rho_1(t)$ are general folded-concave penalty functions, and λ_0 and λ_1 are positive tuning parameters. In the **glmmPen** package, the $\rho_0(t)$ penalty function options include the LASSO L_1 penalty, the SCAD penalty, and the MCP penalty (Friedman *et al.*, 2010; Breheny and Huang, 2011). For the $\rho_1(t)$ penalty, we treat the elements of $\boldsymbol{\gamma}_t$ as a group and penalize them in a groupwise manner using the group LASSO, group MCP, or group SCAD penalties presented by Breheny and Huang (2015). These groups of $\boldsymbol{\gamma}_t$ are then estimated to be either all zero or all nonzero. In this way, we select covariates to have varying effects ($\widehat{\boldsymbol{\gamma}}_{\boldsymbol{t}} \neq \mathbf{0}$) or fixed effects ($\widehat{\boldsymbol{\gamma}}_{\boldsymbol{t}} = \mathbf{0}$) across the K groups.

Similar to other variable selection packages such as package **ncvreg** (Breheny and Huang, 2011), we standardize the fixed effects covariates matrix $\boldsymbol{X} = (\boldsymbol{X}_1^T, ..., \boldsymbol{X}_K^T)^T$ such that $\sum_{k=1}^K \sum_{i=1}^{n_k} x_{ki,j} = 0$ and $N^{-1} \sum_{k=1}^K \sum_{i=1}^{n_k} x_{ki,j}^2 = 1$ for $j = 1, ..., p$. Although the package **grpreg** (Breheny and Huang, 2015) orthogonalizes grouped effects, we have found through simulations that first standardizing the fixed effects and then using subsets of these standardized fixed effects for the random effects (recall: $\boldsymbol{z}_{ki}$ is a q-dimensional subvector of $\boldsymbol{x}_{ki}$) is sufficient. During the selection procedure, the fixed effects intercept and the random effects intercept remain unpenalized.

3.3 MCECM algorithm

We solve equation (3.4) for a specific (λ_0, λ_1) penalty parameter combination using a Monte Carlo Expectation Conditional Minimization (MCECM) algorithm (Garcia *et al.*, 2010). The MCECM algorithm described in this section uses many of the steps and assumptions described in Rashid *et al.* (2020), but here we provide further practical details about the E-step, M-step, and initialization. Additionally, the implementation outlined in this chapter has several improvements to the implementation used in Rashid *et al.* (2020). Compared to the Rashid *et al.* (2020) implementation, the E-step in **glmmPen** allows for several possible sampling schemes, including the fast and efficient No-U-Turn Hamiltonian Monte Carlo sampling procedure (NUTS) from the Stan software (Carpenter *et al.*, 2017; Hoffman and Gelman, 2014). The **glmmPen** package was also able to reduce the required memory usage of the MCECM algorithm. In the M-step, we utilized the fast looping capability of packages **Rcpp** and **RcppArmadillo**, which allowed for the fast recalculation of rows of the large matrix (Step 3 of the M-step discussed in Section 3.3.2) so this large matrix did not have to be stored.

During the MCECM algorithm, we aim to evaluate the expected value of (E-step) and minimize (M-step) the following penalized Q-function in the s^{th} iteration of the algorithm:

$$\begin{aligned} Q_\lambda(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s)}) &= \sum_{k=1}^{K} E\left\{-\log\big(f(\boldsymbol{y}_k, \boldsymbol{X}_k, \boldsymbol{\alpha}_k; \boldsymbol{\theta}|\boldsymbol{d}_o; \boldsymbol{\theta}^{(s)})\big)\right\} + \lambda_0 \sum_{j=1}^{p} \rho_0\left(\beta_j\right) + \lambda_1 \sum_{t=1}^{q} \rho_1\left(||\boldsymbol{\gamma}_t||_2\right) \\ &= Q_1(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s)}) + Q_2(\boldsymbol{\theta}^{(s)}) + \lambda_0 \sum_{j=1}^{p} \rho_0\left(\beta_j\right) + \lambda_1 \sum_{t=1}^{q} \rho_1\left(||\boldsymbol{\gamma}_t||_2\right), \end{aligned} \tag{3.5}$$

where $(\boldsymbol{y}_k, \boldsymbol{X}_k, \boldsymbol{\alpha}_k)$ gives the complete data for group k, $d_{k,o} = (\boldsymbol{y}_k, \boldsymbol{X}_k)$ gives the observed data for group k, $\boldsymbol{d}_o$ represents the entirety of the observed data, and

$$Q_1(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s)}) = -\sum_{k=1}^{K} \int \log[f(\boldsymbol{y}_k|\boldsymbol{X}_k, \boldsymbol{\alpha}_k; \boldsymbol{\theta})]\phi(\boldsymbol{\alpha}_k|\boldsymbol{d}_{k,o}; \boldsymbol{\theta}^{(s)})d\boldsymbol{\alpha}_k, \tag{3.6}$$

$$Q_2(\boldsymbol{\theta}^{(s)}) = -\sum_{k=1}^{K} \int \log \phi(\boldsymbol{\alpha}_k) \phi(\boldsymbol{\alpha}_k | \boldsymbol{d}_{k,o}; \boldsymbol{\theta}^{(s)}) d\boldsymbol{\alpha}_k \tag{3.7}$$

3.3.1 Monte-Carlo E-step

The integrals in the Q-function do not have closed forms when $f(\boldsymbol{y}_k|\boldsymbol{X}_k, \boldsymbol{\alpha}_k^{(s,m)}; \boldsymbol{\theta})$ is assumed to be non-Gaussian, and become difficult to approximate as q increases. Consequently, we approximate these integrals using a Markov Chain Monte Carlo (MCMC) sample of size M from the posterior density $\phi(\boldsymbol{\alpha}_k|\boldsymbol{d}_{k,o}; \boldsymbol{\theta}^{(s)})$. The **glmmPen** package can draw samples from this posterior using one of several techniques: the No-U-Turn Hamiltonian Monte Carlo sampling procedure (NUTS) implemented by the Stan software, which **glmmPen** calls using the **rstan** package (Carpenter *et al.*, 2017) (default, and strongly recommended for its speed and efficiency); Metropolis-within-Gibbs with an adaptive random walk sampler (Roberts and Rosenthal, 2009); and Metropolis-within-Gibbs with an independence sampler (Givens and Hoeting, 2012). Each sampler type uses a standard normal candidate distribution. Let $\boldsymbol{\alpha}_k^{(s,m)}$ be the m^{th} simulated value, $m = 1, ..., M$, at the s^{th} iteration of the algorithm for group k. The integral in equation (3.6) can be approximated as

$$Q_1(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s)}) \approx -\frac{1}{M} \sum_{m=1}^{M} \sum_{k=1}^{K} \log f(\boldsymbol{y}_k|\boldsymbol{X}_k, \boldsymbol{\alpha}_k^{(s,m)}; \boldsymbol{\theta}).$$

Although the optimal number of MCMC samples $M^{(s)}$ in the E-step at EM iteration s is not well defined, the general consensus is that a smaller sample size of the posterior is suitable for the start of the algorithm but larger sample sizes are needed later in the algorithm (Booth and Hobert, 1999). Then in a manner similar to the **mcemGLM** package (Archila, 2020), the MCMC sample size is increased by a multiplicative factor f at each step of the algorithm such that $M^{(s)} = f * M^{(s-1)}$ until either the value of $M^{(s)}$ reaches its maximum allowed value or the EM algorithm converges.

3.3.2 M-step

In the M-step of the algorithm, we aim to minimize

$$Q_{1,\lambda}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s)}) = Q_1(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s)}) + \lambda_0 \sum_{j=1}^{p} \rho_0(\beta_j) + \lambda_1 \sum_{t=1}^{q} \rho_1(||\boldsymbol{\gamma}_t||_2) \tag{3.8}$$

with respect to $\boldsymbol{\theta} = (\boldsymbol{\beta}^T, \boldsymbol{\gamma}^T, \tau)^T$. The minimization of equation (3.8) with respect to $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ is performed using a Majorization-Minimization approach. For the general exponential family, Rashid et al. (2014) suggested minimizing with respect to τ using the standard optimization algorithm Newton-Raphson. In **glmmPen**, the only family implemented with a dispersion parameter is the Gaussian family, and the variance σ^2 can be estimated directly from a derivation of the Q-function conditional on the most recent updates of $\boldsymbol{\beta}^{(s)}$ and $\boldsymbol{\gamma}^{(s)}$:

$$\sigma^2 = \frac{1}{M * N} \sum_{m=1}^{M} \sum_{k=1}^{K} \sum_{i=1}^{n_k} (y_{ki} - \eta_{ki}^{(s,m)})^2, \tag{3.9}$$

where $\eta_{ki}^{(s,m)}$ is the linear predictor η_{ki} evaluated with $\boldsymbol{\beta}^{(s)}$, $\boldsymbol{\gamma}^{(s)}$, and sample $\boldsymbol{\alpha}_k^{(s,m)}$.

Let s represent the iteration of the MCECM algorithm, and f represent the iteration within a particular M-step of the MCECM algorithm. The M-step of the s^{th} iteration of the MCECM algorithm proceeds as given in the steps of Algorithm 1.

The algorithm recomputes the augmented matrices $\tilde{\boldsymbol{z}}_{ki}$ for $k = 1, ..., K$ and $i = 1, ..., n_k$ in step 3 of every M-step iteration f for several reasons. These repeat calculations prevent the algorithm from having to store the augmented matrix $\tilde{\boldsymbol{Z}} = (\tilde{\boldsymbol{Z}}_1^T, ..., \tilde{\boldsymbol{Z}}_K^T)^T$ where $\tilde{\boldsymbol{Z}}_k = (\tilde{\boldsymbol{z}}_{k1}^T, ..., \tilde{\boldsymbol{z}}_{kn_k}^T)^T$. This full augmented matrix is of dimension $(M * N) \times q(q+1)/2$ or $(M * N) \times q$ depending on whether the random effect covariance matrix is unstructured or independent, respectively. As the MCMC sample size increases throughout the algorithm and as q increases, saving this $\tilde{\boldsymbol{Z}}$ becomes more and more memory prohibitive even when utilizing large matrix implementation tools such as the package **bigmemory** (Kane *et al.*, 2013). Through simulations not shown here, we found that recomputing the $\tilde{\boldsymbol{z}}_{ki}$ matrices during each M-step iteration utilizing **Rcpp** (Eddelbuet-

Algorithm 1 M-step of the MCECM algorithm

```
1. The parameters $\boldsymbol{\theta}^{(s,0)}$ for M-step iteration $f = 0$ are initialized using the results from the previous M-step, $\boldsymbol{\theta}^{(s-1)}$.
2. Conditional on $\boldsymbol{\gamma}^{(s,f-1)}$ and $\tau^{(s-1)}$, each $\beta_j^{(s,f)}$ for $j = 1, ..., p$ is given a single update using the Majorization-Minimization algorithm specified by Breheny and Huang (2015).
3. For each group k in $k = 1, ..., K$, the augmented matrix $\tilde{\boldsymbol{z}}_{ki} = (\tilde{\boldsymbol{\alpha}}_k^{(s)} \otimes \boldsymbol{z}_{ki})J_q$ is created for $i = 1, ..., n_k$ where $\tilde{\boldsymbol{\alpha}}_k^{(s)} = ((\boldsymbol{\alpha}_k^{(s,1)})^T, ..., (\boldsymbol{\alpha}_k^{(s,M)})^T)^T$. This augmented matrix is used in the random effect portion of the linear predictor specified in equation (3.2). The dimension of $\tilde{\boldsymbol{z}}_{ki}$ is $M \times q(q+1)/2$ for an unstructured covariance matrix and $M \times q$ for an independent covariance matrix. This augmented matrix is used to calculate equation (2.9) in Breheny and Huang (2015).
4. Conditional on the $\tau^{(s-1)}$ and the recently updated $\boldsymbol{\beta}^{(s,f+1)}$, each $\boldsymbol{\gamma}_t^{(s,f)}$ for $t = 1, ..., q$ is updated using the Majorization-Minimzation coordinate descent grouped variable selection algorithm specified by Breheny and Huang (2015), except the residuals are not updated after every $\boldsymbol{\gamma}_t^{(s,f)}$ coefficient update.
5. Steps 2 through 4 are repeated until the M-step convergence criteria are reached or until the M-step reaches its maximum number of iterations.
6. Conditioning on the newly updated $\boldsymbol{\beta}^{(s)}$ and $\boldsymbol{\gamma}^{(s)}$, $\tau^{(s)}$ is updated (generically, using the Newton-Raphson algorithm; for Gaussian family, using equation (3.9)).
```

tel and François, 2011) and **RcppArmadillo** (Eddelbuettel and Sanderson, 2014) significantly reduced the time and memory required to compute each M step.

In step 4 of the M-step, the residuals are not updated after every update to the random effects coefficients $\boldsymbol{\gamma}_t^{(s,f)}$ for $t = 1, ..., q$ in order to speed up computation. Otherwise, this would require re-calculation of the augmented matrix specified in step 3 for each random effect (q) for each M-step. When q is large, this makes the M-step prohibitively time-consuming. Simplifying step 4 with no residual updates speeds up the computation time in high dimensional settings and was found to have negligible impact on estimation accuracy.

The full MCECM algorithm then proceeds as given in the steps of Algorithm 2.

For details about initialization of the first model in the variable selection model sequence, see Section B.1 in the Appendix for Chapter 3. Details about the initialization of subsequent models in the model sequence are given in Section 3.4.

This MCECM algorithm is able to handle large dimensions of p and q, where p and q are much larger than prior methods for simultaneous fixed and random effects variable selection

Algorithm 2 Full MCECM algorithm for single (λ_0, λ_1) penalty combination

1. Fixed and random effects $\boldsymbol{\beta}^{(0)}$ and $\boldsymbol{\gamma}^{(0)}$ are initialized as discussed later in Section B.1.
2. In each E-step for EM iteration s, a burn-in sample from the posterior distribution of the random effects is run and discarded. A sample of size $M^{(s)}$ from the posterior is then drawn and retained for the M-step.
3. Parameter estimates of $\boldsymbol{\beta}^{(s)}$, $\boldsymbol{\gamma}^{(s)}$, and $\tau^{(s)}$ are then updated as described in the M-step procedure given above.
4. Steps 2 and 3 are repeated until the convergence condition is met at least 2 consecutive times (default) or until the maximum number of EM iterations is reached.
5. Using the estimates of $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$, and τ at EM convergence, a final sample from the posterior distribution of the random effects is drawn for use in the calculation of the marginal log-likelihood as well as for diagnostics of the MCMC chain. The marginal log-likelihood is used for model selection and is discussed in detail in Section 3.4.

(Rashid *et al.*, 2020). When the number of random effect predictors is greater than or equal to 10, we recommend approximating the random effect covariance matrix $\boldsymbol{\Gamma}\boldsymbol{\Gamma}^T$ as a diagonal matrix. In the mixed model setting, Fan and Li (2012) demonstrated both theoretical and empirical advantages to estimating the random effects covariance matrix this way in the high-dimensional case. This simplification also has the advantage of enabling the package to have greater computational efficiency when fitting high-dimensional models.

The MCECM algorithm outlined above describes how the **glmmPen** package estimates the model parameters for a single set of penalty parameters (λ_0, λ_1). Section 3.4 discusses how the package chooses the best model during the model selection procedure.

3.4 Model selection

This section provides details on how the **glmmPen** algorithm selects the optimal tuning parameter combination. When there is no *a priori* knowledge of the true random effects, we recommend that the user set the random effects equal to the fixed effects (i.e. $p = q$) and let the algorithm select the fixed and random effects using the procedure outlined in Section 3.4. However, if the user has prior knowledge about restrictions on the random effects, they can restrict the potential random effects to an appropriate subset. As discussed in the previous section, the package requires that the random effects be a subset of the fixed effects.

We assume in this section and the rest of the paper that p represents the number of fixed effect predictors (not including the fixed intercept), and q represents the number of random effect predictors (not including the random intercept).

3.4.1 Penalty sequence specification

In the selection process, the fixed effects are penalized concurrently with the random effects. By default, we set the sequence of penalty parameters $\boldsymbol{\lambda}_0 = \boldsymbol{\lambda}_1$. The sequence $\boldsymbol{\lambda}_0 = (\lambda_{0,1}, ..., \lambda_{0,\omega_0})$ and $\boldsymbol{\lambda}_1 = (\lambda_{1,1}, ..., \lambda_{1,\omega_1})$ are calculated in a similar manner to the approach used by the package **ncvreg** (Breheny and Huang, 2011). The maximum penalty parameter λ_{max} is calculated using the same procedure as **ncvreg**; this value penalizes all fixed and random effects coefficients to 0. The minimum penalty parameter λ_{min} is a small portion of the λ_{max}.

3.4.2 Tuning parameter search strategy

By default, the algorithm runs a computationally efficient two-stage approach to pick the optimal set of tuning parameters. In the first stage of this approach, the algorithm fits a sequence of models where the fixed effect penalty is kept constant at the minimum value of $\boldsymbol{\lambda}_0$, $\lambda_{0,min}$, and the random effects penalty proceeds from the minimum value of $\boldsymbol{\lambda}_1$, $\lambda_{1,min}$, to the maximum value $\lambda_{1,max}$. The best model from this first stage is then identified using BIC-based selection criteria, described in more detail later in this section. This first stage identifies the optimal random effect penalty value, $\lambda_{1,opt}$. In the second stage, the algorithm fits a sequence of models where the random effects penalty is kept fixed at $\lambda_{1,opt}$ and the fixed effects penalty $\boldsymbol{\lambda}_0$ proceeds from $\lambda_{0,min}$ to $\lambda_{0,max}$. The overall best model is chosen from the models in the second stage. In both stages, the results from each model are used to initialize the coefficients in the subsequent model in the sequence.

Unlike other packages that perform variable selection, such as **ncvreg** and **grpreg**, we run the λ sequence from λ_{min} to λ_{max} and not the traditional progression of λ_{max} to λ_{min}. In this mixed model setting, this approach provides better initialization of subsequent models in the tuning parameter sequence, giving an overall better performance to the algorithm and improving algorithm speed. The algorithm also speeds up the algorithm by strategically restricting the random effects as the algorithm proceeds. In stage one, if a previous model penalized out random effects from the model, the following model will automatically ignore these random effects. In stage two, the random effects considered are restricted to the non-zero random effects from the best model in stage one.

We have found this two-stage approach, which we also refer to as an 'abbreviated grid search', to work very well in practice (see Section 3.5 for performance results).

3.4.3 Optimal tuning parameter selection

Once models have been fit for all (λ_0, λ_1) combinations within the first and second stages of the tuning parameter search strategy (or over the full tuning parameter grid search), the **glmmPen** package chooses the best model from one of several BIC-type selection criteria options. By default, the package uses the BIC-ICQ criterion (Ibrahim *et al.*, 2011), which is expressed below:

$$\begin{aligned} \text{BICq}(\boldsymbol{\theta}_\lambda) &= 2\{Q_1(\boldsymbol{\theta}_\lambda|\boldsymbol{\alpha}_0) + Q_2(\boldsymbol{\alpha}_0)\} + d_\lambda * \log(N) \\ &\approx \left\{ -\frac{2}{M} \sum_{m=1}^{M} \sum_{k=1}^{K} \left[\log f(\boldsymbol{y}_k|\boldsymbol{X}_k, \boldsymbol{\alpha}_{0,k}^{(m)}; \boldsymbol{\theta}_\lambda) + \log \phi(\alpha_{0,k}^{(m)}) \right] \right\} + d_\lambda * \log(N), \end{aligned} \tag{3.10}$$

where $\boldsymbol{\theta}_\lambda$ are the coefficients of the penalized model, $\boldsymbol{\alpha}_0$ are the posterior samples from a 'full' model with either no penalty or a minimum penalty used on the fixed and random effects, and $\alpha_{0,k}^{(m)}$ is the m^{th} posterior sample for group k from such a full model, Q_1 and Q_2 were defined in Section 3.3, d_λ is the number of nonzero coefficients for the model (all nonzero fixed effects coefficients $\boldsymbol{\beta}$ plus all nonzero random effects coefficients $\boldsymbol{\gamma}$), and N is the total number of observations in the data (N_{obs}).

The package can also calculate the traditional BIC criterion, specifying N in the penalty term as either the total number of observations in the data (N_{obs}) or the total number of independent observations (i.e. number of levels within the grouping factor, N_{grps}) as well as the hybrid

BICh given by Delattre *et al.* (2014). These terms are defined in Section 1.2.3. We use the Corrected Arithmetic Mean Estimator (CAME) defined by Pajor (2017) to calculate the marginal log-likelihood used within the BIC and BICh calculations. See Section 1.2.3 for further details on this calculation.

In simulations not shown here, we found that the BIC-ICQ gave the best performance in choosing the correct fixed effects model. The BIC and BICh methods tended to underestimate the number of true fixed effects compared to BIC-ICQ in the simulations we considered. However, in order to calculate the BIC-ICQ, a 'full' model needs to be fit using either no penalty or a small penalty on the fixed and random effects. Posterior samples from this full model are then used to calculate the BIC-ICQ value for each model fit in the variable selection procedure. Depending on the size of the full model, this calculation could take a lot of time. Alternatively, the calculation of the BIC and BICh criteria require a calculation of the marginal log-likelihood $\ell(\boldsymbol{\theta})$ for each model. Since the integrals within $\ell(\boldsymbol{\theta})$ are intractable, we estimate the marginal log-likelihood using the Corrected Arithmetic Mean Estimator (CAME) described by Pajor (2017). We have found this CAME estimator to be relatively quick and easy to calculate, as well as consistent with the marginal log-likelihood estimate calculated by the package **lme4** (Bates *et al.*, 2015) for a range of conditions (results not shown here).

3.5 Simulations

In this section, we present results from simulations in order to examine the performance of our package. We use the **glmmPen** package to perform variable selection on logistic mixed effects models and examine the resulting fixed effects estimates as well as the true and false positives for the fixed and random effects. All simulations are performed using the default optimization settings specified in the **glmmPen** package documentation.

3.5.1 Simulation set-up

We simulated binary responses from a logistic mixed effects model with p predictors. Of p total predictors, we assume that two predictors have truly non-zero fixed and random effects, and the other $p - 2$ predictors have zero-valued fixed and random effects. Our aim in the simulations was to select the true predictors.

In these simulations, we consider the following situations: predictor dimensions of $p = \{10, 50\}$, sample size $N = 500$, number of groups $K = \{5, 10\}$, and standard deviation of the random effects $\sigma = \{1, \sqrt{2}\}$. As discussed in Section 3.2, we approximate the covariance matrix of the random effects as a diagonal matrix for these higher dimensions. We further consider the scenarios of moderate predictor effects, where the true fixed effects are $\boldsymbol{\beta} = (0, 1, 1)^T$, and strong predictor effects, where the true fixed effects are $\boldsymbol{\beta} = (0, 2, 2)^T$.

For group k, we generated the binary response y_{ki}, $i = 1, ..., n_k$ such that $y_{ki} \sim Bernoulli(p_{ki})$ where $p_{ki} = P(y_{ki} = 1|\boldsymbol{x}_{ki}, \boldsymbol{z}_{ki}, \boldsymbol{\alpha}_k, \boldsymbol{\theta}) = exp(\boldsymbol{x}_{ki}^T\boldsymbol{\beta} + \boldsymbol{z}_{ki}^T\boldsymbol{\alpha}_k)/\{1 + exp(\boldsymbol{x}_{ki}^T\boldsymbol{\beta} + \boldsymbol{z}_{ki}^T\boldsymbol{\alpha}_k)\}$, and $\boldsymbol{\alpha}_k \sim N_3(0, \sigma^2\boldsymbol{I}_3)$. The fixed effect coefficients were set to $\boldsymbol{\beta} = (0, 1, 1)^T$ (moderate predictor effects) and $(0, 2, 2)^T$ (strong predictor effects). We also simulated imbalance in sample sizes between the groups. Of the N samples, $N/3$ samples were given to study $k = 1$ and the remaining $2N/3$ samples were evenly distributed among the remaining studies. Each condition was evaluated using 100 total simulated datasets.

For individual i in group k, the vector of predictors for the fixed effects was $\boldsymbol{x}_{ki} = (1, x_{ki,1}, ..., x_{ki,p})^T$, and we set the random effects $\boldsymbol{z}_{ki} = \boldsymbol{x}_{ki}$, where $x_{ki,j} \sim N(0, 1)$ for $j = 1, ..., p$.

Setting the input random effects equal to the fixed effects represents the worst-case scenario where we have no idea what predictors do or do not have random effects. This is an extreme assumption; in many real-world scenarios, users will have reason to set the input random effects to a strict subset of the fixed effects.

In all of these simulations, we use BIC-ICQ for the selection criteria, pre-screening, and the MCP penalty. For all simulations, we performed the abbreviated two-stage grid search as described in Section 3.4. The results for these simulations are presented in Tables 3.1 and 3.2. These results include the average coefficients, true positive and false positive percentages for both fixed and random effects, and the median time for the simulations to complete. The true positive percentages reflect the average percent of the true predictors included in the best models chosen by the BIC-ICQ model selection criteria, which should ideally be near 100%. Likewise, the false positive percentages reflect the average percent of the false predictors included in the best models, which should ideally be near 0%. All simulations were completed on the UNC Longleaf computing cluster (CPU Intel processors between 2.3Ghz and 2.5GHz).

N	p	K	σ	$\hat{\beta}_1$	$\hat{\beta}_2$	TP % Fixed	FP % Fixed	TP % Random	FP % Random	T^{median} (hours)
500	10	5	1	1.02	1.12	89.0	2.1	90.5	3.5	0.20
			$\sqrt{2}$	1.12	1.18	83.0	1.4	96.0	3.6	0.26
		10	1	0.99	1.04	99.0	3.0	95.0	4.8	0.24
			$\sqrt{2}$	1.02	1.11	91.0	1.8	99.5	7.0	0.32
500	50	5	1	1.18	1.14	84.5	1.2	83.5	2.2	8.07
			$\sqrt{2}$	1.42	1.43	75.5	2.5	89.0	2.5	12.20
		10	1	1.12	1.11	95.0	1.8	93.0	3.9	10.67
			$\sqrt{2}$	1.33	1.31	84.5	2.4	95.5	6.2	15.75

Table 3.1: Variable selection simulation results with moderate predictor effects (slopes equal to 1). Results include the estimated coefficients for true non-zero fixed effects, true positive (TP) percentages for fixed and random effects, false positive (FP) percentages for fixed and random effects, and the median time in hours for the algorithm to complete.

By examining the simulation results, we can observe that the performance of the variable selection procedure in **glmmPen** is impacted by the underlying structure of the data. As the magnitude of the random effect variance increases, the true positive percentage of the fixed effects decreases and the true positive percentage of the random effects increases. Additionally, as the number of groups K increases, the true positive percentage of both the fixed and random effects increases. We see that as the dimension of the total number of predictors increases ($p = 10$ to $p = 50$), the true positive percentages of both the fixed and random effects decreases. In regards

N	p	K	σ	$\hat{\beta}_1$	$\hat{\beta}_2$	TP %	FP %	TP %	FP %	T^{median}
						Fixed	Fixed	Random	Random	(hours)
500	10	5	1	1.95	2.08	100.0	1.4	76.5	2.6	0.24
			$\sqrt{2}$	1.93	2.12	99.5	3.1	90.0	3.1	0.33
		10	1	2.05	2.11	100.0	0.9	87.0	5.5	0.31
			$\sqrt{2}$	2.07	2.12	100.0	2.5	94.5	5.6	0.38
500	50	5	1	2.03	2.01	100.0	2.9	70.0	1.0	13.70
			$\sqrt{2}$	2.08	2.05	99.5	1.8	82.0	1.6	16.90
		10	1	2.19	2.25	100.0	2.2	77.0	4.5	17.29
			$\sqrt{2}$	2.06	2.18	100.0	1.0	91.0	4.2	23.48

Table 3.2: Variable selection simulation results with strong predictor effects (slopes equal to 2). Results include the estimated coefficients for true non-zero fixed effects, true positive (TP) percentages for fixed and random effects, false positive (FP) percentages for fixed and random effects, and the median time in hours for the algorithm to complete.

to the run time, Tables 3.1 and 3.2 show that increases in the number of groups and increases in the variance of the random effects generally increases the time for the algorithm to complete.

3.5.2 Pre-screening performance

The time it takes the package to complete the tuning parameter selection procedure depends strongly on the number of random effects considered by the algorithm. Therefore, the pre-screening procedure, which reduces the number of random effects considered within the variable selection algorithm, speeds up the algorithm. Tables 3.3 and 3.4 report the average percent of true positive and false positive random effect predictors that remain under consideration within the variable selection procedure after the pre-screening step has completed. The random effect penalty in the pre-screening step was $0.01\lambda_{max}$ for $p = 10$ and $0.05\lambda_{max}$ for $p = 50$.

Using this higher penalty in the $p = 50$ simulations helps reduce the false positive percentage of the random effects after pre-screening and consequently helps speed up the time of the algorithm to complete. However, we can see by comparing the $p = 50$ and $p = 10$ simulations that this approach can also slightly decrease the true positive percentage. In general, increasing the random effect penalty will help decrease the number of false positive non-zero random effects

N	p	K	σ	TP %	FP %
500	10	5	1	98.0	25.8
			$\sqrt{2}$	100.0	26.1
		10	1	100.0	33.0
			$\sqrt{2}$	100.0	32.2
500	50	5	1	96.0	24.6
			$\sqrt{2}$	96.5	25.7
		10	1	97.5	25.9
			$\sqrt{2}$	98.5	27.3

Table 3.3: Pre-screening results for variable selection simulations with moderate predictor effects (slopes equal to 1). Results include the true positive percentages and false positive percentages of the random effects remaining after the pre-screening procedure.

N	p	K	σ	TP	FP
500	10	5	1	93.5	26.0
			$\sqrt{2}$	97.5	25.5
		10	1	96.0	30.9
			$\sqrt{2}$	98.5	26.8
500	50	5	1	85.5	21.8
			$\sqrt{2}$	94.0	21.6
		10	1	90.5	24.6
			$\sqrt{2}$	97.0	24.0

Table 3.4: Pre-screening results for variable selection simulations with strong predictor effects (slopes equal to 2). Results include the true positive percentages and false positive percentages of the random effects remaining after the pre-screening procedure.

in the pre-screening step, but it may also decrease the number of true positive non-zero random effects. Decreasing this penalty will generally have the opposite effect.

Looking at these pre-screening results, we can also see some general patterns in the performance of the pre-screening step based on the magnitudes of the underlying model parameters. When we compare the pre-screening results between the two predictor effect magnitudes, we see that the true positive percentage of the random effects after pre-screening are lower in the strong predictor effect simulations compared to the moderate predictor effect simulations. We also see that the true positive percentage is generally higher when the magnitude of the true random effect variance is higher.

3.6 Application to real data

The main motivation for the development of these methods was to enhance replicability and generalizability of gene signature selection and subsequent downstream prediction tasks in genomic studies. Multiple approaches have been proposed to combine gene expression data from different studies and calculate improved prediction models (Riester *et al.*, 2014; Ma *et al.*, 2018; Patil and Parmigiani, 2018). Here, we fit a GLMM that can accommodate a high number of features in the model, where it is assumed that the effects of some of these features may randomly vary across studies. By fitting such a model, we can account for between-study heterogeneity and identify more accurate fixed effects estimates for prediction without knowing *a priori* which features will be fixed in their effects across studies versus randomly varying.

We will use the `basal` dataset to demonstrate the utility of the **glmmPen** package. This dataset is composed of four datasets combined from studies that contain gene expression data from subjects with several types of cancer (Moffitt *et al.*, 2015; Weinstein *et al.*, 2013). Two of these datasets contain gene expression data for subjects with Pancreatic Ductal Adenocarcinoma (PDAC), one dataset contains data for subjects with Breast Cancer, and the fourth dataset contains data for subjects with Bladder Cancer. In these datasets, tumor samples are classified as basal-like or non-basal-like. We aim to create a prediction model to correctly classify patient tumors into these categories, which have implications for patient survival (Moffitt *et al.*, 2015).

The data in these studies arises from various methods to measure gene expression, which traditionally required the use of between-sample expression normalization techniques. Due to the complexity of performing this technique across a variety of expression platforms, Rashid *et al.* (2020) integrated the data together using the data integration rank transformation technique. This integration technique creates top scoring pairs (TSPs). To illustrate the interpretation of TSPs, let $g_{ki,A}$ and $g_{ki,B}$ be the raw expression of genes A and B in subject i of group k. For each gene pair $(g_{ki,A}, g_{ki,B})$, the TSP is an indicator $I(g_{ki,A} > g_{ki,B})$ which specifies which gene of the two has higher expression in the subject. We denote a TSP predictor as "GeneA_GeneB". We use the top

50 TSPs as covariates in this example, as determined by Rashid *et al.* (2020). In order to create our desired prediction model, we will use the **glmmPen** package algorithm to fit a logistic mixed effects model with these 50 TSPs as the covariates.

Figure 3.1 shows the TSPs of the most value for predicting the basal-like subtype (the TSPs with non-zero fixed effects coefficient estimates) and the TSPs with the greatest variation between studies. We see from the figure that there are 31 TSP predictors identified as being important for the prediction of the basal-like subtype. For TSPs with positive fixed effects coefficients, we can conclude that if the TSP indicator is 1 for a subject (the first gene has greater raw expression than the second gene), it increases the odds that their subtype is basal-like. Alternatively, if the TSP has a negative fixed effects coefficient, we can conclude that if the TSP indicator is 1 for a subject, it decreases the odds that their subtype is basal-like (i.e. increases the odds that their subtype is non-basal-like). We also see that there are 11 TSP predictors with varying effects across the studies, meaning that these 11 predictors have low replicability across the fours studies.

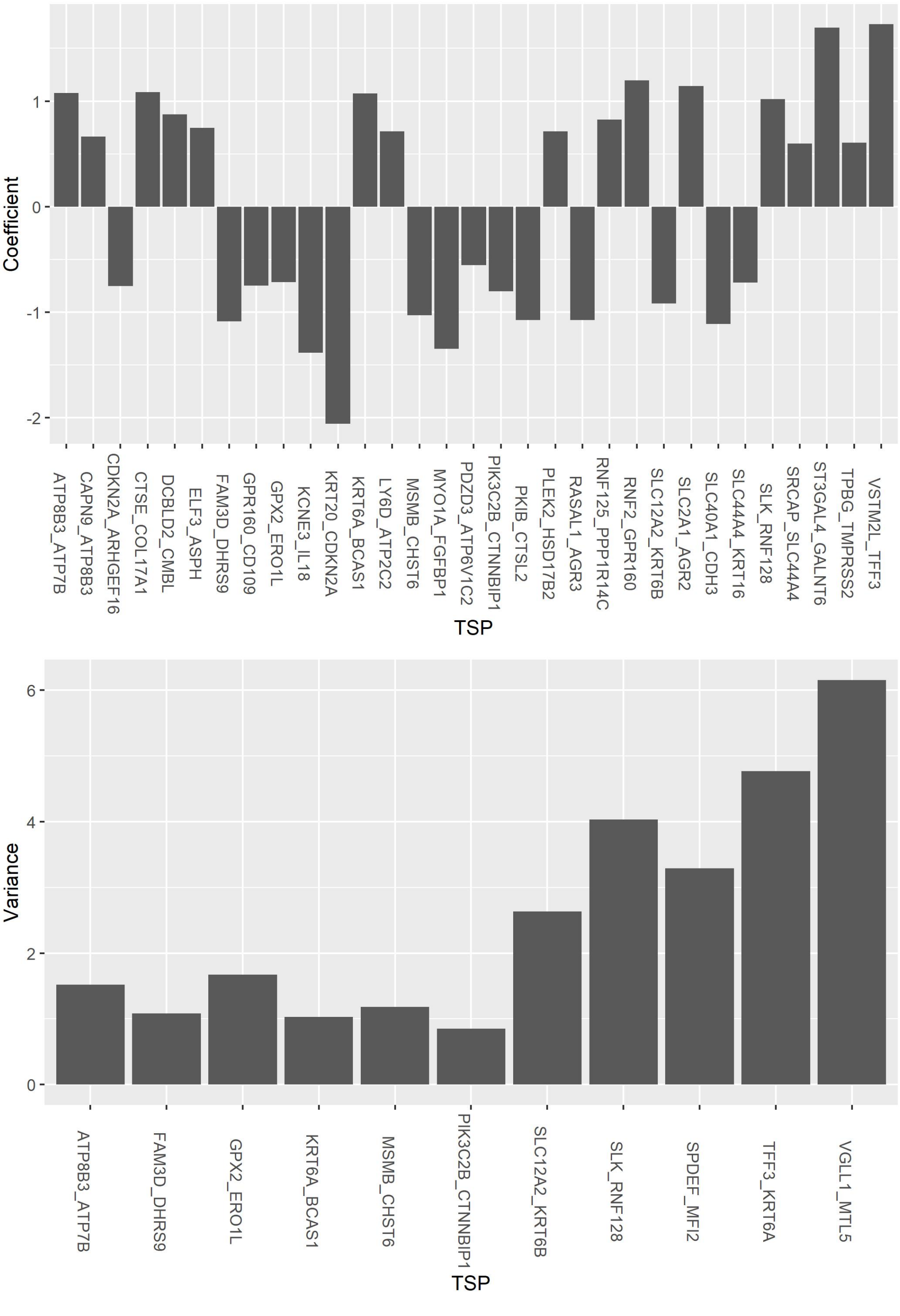


Figure 3.1: Logistic mixed effects model coefficient summaries from glmmPen applied to the Basal dataset. Top: Value of fixed effect coefficients (log odds ratios) for all TSPs with non-zero fixed effects across the four studies. Bottom: Value of random effect variances for all TSPs with non-zero random effects.

3.7 Discussion

This paper introduces the **R** package **glmmPen** for fitting penalized generalized linear mixed models, including Binomial, Gaussian, and Poisson models. The **glmmPen** package's main advantage over other packages that estimate GLMMs is that it can perform variable selection on the fixed and random effects simultaneously. The algorithm utilizes a Monte Carlo Expectation Conditional Minimization (MCECM) algorithm. Several established MCMC sampling techniques are available for the E-step, and a Majorization-Minimization coordinate descent algorithm is used in the M-step. The package utilizes the established methods of Stan and **RcppArmadillo** to increase the computational efficiency of the E-step and M-step, respectively. As a result, the **glmmPen** package can fit models with higher dimensions compared to other packages that fit GLMMs, supporting models with 50 or more fixed and random effects.

Besides the MCECM algorithm used in the model fit, the **glmmPen** package utilizes several techniques to improve the speed of the algorithm. Such techniques include initialization of subsequent models with the coefficients from the previous model fit and pre-screening to remove unnecessary random effects.

The **glmmPen** has several attributes that make it user-friendly. For one, the package was designed to have an interface that is similar to the well-known **lme4** package. Additionally, the **glmmPen** package has several automated processes that make it user-friendly. The **glmmPen** package provides automated data-dependent initialization of the random effect covariance matrix. The package also provides automated recommendations for the penalization parameters.

A unique aspect of the package is the calculation of the marginal log-likelihood. The Corrected Arithmetic Mean Estimate (CAME) calculation described by Pajor (Pajor, 2017) is relatively simple and fast to calculate, and we have found that it performs well when compared with the log-likelihood estimate used in the **lme4** package (results not shown here). This marginal log-likelihood calculation allows the algorithm to perform tuning parameter selection using traditional BIC selection criterion as well as other BIC-derived selection criteria. This gives users the option to forgo calculating the BIC-ICQ selection criterion, which requires the ‘full model’ fit where a minimum penalty is applied to both the fixed and random effects.

CHAPTER 4: EFFICIENT COMPUTATION OF HIGH-DIMENSIONAL PENALIZED GENERALIZED LINEAR MIXED MODELS BY LATENT FACTOR MODELING OF THE RANDOM EFFECTS

4.1 Introduction

Modern biomedical datasets are increasingly high dimensional and exhibit complex correlation structures. Generalized Linear Mixed Models (GLMMs) have long been employed to account for such dependencies. However, proper specification of the fixed and random effects is a critical step in estimation of GLMMs. For instance, omitting important random effects can lead to bias in the estimated variance of the fixed effects, whereas including unnecessary random effects could increase the computational difficulty of fitting the GLMM (Thompson *et al.*, 2017; Gurka *et al.*, 2011; Bondell *et al.*, 2010).

Despite the importance of properly specifying the set of fixed and random effects in such models, it is often unknown *a priori* which variables should be specified as fixed or random in the model, particularly in high dimensional settings in which the feature space is generally assumed to be sparse. As a solution to this problem, Rashid *et al.* (2020) used a Monte Carlo Expectation Conditional Minimization (MCECM) algorithm to perform variable selection in high dimensional GLMMs. Their method has since been incorporated into the **glmmPen** R package (Heiling *et al.*, 2023c). In this **glmmPen** framework, performing simultaneous variable selection on fixed and random effects in GLMMs with input dimensions of 50 or so predictors is feasible. Although this **glmmPen** framework extends the feasible dimensionality of performing variable selection within GLMMs relative to existing methods, new methodology is needed to alleviate the computational burden as the dimension increases even further and allow scalability to hundreds or thousands of predictors.

We present a novel reformulation of the GLMM, which we call **glmmPen_FA**, using a factor model decomposition of the random effects. This factor model is used as a dimension reduction tool to represent a large number of latent random effects as a function of a smaller set of latent factors. By reducing the latent space of the random effects, this new model formulation enables us to extend the feasible dimensionality of performing variable selection in GLMMs to hundreds of predictors. We estimate model parameters and perform simultaneous selection of fixed and random effects using a Monte Carlo Expectation Conditional Minimization (MCECM) algorithm. We show through simulations that through this factor model decomposition, our method can fit high dimensional penalized GLMMs (pGLMMs) faster than comparable methods and more easily scale to larger dimensions not previously seen in existing approaches.

The remainder of this paper is organized as follows. Section 4.2 reviews the statistical models and algorithm used to estimate pGLMMs in our new factor model decomposition framework, termed **glmmPen_FA**. In section 4.3, simulations are conducted to assess the performance of the new **glmmPen_FA** method. Section 4.4 describes a motivating case study for the prediction of pancreatic cancer subtypes using gene expression data and provides results from the application of our new method to the case study. We close the article with some discussion in Section 4.5.

4.2 Methods

4.2.1 Model formulation

We consider the case where we want to analyze data from K independent groups of observations. For each group $k = 1, ..., K$, there are n_k observations for a total sample size of $N = \sum_{k=1}^K n_k$. For group k, let $\boldsymbol{y}_k = (y_{k1}, ..., y_{kn_k})^T$ be the vector of n_k independent responses, $\boldsymbol{x}_{ki} = (x_{ki,1}, ..., x_{ki,p})^T$ be the p-dimensional vector of predictors, and $\boldsymbol{X}_k = (\boldsymbol{x}_{k1}, ..., \boldsymbol{x}_{kn_k})^T$. For simplification of notation, we will set $n_1 = ... = n_K = n$ without loss of generality. In GLMMs, we assume that the conditional distribution of $\boldsymbol{y}_k$ given $\boldsymbol{X}_k$ belongs to the exponential family and has the following density:

$$f(\boldsymbol{y}_k|\boldsymbol{X}_k, \boldsymbol{\alpha}_k; \theta) = \prod_{i=1}^{n} c(y_{ki}) \exp\big[\tau^{-1}\{y_{ki}\eta_{ki} - b(\eta_{ki})\}\big], \tag{4.1}$$

where $c(y_{ki})$ is a constant that only depends on y_{ki}, τ is the dispersion parameter, $b(\cdot)$ is a known link function, and η_{ki} is the linear predictor.

As outlined in Rashid *et al.* (2020), the traditional GLMM formulation of the linear predictor can be represented as

$$\eta_{ki} = \boldsymbol{x}_{ki}^T\boldsymbol{\beta} + \boldsymbol{z}_{ki}^T\boldsymbol{\gamma}_k = \boldsymbol{x}_{ki}^T\boldsymbol{\beta} + \boldsymbol{z}_{ki}^T\boldsymbol{\Gamma}\boldsymbol{\delta}_k, \tag{4.2}$$

where $\boldsymbol{\beta} = (\beta_1, ..., \beta_p)^T$ is a p-dimensional vector for the fixed effects coefficients (including the intercept), $\boldsymbol{\Gamma}$ is the Cholesky decomposition of the random effects covariance matrix $\boldsymbol{\Sigma}$ such that $\boldsymbol{\Gamma}\boldsymbol{\Gamma}^T = \boldsymbol{\Sigma}$, $\boldsymbol{\gamma}_k = \boldsymbol{\Gamma}\boldsymbol{\delta}_k$, where $\boldsymbol{\delta}_k \sim N_q(0, I)$, is a q-dimensional vector of unobservable random effects (including the random intercept) for group k, and $\boldsymbol{z}_{ki}$ is a q-dimensional subvector of $\boldsymbol{x}_{ki}$.

In its current representation, the model assumes a q-dimensional latent space. When q is large, the estimation of the covariance matrix $\boldsymbol{\Sigma}$ = Var($\boldsymbol{\gamma}_k$), can be computationally burdensome to compute due to both the number of parameters needed to estimate this matrix (for an unstructured covariance matrix, $q(q + 1)/2$ parameters are needed) as well as the need to approximate a q-dimensional integral (see Section 4.2.3 for details). Prior work such as Fan *et al.* (2013) and Tran *et al.* (2020) have assumed a factor model structure in order to estimate high-dimensional covariance matrices in other settings, such as the estimation of sample covariance matrices for time series data and the covariance matrix in variational inference used to approximate the posterior distribution, respectively. Here we introduce a novel formulation of the GLMM where we decompose the random effects $\boldsymbol{\gamma}_k$ into a factor model with r latent common factors ($r \ll q$) such that $\boldsymbol{\gamma}_k = \boldsymbol{B}\boldsymbol{\alpha}_k$, where $\boldsymbol{B}$ is the $q \times r$ loading matrix and $\boldsymbol{\alpha}_k$ represents the r latent common factors. We assume the latent factors $\boldsymbol{\alpha}_k$ are uncorrelated and follow a $N_r(\boldsymbol{0}, \boldsymbol{I})$ distribution. We re-write the linear predictor as

$$\eta_{ki} = \boldsymbol{x}_{ki}^T\boldsymbol{\beta} + \boldsymbol{z}_{ki}^T\boldsymbol{B}\boldsymbol{\alpha}_k. \tag{4.3}$$

In this representation, the random component of the linear predictor has variance Var($\boldsymbol{B}\boldsymbol{\alpha}_k$) = $\boldsymbol{B}\boldsymbol{B}^T$ = $\boldsymbol{\Sigma}$. By assuming that $\boldsymbol{\Sigma}$ is low rank, we also reduce the dimension of the latent space from q to r, which reduces the dimension of the integral in the likelihood and thereby reduces the computational complexity of the E-step in the EM algorithm. Further details are given in Section 4.2.3.

In order to estimate $\boldsymbol{B}$, let $\boldsymbol{b}_t \in \mathbb{R}^r$ be the t-th row of $\boldsymbol{B}$ and $\boldsymbol{b} = (\boldsymbol{b}_1^T, ..., \boldsymbol{b}_q^T)^T$. We can then reparameterize the linear predictor as

$$\eta_{ki} = \boldsymbol{x}_{ki}^T\boldsymbol{\beta} + \boldsymbol{z}_{ki}^T\boldsymbol{B}\boldsymbol{\alpha}_k = \begin{pmatrix}\boldsymbol{x}_{ki}^T & (\boldsymbol{\alpha}_k \otimes \boldsymbol{z}_{ki})^T\boldsymbol{J}\end{pmatrix}\begin{pmatrix}\boldsymbol{\beta} \\ \boldsymbol{b}\end{pmatrix} \tag{4.4}$$

in a manner similar to Chen and Dunson (2003) and Ibrahim *et al.* (2011), where $\boldsymbol{J}$ is a matrix that transforms $\boldsymbol{b}$ to vec($\boldsymbol{B}$) such that vec$(\boldsymbol{B}) = \boldsymbol{J}\boldsymbol{b}$ and $\boldsymbol{J}$ is of dimension $(qr) \times (qr)$. The vector of parameters $\boldsymbol{\theta} = (\boldsymbol{\beta}^T, \boldsymbol{b}^T, \tau)^T$ are the main parameters of interest. We denote the true value of $\boldsymbol{\theta}$ as $\boldsymbol{\theta}^* = (\boldsymbol{\beta}^{*T}, \boldsymbol{b}^{*T}, \tau^*)^T = \text{argmin}_{\boldsymbol{\theta}}\text{E}_{\boldsymbol{\theta}}[-\ell(\boldsymbol{\theta})]$ where $\ell(\boldsymbol{\theta})$ is the observed log-likelihood across all K groups such that $\ell(\boldsymbol{\theta}) = \sum_{k=1}^K \ell_k(\boldsymbol{\theta})$, where $\ell_k(\boldsymbol{\theta}) = (1/n)\log\int f(\boldsymbol{y}_k|\boldsymbol{X}_k, \boldsymbol{\alpha}_k; \boldsymbol{\theta})\phi(\boldsymbol{\alpha}_k)d\boldsymbol{\alpha}_k$.

Our main interest lies in selecting the true nonzero fixed and random effects. In other words, we aim to identify the set

$$S = S_1 \cup S_2 = \{j : \beta_j^* \neq 0\} \cup \{t : ||\boldsymbol{b}_t^*||_2 \neq 0\},$$

where S_1 and S_2 represent the true fixed and random effects, respectively. When $\boldsymbol{b}_t = \boldsymbol{0}$, this indicates that effect of covariate t is fixed across the K groups (i.e. the corresponding t-th row and column of $\boldsymbol{\Sigma}$ is set to $\boldsymbol{0}$).

We aim to solve the following penalized likelihood problem:

$$\widehat{\boldsymbol{\theta}} = \text{argmin}_{\boldsymbol{\theta}} - \ell(\boldsymbol{\theta}) + \lambda_0 \sum_{j=1}^{p} \rho_0\left(\beta_j\right) + \lambda_1 \sum_{t=1}^{q} \rho_1\left(||\boldsymbol{b}_t||_2\right), \tag{4.5}$$

where $\ell(\boldsymbol{\theta})$ is the observed log-likelihood for all K groups defined earlier, $\rho_0(t)$ and $\rho_1(t)$ are general folded-concave penalty functions, and λ_0 and λ_1 are positive tuning parameters. The $\rho_0(t)$ penalty functions could include the L_1 penalty, the SCAD penalty, and the MCP penalty (Friedman *et al.*, 2010; Breheny and Huang, 2011). For the $\rho_1(t)$ penalty, we treat the elements of $\boldsymbol{b}_t$ as a group and penalize them in a groupwise manner using the group LASSO, group MCP, or group SCAD penalties presented by Breheny and Huang (2015). These groups of $\boldsymbol{b}_t$ are then estimated to be either all zero or all nonzero. In this way, we select covariates to have varying effects ($\widehat{\boldsymbol{b}}_t \neq \mathbf{0}$) or fixed effects ($\widehat{\boldsymbol{b}}_t = \mathbf{0}$) across the K groups.

We standardize the fixed effects covariates matrix $\boldsymbol{X} = (\boldsymbol{X}_1^T, ..., \boldsymbol{X}_K^T)^T$ such that $\sum_{k=1}^K \sum_{i=1}^{n_k} x_{ki,j} = 0$ and $N^{-1} \sum_{k=1}^K \sum_{i=1}^{n_k} x_{ki,j}^2 = 1$ for $j = 1, ..., p$.

4.2.2 MCECM algorithm

We solve (4.5) for some specific (λ_0, λ_1) using a Monte Carlo Expectation Conditional Minimization (MCECM) algorithm (Garcia *et al.*, 2010).

In the s^{th} iteration of the MCECM algorithm, we aim to evaluate the expectation of (E-step) and minimize (M-step) the following penalized Q-function:

$$\begin{aligned} Q_\lambda(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s)}) &= \sum_{k=1}^K E\left\{-\log\bigl(f(\boldsymbol{y}_k, \boldsymbol{X}_k, \boldsymbol{\alpha}_k; \boldsymbol{\theta}|\boldsymbol{d}_o; \boldsymbol{\theta}^{(s)})\bigr)\right\} + \lambda_0 \sum_{j=1}^p \rho_0\left(\beta_j\right) + \lambda_1 \sum_{t=1}^q \rho_1\left(||\boldsymbol{b}_t||_2\right) \\ &= Q_1(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s)}) + Q_2(\boldsymbol{\theta}^{(s)}) + \lambda_0 \sum_{j=1}^p \rho_0\left(\beta_j\right) + \lambda_1 \sum_{t=1}^q \rho_1\left(||\boldsymbol{b}_t||_2\right), \end{aligned} \tag{4.6}$$

where $(\boldsymbol{y}_k, \boldsymbol{X}_k, \boldsymbol{\alpha}_k)$ gives the complete data for group k, $d_{k,o} = (\boldsymbol{y}_k, \boldsymbol{X}_k)$ gives the observed data for group k, $\boldsymbol{d}_o$ represents the entirety of the observed data, and

$$Q_1(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s)}) = -\sum_{k=1}^K \int \log[f(\boldsymbol{y}_k|\boldsymbol{X}_k, \boldsymbol{\alpha}_k; \boldsymbol{\theta})]\phi(\boldsymbol{\alpha}_k|\boldsymbol{d}_{k,o}; \boldsymbol{\theta}^{(s)})d\boldsymbol{\alpha}_k, \tag{4.7}$$

$$Q_2(\boldsymbol{\theta}^{(s)}) = -\sum_{k=1}^{K} \int \log \phi(\boldsymbol{\alpha}_k)\phi(\boldsymbol{\alpha}_k|\boldsymbol{d}_{k,o};\boldsymbol{\theta}^{(s)})d\boldsymbol{\alpha}_k \tag{4.8}$$

In the E-step of the algorithm, we aim to approximate the r-dimensional integral expressed in (4.7).

4.2.2.1 Monte-Carlo E-step

The integrals in the Q-function do not have closed forms when $f(\boldsymbol{y}_k|\boldsymbol{X}_k, \boldsymbol{\alpha}_k^{(s,m)};\boldsymbol{\theta})$ is assumed to be non-Gaussian. We approximate these integrals using a Markov Chain Monte Carlo (MCMC) sample of size M from the posterior density $\phi(\boldsymbol{\alpha}_k|\boldsymbol{d}_{k,o};\boldsymbol{\theta}^{(s)})$. Let $\boldsymbol{\alpha}_k^{(s,m)}$ be the m^{th} simulated r-dimensional vector from the posterior of the latent common factors, $m = 1, ..., M$, at the s^{th} iteration of the algorithm for group k. The integral in (4.7) can be approximated as

$$Q_1(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s)}) \approx -\frac{1}{M}\sum_{m=1}^{M}\sum_{k=1}^{K} \log f(\boldsymbol{y}_k|\boldsymbol{X}_k, \boldsymbol{\alpha}_k^{(s,m)};\boldsymbol{\theta}).$$

We use the fast and efficient No-U-Turn Hamiltonian Monte Carlo sampling procedure (NUTS) from the Stan software (Carpenter *et al.*, 2017; Hoffman and Gelman, 2014) in order to perform the E-step efficiently.

4.2.2.2 Monte-Carlo M-step

In the M-step of the algorithm, we aim to minimize

$$Q_{1,\lambda}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s)}) = Q_1(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s)}) + \lambda_0 \sum_{j=1}^{p} \rho_0\left(\beta_j\right) + \lambda_1 \sum_{t=1}^{q} \rho_1\left(||\boldsymbol{b}_t||_2\right) \tag{4.9}$$

with respect to $\boldsymbol{\theta} = (\boldsymbol{\beta}^T, \boldsymbol{b}^T, \tau)^T$. We do this by using a Majorization-Minimization algorithm with penalties applied to the fixed effects and the rows of $\boldsymbol{B}$. Let s represent the iteration of the MCECM algorithm, and let h represent the iteration within a particular M-step of the MCECM algorithm. The M-step of the s^{th} iteration of the MCECM algorithm proceeds as described in Algorithm 5 given in the Chapter 4 Appendix.

4.2.2.3 MCECM algorithm

Algorithm 6 in the Chapter 4 Appendix describes the full MCECM algorithm for estimating the parameters with a particular (λ_0, λ_1). The process of model selection and finding optimal tuning parameters are described further in the Chapter 4 Appendix Sections C.2.1 and C.2.2. For further details on initialization and convergence, also see the Chapter 4 Appendix Section C.2.3.

4.2.3 Advantages of glmmPen_FA model formulation

There are several advantages to our proposed factor model decomposition of the random effects. By representing the random effects with a factor model, we reduce the latent space from a high q-dimensional space to a low r-dimensional space. In the more traditional GLMM model formulation, $Q_1(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s)})$ would be represented as

$$Q_1(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s)}) = -\sum_{k=1}^{K} \int \log[f(\boldsymbol{y}_k|\boldsymbol{X}_k, \boldsymbol{\delta}_k; \boldsymbol{\theta})]\phi(\boldsymbol{\delta}_k|\boldsymbol{d}_{k,o}; \boldsymbol{\theta}^{(s)})d\boldsymbol{\delta}_k, \tag{4.10}$$

where $\boldsymbol{\theta}$ includes the fixed effects $\boldsymbol{\beta}$, the non-zero elements of $\boldsymbol{\Gamma}$ given in (4.2), and τ, and the $\boldsymbol{\delta}_k$ are q-dimensional latent variables. However, by using the novel model formulation given in equation (4.3), this changes the integral of interest such that now $Q_1(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s)})$ expressed in (4.7) is of dimension $r \ll q$. This significantly reduces the computational complexity of estimating this integral in the E-step of the algorithm since we only have to estimate a latent space of dimension r. Consequently, this reduces the computational time. This also enables us to scale our method to hundreds of predictors since the practical dimension of the latent space will be much smaller than the total possible random effects predictors.

Furthermore, this proposed formulation allows for more complex correlation structures in higher dimensions. In Rashid *et al.* (2020), the authors approximated the random effect covariance matrix $\boldsymbol{\Sigma}$ as a diagonal matrix when the dimensions are large as recommended by Fan and Li (2012). This approximation was employed in order to reduce the computational complexity of the algorithm and therefore increase the speed of the model fit. However, in our new formulation, we do not need to assume $\mathbf{\Sigma}$ is a diagonal matrix when the dimension is high.

4.2.4 Estimation of the number of latent factors

Performing our proposed **glmmPen_FA** method requires specifying the number of latent factors r. Since r is typically unknown *a priori*, this value needs to be estimated. There have been several proposed methods of estimating r for the approximate factor model (Bai and Ng, 2002; Ahn and Horenstein, 2013; Onatski, 2010; Kapetanios, 2010). We tried the Eigenvalue Ratio method and Growth Ratio method developed by Ahn and Horenstein (2013) as well as the method proposed in Bai and Ng (2002). We found the Growth Ratio (GR) method gave the most accurate estimates of r. Therefore, in this section, we will describe how we implement the GR method to estimate r. The GR method is used in all of our numerical works.

To apply the GR method to our problem, we need a $q \times K$ matrix of observed random effects. Since we can never observe the random effects, we instead calculate pseudo random effects by first fitting a penalized generalized linear model with a small penalty to each group individually. We then take these group-specific estimates and center them so that all features have a mean of 0. Let these q-dimensional group-specific estimates be denoted as $\hat{\boldsymbol{\gamma}}_{\boldsymbol{k}}$ for each group $k = 1, ..., K$. We then define $\boldsymbol{G} = (\hat{\boldsymbol{\gamma}}_{\mathbf{1}}, ..., \hat{\boldsymbol{\gamma}}_{\boldsymbol{K}})$ as the final $q \times K$ matrix of pseudo random effects.

Let $\psi_j(A)$ be the j-th largest eigenvalue of the positive semidefinite matrix A, and let $\tilde{\mu}_{qK,j} \equiv \psi_j(\boldsymbol{G}\boldsymbol{G}^T/(qK)) = \psi_j(\boldsymbol{G}^T\boldsymbol{G}/(qK))$.

To find the GR estimator, we first order the eigenvalues of $\boldsymbol{G}\boldsymbol{G}^T/(qK)$ from largest to smallest. Then, we calculate the following ratios:

$$GR(j) \equiv \frac{\log[V(j-1)/V(j)]}{\log[V(j)/V(j+1)]} = \frac{\log\big(1 + \tilde{\mu}^*_{qK,j}\big)}{\log\big(1 + \tilde{\mu}^*_{qK,j+1}\big)}, \quad j = 1, 2, ..., U \tag{4.11}$$

where $V(j) = \sum_{l=j+1}^{\min(q,K)} \tilde{\mu}_{qK,l}$, $\tilde{\mu}^*_{qK,j} = \tilde{\mu}_{qK,j}/V(j)$, and U is a pre-defined constant. Then, we estimate r by

$$\widehat{r}_{GR} = \text{max}_{1 \leqslant j \leqslant U} GR(j) \tag{4.12}$$

4.3 Simulations

In this section, we examine the performance of the **glmmPen_FA** algorithm in performing variable selection under several different conditions. In all of these simulations, we use a pre-screening step to remove some random effects at the start of the algorithm, the BIC-ICQ (Ibrahim *et al.*, 2011) criterion for tuning parameter selection, the MCP penalty (MCP penalty for the fixed effects, group MCP penalty for the rows of the $\boldsymbol{B}$ matrix), and the abbreviated two-stage grid search as described in the Chapter 4 Appendix Section C.2.1. In order to determine the robustness of our variable selection procedure based on the assumed value of r, we fit models in one of two ways: we estimated the number of common factors r using the Growth Ratio estimation procedure discussed in Section 4.2.4, or we input the true value of r for the algorithm to use.

4.3.1 Variable selection in binomial data with 100 predictors

We examine the performance of the **glmmPen_FA** algorithm when performing variable selection in high dimensions of $p = 100$ total predictors. We simulated binary responses from a logistic mixed effects model with $p = 100$ predictors. Of p total predictors, we assume that the first 10 predictors have truly non-zero fixed and random effects, and the other $p - 10$ predictors have zero-valued fixed and random effects. We specified a full model for the algorithm such the random effect predictors equalled the fixed effect predictors (e.g. $q = p$), and our aim was to select the set of true predictors and random effects.

To simulate the data, we set the sample size to $N = 2500$ and number of groups to $K = 25$, with an equal number of subjects per group. We set up the random effects covariance matrix by specifying a $\boldsymbol{B}$ matrix with dimensions $(p + 1) \times r$, where $p + 1$ represents the p predictors plus the random intercept, and the number of latent common factors $r = \{3, 5\}$. Eleven of these $p + 1$

rows—corresponding to the true 10 predictors plus the intercept—had non-zero elements, while the remaining $p - 10$ rows were set to zero. For each value of r, we considered $\boldsymbol{B}$ matrices that produced covariance matrices $\boldsymbol{\Sigma} = \boldsymbol{B}\boldsymbol{B}^T$ with moderate variances and eigenvalues and large variances and eigenvalues (see the Chapter 4 Appendix for further details). In the simulation results presented in this section, these $\boldsymbol{B}$ matrices are referred to as the 'moderate' and 'large' $\boldsymbol{B}$ matrices, respectively. We use both moderate predictor effects and strong predictor effects, where all 10 of the true fixed effects have coefficient values of 1 or 2, respectively.

For group k, we generated the binary response y_{ki}, $i = 1, ..., n_k$ such that $y_{ki} \sim Bernoulli(p_{ki})$ where $p_{ki} = P(y_{ki} = 1|\boldsymbol{x}_{ki}, \boldsymbol{z}_{ki}, \boldsymbol{\gamma}_k, \boldsymbol{\theta}) = \exp\big(\boldsymbol{x}_{ki}^T\boldsymbol{\beta} + \boldsymbol{z}_{ki}^T\boldsymbol{\gamma}_k\big)/\{1 + \exp\big(\boldsymbol{x}_{ki}^T\boldsymbol{\beta} + \boldsymbol{z}_{ki}^T\boldsymbol{\gamma}_k\big)\}$, and $\boldsymbol{\gamma}_k \sim N_{11}(0, \boldsymbol{B}\boldsymbol{B}^T)$. Each condition was evaluated using 100 total simulated datasets.

For individual i in group k, the vector of predictors for the fixed effects was $\boldsymbol{x}_{ki} = (1, x_{ki,1}, ..., x_{ki,p})^T$, and we set the random effects $\boldsymbol{z}_{ki} = \boldsymbol{x}_{ki}$, where $x_{ki,j} \sim N(0, 1)$ for $j = 1, ..., p$, and each $\boldsymbol{x}_j$ was standardized as described in Section 4.2.1.

The results for these simulations are presented in Tables 4.1 and 4.2. Table 4.1 provides the average true and false positives for both the fixed and random effects variable selection, the median time in hours to complete the variable selection procedure, and the average of the mean absolute deviation between the coefficient estimates and the true coefficients across all simulation replicates. Table 4.2 gives the Growth Ratio r estimation procedure results, including the average estimate of r and the proportion of times that the Growth Ratio estimate of r was underestimated, correct, or overestimated. All simulations were completed on a Longleaf computing cluster (CPU Intel processors between 2.3Ghz and 2.5GHz).

We see from Table 4.1 that the **glmmPen_FA** method is able to accurately select both the fixed and random effects across a variety of conditions, which is supported by the true positives generally being above 90% for both the fixed and random effects and the false positives generally being small: across all conditions, less than 3.5% for fixed effects and less than 1% for random effects.

True r	β	$\boldsymbol{B}$	r Est.	TP % Fixef	FP % Fixef	TP % Ranef	FP % Ranef	T^{med}	Abs. Dev. (Mean)
3	1	Moderate	GR	98.50	2.00	97.20	0.22	2.05	0.26
			True	99.00	2.14	98.40	0.16	2.36	0.26
		Large	GR	95.50	2.19	98.60	0.18	2.52	0.33
			True	95.50	2.31	98.90	0.17	2.42	0.33
	2	Moderate	GR	100.00	2.46	89.00	0.53	1.45	0.37
			True	100.00	2.78	90.10	0.50	2.07	0.31
		Large	GR	100.00	3.39	94.60	0.80	2.39	0.43
			True	100.00	3.40	96.20	0.49	2.41	0.41
5	1	Moderate	GR	96.80	2.02	96.20	0.04	3.56	0.35
			True	96.70	1.86	96.80	0.03	3.60	0.35
		Large	GR	90.40	2.22	96.80	0.08	4.39	0.44
			True	90.50	1.97	96.90	0.07	4.44	0.44
	2	Moderate	GR	100.00	2.11	89.00	0.18	2.29	0.52
			True	100.00	2.42	88.40	0.24	2.99	0.44
		Large	GR	99.90	3.28	93.10	0.50	3.03	0.57
			True	99.90	3.36	93.40	0.47	3.98	0.55

Table 4.1: Variable selection results for the $p = 100$ logistic regression simulations, including true positive (TP) percentages for fixed and random effects, false positive (FP) percentages for fixed and random effects, the median time in hours for the algorithm to complete (T^{med}), and the average of the mean absolute deviation (Abs. Dev. (Mean)) between the coefficient estimates and the true β values across all simulation replicates. Column $\boldsymbol{B}$ describes the general size of both the variances and eigenvalues of the resulting $\boldsymbol{\Sigma} = \boldsymbol{B}\boldsymbol{B}^T$ random effects covariance matrix. Column 'r Est.' refers to the method used to specify r in the algorithm: the Growth Ratio estimate or the true value of r.

We can see from Table 4.2 that the Growth Ratio estimation procedure applied to the pseudo random effect estimates described in Section 4.2.4 has varying levels of accuracy depending on the structure of the underlying data. Generally, the Growth Ratio estimation procedure becomes more accurate as the eigenvalues of the covariance matrix increase and the true predictor effects are moderate. From simulations not shown, the estimation of r also generally improves when either the sample size per group increases, or the total number of predictors used in the GR estimation procedure decreases. When the eigenvalues of the covariance matrix decrease or the true predictor effects increase, the Growth Ratio procedure underestimates r on average. However, when we compare the true and false positives for the fixed and random effects given in Table 4.1

True r	β	$\boldsymbol{B}$	Avg. r	r Underestimated %	r Correct %	r Overestimated %
3	1	Moderate	2.79	21	79	0
		Large	2.95	5	95	0
	2	Moderate	2.21	80	19	1
		Large	2.51	49	51	0
5	1	Moderate	4.60	26	72	2
		Large	4.83	15	83	2
	2	Moderate	3.83	70	28	2
		Large	4.43	46	50	4

Table 4.2: Results of the Growth Ratio r estimation procedure for $p = 100$ logistic mixed effects simulation results, including the average estimate of r across simulations and percent of times that the estimation procedure underestimated r, gave the true r, or overestimated r. Column $\boldsymbol{B}$ describes the general size of both the variances and eigenvalues of the resulting $\boldsymbol{\Sigma} = \boldsymbol{B}\boldsymbol{B}^T$ random effects covariance matrix.

for these less ideal cases, we see that using the estimated r gave very similar results to when the true value of r was utilized by the algorithm. From the average of the mean absolute deviation values, we see that the mis-specification of r also does not significantly impact the estimation of the fixed effects coefficients.

4.3.2 Comparison of the glmmPen and glmmPen_FA methods

As far as we are aware, the **glmmPen** method developed by Rashid *et al.* (2020) and implemented in the **glmmPen** R package available on CRAN is the only other method that performs simultaneous fixed and random effects variable selection in high dimensional GLMMs.

We next compare the performance of this **glmmPen** method and our novel **glmmPen_FA** method developed in this chapter. We first compared the performance of these methods in moderate dimensions. We simulated binary responses from a logistic mixed effects model much like the procedure described in Section 4.3.1, except the total number of predictors used in the analyses was $p = 25$ and we restricted our consideration to $r = 3$ common factors for all simulation scenarios. For the **glmmPen_FA** method, all values of r used in the algorithm were from the Growth Ratio estimates of r. In these moderate dimensions of $p = 25$ with our given sample size of $N = 2500$, it is reasonable to use **glmmPen** to perform variable selection in logistic mixed effects models assuming an unstructured random effects covariance matrix, allowing us to use as directly comparable model assumptions as possible for these method comparisons.

Table 4.3 gives the average true and false positives for both the fixed and random effects, the median time in hours to complete the variable selection procedure, and the average of the mean absolute deviation between the coefficient estimates and the true coefficients across all simulation replicates. Table 4.4 gives the Growth Ratio r estimation procedure results for the **glmmPen_FA** method.

β	$\boldsymbol{B}$	Method	TP % Fixef	FP % Fixef	TP % Ranef	FP % Ranef	T^{med}	Abs. Dev. (Mean)
1	Moderate	glmmPen_FA	99.50	3.47	99.80	0.53	0.59	0.26
		glmmPen	100.00	37.40	100.00	0.00	2.62	0.27
	Large	glmmPen_FA	97.20	3.80	99.90	0.80	0.49	0.33
		glmmPen	99.00	60.60	100.00	0.00	3.18	0.34
2	Moderate	glmmPen_FA	100.00	2.27	98.40	0.47	0.47	0.27
		glmmPen	100.00	13.67	99.20	0.00	2.84	0.43
	Large	glmmPen_FA	99.80	3.53	99.80	0.73	0.48	0.35
		glmmPen	100.00	30.93	100.00	0.00	2.53	0.50

Table 4.3: Results of the variable selection procedure for the $p = 100$ logistics mixed effects simulations, including true positive (TP) percentages for fixed and random effects, false positive (FP) percentages for fixed and random effects, the median time in hours for the algorithm to complete (T^{med}), and the average of the mean absolute deviation (Abs. Dev. (Mean)) between the coefficient estimates and the true β values across all simulation replicates. Column $\boldsymbol{B}$ describes the general size of both the variances and eigenvalues of the resulting $\boldsymbol{\Sigma} = \boldsymbol{B}\boldsymbol{B}^T$ random effects covariance matrix. All values of r used in the glmmPen_FA method were from the Growth Ratio estimates of r.

When comparing the **glmmPen_FA** and **glmmPen** results in these $p = 25$ simulations, we see that the median time for **glmmPen** to complete the variable selection procedures ranged from 2.53 to 3.28 hours for all four simulation scenarios considered. On the other hand, the **glmmPen_FA** method was able to fit these variable selection procedures about 5-6 times faster, where the median running time ranged from 0.47 to 0.59 hours.

True r	β	$\boldsymbol{B}$	Avg. r	r Underestimated %	r Correct %	r Overestimated %
3	1	Moderate	3.00	0	100	0
		Large	3.00	0	100	0
	2	Moderate	2.76	24	76	0
		Large	2.92	8	92	0

Table 4.4: Results of the Growth Ratio r estimation procedure for glmmPen_FA $p = 25$ logistic mixed effects simulation results, including the average estimate of r across simulations and percent of times that the estimation procedure underestimated r, gave the true r, or overestimated r. Column $\boldsymbol{B}$ describes the general size of both the variances and eigenvalues of the resulting $\boldsymbol{\Sigma} = \boldsymbol{B}\boldsymbol{B}^T$ random effects covariance matrix.

Table 4.3 also shows that there is little difference in the true positives for both the fixed and random effects between the two methods. However, **glmmPen** tends to have more false positives in the fixed effects.

We also performed variable selection using **glmmPen** on the $r = 3, p = 100$ simulations described in Section 4.3.1. In these larger dimensions, we simplified the **glmmPen** estimation procedure by assuming an independent covariance matrix to reduce the number of random effects covariance parameters. We let the **glmmPen** variable selection procedure proceed for 100 hours. In that time, **glmmPen** was able to complete the following number of replicates out of the 100 total replicates: 83 for ($\beta = 1, \boldsymbol{B} =$ Moderate), 71 for ($\beta = 1, \boldsymbol{B} =$ Large), 100 for ($\beta = 2, \boldsymbol{B} =$ Moderate), and 96 for for ($\beta = 2, \boldsymbol{B} =$ Large). The minimum times needed to complete the **glmmPen** variable selection procedures: 39.91, 57.60, 23.63, and 42.79 hours, respectively; in summary, it took a day or more for the fastest simulation replicates to complete when using the **glmmPen** method. In cases where we desire to select true random effects from a large number of total predictors, it is clear that the **glmmPen_FA** estimation procedure significantly reduces the required time to perform variable selection.

4.3.3 Variable selection in binomial data with 500 predictors

In order to further illustrate the scalability of our method, we applied our method to binary outcome simulations with $p = 500$ covariates. We simulated the binary responses from a logistic mixed effects model much like the procedure described in Section 4.3.1, except the total number of predictors used in the analyses was $p = 500$ instead of $p = 100$. All simulations assumed the true number of latent factors r was 3 and the Growth Ratio method was used to estimate r. Just as in the $p = 100$ binary outcome simulations, we specified a full model for the algorithm such the random effect predictors equalled the fixed effect predictors (e.g. $q = p$), and our aim was to select the set of true predictors and random effects. The variable selection results to these simulations are given in Table 4.5. The median times needed to complete these simulations took between 10.37 and 17.57 hours.

True r	β	$\boldsymbol{B}$	Avg. r	TP % Fixef	FP % Fixef	TP % Ranef	FP % Ranef	T^{med}	Abs. Dev. (Mean)
3	1	Moderate	2.67	97.40	0.97	93.90	0.05	10.37	0.27
		Large	2.85	88.50	1.73	94.30	0.33	17.57	0.37
	2	Moderate	2.37	100.00	0.06	77.40	0.00	11.44	0.48
		Large	2.41	99.60	0.19	88.10	0.03	13.22	0.55

Table 4.5: Results of the variable selection procedure for $p = 500$ logistic mixed effects simulation results, including true positive (TP) percentages for fixed and random effects, false positive (FP) percentages for fixed and random effects, the median time in hours for the algorithm to complete (T^{med}), and the average of the mean absolute deviation (Abs. Dev. (Mean)) between the coefficient estimates and the true β values across all simulation replicates. Column 'Avg. r' gives the average Growth Ratio r estimate used within the algorithm. Column $\boldsymbol{B}$ describes the general size of both the variances and eigenvalues of the resulting $\boldsymbol{\Sigma} = \boldsymbol{B}\boldsymbol{B}^T$ random effects covariance matrix.

4.3.4 Variable selection in Poisson data with 100 predictors

While we have previously focused on binary outcome data in our simulations, our proposed method also applies to other members of the generalized linear model family, including the Poisson model for count outcome data. To illustrate this, we simulated a Poisson mixed effects model with $p = 100$ predictors, 5 of which had truly non-zero fixed and random effects, and the other $p - 5$ predictors had zero-valued fixed and random effects. As in the previous Binomial simulations, we set the sample size to $N = 2500$ and the number of groups to $K = 25$, with equal numbers of subjects per group. We set r to 3, assigned moderate predictor effects ($\beta = 1$), and specified a $\boldsymbol{B}$ matrix with 6 non-zero rows (for the 5 predictors plus an intercept) that produced a 'moderate' covariance matrix (see the Chapter 4 Appendix for details). Unlike in the previous Binomial simulations, we simulated $x_{ki,j} \sim N(0, \sigma = 0.10)$ for $j = 1, ..., p$ to reduce the overall spread of the simulated $y_{ki} \sim Poisson(\mu_{ki})$ outcome values, where $\mu_{ki} = \exp\big(\boldsymbol{x}_{ki}^T \boldsymbol{\beta} + \boldsymbol{z}_{ki}^T \boldsymbol{\gamma}_k\big)$.

Using the Growth Ratio estimation procedure to estimate r, the average true positive percentages were 83.40% for the fixed effects and for 77.20% the random effects, and the average false positives were 5.92% for the fixed effects and 3.23% for the random effects. The average estimate of r across the simulation replicates was 2.35.

4.4 Case study: Pancreatic Ductal Adenocarcinoma

Patients diagnosed with Pancreatic Ductal Adenocarcinoma (PDAC) generally face a very poor prognosis, where the 5-year survival rate is 6% (Khorana *et al.*, 2016). The study by Moffitt *et al.* (2015) identified genes that are expressed exclusively in pancreatic tumor cells. Using these tumor-specific genes, Moffitt *et al.* (2015) was able to identify and validate two novel tumor subtypes, termed 'basal-like' and 'classical'. It was found that patients diagnosed with basal-like tumors had significantly worse median survival than those diagnosed with the classical tumors. Consequently, it is of clinical interest to robustly predict this basal-like subtype in order to make and improve tailored treatment recommendations.

A common problem observed in the application of gene signatures is the inconsistency of gene signature selection across biomedical studies, where gene signatures identified in one study may have little or no overlap with ones identified in other studies (Waldron *et al.*, 2014). This lack of replicability of gene signatures also translates to a lack of replicability of other findings, where models based upon these different gene signatures have variable accuracy in predicting clinical outcomes in new studies (Sotiriou and Piccart, 2007; Waldron *et al.*, 2014) or provide contradictory effect estimates relating genes to the outcome (Swisher *et al.*, 2012). This lack of replicability across studies comes from several sources, including small sample size (Sotiriou and Piccart, 2007), low frequencies of one or more molecular subtypes (Lusa *et al.*, 2007), and differences in data pre-processing steps (Lusa *et al.*, 2007; Paquet and Hallett, 2015).

In order to improve replicability in the prediction of subtypes in PDAC, we combine PDAC gene expression data from five different studies. The studies used in these analyses are summarized in Chapter 4 Appendix Table C.3. In order to account and adjust for between-study heterogeneity, we apply our new method **glmmPen_FA** to fit a penalized logistic mixed effects model to our data to select predictors with study-replicable effects, where we assume that predictor effects may vary between studies.

Moffitt *et al.* (2015) identified a 500-member gene list relevant to classifying the PDAC tumor subtypes. Of these 500 genes, the five studies described in Chapter 4 Appendix Table C.3 had RNA-seq gene expression data for 432 of these genes. There were some significant correlations between some of these 432 genes, as evaluated by Spearman correlations. In order to avoid having very highly correlated covariates in the analyses, we decided to combine highly correlated genes together into meta-genes. The clustering process used to create these meta-genes is described in Chapter 4 Appendix Section C.2.6. The final dataset included 110 cluster covariates. If clusters were composed of two or more genes, then the raw RNA-seq gene expression for these genes was summed together to create a meta-gene. The final covariates used in the analyses were subject-level rank transformations of the gene expression corresponding to these clusters.

Due to the presence of several pairwise Spearman correlation values greater than 0.5 in this final dataset, we used the Elastic Net penalization procedure (Friedman *et al.*, 2010) to balance between ridge regression and the MCP penalty. We let π represent the balance between ridge regression and the MCP penalty, where $\pi = 0$ represents ridge regression and $\pi = 1$ represents the MCP penalty. In these analyses, we let $\pi = 0.7$, and we estimated the number of common factors r using the Growth Ratio procedure. The same value of π was used for both the fixed effects and random effects penalization. The sequence of λ penalties used the the variable selection procedure was the same as those used in the Binomial variable selection simulations for $p = 100$ as discussed in Chapter 4 Appendix Section C.2.2.

To see the general performance of glmmPen_FA under different covariate correlation structures and under different values of π, see Chapter 4 Appendix Section C.2.5.

The basal or classical subtype outcome was calculated using the clustering algorithm specified in Moffitt *et al.* (2015). Further details are provided in Chapter 4 Appendix Section C.2.6; the code for this procedure is provided in a GitHub repository, see Supporting Information for more details.

In the final results, 8 of the original 110 cluster covariates had non-zero fixed effect values in the best model, implying these covariates were important for the prediction of the basal outcome. These 8 cluster covariates represented 33 genes in total. Table 4.6 includes the cluster label for these 8 clusters, the sign of the associated fixed effect coefficient, and the gene symbols of the genes that make up the cluster. Clusters with positive coefficients indicate that having greater relative expression of these meta-genes increases the odds of a subject being in the basal subtype, and vice versa for negative coefficients. The best model contained a random intercept (variance value 1.57) and no other random slopes. The Growth Ratio procedure estimated $r = 2$ latent common factors to model the random effects.

Cluster No.	Coefficient Sign	Gene Component Symbols
21	+	C16orf74, HES2, S100A2
25	+	CCNG2, MBOAT2, MET, TGFB2
44	+	FAM83A, MUC16, SCEL
45	+	FANK1, HS3ST1, LEMD1
58	+	KRT15, PAK6, PROM2, ST3GAL4, TRIM29
64	+	MMP13, PMAIP1
70	+	PSCA, VGLL1, WNT10A
91	-	ANXA10, BTNL8, CLDN18, CYP2C18, GUCY2C, LRRC31, MYO1A, NR1I2, PIP5K1B, REG4

Table 4.6: Covariate cluster label within the case study dataset of the clusters that had non-zero fixed effects in the final best model, the sign of the fixed effect coefficient associated with the cluster, and the gene symbols of the genes within the cluster.

Sensitivity analyses were conducted, which included additional values for π in the Elastic Net procedure and a larger value of the number of assumed latent common factors r. Details are included in Chapter 4 Appendix Section C.2.7.

We also applied the **glmmPen** variable selection procedure to this data (assuming an independent random effects covariance matrix). The 8 cluster covariates selected by **glmmPen_FA** were also consistently selected by **glmmPen**. One difference in the model results is that **glmmPen** also consistently selected clusters 25 and 58 to have non-zero random effects. The main difference in these variable selection procedures were the time to complete the variable selection procedure, where **glmmPen_FA** completed the procedure within 0.4 hours and **glmmPen** completed the procedure within 37.8 hours. More details about **glmmPen** sensitivity analyses are provided in Chapter 4 Appendix Section C.2.7.

4.5 Discussion

By representing the random effects with a factor model, we reduce the latent space from a large number of random effects to a smaller set of latent factors. We have shown through simulations that by reducing the complexity of the integral in the E-step, we can significantly improve the overall time needed to perform variable selection in high dimensional generalized linear mixed models.

The simulations in Section 4.3 also show how reducing the latent space increases the feasible dimensionality of performing variable selection in generalized linear mixed models. By using our novel formulation of the random effects, we can perform variable selection on mixed models with hundreds of predictors within a reasonable time-frame without any *a priori* knowledge of which predictors are relevant for the model, either in terms of fixed or random effects. From the simulation results, we see that the **glmmPen_FA** method results in accurate selection of the fixed and random effects across several conditions.

In general, our method is limited by the need to provide an estimate for the number of latent common factors. However, this limitation is tempered by several observations from the simulation results. The simulation results show that our data-driven estimation of the number of latent factors, based on the Growth Ratio estimation procedure by Ahn and Horenstein (2013), provides reasonable estimates. Even when it was estimated incorrectly by this procedure, this mis-specification had very little impact on the general variable selection performance or the fixed effects coefficient estimates. Therefore, our method is not sensitive to the estimation of the number of latent factors.

Supporting Information

Chapter 4 Appendices and Tables referenced in Sections 4.3 and 4.4 are given at the end of this document. The Chapter 4 Appendix contains addition details about the variable selection procedure, the simulations, and the case study. The **glmmPen** R package, which contains the code for the original glmmPen formulation and the new glmmPen_FA method, is available for download through CRAN at `https://cran.r-project.org/web/packages/glmmPen/index.html`. The GitHub repository `https://github.com/hheiling/paper_glmmPen_FA` contains the following materials: the code to run the simulations, the code to create the cleaned data for the case study, the code to analyze the case study data, the data used in the case study, and the code to recreate the tables provided within this chapter. Please see the README files within this repository for more details.

CHAPTER 5: EFFICIENT COMPUTATION OF HIGH-DIMENSIONAL PIECEWISE CONSTANT HAZARD PENALIZED RANDOM EFFECTS SURVIVAL MODELS

5.1 Introduction

Modeling survival outcomes has great clinical significance in medical and public health research. In particular, the Cox proportional hazards model has been widely utilized in order to characterize the relationship between treatments, exposures, or other covariates and patient time-to-event outcomes. However, modern biomedical datasets are increasingly high dimensional, and groups of samples within the data can exhibit complex correlations. For example, when studying survival outcomes with respect to multi-center clinical trials, recurrent events, and genetic studies, proportional hazards mixed effects models are used to account for correlations among groups within the data and model the heterogeneity of treatment and predictor effects across groups (Vaida and Xu, 2000; Ripatti and Palmgren, 2000). These proportional hazards mixed effects models are traditionally referred to as frailty models when the model contains a single random effect applied to the baseline hazard.

In high dimensional settings, in which the covariate effects are generally assumed to be sparse, it is often unknown *a priori* which covariates should be specified as fixed or random in the model. Variable selection methods such as LASSO and SCAD exist for high dimensional proportional hazards models or frailty models (Tibshirani, 1997; Bradic *et al.*, 2011; Simon *et al.*, 2011; Fan and Li, 2002), but they do not allow for the selection of random effects. Several mixed effects model selection methods that rely on the specification of candidate models have been proposed, including likelihood ratios, profile Akaike information criterion (AIC) (Xu *et al.*, 2009), and conditional AIC (Donohue *et al.*, 2011). However, specifying all 2^p possible candidate models in high dimensions is impractical. Lee *et al.* (2014) developed a stochastic search variable selection (SSVS) method that selects both fixed and random effects in proportional hazards mixed effects models in a Bayesian framework, but their method is only computationally feasible for small or moderate dimensions.

Rashid *et al.* (2020) developed a method that simultaneously selects both fixed and random effects in high dimensional generalized linear mixed models (GLMMs), which has since been developed into an R package available on CRAN (Heiling *et al.*, 2023a,c), recall Chapter 3. This method broadened the feasible dimensionality of performing variable selection in GLMMs to greater dimensions than previously existing methods. This method was extended by Heiling *et al.* (2023b) (recall Chapter 4), who proposed a new formulation of the GLMM using a factor model decomposition of the random effects. As a result of this new formulation, they were able to improve the scalability of their method and perform variable selection within GLMMs in cases with much larger dimensions. However, these methods do not apply to survival data.

In this chapter, we propose a method that simultaneously selects fixed and random effects within clustered survival data. We approximate the proportional hazards mixed effects model using a piecewise constant hazard mixed effects model (Austin, 2017) and utilize the factor model decomposition of the random effects proposed in Heiling *et al.* (2023b) (recall Chapter 4), allowing us to scale our method to cases with hundreds of predictors. We label our method as **phmmPen_FA**, which reflects our goal of estimating penalized proportional hazards mixed effects models using factor analysis on the random effects. In order to extend the methods of Rashid *et al.* (2020) and Heiling *et al.* (2023b) to survival data, we approximate the Cox proportional hazards model using a fully parametric model. The proportional hazards model can be approximated using the piecewise constant hazard survival model, in which the follow-up time of the study is split into time intervals where the baseline hazard is assumed to be constant within these intervals (Friedman, 1982; Laird and Olivier, 1981; Holford, 1980; Rodriguez, 2010). This piecewise constant hazard survival model can be fit using a log-linear model which incorporates the duration of exposure within each interval.

The remainder of this paper is organized as follows. Section 5.2 reviews the statistical models and algorithm used to estimate piecewise constant hazard mixed models. In section 5.3, simulations are conducted to assess the performance of our method. Section 5.4 describes a motivating case study for the prediction of survival in pancreatic ductal adenocarinoma cancer using gene expression data from multiple trials, and provides results from the application of our new method to the case study. We close the article with some discussion in Section 5.5.

5.2 Methods

5.2.1 Model formulation

In this section, we review the notation and model formulation of our approach. We consider the case where we want to analyze data from K independent groups of subjects. For each group $k = 1, ..., K$, there are n_k subjects for a total sample size of $N = \sum_{k=1}^{K} n_k$. For group k, let $\boldsymbol{y}_k = (y_{k1}, ..., y_{kn_k})^T$ be the vector of n_k observed times, where $y_{ki} = \min(T_{ki}, C_{ki})$, T_{ki} represents the event time, and C_{ki} represents censoring time; let $\boldsymbol{\delta}_k = (\delta_{k1}, ..., \delta_{kn_k})$ where $\delta_{ki} = I(T_{ki} < C_{ki})$ represents the indicator that a subject's event time was observed; and let $\boldsymbol{x}_{ki} = (x_{ki,1}, ..., x_{ki,p})^T$ be the p-dimensional vector of predictors, and $\boldsymbol{X}_k = (\boldsymbol{x}_{k1}, ..., \boldsymbol{x}_{kn_k})^T$.

We would like to estimate the proportional hazards mixed effects model

$$h(t|\eta_{ki}) = h_0(t) \exp(\eta_{ki}), \tag{5.1}$$

where $h(t|\eta_{ki})$ is the subject's individual hazard at time t, $h_0(t)$ represents the baseline hazard at time t, and η_{ki} represents the linear predictor containing the fixed effects log hazard ratio coefficients, the group-specific random effects, and the subject's individual covariates. The exact form of the linear predictor η_{ki} assumed in our model is described later in this section.

We approximate (5.1) with the piecewise constant hazard mixed effects model (Austin, 2017). When survival models include random effects, it is necessary to fully model the baseline hazard function. Approximating this baseline hazard using a piecewise constant function allows for relatively convenient computation. We first partition the time of the study into J intervals, where we assume that the baseline hazard within a particular time interval is constant. Let us define the cut points $0 = \tau_0 < \tau_1 < ... < \tau_J = \infty$, and let h_j be the constant baseline hazard within interval j, $[\tau_{j-1}, \tau_j)$. We then write the model as

$$h_{kij} = h_j \exp(\eta_{ki}), \tag{5.2}$$

where h_j is the baseline hazard for interval j and h_{kij} is the constant hazard corresponding to subject i in group k within interval j.

The observed data for each subject includes their observed time y_{ki} and their event indicator δ_{ki}. We extend these to define analogous measures for each interval, where $t^*_{kij} = \max[\min(y_{ki}, \tau_j) - \tau_{j-1}, 0]$ is the amount of time subject i in group k survived within interval j, and $d_{kij} = I(\tau_{j-1} \leqslant y_{ki} < \tau_j, \delta_{ki} = 1)$ is the indicator of whether the subject died during interval j. To better clarify t^*_{kij}, this term has three possible values, determined by the relative value of their observed time y_{ki} to the interval cut points:

$$t^*_{kij} = \begin{cases} \tau_j - \tau_{j-1}, & y_{ki} > \tau_j; \\ y_{ki} - \tau_{j-i}, & \tau_{j-1} < y_{ki} \leqslant \tau_j; \\ 0, & y_{ki} \leqslant \tau_{j-1}. \end{cases}$$

We can then treat the death indicators d_{kij} as if they are independent Poisson observations with means $\mu_{kij} = t^*_{kij} h_{kij}$, allowing us to fit the data using the log-linear model

$$\log \mu_{kij} = \log t^*_{kij} + \psi_j + \eta_{ki}, \tag{5.3}$$

where $\psi_j = \log(h_j)$ is the logarithm of the constant hazard within interval j and $\log\big(t^*_{kij}\big)$, the log of the time a subject survived within interval j, is treated as an offset to the model.

Let us define $\boldsymbol{d}_k = (d_{k11}, ..., d_{k1J}, ..., d_{kn_k1}, ..., d_{kn_kJ})^T$ as the vector of death indicator values for all subjects in group k and all J time intervals. Then, the piecewise constant hazard likelihood is defined as

$$f(\boldsymbol{d}_k | \boldsymbol{X}_k, \boldsymbol{\alpha}_k; \theta) = \prod_{i=1}^{n_k} \prod_{j=1}^{J} \big[I(t_{kij}^* > 0)\mu_{kij}\big]^{d_{kij}} \exp\big[-I(t_{kij}^* > 0)\mu_{kij}\big], \tag{5.4}$$

where μ_{kij} is defined in (5.3), and $I(t_{kij}^* > 0) = 1$ indicates that a subject i in group k survived at least part way through interval j, 0 if the subject died or was censored before interval j.

Now we may defined the form of the linear predictor term η_{ki} used within this model, similar to the one used in the traditional generalized linear mixed model (Chen and Dunson, 2003; Ibrahim *et al.*, 2011; Rashid *et al.*, 2020)

$$\eta_{ki} = \boldsymbol{x}_{ki}^T \boldsymbol{\beta} + \boldsymbol{z}_{ki}^T \boldsymbol{\gamma}_k = \boldsymbol{x}_{ki}^T \boldsymbol{\beta} + \boldsymbol{z}_{ki}^T \boldsymbol{\Gamma} \boldsymbol{\epsilon}_k, \tag{5.5}$$

where $\boldsymbol{\beta} = (\beta_1, ..., \beta_p)^T$ is a p-dimensional vector for the fixed effects coefficients (β represents the log hazard ratio values for each predictor and excludes an intercept), $\boldsymbol{\Gamma}$ is the Cholesky decomposition of the random effects covariance matrix $\boldsymbol{\Sigma}$ such that $\boldsymbol{\Gamma}\boldsymbol{\Gamma}^T = \boldsymbol{\Sigma}$, $\boldsymbol{\gamma}_k = \boldsymbol{\Gamma}\boldsymbol{\epsilon}_k$, where $\boldsymbol{\epsilon}_k \sim N_q(0, \boldsymbol{I})$, is a q-dimensional vector of unobservable random effects (including the random intercept) for group k, and $\boldsymbol{z}_{ki}$ is a q-dimensional vector that includes an intercept term and a subset of $\boldsymbol{x}_{ki}$.

We reformulate the linear predictor as described in Heiling *et al.* (2023b) (recall Chapter 4) by decomposing the random effects $\boldsymbol{\gamma}_k$ into a factor model with r latent common factors, where we assume $r \ll q$. As a result, we assume $\boldsymbol{\gamma}_k = \boldsymbol{B}\boldsymbol{\alpha}_k$, where $\boldsymbol{B}$ is the $q \times r$ loading matrix and $\boldsymbol{\alpha}_k$ represents the r latent common factors. We assume the latent factors $\boldsymbol{\alpha}_k$ are uncorrelated and follow a $N_r(\boldsymbol{0}, \boldsymbol{I})$ distribution. We re-write the linear predictor as

$$\eta_{ki} = \boldsymbol{x}_{ki}^T \boldsymbol{\beta} + \boldsymbol{z}_{ki}^T \boldsymbol{B} \boldsymbol{\alpha}_k. \tag{5.6}$$

In the representation of (5.6), the random component of the linear predictor has variance Var($\boldsymbol{B}\boldsymbol{\alpha}_k$) = $\boldsymbol{B}\boldsymbol{B}^T$ = $\boldsymbol{\Sigma}$, which is low rank. By using this representation, we reduce the dimension of the latent space from q to r; this reduces the dimension of the integral in the likelihood, which reduces the computational complexity of the E-step in the EM algorithm described in Section 5.2. Consequently, this factor decomposition reduces the computational time of the algorithm and enables our method to scale to hundreds of predictors (Heiling *et al.*, 2023b).

In order to estimate $\boldsymbol{B}$, let $\boldsymbol{b}_t \in \mathbb{R}^r$ be the t-th row of $\boldsymbol{B}$ and $\boldsymbol{b} = (\boldsymbol{b}_1^T, ..., \boldsymbol{b}_q^T)^T$. We then reparameterize the linear predictor as

$$\eta_{ki} = \boldsymbol{x}_{ki}^T\boldsymbol{\beta} + \boldsymbol{z}_{ki}^T\boldsymbol{B}\boldsymbol{\alpha}_k = \begin{pmatrix}\boldsymbol{x}_{ki}^T & (\boldsymbol{\alpha}_k \otimes \boldsymbol{z}_{ki})^T\boldsymbol{J}\end{pmatrix}\begin{pmatrix}\boldsymbol{\beta} \\ \boldsymbol{b}\end{pmatrix} \tag{5.7}$$

in a manner similar to Chen and Dunson (2003) and Ibrahim *et al.* (2011), where $\boldsymbol{J}$ is a matrix that transforms $\boldsymbol{b}$ to vec($\boldsymbol{B}$) such that vec$(\boldsymbol{B}) = \boldsymbol{J}\boldsymbol{b}$ and $\boldsymbol{J}$ is of dimension $(qr) \times (qr)$. The vector of parameters $\boldsymbol{\theta} = (\boldsymbol{\beta}^T, \boldsymbol{b}^T, \boldsymbol{\psi}^T)^T$ are the main parameters of interest.

We denote the true value of $\boldsymbol{\theta}$ as $\boldsymbol{\theta}^* = (\boldsymbol{\beta}^{*T}, \boldsymbol{b}^{*T}, \boldsymbol{\psi}^{*T})^T = \text{argmin}_{\boldsymbol{\theta}}\text{E}_{\boldsymbol{\theta}}[-\ell(\boldsymbol{\theta})]$ where $\ell(\boldsymbol{\theta})$ is the observed log-likelihood across all K groups such that $\ell(\boldsymbol{\theta}) = \sum_{k=1}^K \ell_k(\boldsymbol{\theta})$, where $\ell_k(\boldsymbol{\theta}) = (1/n_k)\log\int f(\boldsymbol{d}_k|\boldsymbol{X}_k, \boldsymbol{\alpha}_k; \boldsymbol{\theta})\phi(\boldsymbol{\alpha}_k)d\boldsymbol{\alpha}_k$.

Our primary goal is to select the true nonzero fixed and random effects, i.e. identify the set

$$S = S_1 \cup S_2 = \{j : \beta_j^* \neq 0\} \cup \{t : ||\boldsymbol{b}_t^*||_2 \neq 0\},$$

where S_1 and S_2 represent the true fixed and random effects, respectively. When $\boldsymbol{b}_t = \boldsymbol{0}$, this indicates that the effect of covariate t is fixed across the K groups (i.e. the corresponding t-th row and column of $\boldsymbol{\Sigma}$ is set to $\boldsymbol{0}$).

Our objective is to solve the penalized likelihood problem of (5.8):

$$\widehat{\boldsymbol{\theta}} = \text{argmin}_{\boldsymbol{\theta}} - \ell(\boldsymbol{\theta}) + \lambda_0 \sum_{j=1}^p \rho_0(\beta_j) + \lambda_1 \sum_{t=1}^q \rho_1(||\boldsymbol{b}_t||_2), \tag{5.8}$$

where $\ell(\boldsymbol{\theta})$ is the observed log-likelihood for all K groups, $\rho_0(t)$ and $\rho_1(t)$ are general folded-concave penalty functions, and λ_0 and λ_1 are positive tuning parameters. The penalty functions applied to the fixed effects, represented by $\rho_0(t)$, could include the L_1 penalty (LASSO), the SCAD penalty, and the MCP penalty (Friedman *et al.*, 2010; Breheny and Huang, 2011). The penalty functions applied to the random effects, represented by $\rho_1(t)$, could include the group LASSO, the group MCP, or the group SCAD penalties presented by Breheny and Huang (2015) since we treat the elements of $\boldsymbol{b}_t$ as a group and penalize them in a groupwise manner. As a result, these groups of $\boldsymbol{b}_t$ are estimated to be either all zero or all nonzero, which means that we select covariates to have random effects ($\widehat{\boldsymbol{b}}_t \neq \mathbf{0}$) or fixed effects ($\widehat{\boldsymbol{b}}_t = \mathbf{0}$) across the K groups.

We standardize the fixed effects covariates matrix $\boldsymbol{X} = (\boldsymbol{X}_1^T, ..., \boldsymbol{X}_K^T)^T$ such that $\sum_{k=1}^K \sum_{i=1}^{n_k} x_{ki,j} = 0$ and $N^{-1} \sum_{k=1}^K \sum_{i=1}^{n_k} x_{ki,j}^2 = 1$ for $j = 1, ..., p$.

5.2.2 MCECM algorithm

We solve (5.8) for some specific (λ_0, λ_1) using a Monte Carlo Expectation Conditional Minimization (MCECM) algorithm (Garcia *et al.*, 2010).

Our objective within the s^{th} iteration of the MCECM algorithm is to evaluate the expectation of (E-step) and minimize (M-step) the penalized Q-function defined in (5.9):

$$\begin{aligned} Q_\lambda(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s)}) &= \sum_{k=1}^{K} E\left\{-\log\big(f(\boldsymbol{d}_k, \boldsymbol{X}_k, \boldsymbol{\alpha}_k; \boldsymbol{\theta}|\boldsymbol{D}_o; \boldsymbol{\theta}^{(s)})\big)\right\} + \lambda_0 \sum_{j=1}^{p} \rho_0\left(\beta_j\right) + \lambda_1 \sum_{t=1}^{q} \rho_1\left(||\boldsymbol{b}_t||_2\right) \\ &= Q_1(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s)}) + Q_2(\boldsymbol{\theta}^{(s)}) + \lambda_0 \sum_{j=1}^{p} \rho_0\left(\beta_j\right) + \lambda_1 \sum_{t=1}^{q} \rho_1\left(||\boldsymbol{b}_t||_2\right), \end{aligned} \tag{5.9}$$

where $(\boldsymbol{d}_k, \boldsymbol{X}_k, \boldsymbol{\alpha}_k)$ gives the complete data for group k, $\boldsymbol{D}_{k,o} = (\boldsymbol{d}_k, \boldsymbol{X}_k)$ gives the observed data for group k, $\boldsymbol{D}_o$ represents the entirety of the observed data, and

$$Q_1(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s)}) = -\sum_{k=1}^{K} \int \log[f(\boldsymbol{d}_k|\boldsymbol{X}_k, \boldsymbol{\alpha}_k; \boldsymbol{\theta})]\phi(\boldsymbol{\alpha}_k|\boldsymbol{D}_{k,o}; \boldsymbol{\theta}^{(s)})d\boldsymbol{\alpha}_k, \tag{5.10}$$

$$Q_2(\boldsymbol{\theta}^{(s)}) = -\sum_{k=1}^{K} \int \log[\phi(\boldsymbol{\alpha}_k)]\phi(\boldsymbol{\alpha}_k|\boldsymbol{D}_{k,o};\boldsymbol{\theta}^{(s)})d\boldsymbol{\alpha}_k \tag{5.11}$$

Our goal in the E-step of the algorithm is to approximate the r-dimensional integral expressed in (5.10). We first specify J time intervals defined so that there are an approximately equal number of events within each time interval. If a subject survived at least part-way through j^* intervals (i.e. $t^*_{kij} > 0$ for $j = 1, ..., j^* \leqslant J$), the long-form dataset contains j^* observations for that subject. For subject i in group k that survived at least part-way through j^* time intervals, we define $d_{kij} = I(\tau_{j-1} \leqslant y_{ki} < \tau_j, \delta_{ki} = 1)$ for $j = 1, ..., j^* \leqslant J$, the subject's $\boldsymbol{x}_{ki}$ and $\boldsymbol{z}_{ki}$ covariates are repeated for all j^* observations, the $\log(t^*_{kij})$ offset term is calculated for each interval, and additional reference coded indicator values $\boldsymbol{v}_{kij}$ for the time interval $j = 1, ..., j^*$ are specified. The first element of $\boldsymbol{v}_{kij}$ is always 1, encoding a fixed effect intercept which represents time interval 1. For time interval $j > 1$, the j-th element of v_{kij} is also 1. All other values of v_{kij} are set to 0.

Instead of estimating $\boldsymbol{\psi}^*$ directly, we reformulate this quantity as $\tilde{\boldsymbol{\psi}}$, where $\psi^*_1 = \tilde{\psi}_1$ and $\psi^*_j = \tilde{\psi}_1 + \tilde{\psi}_j$. In this formulation, $\tilde{\psi}_1$ estimates the log of the baseline hazard for time interval $[\tau_0, \tau_1)$, and $\tilde{\psi}_1 + \tilde{\psi}_j$ estimates the log of the baseline hazard for time interval $[\tau_{j-1}, \tau_j)$ for $j = 2, ..., J$. By estimating the log baseline hazard parameters in this way, we are including a fixed effect intercept in our model. By including a fixed effect intercept, we ensure that the full $\boldsymbol{z}_{ki}$ vector, which includes a random intercept, is a subset of the subject's fixed effects.

We can re-write the log-linear model of (5.3) as

$$\log \mu_{kij} = \log t^*_{kij} + \boldsymbol{v}^T_{kij}\tilde{\boldsymbol{\psi}} + \boldsymbol{x}^T_{ki}\boldsymbol{\beta} + \boldsymbol{z}^T_{ki}\boldsymbol{B}\boldsymbol{\alpha}_k. \tag{5.12}$$

5.2.2.1 Monte-Carlo E-step

The integrals in the Q-function do not have closed forms. We approximate these integrals using a Markov Chain Monte Carlo (MCMC) sample of size M from the posterior density

$\phi(\boldsymbol{\alpha}_k|\boldsymbol{D}_{k,o};\boldsymbol{\theta}^{(s)})$. Let $\boldsymbol{\alpha}_k^{(s,m)}$ be the m^{th} simulated r-dimensional vector from the posterior of the latent common factors, $m = 1, ..., M$, at the s^{th} iteration of the algorithm for group k. The integral in (5.10) can be approximated as

$$\begin{aligned} Q_1(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s)}) &\approx -\frac{1}{M}\sum_{m=1}^{M}\sum_{k=1}^{K}\log f(\boldsymbol{d}_k|\boldsymbol{X}_k,\boldsymbol{\alpha}_k^{(s,m)};\boldsymbol{\theta}) \\ &= -\frac{1}{M}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{i=1}^{n_k}\sum_{j=1}^{J} I(t^*_{kij}>0)\left[d_{kij}\log\mu_{kij}^{(s,m)} - \mu_{kij}^{(s,m)}\right], \end{aligned}$$

where $\log\mu_{kij}^{(s,m)} = \log t^*_{kij} + \boldsymbol{v}_{kij}^T\tilde{\boldsymbol{\psi}} + \boldsymbol{x}_{ki}^T\boldsymbol{\beta} + \boldsymbol{z}_{ki}^T\boldsymbol{B}\boldsymbol{\alpha}_k^{(s,m)}$. We use the No-U-Turn Sampler Hamiltonian Monte Carlo sampling procedure (NUTS HMC) from the Stan software (Carpenter *et al.*, 2017; Hoffman and Gelman, 2014) so that we can perform the E-step quickly and efficiently.

5.2.2.2 M-step

In the M-step of the algorithm, we aim to minimize

$$Q_{1,\lambda}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s)}) = Q_1(\boldsymbol{\theta}|\boldsymbol{\theta}^{(s)}) + \lambda_0\sum_{j=1}^{p}\rho_0(\beta_j) + \lambda_1\sum_{t=1}^{q}\rho_1(||\boldsymbol{b}_t||_2) \tag{5.13}$$

with respect to $\boldsymbol{\theta} = (\boldsymbol{\beta}^T, \boldsymbol{b}^T, \tilde{\boldsymbol{\psi}}^T)^T$. We do this by using a Majorization-Minimization algorithm with penalties applied to the fixed effects $\boldsymbol{\beta}$ and the rows of $\boldsymbol{B}$.

Let s represent the iteration of the MCECM algorithm, and let g represent the iteration within a particular M-step of the MCECM algorithm. Algorithm 5 describes the M-step of the s^{th} iteration of the MCECM algorithm.

5.2.2.3 MCECM algorithm

The full MCECM algorithm for estimating the parameters with a particular (λ_0, λ_1) proceeds as described in Algorithm 6. Supplementary Material Sections D.1.2 and D.1.3 outline the process of model selection and finding optimal tuning parameters. The Supplementary Material Section D.1.4 gives further details on initialization and convergence.

Algorithm 3 M-step of the MCECM algorithm

1. The parameters $\boldsymbol{\theta}^{(s,0)}$ for M-step iteration $g = 0$ are initialized using the results from the previous M-step, $\boldsymbol{\theta}^{(s-1)}$. The step size for the Majorization-Minimization algorithm c^0 is also initialized using the step size at the end of the previous M-step.
2. Conditional on $\boldsymbol{\beta}^{(s,g-1)}$ and $\boldsymbol{b}^{(s,g-1)}$, each $\psi_j^{*(s,g)}$ for $j = 1, ..., J$ is given a single update using the Majorization-Minimization algorithm specified by Breheny and Huang (2015) with no penalization applied.
3. Conditional on $\boldsymbol{b}^{(s,g-1)}$ and $\psi^{*(s-1)}$, each $\beta_l^{(s,g)}$ for $l = 1, ..., p$ is given a single update using the Majorization-Minimization algorithm specified by Breheny and Huang (2015).
4. For each group k in $k = 1, ..., K$, the augmented matrix $\tilde{\boldsymbol{z}}_{ki} = (\tilde{\boldsymbol{\alpha}}_k^{(s)} \otimes \boldsymbol{z}_{ki})J$ is created for $i = 1, ..., n_k$ where $\tilde{\boldsymbol{\alpha}}_k^{(s)} = ((\boldsymbol{\alpha}_k^{(s,1)})^T, ..., (\boldsymbol{\alpha}_k^{(s,M)})^T)^T$.
5. Conditional on the recently updated $\boldsymbol{\beta}^{(s,g)}$ and $\boldsymbol{\psi}^{*,(s,g)}$, each $\boldsymbol{b}_t^{(s,g)}$ for $t = 1, ..., q$ is updated using the Majorization-Minimization coordinate descent grouped variable selection algorithm specified by Breheny and Huang (2015).
6. The step size c^{g+1} is updated (if necessary) using the Proximal Gradient algorithm (Parikh *et al.*, 2014).
7. Steps 2 through 6 are repeated until the M-step convergence criteria are reached or until the M-step reaches its maximum number of iterations.

Algorithm 4 Full MCECM algorithm for single (λ_0, λ_1) penalty combination

1. Fixed effects $\boldsymbol{\psi}^*$ and $\boldsymbol{\beta}^{(0)}$ and the random effects $\boldsymbol{b}^{(0)}$ are initialized as discussed the Web Appendix.
2. In each E-step for EM iteration s, a burn-in sample from the posterior distribution of the random effects is run and discarded. A sample of size $M^{(s)}$ from the posterior is then drawn and retained for the M-step.
3. Parameter estimates of $\boldsymbol{\beta}^{(s)}$, $\boldsymbol{b}^{(s)}$, and $\tau^{(s)}$ are then updated as described in the M-step procedure given above.
4. Steps 2 and 3 are repeated until the convergence condition is met a pre-specified consecutive number of times or until the maximum number of EM iterations is reached.

5.2.3 Estimation of the number of latent factors

Performing our proposed **phmmPen_FA** method requires specifying the number of latent factors r. Since r is typically unknown *a priori*, this value needs to be estimated. Here, we use the Growth Ratio (GR) procedure (Ahn and Horenstein, 2013).

The GR method for our application requires a $q \times K$ matrix of observed random effects. Since these random effects cannot be directly observed, we instead calculate pseudo random effects by first fitting a penalized piecewise constant survival model with a small penalty to each group individually. We then take these group-specific estimates and center them so that all features have a mean of 0. Let these q-dimensional group-specific estimates be denoted as $\hat{\boldsymbol{\gamma}}_{\boldsymbol{k}}$ for each group $k = 1, ..., K$. We then define $\boldsymbol{G} = (\hat{\boldsymbol{\gamma}}_{\boldsymbol{1}}, ..., \hat{\boldsymbol{\gamma}}_{\boldsymbol{K}})$ as the final $q \times K$ matrix of pseudo random effects.

Let $\psi_j(A)$ be the j-th largest eigenvalue of the positive semidefinite matrix A, and let $\tilde{\mu}_{qK,j} \equiv \psi_j(\boldsymbol{G}\boldsymbol{G}^T/(qK)) = \psi_j(\boldsymbol{G}^T\boldsymbol{G}/(qK))$. To find the GR estimator, we first order the eigenvalues of $\boldsymbol{G}\boldsymbol{G}^T/(qK)$ from largest to smallest. Then, we calculate the following ratios:

$$GR(j) \equiv \frac{\log[V(j-1)/V(j)]}{\log[V(j)/V(j+1)]} = \frac{\log\left(1+\tilde{\mu}^*_{qK,j}\right)}{\log\left(1+\tilde{\mu}^*_{qK,j+1}\right)}, \quad j = 1, 2, ..., U \tag{5.14}$$

where $V(j) = \sum_{l=j+1}^{\min(q,K)} \tilde{\mu}_{qK,l}$, $\tilde{\mu}^*_{qK,j} = \tilde{\mu}_{qK,j}/V(j)$, and U is a pre-defined constant. Then, we estimate r by

$$\widehat{r}_{GR} = \max\nolimits_{1 \leqslant j \leqslant U} GR(j) \tag{5.15}$$

5.3 Simulations

In this section, we examine how well the **phmmPen_FA** algorithm performs variable selection on the fixed and random effects covariates for piecewise constant hazard mixed effects models under several different conditions. In all of these simulations, we use the MCP penalty (MCP penalty for the fixed effects, group MCP penalty for the rows of the $\boldsymbol{B}$ matrix) and the BIC-ICQ (Ibrahim *et al.*, 2011) model selection criterion with the abbreviated two-stage grid search as described in the Supplementary Material Section D.1.2. In order to determine the robustness of our variable selection procedure based on the assumed value of r, we fit models in one of two ways: we estimated the number of common factors r using the Growth Ratio estimation procedure, or we use the true value of r.

5.3.1 Variable selection in survival data with 100 predictors

We examine the performance and scalability of the **phmmPen_FA** algorithm when performing variable selection in high dimensions of $p = 100$ total predictors. We simulated survival data from a piecewise constant hazard mixed effect model with p predictors. Of p total predictors, we assume that the first 5 predictors have truly non-zero fixed and random effects, and the other $p - 5$ predictors have zero-valued fixed and random effects. We specified a full model for the algorithm such that the random effect predictors equalled the fixed effect predictors (e.g. $q = p$), and our aim was to select the set of true predictors and random effects.

To simulate the data, we set the total sample size to $N = 1000$ and considered the number of groups K to be either 5 or 10, with an equal number of subjects per group. We set up the random effects covariance matrix by specifying a $\boldsymbol{B}$ matrix with dimensions $(p + 1) \times r$, where $p + 1$ represents the p predictors specified in the $\boldsymbol{X}$ matrix plus the random intercept, and the number of latent common factors r was set to three. Six of these $p + 1$ rows—corresponding to the true 5 predictors plus the random intercept—had non-zero elements, while the remaining $p - 5$ rows were set to zero. For each value of r, we considered a $\boldsymbol{B}$ matrix that produced $\boldsymbol{\Sigma} = \boldsymbol{B}\boldsymbol{B}^T$ with either small or moderate variances and eigenvalues; see Section D.1.1 of the Supplementary Material for further details. These two cases are referred to as the 'small' or 'moderate' $\boldsymbol{B}$ matrices in the simulation results presented in this section. We generate both moderate and strong predictor effects, where all 5 of the true fixed effects have coefficient values of 0.5 or 1.0, respectively. Each condition was evaluated using 100 total simulated datasets.

In order to sample event times $\boldsymbol{T} = (\boldsymbol{T}_1^T, ..., \boldsymbol{T}_k^T)^T$ where $\boldsymbol{T}_k = (T_{k1}, ..., T_{kn_k})^T$, we defined five half-year time intervals as $\{[0, 0.5), [0.5, 1.0), [1.0, 1.5), [1.5, 2.0), [2.0, \infty)\}$. The corresponding log baseline hazard values for these intervals were $\boldsymbol{\psi}_j^* = (-1.5, 1.0, 2.7, 3.7, 6.8)$.

For group k, we generated the event times T_{ki}, $i = 1, ..., n_k$, using the following procedure: We first simulated values from the exponential distribution $e_{kij} \sim Exp(R_{kij})$ starting with $j = 1$, where the exponential rate $R_{kij} = \exp\big(\psi_j + \boldsymbol{x}_{ki}^T\boldsymbol{\beta} + \boldsymbol{z}_{ki}^T\boldsymbol{\gamma}_k\big)$, where $\boldsymbol{\gamma}_k \sim N_6(0, \boldsymbol{B}\boldsymbol{B}^T)$. If the inequality $\tau_j < \tau_{j-1} + e_{kij}$ was true, then we simulated e_{kij} using the $j + 1$ interval parameters until either the inequality $\tau_j >= \tau_{j-1} + e_{kij}$ held for a particular j^* or the last time interval J was reached. We then defined $T_{ki} = e_{kij*} + \tau_{j*-1}$.

For individual i in group k, the vector of predictors for the fixed effects is given as $\boldsymbol{x}_{ki} = (x_{ki,1}, ..., x_{ki,p})^T$, which does not inlcude an intercept, and we define the random effects $\boldsymbol{z}_{ki} = (1, \boldsymbol{x}_{ki})$, where $x_{ki,l} \sim N(0,1)$ for $l = 1, ..., p$, and each $\boldsymbol{x}_l$ was standardized as described in Section 5.2.1. We include a random intercept in the random effects predictors $\boldsymbol{z}_{ki}$ to allow for the baseline hazard to vary across groups.

We prepared the data to be fit with a piecewise constant hazard survival model by calculating eight time intervals—specified such that there were an approximately equal number of events within each time interval—and then creating the long-form dataset specified in Section 5.2.2 using the `survival::survSplit()` function from the **survival** R package (Therneau, 2021; Terry M. Therneau and Patricia M. Grambsch, 2000).

The results for these simulations are presented in Tables 5.1 and 5.2. Table 5.1 provides the average true and false positive percentages for both the fixed and random effects variable selection, the median time in hours to complete the variable selection procedure, and the average of the mean absolute deviation between the coefficient estimates and the true coefficients across all simulation replicates. The true positive percentages express the average percent of the true predictors selected in the best models across simulation replicates, and the false positive percentages express the average percent of false predictors selected in the best models. Table 5.2 gives the Growth Ratio estimation procedure results, including the average estimate of r and the proportion of times that the Growth Ratio estimate of r was underestimated, correct, or overestimated. All simulations were completed on a high performance computing cluster with CPU Intel processors between 2.3Ghz and 2.5GHz.

We see from Table 5.1 that the **phmmPen_FA** method is able to accurately select both the fixed and random effects within the piecewise constant hazard mixed effects model across a variety of conditions. The true positive rates of the **phmmPen_FA** method are generally above 90% for both fixed and random effects; the fixed effects true positives increase when the true predictor effects are larger, and the random effects true positives increase when the number of

β	K	$\boldsymbol{B}$	r Est.	TP %	FP %	TP %	FP %	T^{med}	Abs. Dev.
				Fixef	Fixef	Ranef	Ranef		(Mean)
0.5	5	Small	True	91.80	2.39	93.00	0.65	3.81	0.23
			GR	90.80	2.46	91.60	1.40	2.34	0.22
		Moderate	True	91.00	4.85	94.40	1.31	6.96	0.34
			GR	91.00	4.18	92.40	2.13	3.78	0.32
	10	Small	True	94.60	2.18	98.60	0.99	4.81	0.17
			GR	94.00	3.28	95.80	1.63	2.66	0.17
		Moderate	True	90.00	5.25	94.60	3.57	6.08	0.24
			GR	86.20	6.42	93.40	3.83	3.02	0.23
1.0	5	Small	True	99.20	1.05	96.00	0.16	6.01	0.26
			GR	99.00	1.09	93.20	0.54	3.50	0.25
		Moderate	True	97.60	2.92	95.20	0.65	8.72	0.36
			GR	95.20	2.75	94.20	1.22	3.63	0.34
	10	Small	True	100.00	1.02	99.60	0.15	5.14	0.20
			GR	99.80	1.20	97.00	0.34	2.97	0.23
		Moderate	True	98.80	2.39	99.60	1.01	7.55	0.27
			GR	98.20	4.22	98.80	0.66	4.02	0.33

Table 5.1: Variable selection results for the $p = 100$ piecewise constant hazard mixed effects simulations, including true positive (TP) percentages for fixed and random effects, false positive (FP) percentages for fixed and random effects, the median time in hours for the algorithm to complete (T^{med}), and the average of the mean absolute deviation (Abs. Dev. (Mean)) between the coefficient estimates and the true β values across all simulation replicates. Column $\boldsymbol{B}$ describes the general size of both the variances and eigenvalues of the resulting $\boldsymbol{\Sigma} = \boldsymbol{B}\boldsymbol{B}^T$ random effects covariance matrix. Column 'r Est.' refers to the method used to specify r in the algorithm: the Growth Ratio (GR) estimate or the true value of r.

groups in the data increase. The false positive rates are less than 6.5% for fixed effects and less than 3.9% for the random effects across all conditions.

We can see from Table 5.2 that the Growth Ratio estimation procedure generally underestimates the number of latent factors r for the simulated data set-ups used in this section. We expect that this is a result of a combination of reasons, including relatively low numbers of groups K in the data and $\boldsymbol{B}$ matrices that created $\boldsymbol{\Sigma}$ matrices with relatively low eigenvalues. This is supported by the results that show an improvement in the accuracy of the estimation as the number of groups and the relative size of the $\boldsymbol{B}$ matrix increases. Additionally, the Growth Ratio utilizes group-specific penalized piecewise constant hazard coefficient estimates, and these estimates

β	K	$\boldsymbol{B}$	Avg. r	r Underestimated %	r Correct %	r Overestimated %
0.5	5	Small	2.00	100	0	0
		Moderate	2.00	100	0	0
	10	Small	2.07	95	3	2
		Moderate	2.20	85	10	5
1.0	5	Small	2.00	100	0	0
		Moderate	2.00	100	0	0
	10	Small	2.13	88	11	1
		Moderate	2.19	83	15	2

Table 5.2: Results of the Growth Ratio r estimation procedure for $p = 100$ piecewise constant hazard mixed effects simulation results, including the average estimate of r across simulations and percent of times that the estimation procedure underestimated r, gave the true r, or overestimated r. Column $\boldsymbol{B}$ describes the general size of both the variances and eigenvalues of the resulting $\boldsymbol{\Sigma} = \boldsymbol{B}\boldsymbol{B}^T$ random effects covariance matrix.

might be sensitive to the fact that for some simulated datasets, not all groups had a sufficient number of events within each time interval to get reasonable $\boldsymbol{\psi}^*$ estimates, possibly leading to less than stable pseudo random effect estimates.

Even though the Growth Ratio procedure consistently underestimated r, this did not strongly impact the variable selection results nor the bias of the fixed effects estimates selected in the best models. When the algorithm used the Growth Ratio estimate of r instead of the true estimate of r, the true and false positive rates remained very consistent, with only slight decreases in true positive rates for the fixed and random effects when the Growth Ratio procedure is used. The largest impact that underestimating r had on the bias of the fixed effects estimates was when $K = 10$ and $\beta = 1.0$.

5.3.2 Variable selection in survival data with 500 predictors

In order to further illustrate the scalability of our method, we applied our method to survival simulations with $p = 500$ covariates. We simulated the event and censoring times from a piecewise constant hazard mixed effects model much like the procedure described in Section 5.3.1, except the total number of predictors used in the analyses was $p = 500$ instead of $p = 100$. All simulations assumed the true number of latent factors r was 3 and the Growth Ratio method was used to estimate r. Just as in the $p = 100$ survival simulations, we specified a full model for the algorithm such the random effect predictors equalled the fixed effect predictors (e.g. $q = p$), and our aim was to select the set of true predictors and random effects. The variable selection results to these simulations are given in Table 5.3. The median times needed to complete these simulations took between 11.9 and 21.3 hours.

β	K	$\boldsymbol{B}$	Avg. r	TP %	FP %	TP %	FP %	T^{med}	Abs. Dev.
				Fixef	Fixef	Ranef	Ranef		(Mean)
0.5	5	Small	2.00	86.40	1.97	77.40	0.81	14.19	0.18
		Moderate	2.00	80.60	3.91	76.40	1.63	19.40	0.24
	10	Small	2.01	91.40	1.79	88.00	0.53	16.21	0.16
		Moderate	2.04	81.60	5.82	80.80	2.44	23.62	0.20
1.0	5	Small	2.00	99.40	0.36	91.40	0.02	18.20	0.30
		Moderate	2.00	93.60	0.79	85.00	0.12	19.77	0.39
	10	Small	2.06	100.00	0.56	95.40	0.04	16.78	0.30
		Moderate	2.03	97.20	1.31	92.80	0.18	24.31	0.40

Table 5.3: Variable selection results for the $p = 500$ piecewise constant hazard mixed effects simulations, including true positive (TP) percentages for fixed and random effects, false positive (FP) percentages for fixed and random effects, the median time in hours for the algorithm to complete (T^{med}), and the average of the mean absolute deviation (Abs. Dev. (Mean)) between the coefficient estimates and the true β values across all simulation replicates. Column $\boldsymbol{B}$ describes the general size of both the variances and eigenvalues of the resulting $\boldsymbol{\Sigma} = \boldsymbol{B}\boldsymbol{B}^T$ random effects covariance matrix. Column 'r Est.' refers to the method used to specify r in the algorithm: the Growth Ratio (GR) estimate or the true value of r.

5.4 Case study: Pancreatic Ductal Adenocarcinoma

Patients diagnosed with Pancreatic Ductal Adenocarcinoma (PDAC) generally face a very poor prognosis, where the 5-year survival rate is 6% (Khorana *et al.*, 2016). Consequently, it is of clinical interest to robustly identify gene signatures that are associated with overall survival to better predict patient prognosis in the clinic.

Selecting gene signatures for the prediction of clinical outcomes, including survival outcomes, can often be inconsistent across biomedical studies, where gene signatures identified in one study may have little or no overlap with ones identified in other studies (Waldron *et al.*, 2014). Consequences of this lack of replicability in gene signature selection include variable accuracy in predicting clinical outcomes in new studies using these models (Sotiriou and Piccart, 2007; Waldron *et al.*, 2014) and contradictory effect estimates relating genes to the outcome (Swisher *et al.*, 2012). This lack of replicability across studies can come from small sample size (Sotiriou and Piccart, 2007) and differences in data pre-processing steps (Lusa *et al.*, 2007; Paquet and Hallett, 2015), among other sources.

In order to improve replicability in the prediction of survival in PDAC, we combine PDAC gene expression data from seven different studies. The studies used in these analyses are summarized in Supplementary Material Table D.1. The seven combined studies resulted in a sample size of 879 subjects with 539 events. In order to account and adjust for between-study heterogeneity, we apply our new method **phmmPen_FA** to fit a penalized piecewise constant hazard mixed effects model to our data to select predictors with study-replicable effects, where we assume that predictor effects may vary between studies.

Moffitt *et al.* (2015) identified a 500-member gene list relevant to classifying two PDAC tumor subtypes they identified—basal and classical—which were prognostic of survival. Therefore, we decided to limit our initial interest to these 500 genes. Of these 500 genes, 420 of these genes were common among all of the datasets. We removed 20% of the genes with the lowest gene expression based on their average rank, leaving 336 genes.

We integrated gene expression data from multiple studies by first using the data integration rank transformation technique as specified by Rashid *et al.* (2020), allowing us to sidestep complex questions regarding how to cross-normalize data at training time. This integration technique creates top scoring pairs (TSPs). To illustrate the interpretation of TSPs, let $g_{ki,A}$ and $g_{ki,B}$ be the raw expression of genes A and B in subject i of group k. For each gene pair ($g_{ki,A}$, $g_{ki,B}$), the TSP is an indicator $I(g_{ki,A} > g_{ki,B})$ which specifies which of the two genes has higher expression in the subject. We denote a TSP predictor as "GeneA_GeneB". In the dataset, we use 168 TSP predictors. The Supplementary Material Section D.1.5 provides additional details on the data processing and selection of the TSPs used in the analysis.

Due to the presence of several pairwise Spearman correlation values greater than 0.5 between the TSPs used in the analyses, we used the Elastic Net penalization procedure (Friedman *et al.*, 2010) to balance between ridge regression and the MCP penalty. We let π represent the balance between ridge regression and the MCP penalty, where $\pi = 0$ represents ridge regression and $\pi = 1$ represents the MCP penalty. In these analyses, we let $\pi = 0.9$, and we estimated the number of common factors r using the Growth Ratio procedure. The same value of π was used for both the fixed effects and random effects penalization. Sensitivity analyses for different π values and for different values of r are presented in Supplementary Material Section D.1.6, and the sequence of λ penalties used the the variable selection procedure is described in Supplementary Material Section D.1.3.

In the best model according to the BIC-ICQ model selection criteria, 19 of the 168 TSP covariates were selected to have non-zero fixed effect values and were therefore considered important for the prediction of survival in PDAC subjects. The log of the hazard ratios for these 19 TSP covariates are presented in Figure 5.1. Log hazard ratios that are positive imply that having a higher expression in the first gene of the TSP compared to the second gene increases a subject's risk of death, and negative hazard ratios conversely imply that having a higher expression in the first gene of the TSP compared to the second gene decreases a subject's risk of death. The best model contained a random intercept and a random slope for the TSP CYP2C18_COX6B2. The Growth Ratio procedure estimated $r = 2$ latent common factors to model the random effects. The time to complete the variable selection procedure was 2.1 hours.

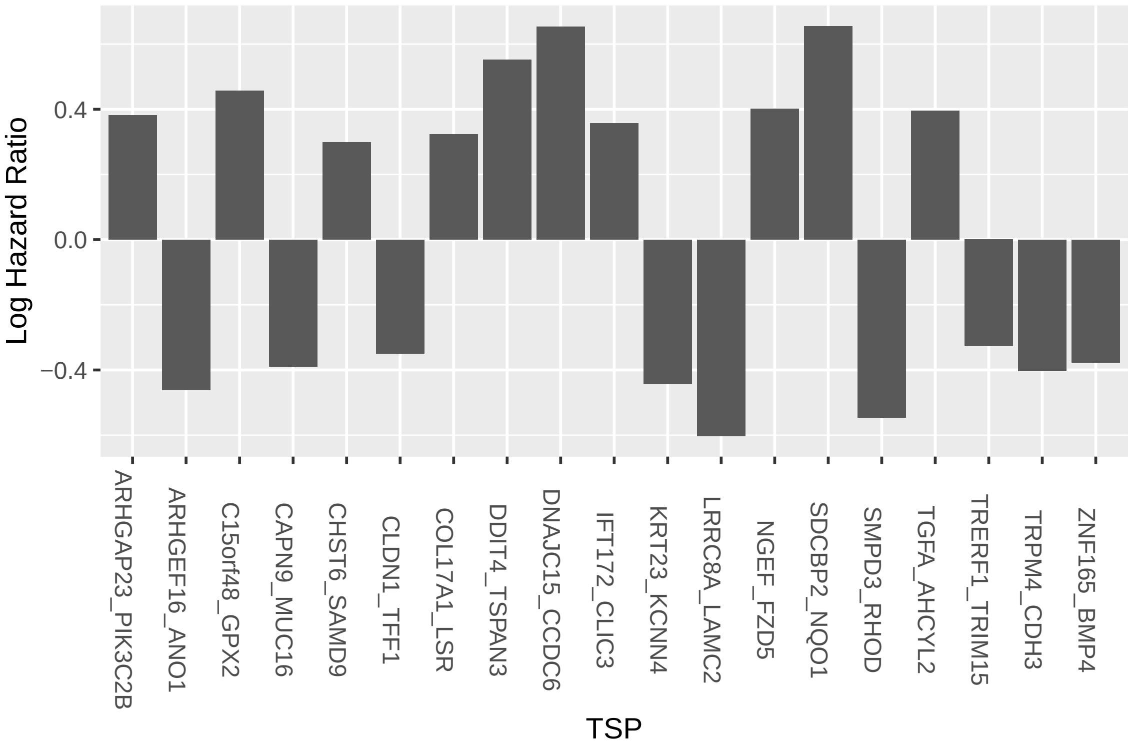


Figure 5.1: Log hazard ratios for the TSP covariates selected during the Chapter 5 case study analysis.

5.5 Discussion

We have shown through simulations and a case study of pancreatic ductal adenocarcinoma patients that we can extend the method to perform variable selection in high dimensional mixed effects models to survival data. We accomplish this by approximating proportional hazards mixed effects models using a piecewise constant hazard mixed effects model and then applying the Monte Carlo Expectation Conditional Minimization (MCECM) algorithm to simultaneously select for fixed and random effects. We incorporate the factor model decomposition of the random effects proposed in Heiling *et al.* (2023b) (recall Chapter 4) in order to scale this method to larger dimensions, e.g. hundreds of predictors.

The simulations presented in Section 5.3 show that the **phmmPen_FA** method can accurately select both fixed and random effects even for small or moderate effect sizes, which reflects hazard values and variations in typical survival data. By using the factor model decomposition of the random effects, this model selection procedure can be accomplished within reasonable time frames.

Our method is limited by the need to provide an estimate for the number of latent factors that model the random effects. The simulation results showed that the Growth Ratio procedure tended to underestimate this value for the simulation conditions that we considered. However, even when the number of latent factors was estimated incorrectly by the Growth Ratio procedure, this mis-specification had very little impact on the general variable selection performance or the fixed effects coefficient estimates. Therefore, our method is not sensitive to the estimation of the number of latent factors.

5.6 Software

Software in the form of R code is available through the GitHub repository `https://github.com/hheiling/glmmPen`. Code to run the simulations and the case study analysis is available through the GitHub repository `https://github.com/hheiling/paper_phmmPen_FA`.

5.7 Conclusion

In Chapters 3, 4, and 5, we introduced the methods **glmmPen**, **glmmPen_FA**, and **phmmPen_FA**. These methods have increased the feasible dimensionality of performing variable selection of both fixed and random effects within mixed models. Our variable selection of mixed effects models can be applied to Binomial, Gaussian, and Poisson distributional families, as well as survival data. By incorporating a factor model decomposition on the random effects, we were able to extend our procedure to apply to cases with hundreds of covariates. We have developed a user-friendly R package that incorporates the **glmmPen**, **glmmPen_FA**, and **phmmPen_FA** methods; this package is available on CRAN and GitHub. Providing accessible software for our methods helps encourage the interest in and utilization of our methods.

We hope to someday extend our methods even further by enabling our procedure to apply to a larger range of outcome data types, such as negative binomial and multi-categorical outcomes, as well as allow for non-canonical links within the current distributional families of Binomial, Gaussian, and Poisson. We are also interested in continuing to investigate methodological and computational modifications that may allow us to perform variable selection in mixed effects models with thousands of covariates.

APPENDIX A: APPENDIX FOR CHAPTER 2

A.1 Chapter 2 additional simulation details

A.1.1 Transcript cluster selection: Large isoform effect sizes

This section describes the procedure used to select transcript clusters that have isoforms highly expressed in one cell type but minimally expressed or not expressed at all in the other two cell types. In the first part of the section, we describe the specific procedure we utilized to select transcript clusters for use in the *in silico* Blueprint analysis described in Section 2.3; many of the pure sample isoform parameter estimates from these clusters were also used in the simulations presented in Section 2.4. Later in the section, we discuss how these steps can be generalized for those wishing to use the IsoDeconvMM procedure.

A.1.1.1 *In Silico* Blueprint analysis

Ten samples per cell type were selected from the cell type specific gene expression data generated by the Blueprint project Chen *et al.* (2016). These 30 samples were separate from the samples used during the IsoDeconvMM algorithm fit and the samples used to create the mixture files. For each of these samples, the function `isoDetector` from the isoform R package Sun *et al.* (2015) was applied to obtain penalized estimation of isoform level expression for each cluster.

Next we outline the procedure for cluster selection. The transcript clusters were first filtered such that we only considered clusters on chromosomes one through four. Additionally, transcript clusters were filtered such that every cluster had between 3 and 20 isoforms.

For each cell type, we sought to select a cluster if it has at least one isoform with high expression in one cell type, and no or minimal expression in all other cell types. The same procedure is applied to each cell type and here we just use cell type one as an example. For each transcript cluster, we identified isoforms that were sufficiently expressed in cell type one (e.g., it had non-zero abundance values in at least 9 of the 10 samples for cell type one). For each isoform that met this criteria, we calculated the fold change of its average abundance in cell type one vs the other two cell types combined.

In addition to fold change, we also applied hypothesis testing for cluster selection. Again, consider cluster selection for cell type one. We again identified isoforms that were sufficiently expressed in cell type one (e.g., it had non-zero abundance values in at least 9 of the 10 samples for cell type one). For each isoform that was expressed in cell type one, a one-sided Wilcoxon rank sum test was performed to test the hypothesis that this isoform has higher abundance in cell type one than the other two cell types combined.

Isoforms that resulted in Bonferroni-adjusted p-values below the 0.05 threshold from the Wilcoxon rank sum tests were kept for further consideration. Of the isoforms that met this criteria, the 60 isoforms with the largest fold change values from each cell type were selected. The union across all cell types of the clusters associated with these best isoforms gave 130 transcript clusters.

Once the pure sample fit portion of the IsoDeconvMM algorithm was applied to these transcript clusters, some further filtration was applied. Clusters whose pure sample isoform Diriclet parameter values resulted in `NA` values or extremely large and divergent values (more than two values were greater than 500) were excluded from further consideration. In the case when five pure samples were used to estimate the cell type specific parameters, eight clusters met this exclusion criteria.

Once these clusters were excluded, n_s isoforms with the greatest fold change values for each cell type were selected. We adjusted the value of n_s so that the total number of transcript clusters selected was 100, 50, 25, and 10.

A.1.1.2 Generalization of procedure

We provide here a generalization of the above procedure. For general data with K cell types, we recommend obtaining at least 5 pure cell type reference samples from each cell type. On each of the pure cell type reference samples, run the `isoDetector` function in order to obtain the abundance estimates of each isoform within each transcript cluster. For a particular cell type k, perform the following steps:

1. Identify isoforms where no more than one of the pure reference samples for cell type k have an estimated abundance of zero for that isoform.

2. For each isoform that meets the criteria of step 1, sum up the abundance estimates of the isoform within the samples of cell type k and sum up the abundance estimates of the isoform within all other cell type samples. Calculate the fold change estimate

3. For each isoform that meets the criteria of step 1, perform a one-sided Wilcoxon rank sum test to test the hypothesis that this isoform has higher abundance in cell type one than the other two cell types combined. Calculate Bonferroni-adjusted p-values and ignore isoforms that give adjusted p-values above a certain cut-off (e.g., cut-off 0.05).

4. Of the isoforms that meet the criteria of step 3, examine their fold change estimates. At this step, one could either pick the X isoforms with the highest fold change values (e.g. $X = 50$ or $X = 25$) or pick the isoforms with fold change values above a particular threshold.

5. For the isoforms picked after step 4, identify the transcript clusters to which these isoforms belong.

Complete the above procedure for each cell type $k = 1, ..., K$. Use the transcript clusters identified with this procedure in the IsoDeconvMM analysis.

A.1.1.3 Initial points used for *In Silico* Blueprint analysis

Table A.1 comprises a systematic approach to selecting initial points, where the following scenarios are represented: extreme cases where one cell type dominates with a large proportion and the other cell types split the remaining proportion; the equality case where all cell types are

Table A.1: The 10 generic initial points used in the *in silico* Blueprint analysis

CT1	CT2	CT3
0.10	0.10	0.80
0.10	0.80	0.10
0.80	0.10	0.10
0.25	0.25	0.50
0.25	0.50	0.25
0.50	0.25	0.25
0.20	0.40	0.40
0.40	0.20	0.40
0.40	0.40	0.20
0.33	0.33	0.33

represented equally; and moderate cases that fall in between the extreme and equality cases. In the more general case of K cell types, we would also recommend setting up a mix of these three cases for the initial points. For the extreme cases, one could consider setting initial points in the following manner: $K - 1$ cell types initialized with proportion 0.10, and the K^{th} cell type initialized with the remaining proportion $(1 - 0.1 * (K - 1))$. When $K \geqslant 4$, it would be sufficient to leave out moderate cases and instead just add the equality case when each proportion is equal to $1/K$, which would not be much different from any moderate cases that could be specified. In the case of $K = 2$, we recommend adding the moderate cases of cell type proportion combinations $\{$0.25, 0.75$\}$ and $\{$0.33, 0.67$\}$. The IsoDeconvMM R package automatically recommends initial points in the above manner.

A.2 Chapter 2 supplementary methods

A.2.1 An overview

See Table A.2 for a summary of notation that will be refered to throughout Appendix A.2.

Table A.2: Notation for defining the IsoDeconvMM Model.

Pure Sample Expressions		
Value	**Dim.**	**Description**
$Y_{kj} = \{Y_{kjA}\}$	$E \times 1$	Vector of read counts across all E exon sets in the given gene for pure sample j of cell type k.
$Y_{kj(O)}$	1×1	Total read count outside gene of interest in pure sample j of cell type k.
γ_{kj}	$I \times 1$	Isoform expression parameters unique to pure sample j of cell type k.
τ_{kj}	1×1	Probability that a randomly selected read maps to the gene of interest in pure sample j of cell type k.
t_{kj}	1×1	The total read count for gene of interest in pure sample j of cell type k.
Mixture Sample Expressions		
Value	**Dim.**	**Description**
$Z = \{Z_A\}$	$E \times 1$	Vector of read counts across E exon sets belonging to gene of interest in the mixture sample. Denote its sum as $\left(Z_T = \mathbf{1}^T Z = \sum_{e=1}^{E} Z_A\right)$.
Z_{kA*}	1×1	Read count at exon set A in mixture sample attributable to cell type k.
γ_k^*	$I \times 1$	Isoform expression parameters for cell type k within the mixture sample.
τ_k^*	1×1	Probability that a randomly selected read from cell type k in the mixture sample maps to the gene of interest.
Cell-Type Specific and Cluster Level Parameters		
Value	**Dim.**	**Description**
$X = \{X_{Ai}\}$	$E \times I$	Matrix of effective lengths for E exon sets and I isoforms.
$\tilde{l}$	$I \times 1$	Vector of complete effective lengths of each utilized isoform $\left(\tilde{l}_i = \sum_A X_{Ai}\right)$.
$\rho = \{\rho_k\}$	$K \times 1$	Proportions of cell types $k = 1, ..., K$ present in the mixture.
α_k	$I \times 1$	Hyperparameters of isoform expression levels within cell type k.
β_k	2×1	Hyperparameters governing gene expression levels within cell type k.

A.2.2 Lemmas involving a multinomial distribution

Prior to specification of the IsoDeconv model, we develop a set of lemmas for the multinomial distribution which will allow easier specification in the following materials. For completeness, we define a multinomially distributed vector $X = (X_1, ..., X_R)$ with size n and proportions $p = (p_1, ..., p_R)$. The density function of $X \sim \texttt{Multinomial}\,(n, p)$ is given by:

$$P\left\{X_1, ..., X_R \middle| n, p\right\} = \begin{pmatrix} n \\ X_1, ..., X_R \end{pmatrix} \prod_{i=1}^{R} p_i^{X_i}$$

Lemma 1.1: Sum over Groups

W.L.O.G. construct the sum $X. = X_1 + ... + X_g$ and consider the grouped multinomial $X' = f(X) = (X., X_{g+1}, X_{g+2}, ..., X_R)$. Let S represent the set of vectors X such that $X' = f(X) = x$ where x is an arbitrary $(R - g + 1)$-dimensional non-negative vector summing to n. The density of this random variable is given by:

$$\begin{aligned} P\left\{X' = x \middle| n, p\right\} &= \sum_{X' \in S} \begin{pmatrix} n \\ X_1, ..., X_R \end{pmatrix} \prod_{i=1}^{R} p_i^{X_i} \\ &= \begin{pmatrix} n \\ x_1, x_2 ..., x_{R-g+1} \end{pmatrix} \prod_{i=g+1}^{R} p_i^{x_i} \left\{ \sum_{X \in S} \begin{pmatrix} x_1 \\ X_1, ..., X_g \end{pmatrix} \prod_{i=1}^{g} p_i^{X_i} \right\} \\ &= \begin{pmatrix} n \\ x_1, x_2 ..., x_{R-g+1} \end{pmatrix} \left(\sum_{i=1}^{g} p_i \right)^{x_1} \prod_{i=g+1}^{R} p_i^{x_{i-g+1}} \end{aligned}$$

Thus, it is clear that $X' \sim \texttt{Multinomial}\,(n, p')$ where $p' = (p_1 + ... + p_g, p_{g+1}, ..., p_R)$.

Lemma 1.2: Marginal of a Single Element

We extend (1.1) to the case where $X. = X_1 + ... + X_{R-1}$ and consider the distribution of $X' = f(X) = (X., X_R)$. Using (1.1) it is clear that $X' \sim \texttt{Multinomial}\,(n, (1 - p_R, p_R))$. Thus, it is obvious that:

$$X_R \sim \text{Bin}\left(n, p_R\right)$$

Lemma 2.1: Conditional over Multiple Elements

W.L.O.G. consider conditioning on the first g elements. Thus, we seek to specify the conditional density of $X^* = (X_{g+1}, ..., X_R)$ given $(X_1, ..., X_g)$. By lemma (1.1), we know that:

$$P\left\{X_1, ..., X_g\right\} = \begin{pmatrix} n \\ X_1, ..., X_g, n - X_1 - ... - X_g \end{pmatrix} \left\{\prod_{i=1}^{g} p_i^{X_i}\right\} \left(1 - p_1 - ... - p_g\right)^{n - X_1 - ... - X_g}$$

Thus, applying this to the definition of conditional densities, we have:

$$\begin{aligned} P\left\{X^* \middle| X_1, ..., X_g\right\} &= \frac{P\left\{X^* \cap (X_1, ..., X_g)\right\}}{P\left\{X_1, ..., X_g\right\}} \\ &= \frac{\begin{pmatrix} n \\ X_1, ..., X_R \end{pmatrix} \prod_{i=1}^{R} p_i^{X_i}}{\begin{pmatrix} n \\ X_1, ..., X_g, , n - X_1 - ... - X_g \end{pmatrix} \left\{\prod_{i=1}^{g} p_i^{X_i}\right\} \left(1 - p_1 - ... - p_g\right)^{n - X_1 - ... - X_g}} \\ &= \begin{pmatrix} n - X_1 - ... - X_g \\ X_{g+1}, ..., X_R \end{pmatrix} \left\{\prod_{i=g+1}^{R} \left(\frac{p_i}{1 - p_1 - ... - p_g}\right)^{X_i}\right\} \end{aligned}$$

Thus, it is clear that:

$$X^* \middle| \left(X_1, ..., X_g\right) \sim \texttt{Multinomial}\left(n - X_1 - ... - X_g, p^*\right)$$

where $p^* = \left(\frac{p_{g+1}}{1 - p_1 - ... - p_g}, \cdots, \frac{p_R}{1 - p_1 - ... - p_g}\right)$.

Lemma 2.2: Conditional of a Single Element

We consider a specific case of lemma (2.1) where $X^* = (X_2, \cdots, X_R)$. Thus, it is clear that:

$$X^* \big| X_1 \sim \texttt{Multinomial}\left(n - X_1, p_1^*\right) \quad \text{where} \quad p_1^* = \left(\frac{p_2}{1-p_1}, \cdots, \frac{p_R}{1-p_1}\right)$$

Lemma 3: Conditional Over Sums

Under the original framework, consider splitting the R elements of X into K distinct groups. W.L.O.G. we specify:

Table A.3: Group specifications of R and X in Lemma 3

Group 1	**Group 2**	$\cdots$	**Group K**
$X_1, \cdots, X_{k_1}$	$X_{k_1+1}, \cdots, X_{k_2}$	$\cdots$	$X_{k_{K-1}}, \cdots, X_R$
$p_1, \cdots, p_{k_1}$	$p_{k_1+1}, \cdots, p_{k_2}$	$\cdots$	$p_{k_{K-1}}, \cdots, p_R$

For convenience, define $S_j = \sum_{i=k_{j-1}+1}^{k_j} X_i$ where $k_0 = 1$. Additionally, define $p_j^* = \sum_{i=k_{j-1}+1}^{k_j} p_i$. Thus, we examine the following conditional density:

$$\begin{aligned}
P\left\{X_1, ..., X_R \middle| S_1, \cdots, S_K\right\} &= \frac{P\left\{X_1, \cdots, X_R\right\}}{P\left\{S_1, \cdots, S_K\right\}} \\
&= \frac{\displaystyle\binom{n}{X_1, ..., X_R} \prod_{j=1}^{R} p_j^{X_j}}{\displaystyle\binom{n}{S_1, ..., S_K} \prod_{j=1}^{K} p_j^{*S_j}} \\
&= \prod_{j=1}^{K} \left\{ \binom{S_j}{X_{k_{j-1}}, ..., X_{k_j}} \prod_{l=k_{j-1}+1}^{k_j} \left(\frac{p_l}{p_j^*}\right)^{X_l} \right\}
\end{aligned}$$

The second equality holds through repeated application of Lemma 1.1. The final equality demonstrates that the desired conditional distribution is the product of independent multinomials. Symbolically, we have:

$$X_1, \cdots, X_R \big| S_1, \cdots, S_K \sim \prod_{j=1}^{K} \texttt{Multinomial}\left(S_j, p'_j\right)$$

where $p'_j = \left(p_{k_{j-1}+1}, \cdots, p_{k_j}\right)/p^*_j$.

A.2.3 Stage 1 estimation: Pure sample necessities

For the following, refer to Table A.2 regarding notation. Additionally, note that the following specification is performed for a single gene (or single transcript cluster) only; subscripts related to gene identity are omitted for clarity. The following structure holds for a single purified reference sample and gene:

$$\begin{bmatrix} Y_{kj(O)} \\ Y_{kj} \end{bmatrix} \sim \texttt{Multinomial}\left(t_{kj}, \begin{bmatrix} 1 - \tau_{kj} \\ \tau_{kj} X \gamma_{kj} \end{bmatrix}\right)$$

Implicit in this construction are restrictions upon the τ_{kj} and γ_{kj}. As a single probability value, it must be that $0 \leqslant \tau_{kj} \leqslant 1$. However, the γ_{kj} pose a more complicated set of restrictions. Consider the following:

$$\begin{aligned} 1 &= (1 - \tau_{kj}) + \tau_{kj} \mathbf{1}^T X \gamma_{kj} \\ &= \mathbf{1}^T X \gamma_{kj} \end{aligned}$$

It is clear from the above that the $X\gamma_{kj}$ are conditional probabilities and thus must be non-negative. To ensure this, we restrict the γ_{kj} to be non-negative since the elements of X_{kj} are non-negative by definition. Using our summation constraints, we have:

$$1 = \mathbf{1}^T X \gamma_{kj} = \sum_{i=1}^{I} \tilde{l}_i \gamma_{kji}$$

This shows that the $\tilde{l}_i\gamma_{kji}$ are probabilities that a randomly selected read is attributable to isoform i for reference j of cell type k. Thus, it is clear that the γ_{kj} are collections of per-unit of effective length conditional probabilities that a read belongs to isoform i given that it maps to the gene of interest.

Thus, the likelihood for sample j of cell type k is given by:

$$\begin{aligned}\ell_{kj} &= Y_{kj(O)} \log\left(1 - \tau_{kj}\right) + \sum_{e=1}^{E} Y_{kje} \log\left(\tau_{kj} X_e^T \gamma_{kj}\right) \\ &= Y_{kj(O)} \log\left(1 - \tau_{kj}\right) + \left(\mathbf{1}^T Y_{kj}\right) \log\left(\tau_{kj}\right) + \sum_{e=1}^{E} Y_{kje} \log\left(X_e^T \gamma_{kj}\right)\end{aligned}$$

Given the gene and isoform expressions, the reference samples within and across cell types are independent. Thus, we may estimate the τ_{kj} and γ_{kj} separately within each sample.

Estimate τ_{kj}:

The maximum likelihood estimate of τ_{kj} is given by:

$$\hat{\tau}_{kj} = \frac{\mathbf{1}^T Y_{kj}}{t_{kj}}$$

Estimate γ_{kj}:

In order to estimate the isoform expressions for a single subject, we make some simplifying alterations to the effective length matrix X and reparametrize the isoform expression parameters. Consider the following, where X_{cj} refers to the j-th column of X.

$$\begin{aligned}\begin{bmatrix} X_{c1} & X_{c2} & \cdots & X_{cI} \\ \vert & \vert & & \vert \end{bmatrix} \begin{bmatrix} \gamma_{kj1} \\ \vdots \\ \gamma_{kjI} \end{bmatrix} &= \begin{bmatrix} X_{c1}/\tilde{l}_1 & X_{c2}/\tilde{l}_2 & \cdots & X_{cI}/\tilde{l}_I \\ \vert & \vert & & \vert \end{bmatrix} \begin{bmatrix} \tilde{l}_1\gamma_{kj1} \\ \vdots \\ \tilde{l}_1\gamma kjI \end{bmatrix} \\ &= \left[\sum_{i=1}^{I-1} \left(\frac{X_{ci}}{\tilde{l}_i}\right) \left(\tilde{l}_i\gamma_{kji}\right)\right] + \left(1 - \tilde{l}_i\gamma_{kj1} - \cdots - \tilde{l}_{I-1}\gamma_{kj,I-1}\right) \left(\frac{X_{cI}}{\tilde{l}_I}\right) \\ &= \left[\sum_{i=1}^{I-1} \left(\frac{X_{ci}}{\tilde{l}_i} - \frac{X_I}{\tilde{l}_I}\right) \left(\tilde{l}_i\gamma_{kji}\right)\right] + \left(\frac{X_{cI}}{\tilde{l}_I}\right)\end{aligned}$$

$$= \begin{bmatrix} X^*_{c1} & X^*_{c2} & \cdots & X^*_{cI} \\ | & | & & | \end{bmatrix} \begin{bmatrix} e^{-\gamma_{kj1}} \\ \vdots \\ e^{-\gamma_{kj,I-1}} \\ 1 \end{bmatrix}$$

where:

$$\begin{aligned} X^*_{cs} &= \left[X_s - \left(\tilde{l}_s/\tilde{l}_I\right) X_{cI}\right] && \text{for } j \in \{1, 2, ..., I-1\} \\ X^*_{cI} &= \frac{X_{cI}}{\tilde{l}_I} \\ X^* &= \begin{bmatrix} X^*_{c1} & \cdots & X^*_{cI} \end{bmatrix} \\ \gamma_{kji} &= e^{-\gamma^r_{kji}} && \text{for } i \in \{1, 2, ..., I-1\} \\ \gamma'_{kj} &= (\gamma_{kj1}, ..., \gamma_{kj,I-1}, 1) \end{aligned}$$

We optimize the likelihood with respect to these isoform expression parameters using R's `constrOptim` from the `alabama` package. To this end, we specify the derivative to improve efficiency of the routine.

In the following, let X^*_e refer to the e-th row of the matrix X^* and $X_{e,(I)}$ be the truncated version of this row excluding the last column entry. :

$$\frac{\mathrm{d}\ell_{kj}}{\mathrm{d}\gamma'_{kj}} = -\sum_{e=1}^{E} \left(\frac{Y_{kje}}{X^{*T}_e \gamma'_{kj}}\right) \left[X^*_{e,(I)} \circ e^{-\gamma^r_{kj}}\right]$$

A.2.4 Stage 2 estimation: Defining penalties

We must now incorporate the estimates from purified reference samples to guide estimation within the mixture. We choose to accomplish this using a penalty function over the isoform expression parameters within the mixture. As we have allowed for biological variance in gene and expression parameters across subjects and because these parameters are probabilities, it is natural to propose a dirichelet distribution over these parameters.

Normally, by placing a dirichelet distribution over these parameters, one would construct a likelihood function containing both pieces simultaneously. This likelihood would then be optimized with respect to all parameters, including hyperparameters, at the same time. However, we found this approach to be unstable. Thus, we separate the estimation of individual expression parameters from the hyperparameters to improve results. Fixing the individual gene and isoform expression parameters, we construct a likelihood optimization using the dirichelet piece. Optimization of this likelihood proceeds numerically using quasi-Newton methods and non-negativity constraints. The following derivatives improve accuracy of the estimates obtained from R's `nlminb`.

Gene Expression Penalty:

The likelihood for this penalty is given below

$$\ell_k^\tau = \sum_{j=1}^{n_k} \left\{ \ln\Gamma\left(\alpha_{k1}+\alpha_{k2}\right) - \ln\Gamma\left(\alpha_{k1}\right) - \ln\Gamma\left(\alpha_{k2}\right) + \left(\alpha_{k1}-1\right)\log(\tau_{kj}) + \left(\alpha_{k2}-1\right)\log(1-\tau_{kj}) \right\}$$

The necessary derivatives are provided here. Denote the digamma function by $\varphi()$ and trigamma by $\varphi_1()$ for this derivatives.

$$\nabla \ell_k^\tau = n_k \begin{bmatrix} \varphi(\alpha_{k1}+\alpha_{k2}) - \varphi(\alpha_{k1}) \\ \varphi(\alpha_{k1}+\alpha_{k2}) - \varphi(\alpha_{k2}) \end{bmatrix} + \begin{bmatrix} \sum_{j=1}^{n_k} \log(\tau_{kj}) \\ \sum_{j=1}^{n_k} \log(1-\tau_{kj}) \end{bmatrix}$$

$$\text{Hess}\left(\ell_k^\tau\right) = n_k \begin{bmatrix} \varphi_1(\alpha_{k1}+\alpha_{k2}) - \varphi_1(\alpha_{k1}) & \varphi(\alpha_{k1}+\alpha_{k2}) \\ \varphi(\alpha_{k1}+\alpha_{k2}) & \varphi(\alpha_{k1}+\alpha_{k2}) - \varphi(\alpha_{k2}) \end{bmatrix}$$

Isoform Expression Penalty:

As for the gene expression penalty, we define the likelihood here. To clarify the following terms, define $\beta_{k\cdot} = \sum_{i=1}^{I} \beta_{ki}$ and utilize the same definitions for φ and φ_1.

$$\ell_k^{\gamma} = \sum_{j=1}^{n_k} \left\{ \ln\Gamma\left(\beta_{k\cdot} - \sum_{i=1}^{I} \ln\Gamma\left(\beta_{ki}\right) \right) + \sum_{i=1}^{I} \left(\beta_{ki} - 1\right) \log\left(\tilde{l}_i \gamma_{kji} \right) \right\}$$

The necessary derivatives are specified below:

$$\nabla \ell_k^{\gamma} = n_k \begin{bmatrix} \varphi(\beta k\cdot) - \varphi(\beta_{k1}) \\ \vdots \\ \varphi(\beta k\cdot) - \varphi(\beta_{kI}) \end{bmatrix} + \begin{bmatrix} \sum_{j=1}^{n_k} \log\left(\tilde{l}_1 \gamma_{kj1}\right) \\ \vdots \\ \sum_{j=1}^{n_k} \log\left(\tilde{l}_I \gamma_{kjI}\right) \end{bmatrix}$$

$$\text{Hess}\left(\ell_k^{\gamma}\right) = n_k \left(\mathbf{1}\mathbf{1}^T \varphi_1\left(\beta_{k\cdot}\right) - \text{diag}_i\left(\varphi_1(\beta_{ki})\right) \right)$$

A.2.5 Stage 3 estimation: Mixture sample estimation

To structure the likelihood model within the mixture sample, consider the following underlying likelihood model. In this model, we assume that the number of reads mapping to each cell type within each gene and outside of it can be observed and that t_m represents the total read count in the mixture.

$$\begin{bmatrix} Z_{1(E)*} & \cdots & Z_{K(E)*} \\ Z_{11*} & & Z_{K1*} \\ \vdots & & \vdots \\ Z_{1E*} & & Z_{KE*} \end{bmatrix} \Bigg| \tau_k^*, \gamma_k^* \sim \texttt{Multinomial}\left(t_m, \begin{bmatrix} \rho_1(1-\tau_1^*) & \cdots & \rho_K(1-\tau_K^*) \\ \rho_1 \tau_1^* X \gamma_1^* & \cdots & \rho_K \tau_K^* X \gamma_k^* \end{bmatrix} \right)$$

When allowing IsoDeconv to consider genes mapping outside of the gene of interest, initial simulations demonstrated that these terms dominated estimation. This occurs since over 99% of all reads map outside the gene of interest and thus drown out the information within the gene due to sheer abundance. Restricting to reads within the gene of interest only, estimation behavior was seen to improve (not shown). Thus, using lemma 1.1 to combine all contributions of cell types outside the gene and then lemma 2.2 to condition on this quantity, we have:

$$\begin{bmatrix} Z_{11*} & Z_{K1*} \\ \vdots & \vdots \\ Z_{1E*} & Z_{KE*} \end{bmatrix} \Bigg| \tau_k^*, \gamma_k^* \sim \texttt{Multinomial}\left(Z_T, \left[\frac{\rho_1 \tau_1^* X \gamma_1^*}{\sum_{k=1}^K \rho_k \tau_k^*} \quad \cdots \quad \frac{\rho_K \tau_K^* X \gamma_k^*}{\sum_{k=1}^K \rho_k \tau_k^*} \right] \right) \tag{A.1}$$

However, due to the properties of bulk expression datasets, we do not observe the number of reads mapping to each cell type. Thus, we only observe the sums from all cell types at each exon set.

$$Z = \begin{bmatrix} Z_1 \\ \vdots \\ Z_E \end{bmatrix} \Bigg| \tau_k^*, \gamma_k^* \sim \texttt{Multinomial}\left(Z_T, \left[\frac{\sum_{k=1}^K \rho_k \tau_k^* X \gamma_1^*}{\sum_{k=1}^K \rho_k \tau_k^*} \right] \right)$$

The update of such a likelihood is a computationally difficult problem - we have I+2 parameters being measured for each cell type and all must be optimized simultaneously. To improve the tractability of such a numerical optimization technique, we utilize the EM algorithm.

For this problem, the missing data that we will assume is the expression from each individual cell type. Thus, we revert to the likelihood given above in equation (A.1). The complete data log-likelihood is given by:

$$\begin{aligned} \ell = \sum_{k=1}^K \Bigg[\sum_{e=1}^E \Big\{ Z_{ke*} \Big(\log(\rho_1 \tau_1^*) - \log\Big(\sum \rho_r \tau_r^* \Big) + \log\big(X_e^T \gamma_k^* \big) \Big) \Big\} + \\ \ln\Gamma\left(\alpha_{k\cdot}\right) - \ln\Gamma\left(\alpha_{k1}\right) - \ln\Gamma\left(\alpha_{k2}\right) + (\alpha_{k1} - 1)\log(\tau_k^*) + (\alpha_{k2} - 1)\log(1 - \tau_k^*) \\ \ln\Gamma\left(\beta_{k\cdot}\right) - \ln\Gamma\left(\beta_{k1}\right) - \cdots - \ln\Gamma\left(\beta_{kI}\right) + \sum_{i=1}^I (\beta_{ki} - 1)\log\Big(\tilde{l}_i \gamma_{ki}^* \Big) \Bigg] \end{aligned}$$

The EM algorithm utilized to solve this problem is composed of three separate steps.

1 E-Step: Update Posterior Means of Z_{ke*}

2 M-Step (1): Update $(\rho_1, ..., \rho_k, \tau_k^*)$

2 M-Step (2): Update γ_k^*

These steps are outlined below.

E-Step: Update Posterior Means of Z_{ke*}:

Recall that the observed expression values, the Z_e, represent the sum of all counts from each cell type. Thus, $Z_e = \sum_{k=1}^{K} Z_{ke*}$. By grouping elements of the multinomial according to exon set, a simple application of lemma 3 provides:

$$(Z_{11*}, ..., Z_{K1*}, ..., Z_{1E*}, ...Z_{KE*}) \Big| Z_1, ..., Z_e, \tau^*, \gamma^* \sim \prod_{e=1}^{E} \texttt{Multinomial}\left(Z_e, p_e'\right)$$

where

$$p_e' = \left(\frac{\rho_1 \tau_1^* X \gamma_1^*}{\sum_{k=1}^{K} \rho_k \tau_k^* X \gamma_k^*}, \cdots, \frac{\rho_K \tau_K^* X \gamma_K^*}{\sum_{k=1}^{K} \rho_k \tau_k^* X \gamma_k^*} \right)$$

Thus, it becomes clear by property of the multinomial distribution that:

$$E\left[Z_{je*} \Big| Z_1, \cdots, Z_E, \tau^*, \gamma^* \right] = Z_e \left(\frac{\rho_j \tau_j^* X \gamma_j^*}{\sum_{k=1}^{K} \rho_k \tau_k^* X \gamma_k^*} \right)$$

M-Step (1): Update $(\rho_1, ..., \rho_K, \tau_K^*)$:

It is clear from the complete data log-likelihood specified above that the cell type proportions and gene expression parameters must be updated simultaneously. These terms are inextricably linked within the log function. We do note that this set of parameters is separable from the isoform parameters as the likelihood can be partitioned into a sum of two independent pieces, one containing the gene expression parameters and cell type proportions and the other containing the isoform parameters. Thus, we consider recasting the likelihood to include only the cell type proportions and gene expression parameters.

$$
\begin{aligned}
\ell\left(\rho,\tau^*\right) &= \sum_{k=1}^{K}\left[\sum_{e=1}^{E}\left\{Z_{ke*}\left(\log(\rho_k\tau_k^*) - \log\left(\sum\rho_r\tau_r^*\right)\right)\right\} + \right. \\
&\qquad \left.(\beta_{k1}-1)\log(\tau_k^*) + (\beta_{k2}-1)\log(1-\tau_k^*)\right] \\
&= \sum_{k=1}^{K}\left\{Z_{k\cdot*}\left(\log(\rho_k\tau_k^*) - \log\left(\sum\rho_r\tau_r^*\right)\right) + (\beta_{k1}-1)\log(\tau_k^*) + \right. \\
&\qquad \left.(\beta_{k2}-1)\log(1-\tau_k^*)\right\} \\
&= \left\{\sum_{k=1}^{K} Z_{k\cdot*}\log(\rho_k\tau_k^*) + (\beta_{k1}-1)\log(\tau_k^*) + (\beta_{k2}-1)\log(1-\tau_k^*)\right\} - \\
&\qquad Z_T\log\left(\sum\rho_r\tau_r^*\right) \\
&= \left[\sum_{k=1}^{K} Z_{k\cdot*}\log\{\rho_k\exp(-\tau_k')\} + (\beta_{k1}-1)\log\{\exp(-\tau_k')\} + \right. \\
&\qquad \left.(\beta_{k2}-1)\log\{1-\exp(-\tau_k')\}\right] - Z_T\log\left\{\sum\rho_r\exp(-\tau_r')\right\}
\end{aligned}
$$

Taking the expectation of this likelihood will result in the use of quantities found in (1) to replace the Z_{ke*} pieces. In the following, we leave the the $Z_{k\cdot}$ notation for simplicity of notation, but please note that these values have been replaced by their expectations.

Taking the derivative of $\ell(\rho,\tau^*)$ with respect to the reparametrized τ^*, we have:

$$
\dot{\ell}_{\tau_r'}(\rho,\tau^*) = -Z_{r\cdot} - (\beta_{k1}-1) + \frac{(\beta_{k2}-1)\exp\{-\tau_r'\}}{1-\exp\{-\tau_r'\}} + Z_T\left(\frac{\rho_r\exp\{-\tau_r'\}}{\sum\rho_k\exp\{-\tau_k'\}}\right)
$$

To consider the derivatives of the proportions, we consider the natural linearity constraints to rewrite the likelihood as follows and subsequently take the derivative:

$$\begin{aligned}
\ell(\rho, \tau^*) \approx & \left[\sum_{k=1}^{K-1} Z_{k\cdot} \log\left(\rho_k \tau_k^*\right)\right] + Z_{K\cdot} \log\left\{(1-\rho_1-...-\rho_{K-1})\tau_K^*\right\} \\
& - Z_T \log\left(\sum_{s=1}^{K-1} \rho_s(\tau_s^* - \tau_K^*) + \tau_K^*\right) \\
\dot{\ell}_{\rho_r}(\rho, \tau^*) = & \left(\frac{Z_{r\cdot}}{\rho_r}\right) - Z_{K\cdot}\left(\frac{1}{1-\rho_1-...-\rho_{K-1}}\right) - Z_T\left[\frac{\tau_r^* - \tau_K^*}{\sum_{s=1}^{K} \rho_s \tau_s^*}\right]
\end{aligned}$$

The update of the procedures proceeds using a joint, constrained optimization approach using R's `constrOptim`.

M-Step (2): Update γ_k^*:

As noted above, we may update the γ_k^* separately from one another and from the proportion and gene expression parameters. The piece of the likelihood governing the update of isoform expression parameters for cells of type k is given by:

$$\begin{aligned}
\ell\left(\gamma_k^*\right) &= \left(\sum_{e=1}^{E} Z_{ke} \log\left(X_e^T \gamma_k^*\right)\right) + \ln\Gamma\left(\alpha_{k\cdot}\right) - \sum_{i=1}^{I} \ln\Gamma\left(\alpha_{ki}\right) + \sum_{i=1}^{I}\left(\alpha_{ki}-1\right)\log\left(\tilde{l}_i \gamma_{ki}\right) \\
&= \left(\sum_{e=1}^{E} Z_{ke} \log\left(X_e^T \gamma_k^*\right)\right) + \ln\Gamma\left(\alpha_{k\cdot}\right) - \sum_{i=1}^{I} \ln\Gamma\left(\alpha_{ki}\right) + \sum_{i=1}^{I-1}\left(\alpha_{ki}-1\right)\log\left(\tilde{l}_i \gamma_{ki}\right) + \\
&\qquad \left(\alpha_{kI}-1\right)\log\left(1 - \tilde{l}_1\gamma_{k1}^* - \cdots - \tilde{l}_{I-1}\gamma_{k,I-1}^*\right)
\end{aligned}$$

For simplicity of notation in the following, we suppress the notation regarding expectations of the missing parameters. Note, however, that these values are replaced by their expectations derived in the E-step.

Recall the special definitions of X^*, X_e^* and $X_{e,(I)}^*$ from their use in the pure sample expression materials. In addition, we define reparameterized isoform expression parameters for the mixture given by $\gamma_{ki}^{*} = \exp\{-\gamma_{ki}^{*r}\}$ to simplify constraints. Finally, we define $\tilde{l}_{(I)}$ as the $\tilde{l}$ vector with the I-th entry removed. Taking the derivative, we have:

$$\dot{\ell}_{\gamma_k^{r*}}(\gamma_k^*) = \sum_{e=1}^{E} -Z_{ke}\left(\frac{X_{e,(I)}^{*} \circ \exp\{-\gamma_k^{r*}\}}{X_e^{*T}\gamma_k^*}\right) - (\beta_k - 1) + \\ \left(\frac{\beta_{kI} - 1}{1 - \tilde{l}_1 \exp\{-\gamma_{k1}^{*r}\} - \cdots - \tilde{l}_{I-1}\exp\{-\gamma_{k,I-1}^{*r}\}}\right)\left(\tilde{l}_{(I)} \circ \exp\{-\gamma_k^{r*}\}\right)$$

Thus, given the restrictions outlined for the pure sample case, we utilize this derivative in R's `constrOptim` to update the isoform expression parameters.

A.2.6 Explaining modeling decisions

Several facets of the model deserve illumination. First, consider the use of the Dirichlet-Multinomial model instead of a negative binomial model. In order to incorporate the isoform expression parameters as conditional probabilities, the model within a gene must condition on the number of reads mapping to that gene in the purified reference samples. Supposing this conditioning is performed and that an independent negative binomial distribution is assumed at each exon set, the likelihood becomes inconsistent. This arises because the independent negative binomials could theoretically exceed the read count upon which the model is conditioned. The multinomial model maintains its consistency despite the conditioning argument.

Secondly, the use of the described staged estimation approach became necessary after an initial version of the model, which attempted to estimate α_k and γ_k values simultaneously, proved intractable. This approach led to unstable estimates of the γ_k and α_k parameters wherein the α_k parameters became unbounded. This would suggest little to no variability in the isoform expressions, an impossibility in the simulated data upon which the model was tested.

Finally, the incorporation of the $\log(\tau_k^*)$ and $log(\gamma_k^*)$ transformations was performed after initial testing with untransformed parameters proved inaccurate. "Hill-climbing" estimation methods such as Newton Raphson and BFGS require that the likelihood is sufficiently stable across the parameter space so that the crest of the "hill" is not continually overstepped. The proposed optimization approach is more stable with respect to the log parameters since the log scale spreads out the small parameter values. Under these reparametrizations, model accuracy and the mobility of proportion estimates improved.

A.2.7 Staged estimation vs joint estimation

We chose to utilize a staged estimation approach instead of a joint estimation approach for several reasons. Our method, like many other reference-based deconvolution methods, rely on a reference. Our staged estimation procedure separates the estimation for cell-type specific reference and bulk mixture samples. (Note: This approach to make separate estimation for reference and mixture is adopted by all other reference-based deconvolution methods). Researchers who chose to use the IsoDeconvMM method will have their own mixture data, and they may have their own reference (or, more likely, want to use an existing reference). In a joint estimation procedure, the reference estimates will change given the mixture, and thus create unwanted instability in our method. In other words, two users of the method may think they are using the same reference, but indeed the reference estimates were different if they used a joint model.

Furthermore, in most cases, the pure cell type reference data will come from different platforms and different pipelines than the bulk mixture sample data. As a result of this, the variance estimate from the reference data may not be useful or applicable for the bulk mixture data. In addition, there is a variety of platforms and pipelines available for obtaining the reference data, and new technology and pipelines are constantly being developed. These realities make it hard to derive an integrated framework for which we could create a joint estimation model.

A joint estimation procedure has a great advantage when we want to perform hypothesis testing, since the variances of parameter estimates can be more accurate. However, like many other deconvolution methods, our objective is not to perform hypothesis testing, but to simply obtain a point estimate. Therefore, the benefit of a joint estimation procedure is limited

Because we are using a staged estimation approach, the final estimate is the MLE of the third stage of the method, but it is not necessarily the MLE of the whole model.

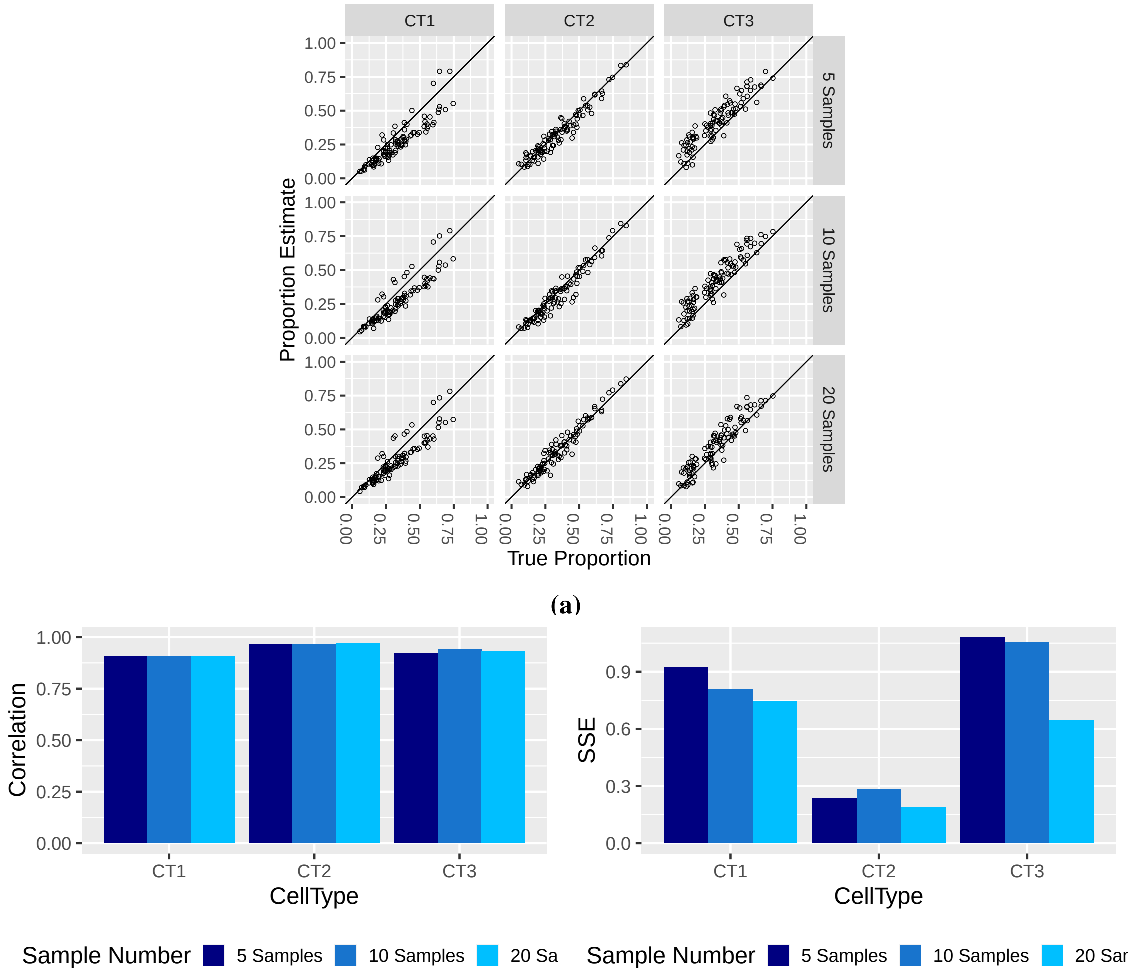
A.3 Chapter 2 supplementary simulation and *In Silico* materials

A.3.1 *In Silico* Blueprint analysis supplement

A.3.1.1 Comparison of number pure reference samples used

We explored what number of pure samples per cell type might be optimal to use. Figure A.1 compares mixture proportion results using 5, 10, and 20 reference samples per cell type. In order to eliminate the possibility that an inadequate selection of initial points contributed to any variation, the initial points included the true proportion for this comparison. Increasing the number of reference samples per cell type provided minimal overall improvements in the correlations and sum-of-squared error (SSE) results as the number of pure reference samples per cell type increased from 5 to 10. We suspect that we didn't observe an improvement in the method when we added additional pure reference samples because although adding additional pure samples per cell type may reduce the variance of the estimation for the pure cell type parameter estimates, it may not improve the actual point estimate of the pure cell type parameters. In the IsoDeconvMM procedure, we only use the point estimates of these pure cell type parameters in later stages. For this reason, adding a larger number of pure samples per cell type may not improve the final cell type proportion estimate.

We also noticed that as the number of pure samples per cell type increased, additional transcript clusters acquired unstable pure sample isoform parameter estimates. Based on these results, we concluded that adding more reference samples per cell type was not worth the additional computational cost and memory burden.



(a)

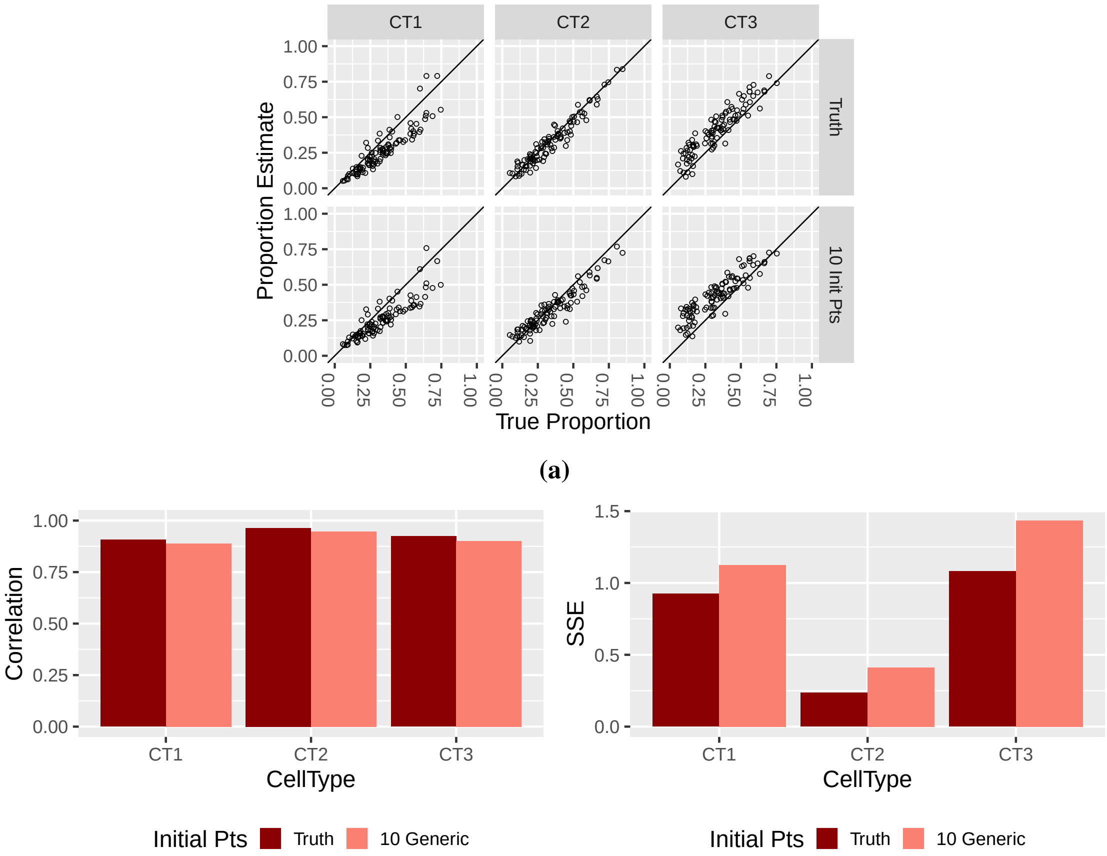
(b)

(c)

Figure A.1: Comparison of proportion estimates vs true proportions when 5, 10, and 20 pure samples per cell type are used in the IsoDeconvMM algorithm fit of the simulated Blueprint mixture samples. Each mixture proportion estimate uses 100 DU transcript clusters. (a) Proportion estimates plotted against the true proportion. Plots separated by cell types (columns) and number of pure samples per cell type (rows). (b) Correlation results compared across number of pure samples. (c) Sum of square error (SSE) results compared across number of pure samples.

A.3.1.2 Comparison of number initial points used

Since IsoDeconvMM algorithm requires multiple initial points in order to optimize the accuracy of the results, we explore how many initial points are sufficient to use. Figure A.2 compares the proportion estimate results when the IsoDeconvMM algorithm uses the truth as the initial point and 10 generic initial points. These 10 initial points are included in Table 1 in the Appendix. There is little difference in the accuracy and precision of the proportion estimate results between using the truth as the initial point and using the 10 generic initial points. This suggests that users of the method could consider these initial points as sufficient for exploring a variety of starting points when there are three cell types in the mixture samples. While it is possible that increasing the number of initial points could improve the results even more, including more initial points is computationally burdensome. Realistically, users of the IsoDeconvMM method would restrict the number of initial points.



(a)

(b)

(c)

Figure A.2: Comparison of proportion estimates vs true proportions when the algorithm fit utilizes the truth as the initial point and 10 generic initial points. Shows results for the 100 mixture samples created in the *in silico* Blueprint analysis, where each mixture proportion estimate uses 100 DU transcript clusters. (a) Proportion estimates plotted against the true proportion. Plots separated by cell types (columns) and different initial point options (rows). (b) Correlation results compared across start points. (c) Sum of square error (SSE) results compared across start points.

Based on the subsections A.3.1.1 and A.3.1.2, we conclude that the ‘best’ IsoDeconvMM settings, balanced for computational efficiency and accuracy of the proportion estimates, include using five pure reference samples per cell type and 10 initial points. We note that as the number of cell types increases, the ideal number of initial points may need to increase.

A.3.1.3 Paired-end vs single-end reads discrepancies

The Blueprint data set (Chen *et al.* (2016)) provided bulk RNA-seq data on purified cell samples for three cell types: CD4-positive, alpha-beta T cells; CD14-positive, CD16-negative classical monocyte; and mature neutrophil. Of the 197 available CD4-positive, alpha-beta T cell RNA-seq samples, 194 samples used paired-end reads. However, only 3 of the 173 CD14-positive, CD16-negative classical monocyte samples and only 9 of the 191 mature neutrophil samples used paired-end reads; the other samples used single-end reads. In order to reduce potential variability within cell types due to this discrepancy in data collection, we only used the paired-end read samples from the CD4-positive, alpha-beta T cell samples, and we only used the single-end read samples from the other two cell types.

A.3.1.4 Combined fragment length distribution file

Because we simulated the mixture files from the pure cell type files, the mixture files did not have an associated fragment length distribution file. Additionally, the IsoDeconvMM algorithm assumes paired-end read fragment distributions, which two of the three cell types used in the analysis did not have. Consequently, a common fragment length distribution file was created by combining the fragment length distribution files of 50 paired-end read samples from the CD4-positive, alpha-beta T cell samples, and this combined fragment length distribution file was used for all mixture and pure reference samples.

A.3.2 Simulation supplement

A.3.2.1 Gene selection from UCLA data

This subsection discusses the selection of the 5,172 genes used in the simulations discussed in Section 4. In the UCLA data set Parikshak *et al.* (2016) described within the paper, there were a total of 57,821 genes. For convenience purposes, these total genes were filtered such that there was a one-to-one correspondence between genes and transcript clusters (a transcript cluster represented only a single gene). This step filtered the genes to 38,851 genes.

Next, the genes were filtered by expression levels. If the third quartile of the counts for a gene were less than 30, a gene was ignored from further consideration. This step left 10,414 genes. In order to limit the number of genes, only the genes on chromosomes one through nine were considered for further analysis, resulting in 5,172 genes total.

Of the total 5,172 genes, we selected 1,000 genes with relatively high expression (the median of the gene expression across the 89 samples was above the 25^{th} percentile of the gene expression medians) and at least three isoforms as possible genes to be used for the mixture sample proportion estimate in the IsoDeconvMM analysis. Additional filtering of these 1,000 genes—excluding genes with over 15 isoforms and selecting genes with relatively high expression from these 1,000 genes—was performed, and 100 genes were randomly selected for differential isoform usage (DU).

A.3.3 Mixture file creation

For both the *in silico* Blueprint data analyses presented in Section 3 and the simulations presented in Section 4, the creation of the mixture files proceeded as follows.

Among the individuals that had pure reference samples from each of the three cell types, 100 individuals were randomly selected. For each individual, all three of their pure cell type samples were extracted, and a mixture sample was created using these pure cell type samples.

A total read count was randomly selected from a normal distribution. The normal distributions were informed by the UCLA and Blueprint data sets. The normal distribution for the total read counts of the Blueprint mixture samples created for Section 3 had a mean of 12 million and a standard deviation of 2.5 million. This approximated the distribution of total read counts for the 47,749 transcript clusters in the Blueprint data. The normal distribution for the total read counts of the mixture files created for Section 4 had a mean of 7 million and a standard deviation of 1 million, which approximated the distribution of total read counts for the 5,172 genes of interest in the 89 UCLA samples.

The pure reference samples for both the simulations and the *in silico* analyses were split into two groups. One group of pure samples was reserved for the IsoDeconvMM algorithm fit, and the other group was reserved for the creation of mixture files. From the group reserved for the creation of mixture files, one pure sample from each cell type was randomly selected. Ratios of the randomly selected total read count for the mixture file and total read counts for the pure sample files were calculated, and the counts in each pure sample were adjusted by its read count ratio.

For each mixture file, a probability was randomly selected from the Dirichlet(2,2,2) distribution (extreme probabilities where one or more cell type proportions was less than 0.05 were excluded). The counts in each pure sample were also multiplied by the appropriate cell type proportion. The exon set counts from each pure cell type reference sample, adjusted by the read count ratio and the cell type proportion, were rounded to the nearest integer and added together to calculate the counts of the exon sets within each mixture file.

A.3.4 Investigation into V-shape of Blueprint cell type 1 scatter plots

We performed some additional simulations to test a theory as to why IsoDeconvMM is biased in CT1 when the number of clusters is large. The bias seen in the in silico Blueprint analyses could be explained by a combination of unlucky randomness in the simulation and potential systematic differences between the pure reference samples used in the algorithm fit and the pure reference samples used to create mixture samples. As discussed in Section A.3.3, the mixture files for the in silico analysis were created by mixing exon set counts from pure reference samples of the same individual. This required that the pure reference samples used for mixture creation came from individuals with complete pure samples from each of the three cell types. On the other hand, the pure reference samples used for the pure sample estimation in IsoDeconvMM were selected from a combination of (a) individuals with a complete set of pure reference samples (samples from each cell type) and (b) individuals who did not have a complete set of pure reference samples. Looking back, it is possible that individuals who did not have a complete set of pure reference samples are somehow systematically different from those who had complete sets. In order to determine if a combination of unlucky randomness and the systematic differences discussed were to blame for the CT1 bias, we ran some simulations.

Simulation set-up: There were 113 individuals with a complete set of pure reference samples. For each simulation replicate (of which there were 10 replicates), we randomly selected 100 individuals to create mixture samples from their pure reference samples, and we randomly selected 5 individuals from the remaining 13 to use their pure reference samples in the pure sample fit part of the IsoDeconvMM algorithm. Each simulation replicate used a different set of 100 proportion estimates to create the mixture files. The creation of mixture files and the running of the IsoDeconvMM algorithm for each simulation replicate then proceeded in the same way as described in the paper. We performed 10 simulation replicates.

Results: Figure A.3 shows that of the 10 simulation replicates, 2 simulation replicates continued to have the V shape in CT1 that we saw in the paper results (simulation replicates 1 and 9). However, the other 8 simulation replicates did not have such a V shape. When we specifically look at the correlations and SSE of cell type 1 (Figure A.4), we see that the correlations and SSE of simulations 1 and 9 are comparable to what was reported in the in silico analyses given in the paper, but the other 8 simulations gave markedly better correlation and SSE results for cell type 1. Based on these results, we conclude that the unusual V shape seen in the paper results were due to unlucky randomness within the simulation. We also conclude that the V shape was not due to systematic biases because we still saw the V shape even when restricting the pure reference samples to come from individuals with complete sets of pure reference samples. We added a comment about these results in the Discussion.

A.3.5 CIBERSORTx vs IsoDeconvMM in simulated Dirichlet-Multinomial data

We ran an additional simulation to show that the same pattern we observed in the in silico Blueprint analyses held in simulated data. We simulated the data using the same Dirichlet-Multinomial simulation set-up described in Section 4, but we added an additional step to simulate 100 genes that were differentially expressed across cell types.

We simulated genes with differential expression using the following procedure: We selected 100 genes for differential expression in the same way that we selected 100 genes for differential isoform usage (see Section A.1.1). Of these 100 genes designated for differential expression, 33 were designated to be up-regulated in CT1 compared to CT2 and CT3, 33 were designated to be up-regulated in CT2 compared to CT1 and CT3, and the remaining 34 were designated to be up-regulated in CT3 compared to CT1 and CT2. Suppose we consider gene g that is supposed to be up-regulated in CT1 compared to the other two cell types. After the original gene counts were simulated assuming no differential expression, the counts of gene g were multiplied by a constant ranging between 1.24 and 1.55 in CT1, and the counts of gene g were multiplied by the constant 0.775 in the other two cell types. Some small variation introduced to replicate samples of each cell type, where the fold change f was adjusted as follows: $f * 2^c$, $c \sim N(0, \sigma = 0.025)$. This resulted in an overall fold change ranging between 1.6 and 2.0 between the cell types. This procedure was applied similarly to all genes that were designated to be up-regulated in CT2 and CT3. When the proportion of gene counts in these 100 genes was compared before and after any fold change adjustments, the ratio of these proportions (before/after) ranged from 0.96 to 1.05.

Twenty pure cell type samples were simulated for each cell type. Of these 20 samples per cell type, 10 were used in the mixture sample creation, and 5 were selected to be used in the pure sample fit part of the IsoDeconvMM algorithm. The mixture samples were created as described in Section A.3.3. CIBERSORTx estimated the cell type proportions using all 100 genes as well as a random subset of the CT1, CT2, and CT3 genes such that the total number of genes were 10 with approximately equal numbers of up-regulated CT1, CT2, and CT3 genes in each subset.

IsoDeconvMM estimated the cell type proportions using all 100 genes as well as a random subset of the CT1, CT2, and CT3 genes such that the total number of genes were 10.

The results to this simulation are in Figure A.5.

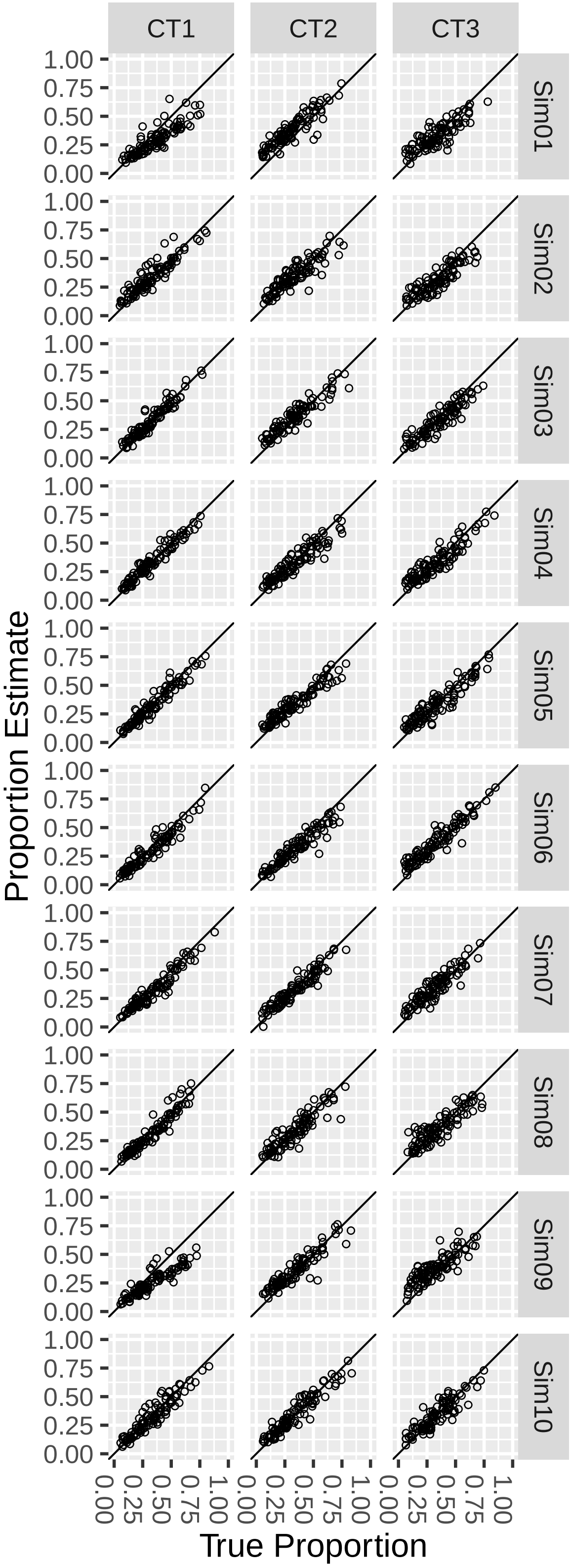
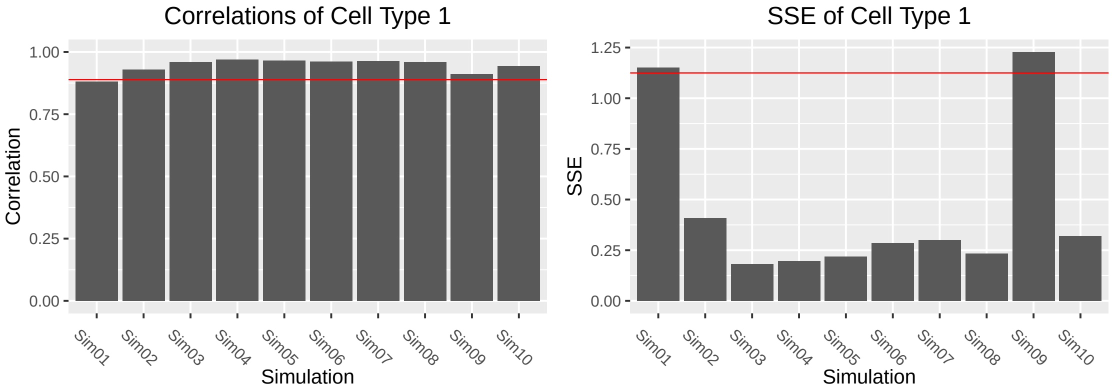
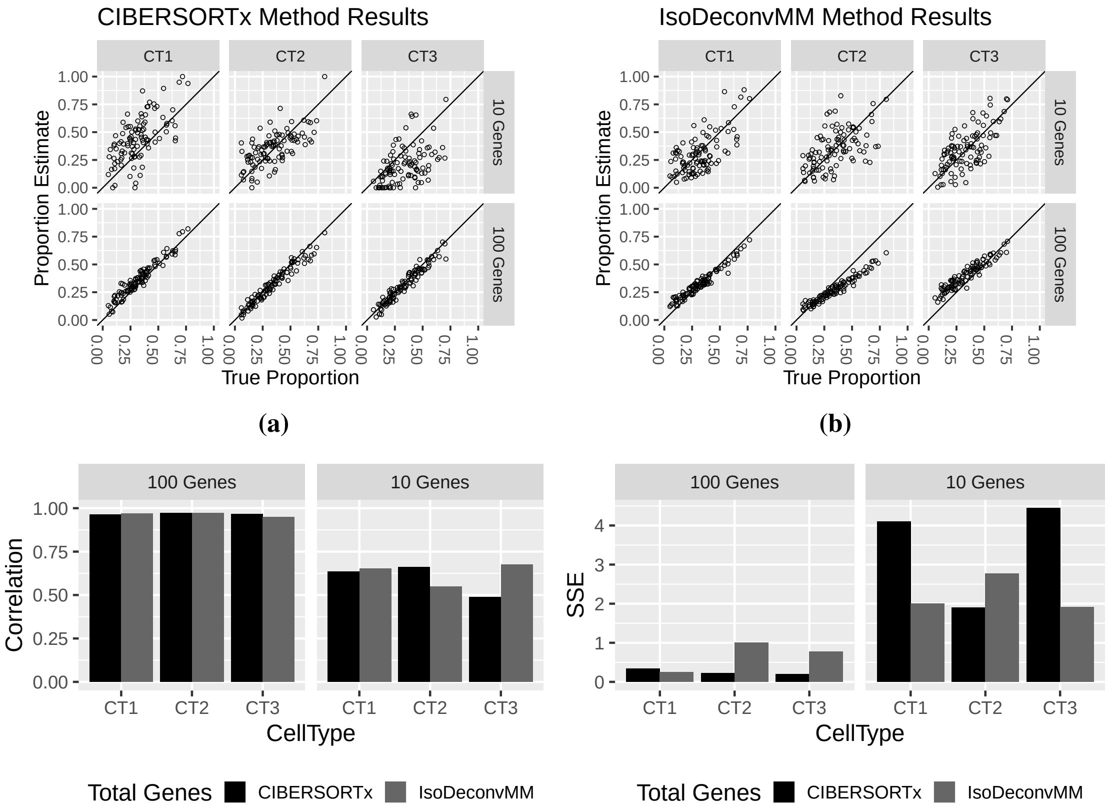


Figure A.3: Scatter plots summarizing results for the additional replicates of the *in silico* Blueprint analyses. Scatter plots show the mixture proportion estimates vs the true cell type proportion for each estimated cell type (column). EAch row represents a simulation replicate.



(a) **(b)**

Figure A.4: Correlations (left) and sum of square error (SSE, right) of cell type 1 for each of the additional replictates of the *in silico* Blueprint analyses. Red line indicates the cell type 1 correlation and cell type 1 SSE reported in the in silico analyses used in the paper.



(a)

(b)

(c)

(d)

Figure A.5: Dirichlet-Multinomial simulation mixture proportion estimate results calculated using the CIBERSORTx and IsoDeconvMM methods on DE and DU genes, respectively. Results separated by cell types (columns) and number of genes used in the analysis (rows). (a) Proportion estimates vs true proportions for CIBERSORTx method (used DE clusters only). (b) Proportion estimates vs true proportions for IsoDeconvMM method (used DU clusters only). (c) Correlation and (d) sum-of-square (SSE) results compared across methods.

APPENDIX B: APPENDIX FOR CHAPTER 3

B.1 Initialization of the glmmPen algorithm

The fixed effects $\boldsymbol{\beta}^{(0)}$ and random effects covariance terms $\boldsymbol{\gamma}^{(0)}$ are initialized at iteration $s = 0$ in one of two ways. We discuss first the initialization procedure used when the package **glmmPen** is used to fit a single model or the first model in the sequence of models fit for variable selection. In this scenario, the fixed effects $\boldsymbol{\beta}^{(0)}$ are initialized by fitting a 'naive' model using the coordinate descent techniques of Breheny and Huang (2011) assuming no random effects and the random effects covariance matrix is initialized as a diagonal matrix with positive variance.

By default, the starting variance is initialized in an automated fashion. The data is fit to a model composed of only a fixed and random intercept using a Laplace approximation. The random intercept variance from this model is then multiplied by 2, and this value is set as the starting variance. We use this approach so that the starting variance of the random effects is sufficiently large. Having a sufficiently large starting variance helps improve the stability of the algorithm.

The E-step MCMC chain of the sample of the posterior density $\phi(\boldsymbol{\alpha}_k|\boldsymbol{d}_{k,o};\boldsymbol{\theta}^{(s)})$ for groups $k = \{1, ..., K\}$ is initialized in iteration $s = 1$ with random draws from the standard normal distribution. For all following iterations $s > 1$, the MCMC chain is initialized with the last draw from the previous iteration $s - 1$.

When the algorithm performs variable selection, we initialize models with previous model results. For all subsequent models after the first model fit in the variable selection procedure, the fixed effects, random effects covariance matrix, and random effects MCMC chain are initialized using results from a previous model fit. More details about initialization for variable selection is discussed in Section 3.4.

APPENDIX C: APPENDIX FOR CHAPTER 4

C.1 Chapter 4 algorithms and B matrices

C.1.1 B matrices used in simulations

The transpose of the first 11 rows of the deterministic ‘large’ $\boldsymbol{B}$ matrices used in the Binomial simulations in Section 4.3 are given in equations (C.1) and (C.2), corresponding to $r = \{3, 5\}$, respectively. The deterministic ‘moderate’ $\boldsymbol{B}$ matrices are these large $\boldsymbol{B}$ matrices multiplied by the constants 0.75 and 0.80 for r equal to 3 and 5, respectively. All other $p - 10$ rows of the $\boldsymbol{B}$ matrices were set to 0, where p is the total number of predictors used in the simulations.

$$\boldsymbol{B}^T_{large,r=3} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 \\ -2 & 2 & -1 & 0 & 1 & -1 & 0 & 1 & -1 & 0 & 1 \end{bmatrix} \tag{C.1}$$

$$\boldsymbol{B}^T_{large,r=5} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 \\ -2 & 2 & -1 & 0 & 1 & -1 & 0 & 1 & -1 & 0 & 1 \\ -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 \\ -1 & -1 & 0 & 1 & 1 & -1 & -1 & 0 & 1 & 1 & -2 \end{bmatrix} \tag{C.2}$$

The transpose of the first 6 rows of the deterministic ‘moderate’ $\boldsymbol{B}$ matrix used in the Poisson simulations in Section 4.3.4 are given in equation (C.3). All other $p - 5$ rows of the $\boldsymbol{B}$ matrices were set to 0, where $p = 100$ in the Poisson simulations.

$$\boldsymbol{B}^T_{poisson,r=3} = 0.75 \times \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & 1 & 1 & 1 \\ -1 & 0 & 1 & -1 & 0 & 1 \end{bmatrix} \tag{C.3}$$

C.1.2 Algorithms

Algorithm 5 M-step of the MCECM algorithm

1. The parameters $\boldsymbol{\theta}^{(s,0)}$ for M-step iteration $h = 0$ are initialized using the results from the previous M-step, $\boldsymbol{\theta}^{(s-1)}$.
2. Conditional on $\boldsymbol{b}^{(s,h-1)}$ and $\tau^{(s-1)}$, each $\beta_j^{(s,h)}$ for $j = 1, ..., p$ is given a single update using the Majorization-Minimization algorithm specified by Breheny and Huang (2015).
3. For each group k in $k = 1, ..., K$, the augmented matrix $\tilde{\boldsymbol{z}}_{ki} = (\tilde{\boldsymbol{\alpha}}_k^{(s)} \otimes \boldsymbol{z}_{ki})J$ is created for $i = 1, ..., n_k$ where $\tilde{\boldsymbol{\alpha}}_k^{(s)} = ((\boldsymbol{\alpha}_k^{(s,1)})^T, ..., (\boldsymbol{\alpha}_k^{(s,M)})^T)^T$.
4. Conditional on the $\tau^{(s-1)}$ and the recently updated $\boldsymbol{\beta}^{(s,h+1)}$, each $\boldsymbol{b}_t^{(s,h)}$ for $t = 1, ..., q$ is updated using the Majorization-Minimzation coordinate descent grouped variable selection algorithm specified by Breheny and Huang (2015).
5. Steps 2 through 4 are repeated until the M-step convergence criteria are reached or until the M-step reaches its maximum number of iterations.
6. Conditioning on the newly updated $\boldsymbol{\beta}^{(s)}$ and $\boldsymbol{b}^{(s)}$, $\tau^{(s)}$ is updated (generically, using the Newton-Raphson algorithm).

Algorithm 6 Full MCECM algorithm for single (λ_0, λ_1) penalty combination

1. Fixed and random effects $\boldsymbol{\beta}^{(0)}$ and $\boldsymbol{b}^{(0)}$ are initialized as discussed Appendix B.
2. In each E-step for EM iteration s, a burn-in sample from the posterior distribution of the random effects is run and discarded. A sample of size $M^{(s)}$ from the posterior is then drawn and retained for the M-step.
3. Parameter estimates of $\boldsymbol{\beta}^{(s)}$, $\boldsymbol{b}^{(s)}$, and $\tau^{(s)}$ are then updated as described in the M-step procedure given above.
4. Steps 2 and 3 are repeated until the convergence condition is met a pre-specified consecutive number of times or until the maximum number of EM iterations is reached.

C.2 Chapter 4 supplementary simulation and case study materials

C.2.1 Model selection

This section provides details on how the **glmmPen_FA** algorithm selects the optimal tuning parameter combination. In all simulations and analyses discussed in this paper, we set the random effects equal to the fixed effects (i.e. $p = q$) and let the algorithm select the fixed and random effects.

The algorithm runs a computationally efficient two-stage approach to pick the optimal set of tuning parameters. In the first stage of this approach, the algorithm fits a sequence of models where the fixed effect penalty is kept constant at the minimum value of the fixed effects penalty sequence, labeled here as $\lambda_{0,min}$, and the random effects penalty proceeds from the minimum random effect penalty, labeled $\lambda_{1,min}$, to the maximum value $\lambda_{1,max}$. The best model from this first stage is then identified using the BIC-ICQ criterion (Ibrahim *et al.*, 2011). This first stage identifies the optimal random effect penalty value, $\lambda_{1,opt}$. In the second stage, the algorithm fits a sequence of models where the random effects penalty is kept fixed at $\lambda_{1,opt}$ and the fixed effects penalty proceeds from its minimum value $\lambda_{0,min}$ to its maximum value $\lambda_{0,max}$. The overall best model is chosen from the models in the second stage.

We have found this two-stage model selection approach to work very well in practice (see Section 3 for performance results).

C.2.2 Tuning parameter selection

The default maximum penalty, labeled here as λ_{max}, was calculated as the penalty that would penalize all of the fixed effects to 0 when no random effects are in the model. We used code from the **ncvreg** R package (Breheny and Huang, 2011) to calculate this value.

For all Binomial outcome variable selection simulations and case study analyses where the total number of predictors was 100 or less, we used the following sequence of penalties for both the fixed effects and the rows of the $\boldsymbol{B}$ matrix: a sequence of 10 penalties from $0.05\lambda_{max}$ to λ_{max}, with penalty values equidistant from each other on the log scale.

For all Binomial outcome variable selection simulations and case study analyses where the total number of predictors was 500, we used the following sequence of penalties: a sequence of 10 penalties from $0.15\lambda_{max}$ to λ_{max} for the fixed effects, and a sequence of 10 penalties from $0.10\lambda_{max}$ to λ_{max} for the rows of the $\boldsymbol{B}$ matrix, with penalty values equidistant from each other on the log scale. In simulations not shown here, using a consistent sequence of 10 penalties from $0.10\lambda_{max}$ to λ_{max} for both sets of parameters resulted in very similar final results, but accomplished in less time; using a consistent sequence of 10 penalties from $0.15\lambda_{max}$ to λ_{max} for both sets of parameters decreased the random effect true positive results.

For the Poisson outcome variable selection simulations, a penalty sequence with larger values was needed for both the fixed effects and rows of the $\boldsymbol{B}$ matrix due to the nature of how the data was simulated and fit. In these simulations, the covariate values $x_{ki,j}$ were simulated from a $N(0, \sigma = 0.10)$ distribution for $j = 1, ..., p$ and left unstandardized in the algorithm, whereas in the binomial simulations, the covariate values were simulated from the standard normal distribution $N(0, 1)$ and then standardized so that $\sum_{k=1}^{K} \sum_{i \in n_k} x_{ki,j} = 0$ and $\boldsymbol{x}_j^T \boldsymbol{x}_j / N = 1$ for each j. The fixed effects penalty sequence included $0.30\lambda_{max}$ and $(\delta_{0,1}, ..., \delta_{0,12}) * \lambda_{max}$, where $\delta_{0,i} = 2 + (i - 1)$. The random effect penalty sequence applied to rows of the $\boldsymbol{B}$ matrix included $0.30\lambda_{max}$ and $(\delta_{1,1}, ..., \delta_{1,11}) * \lambda_{max}$, where $\delta_{1,i} = 0.5 + (i - 1)$.

C.2.3 Initialization and convergence - glmmPen_FA

The fixed effects $\boldsymbol{\beta}^{(0)}$ and random effects covariance terms $\boldsymbol{\gamma}^{(0)}$ are initialized at iteration $s = 0$ in one of two ways. We discuss first the initialization procedure used when the package **glmmPen** is used to fit a single model or the first model in the sequence of models fit for variable selection. In this scenario, the fixed effects $\boldsymbol{\beta}^{(0)}$ are initialized by fitting a 'naive' model using the coordinate descent techniques of Breheny and Huang (2011) assuming no random effects.

Based on the initialized fixed effects $\boldsymbol{\beta}^{(0)}$, the predictors with non-zero initialized fixed effects are also initialized to have non-zero random effects (i.e. the corresponding rows of the $\boldsymbol{B}$ matrix are set to non-zero values), and predictors with zero-valued initialized fixed effects are initialized to have zero-valued random effects (i.e. the corresponding rows of the $\boldsymbol{B}$ matrix are set to zero). By default, the starting $\boldsymbol{B}$ matrix elements are initialized as $\sqrt{0.10/r}$, where r is the estimated number of latent factors. The corresponding initialized covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{B}\boldsymbol{B}^T$ will have all non-zero elements equal to 0.10.

The E-step MCMC chain of the sample of the posterior density $\phi(\boldsymbol{\alpha}_k | \boldsymbol{d}_{k,o}; \boldsymbol{\theta}^{(s)})$ for groups $k = \{1, ..., K\}$ is initialized in iteration $s = 1$ with random draws from the standard normal distribution. For all following iterations $s > 1$, the MCMC chain is initialized with the last draw from the previous iteration $s - 1$. At iteration $s = 0$, we sample $M = 100$ posterior samples from each group, and M increases to a max of 500 as the iteration number s increases.

When the algorithm performs variable selection, we initialize models with previous model results. After the first model is fit in the variable selection procedure, the fixed effects, random effects covariance matrix, and random effects MCMC chain are initialized using results from a previous model fit.

The EM algorithm is considered to have converged when the following condition is met at least 2 consecutive times (default) or until the maximum number of EM iterations (25) is reached:

$$||(\boldsymbol{\beta}^{(s)T}, \boldsymbol{b}^{(s)T})^T - (\boldsymbol{\beta}^{(s-t)T}, \boldsymbol{b}^{(s-t)T})^T||_2^2 / d_n^{s-t} < \epsilon_{EM} \tag{C.4}$$

where the superscript $(s - t)$ indicates t EM iterations back (default $t = 2$), $||.||_2^2$ represents the L_2 norm, and d_n^{s-t} equals the total number of non-zero $(\boldsymbol{\beta}^T, \boldsymbol{b}^T)^T$ coefficients in iteration $(s - t)$. In other words, the algorithm computes the average Euclidean distance between the current coefficient vector $(\boldsymbol{\beta}^T, \boldsymbol{b}^T)^T$ and the coefficient vector from t EM iterations back and compares it with $\epsilon_{EM} = 0.0015$.

The M-step algorithm is considered to have converged when the following condition is met or until the maximum number of iterations (50) is reached:

$$\text{max}_j|\beta_j^{(s,f+1)} - \beta_j^{(s,f)}| \cap \text{max}_{t,h}|b_{th}^{(s,f+1)} - b_{th}^{(s,f)}| < \epsilon_m, \tag{C.5}$$

where b_{th} is an individual element of $\boldsymbol{b}_t$, which is the t-th row of the $\boldsymbol{B}$. The value of ϵ_m was set to 0.001.

C.2.4 Initialization and convergence - glmmPen

The initialization and convergence of glmmPen in these simulations and analyses were very similar to the initialization and convergence of glmmPen_FA with the exception of the initialization of the random effect covariance matrix. This starting variance is initialized in an automated fashion. First, a GLMM composed of only a fixed and random intercept is fit using the **lme4** package. The random intercept variance from this model is then multiplied by 2, and this value is set as the starting values of the diagonal of the random effects covariance matrix. Similar to the **glmmPen_FA** random effect intialization, the predictors are initialized to have non-zero random effects if they are also initialized to have nonzero fixed effects; otherwise, predictors are initialized to have no random effects.

C.2.5 Elastic Net penalty

We extended our simulations on variable selection in Binomial data by adding correlations between the simulated covariates and adjusting for this correlation using the Elastic Net penalization approach. Elastic Net penalization balances the MCP, SCAD, or LASSO penalties with ridge regression. This balance between ridge regression and the other penalty is dictated by a value we label as π, where $\pi = 1.0$ represents the MCP penalty and $\pi = 0$ represents ridge regression.

In these simulations, we set the sample size to $N = 2500$ and number of groups to $K = 25$, with an equal number of subjects per group. There were $p = 100$ total predictors and 10 true predictors with non-zero fixed and random effects. We considered four types of correlations between the predictors. In three of the four correlation types, the correlation between all covariates was set to a common value of 0.2, 0.4, or 0.6, and the variance of the covariates was set to 1.0. In the fourth correlation type, we randomly selected 100 of the 110 covariates used in the case study (see Web Appendix Section C.2.6 for details) and calculated the Spearman correlation of these 100 covariates. In all four correlation cases, we simulated the covariates from a multivariate normal distribution with mean 0 and covariance matrix set to the correlation matrices described above.

We simulated the random effects covariance matrix using $r = 3$ and the corresponding moderate $\boldsymbol{B}$ matrix described in the main manuscript. The 10 true fixed effects $\boldsymbol{\beta}$ coefficients were set to 1. The generation of the binary responses from a logistic mixed effects model proceeded as described in Section 3.1.

We performed variable selection on these simulated data using Elastic Net α values of 0.1, 0.3, 0.5, 0.8, and 1.0, and we estimated the number of common factors r using the default Growth Ratio procedure described in Section 2.4.

A summary of the variable selection results—true positive percentages, false positive percentages, median time in hours to complete the procedure, and average absolute deviation for the fixed effects coefficients—is given in Chapter 4 Appendix Table C.1. A summary of the performance of the Growth Ratio estimation procedure is given in Chapter 4 Appendix Table C.2.

In general, increasing the correlation among the predictors decreases the average true positive percentage. Within a particular correlation set-up, decreasing the value of the Elastic Net π tends to increase both the true positives and the false positives.

The Growth Ratio procedure tends to underestimate the number of common factors r as the correlation between the covariates increases. However, when the correlation between the covariates is high at a value of 0.6 and there is no adjustment for ridge regression (i.e. $\pi = 1.0$, equivalent to the MCP penalty), there are more instances of the Growth Ratio procedure overestimating r.

For low values of π and/or high correlation, some simulation replicates had model fit issues. Specifically, in certain situations, the random effect variances diverged to excessively large values. As a result, the BIC-ICQ model selection criteria could not be calculated for the model, and the model selection procedure was suspended. When $\pi = 0.1$ and the correlation among the predictors was 0.4 or 0.6, this phenomena happened 25% or 26% of the time, respectively. When $\pi = 0.1$ and the correlation was 0.2, this happened 2% of the time; when $\pi = 0.3$ and the correlation was 0.4 or 0.6, this happened 1% or 3% of the time, respectively; when $\pi = 1.0$ and the correlation was 0.6, this happened 1% of the time. The simulations summarized in Chapter 4 Appendix Table C.1 do not include results from these problematic simulation replicates.

Corr	π	TP % Fixef	FP % Fixef	TP % Ranef	FP % Ranef	T^{med}	Abs. Dev. (Mean)
0.2	0.1	97.55	27.39	85.10	12.40	23.34	0.50
	0.3	91.70	13.32	61.90	4.46	22.06	0.54
	0.5	93.30	7.79	65.80	2.64	8.63	0.51
	0.8	94.10	1.08	90.00	0.63	2.58	0.40
	1.0	94.70	4.28	94.40	0.53	1.46	0.31
0.4	0.1	86.53	22.79	74.53	14.33	14.37	0.56
	0.3	87.98	11.93	59.49	1.99	15.10	0.58
	0.5	85.70	4.64	72.70	1.07	4.71	0.53
	0.8	80.30	1.61	83.30	0.37	2.54	0.44
	1.0	80.40	1.98	82.60	0.21	1.17	0.39
0.6	0.1	80.14	17.16	53.24	7.03	13.08	0.63
	0.3	76.39	10.63	55.26	3.13	10.31	0.61
	0.5	75.00	4.20	58.50	0.86	4.10	0.55
	0.8	76.70	1.28	64.20	0.61	2.47	0.45
	1.0	71.31	0.85	56.26	0.24	1.40	0.41
CS	0.1	93.10	45.41	92.00	29.90	28.90	0.49
	0.3	93.80	28.63	74.30	16.32	24.74	0.46
	0.5	87.90	19.19	66.60	11.71	12.78	0.44
	0.8	91.70	3.99	75.00	2.94	3.12	0.40
	1.0	95.90	2.04	86.20	1.38	1.36	0.30

Table C.1: Variable selection results for the Elastic Net simulations, including true positive (TP) percentages for fixed and random effects, false positive (FP) percentages for fixed and random effects, the median time in hours for the algorithm to complete (T^{med}), and the average of the mean absolute deviation (Abs. Dev. (Mean)) between the coefficient estimates and the true β values across all simulation replicates. Column "Corr" describes the correlation between the covariates (equal correlation of values 0.2, 0.4, or 0.6, or correlation based on data from the case study data, labeled as 'CS'). Column π represents the Elastic Net balance between ridge regression ($\pi = 0$) and the MCP penalty ($\pi = 1$).

C.2.6 Case study: Study information and data processing

In this section, we provide more information about the individual studies contained within the dataset and describe how we set up the data for the case study analyses in Section 4. More complete coding details are provided in the GitHub repository `https://github.com/hheiling/paper_glmmPen_FA`.

Corr	π	Avg. r	r Underestimated %	r Correct %	r Overestimated %
0.2	0.1	2.67	35	63	2
	0.3	2.65	35	65	0
	0.5	2.63	37	63	0
	0.8	2.54	46	54	0
	1.0	2.62	39	60	1
0.4	0.1	2.24	76	24	0
	0.3	2.35	67	32	1
	0.5	2.38	68	28	4
	0.8	2.47	69	26	5
	1.0	2.35	73	24	3
0.6	0.1	2.05	95	5	0
	0.3	2.30	82	12	5
	0.5	2.30	81	12	7
	0.8	2.38	84	8	8
	1.0	2.99	72	13	15
CS	0.1	2.47	53	47	0
	0.3	2.45	55	45	0
	0.5	2.42	58	42	0
	0.8	2.44	56	44	0
	1.0	2.43	57	43	0

Table C.2: Results of the Growth Ratio r estimation procedure for the Elastic Net $p = 100$ logistic mixed effects simulation results, including the average estimate of r across simulations and percent of times that the estimation procedure underestimated r, gave the true r, or overestimated r. Column "Corr" describes the correlation between the covariates (equal correlation of values 0.2, 0.4, or 0.6, or correlation based on data from the case study data). Column π represents the Elastic Net balance between ridge regression ($\pi = 0$) and the MCP penalty ($\pi = 1$).

The studies used in these analyses are summarized in Chapter 4 Appendix Table C.3, which contains the gene expression platform information (all RNA-seq), the sample sizes, and the percent of the samples that were classified into the basal subgroup.

Chapter 4 Appendix Table C.3 provides the dataset abbreviations, their respective citations, their sample sizes, and the percent of the subjects within each study that were classified into the basal subtype. The sample sizes listed in the table—and used in the analyses—were smaller than the studies' total sample size as we removed subjects with missing tumor grade information, normal tissue samples, and those who did not have primary tumor samples of sufficient quality.

Dataset	Platform	Sample Size	% Basal	Citation
Aguirre	RNA-seq	28	29	Aguirre *et al.* (2018)
CPTAC	RNA-seq	99	22	Cao *et al.* (2021)
Dijk	RNA-seq	61	39	Dijk *et al.* (2020)
Hayashi	RNA-seq	75	53	Hayashi *et al.* (2020)
TCGA	RNA-seq	97	20	Raphael *et al.* (2017)

Table C.3: Summaries of PDAC gene expression datasets. Used in case study prediction model of basal subtype.

All five datasets had RNA-seq data for 60,230 total gene symbols for each subject. Of those, 432 were also part of the 500 member gene list that Moffitt *et al.* (2015) specified as relevant for classifying subjects into the basal vs classical subtypes (this gene list is also provided within the aforementioned GitHub repository).

There were some significant correlations between some of these 432 genes, as evaluated by Spearman correlations. In order to avoid having very highly correlated covariates in the analyses, we decided to combine highly correlated genes together into meta-genes. We accomplished this by applying a hierarchical clustering algorithm to the the genes using the `pheatmap::pheatmap()` R function (Kolde, 2019) to the absolute values of the Spearman correlation matrix. We then cut the tree using the `stats::cutree()` R function (R Core Team, 2021) at a height of 2. This produced a total of 119 clusters. For clusters that represented two or more genes, we added the raw RNA-seq gene expression of all participating genes to get the new cluster covariate values. We further removed 9 of the 119 of these clusters so that all pairwise Spearman correlations were below 0.9. The remaining 110 clusters were rank-transformed on the subject level, and these rank-transformed covariates were used in the case study analyses.

The cancer subtype outcome—basal or classical—was calculated using the clustering algorithm specified in Moffitt *et al.* (2015). For each study individually, this clustering algorithm was applied to the RNA-seq gene expression for the 432 genes described above, where the distance matrix was the Euclidean distance and the assumed number of clusters was set to two.

C.2.7 Case study: Sensitivity analyses

We performed sensitivity analyses on our case study by running the Elastic Net variable selection procedure with alternative values of π—the value that represents the balance between ridge regression and the MCP penalty ($\pi = 0$ represents ridge regression, $pi = 1$ represents the MCP penalty)—and alternative values for the number of latent common factors r (for the **glmmPen_FA** procedure). Based on the results in Chapter 4 Appendix Table C.1, π values between 0.5 and 1.0 were likely to have good selection results for the correlation structure of the covariates in the dataset. We fit the variable selection procedure using $\pi = \{0.6, 0.7, 0.8, 0.9, 1.0\}$. The **glmmPen** procedure assumed an independent random effects covariance matrix.

We first discuss the **glmmPen_FA** sensitivity results. In addition to estimating the number of latent common factors using the Growth Ratio procedure, which estimated a value of $r = 2$, we also fit the model assuming $r = 3$ because the simulations given in Web Appendix Section C.2.5 indicated that the Growth Ratio method may underestimate r. Regardless of whether r was estimated as 2 using the Growth Ratio procedure or set to 3 manually, the coefficient values and selection results were very consistent for each value of π. The single exception was when the $\pi = 0.9$ selection procedure included another cluster covariate in the best model for $r = 3$ but not $r = 2$; even in this case, the fixed effect coefficient for the additional covariate was relatively small.

In terms of fixed effects, the values π between 0.6 and 1.0 gave very consistent results within the **glmmPen_FA** procedure. The 8 covariate clusters described in the main paper Table 4.6 were consistently chosen across the different values of π, with the exception of cluster 45, which was excluded from the best model when $\pi = \{0.6, 0.8\}$. A 9-th cluster, cluster 75 (genes SERPINB3 and SERPINB4) was included when $\pi = 0.9$ and r was set to 3; the fixed effect coefficient for this covariate was relatively small in comparison with the other fixed effect coefficients.

For random effects, values of $\pi \leqslant 0.8$ consistently selected 0 random effect slopes (random intercept only for random effects) within the **glmmPen_FA** procedure. For $\pi = \{0.9, 1.0\}$, clusters 25, 58, and 91 had non-zero random effects. However, when $\pi = 1.0$, the variances of these random effects became suspiciously large (variance values approximately between 9 and 30), indicating a poor model fit for $\pi = 1.0$. The variances of these same random effects when $\pi = 0.9$ were instead approximately between 1 and 2.

We chose to report the **glmmPen_FA** results of $\pi = 0.7$ in the main manuscript for several reasons. Based on the fact that we had a range of correlations among the covariates in the dataset, including some pairwise correlations greater than 0.6, we felt it was appropriate to fit the variable selection procedure with $\pi < 1.0$; furthermore, based on the best model results, the $\pi = 1.0$ showed poor model fit results for the random effects. When choosing between the other values of π, the $\pi = 0.7$ results contained the consistently selected 8 cluster covariates, and these results did not contain any random slopes, which was also consistent across most of the values of $\pi = \{0.6, 0.7, 0.8, 0.9\}$.

The times to complete the **glmmPen_FA** variable selection procedure was between 0.4 and 1.0 hours for $\pi = \{0.6, 0.7, 0.8, 0.9\}$ (this range includes r either 2 or 3). When $\pi = 1.0$, the time to complete the procedure was 1.3 and 2.1 hours for r equal to 2 or 3, respectively.

The **glmmPen** procedure also consistently selected the 8 covariate clusters described in the main paper Table 4.6 to have non-zero fixed effects with the the following exceptions: procedure that used $\pi = 0.7$ penalized out clusters 25 and 45, and the procedure that used $\pi = 1.0$ penalized out cluster 75. With the exception of $\pi = 0.6$, **glmmPen** consistently selected clusters 25 and 58 to have non-zero random effects. The values of these slopes changed depending on the value of π: variances of approximately 0.5 when $\pi = \{0.7, 0.8\}$, and variances greater than 1.0 when $\pi = 0.9$, and divergent values greater than 9 when $\pi = 1.0$.

The times in hours to complete the **glmmPen** variable selection procedure was 32.4, 37.8, 37.7, 51.2, and 68.2 for π equal to 0.6, 0.7, 0.8, 0.9, and 1.0, respectively.

C.3 Chapter 4 proximal gradient line algorithm

As was discussed in Section 1.5, we use a proximal gradient line search algorithm to estimate the appropriate step size to utilize in the Majorization-Minimization algorithm. In this section, we provide more details about specific quantities needed to analyze this line search algorithm.

We want to define an upper bound of our loss function f, labeled as $\hat{f}_\delta$, which is derived from taking the Taylor series expansion of f about the value of $\beta^{(s)}$:

$$\hat{f}_\delta(\beta, \beta^{(s)}) = f(\beta^{(s)}) + \Delta f(\beta^{(s)})^\top(\beta - \beta^{(s)}) + \frac{1}{2\delta}||\beta - \beta^{(s)}||_2^2. \tag{C.6}$$

Let θ represent all coefficients (both fixed and random effects), let l represent the previous EM iteration, and let s represent the previous M-step iteration. We define our loss function as the Q-function defined in (C.7) for a generic value of θ, and we also consider this same expression evaluated at the most recently accepted coefficient vector $\theta^{(s)}$:

$$f(\theta) = -\frac{1}{M}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{i \in n_k} \log f(y_{ki}|x_{ki}, \alpha^{(l,m)}; \theta) \tag{C.7}$$

$$f(\theta^{(s)}) = -\frac{1}{M}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{i \in n_k} \log f(y_{ki}|x_{ki}, \alpha^{(l,m)}; \theta^{(s)}) \tag{C.8}$$

We further define the following terms for a particular subject i in group k evaluated at the m-th MCMC draw of the E-step:

$$f_{kim}(\theta) = -\log f(y_{ki}|x_{ki}, \alpha^{(l,m)}; \theta)$$

$$\tilde{x}_{kim}^\top = \left(x_{ki}^\top \;\; (\alpha_k^{(l,m)} \otimes z_{ki})^\top J\right)$$

$$\eta_{kim} = \tilde{x}_{kim}^\top \theta$$

$$\mu_{kim} = g(\eta_{kim})$$

where $g(.)$ is a link function and the rest of the notation for these terms are specified in Chapter 4.

We consider the contribution to the loss function provided by a particular subject i in group k evaluated at the m-th MCMC draw from the E-step. We define the Taylor series expansion of this contribution to the loss function about the most recently accepted coefficient update $\theta^{(s)}$ as

$$\begin{aligned}
\hat{f}_{kim}(\theta, \theta^{(s)}) &= f_{kim}(\theta^{(s)}) + \Delta f_{kim}(\theta^{(s)})^{\top}(\theta - \theta^{(s)}) + \frac{1}{2\delta}||\theta - \theta^{(s)}||_2^2 \\
&= f_{kim}(\theta^{(s)}) - (y_{ki} - \mu_{kim}^{(s)})\tilde{x}_{kim}^{\top}(\theta - \theta^{(s)}) + \frac{1}{2\delta}||\theta - \theta^{(s)}||_2^2 \\
&= f_{kim}(\theta^{(s)}) - (y_{ki} - \mu_{kim}^{(s)})(\eta_{kim} - \eta_{kim}^{(s)}) + \frac{1}{2\delta}||\theta - \theta^{(s)}||_2^2
\end{aligned} \tag{C.9}$$

The upper bound of the Q-function is defined as the Taylor series expansion of the Q-function about the most recently accepted coefficient update $\theta^{(s)}$. To calculate this, we add the quantity $\hat{f}_{kim}(\theta, \theta^{(s)})$ over all subjects and all MCMC draws and divide by M:

$$\begin{aligned}
\hat{f}(\theta, \theta^{(s)}) &= \frac{1}{M}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{i \in n_k} \hat{f}_{kim}(\theta, \theta^{(s)}) \\
&= f(\theta^{(s)}) - \frac{1}{M}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{i \in n_k}(y_{ki} - \mu_{kim}^{(s)})(\eta_{kim} - \eta_{kim}^{(s)}) + \frac{N}{2\delta}||\theta - \theta^{(s)}||_2^2
\end{aligned} \tag{C.10}$$

We then use this quantity within our line search algorithm to determine if we need to adjust the step size within the Majorization-Minimization algorithm. See the line search algorithm described in Section 1.5 for details.

APPENDIX C: APPENDIX FOR CHAPTER 5

D.1 Chapter 5 additional simulation and case study details

D.1.1 B matrices used in simulations

The transpose of the first 6 rows of the deterministic ‘small’ and ‘moderate’ $\boldsymbol{B}$ matrix used in the piecewise exponential simulations in Section 3 are given in (D.2) and (D.1). All other $p - 5$ rows of the $\boldsymbol{B}$ matrices were set to 0, where $p \in \{100, 500\}$ in the piecewise exponential mixed effects simulations.

$$\boldsymbol{B}^T_{moderate} = 0.75 \times \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & 1 & 1 & 1 \\ -1 & 0 & 1 & -1 & 0 & 1 \end{bmatrix} \tag{D.1}$$

$$\boldsymbol{B}^T_{small} = 0.50 \times \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & 1 & 1 & 1 \\ -1 & 0 & 1 & -1 & 0 & 1 \end{bmatrix} \tag{D.2}$$

D.1.2 Model selection

This section provides details on how the **phmmPen_FA** algorithm selects the optimal tuning parameter combination. In all simulations and analyses discussed in this paper, we set the random effects equal to the fixed effects (i.e. $p = q$) and let the algorithm select the fixed and random effects.

The algorithm runs a computationally efficient two-stage approach to pick the optimal set of tuning parameters. In the first stage of this approach, the algorithm fits a sequence of models where the fixed effect penalty is kept constant at the minimum value of the fixed effects penalty sequence, labeled here as $\lambda_{0,min}$, and the random effects penalty proceeds from the minimum random effect penalty, labeled $\lambda_{1,min}$, to the maximum value $\lambda_{1,max}$. The best model from this first stage is then identified using the BIC-ICQ criterion (Ibrahim *et al.*, 2011). This first stage identifies the optimal random effect penalty value, $\lambda_{1,opt}$. In the second stage, the algorithm fits a sequence of models where the random effects penalty is kept fixed at $\lambda_{1,opt}$ and the fixed effects penalty proceeds from its minimum value $\lambda_{0,min}$ to its maximum value $\lambda_{0,max}$. The overall best model is chosen from the models in the second stage.

We have found this two-stage model selection approach to work very well in practice (see Section 3 for performance results).

D.1.3 Tuning parameter selection

The default maximum penalty, labeled here as λ_{max}, was calculated as the penalty that would penalize all of the fixed effects to 0 when no random effects are in the model. We used code from the **ncvreg** R package (Breheny and Huang, 2011) to calculate this value.

For all piecewise exponential mixed effect variable selection simulations and case study analyses where the total number of predictors was 100, we used the following sequence of penalties for both the fixed effects and the rows of the $\boldsymbol{B}$ matrix: a sequence of 10 penalties from $0.05\lambda_{max}$ to λ_{max}, with penalty values equidistant from each other on the log scale.

For all piecewise exponential outcome variable selection simulations and case study analyses where the total number of predictors was 500, we used the following sequence of penalties: a sequence of 10 penalties from $0.25\lambda_{max}$ to λ_{max} for the fixed effects, and a sequence of 10 penalties from $0.10\lambda_{max}$ to λ_{max} for the rows of the $\boldsymbol{B}$ matrix, with penalty values equidistant from each other on the log scale.

For the case study that performed variable selection on a piecewise exponential mixed effects model with $p = 168$ TSP covariates, we used a sequence of 10 penalties from $0.10\lambda_{max}$ to λ_{max} for both the fixed effects and the rows of the $\boldsymbol{B}$ matrix, with penalty values equidistant from each other on the log scale.

D.1.4 Initialization and convergence for phmmPen_FA

The fixed effects $\tilde{\boldsymbol{\psi}}^{(0)}$ and $\boldsymbol{\beta}^{(0)}$ and random effects covariance terms $\boldsymbol{b}^{(0)}$ are initialized at MCECM iteration $s = 0$ in one of two ways. We discuss first the initialization procedure used when **phmmPen_FA** is used to fit a single model or the first model in the sequence of models fit for variable selection. In this scenario, the fixed effects $\tilde{\boldsymbol{\psi}}^{(0)}$ and $\boldsymbol{\beta}^{(0)}$ are initialized by fitting a 'naive' piecewise exponential model using the coordinate descent techniques of Breheny and Huang (2011) assuming no random effects.

Based on the initialized fixed effects $\boldsymbol{\beta}^{(0)}$, the predictors with non-zero initialized fixed effects are also initialized to have non-zero random effects (i.e. the corresponding rows of the $\boldsymbol{B}$ matrix are set to non-zero values), and predictors with zero-valued initialized fixed effects are initialized to have zero-valued random effects (i.e. the corresponding rows of the $\boldsymbol{B}$ matrix are set to zero). By default, the starting $\boldsymbol{B}$ matrix elements are initialized as $\sqrt{0.02/r}$, where r is the estimated number of latent factors. The corresponding initialized covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{B}\boldsymbol{B}^T$ will have all non-zero elements equal to 0.02. We found that increasing these initial covariance element values tended to increase the false positives of the fixed and random effects of the algorithm.

The E-step MCMC chain of the sample of the posterior density $\phi(\boldsymbol{\alpha}_k|\boldsymbol{\omega}_{k,o};\boldsymbol{\theta}^{(s)})$ for groups $k = 1, ..., K$ is initialized in iteration $s = 1$ with random draws from the standard normal distribution. For all following iterations $s > 1$, the MCMC chain is initialized with the last draw from the previous iteration $s - 1$. At iteration $s = 0$, we sample $M = 100$ posterior samples from each group, and M increases to a max of 500 as the iteration number s increases.

When the algorithm performs variable selection, we initialize models with previous model results. After the first model is fit in the variable selection procedure, the fixed effects, random effects covariance matrix, and random effects MCMC chain are initialized using results from a previous model fit.

The EM algorithm is considered to have converged when the following condition is met at least 2 consecutive times (default) or until the maximum number of EM iterations (25) is reached:

$$||(\tilde{\boldsymbol{\psi}}^{(s)T}, \boldsymbol{\beta}^{(s)T}, \boldsymbol{b}^{(s)T})^T - (\tilde{\boldsymbol{\psi}}^{(s-t)T}, \boldsymbol{\beta}^{(s-t)T}, \boldsymbol{b}^{(s-t)T})^T||_2^2/d_n^{s-t} < \epsilon_{EM} \tag{D.3}$$

where the superscript $(s-t)$ indicates t EM iterations back (default $t = 2$), $||.||_2^2$ represents the L_2 norm, and d_n^{s-t} equals the total number of non-zero $(\tilde{\boldsymbol{\psi}}^T, \boldsymbol{\beta}^T, \boldsymbol{b}^T)^T$ coefficients in iteration $(s-t)$. In other words, the algorithm computes the average Euclidean distance between the current coefficient vector $(\tilde{\boldsymbol{\psi}}^T, \boldsymbol{\beta}^T, \boldsymbol{b}^T)^T$ and the coefficient vector from t EM iterations back and compares it with $\epsilon_{EM} = 0.0015$.

The M-step algorithm is considered to have converged when the following condition is met or until the maximum number of iterations (50) is reached:

$$\max\{\text{max}_j|\tilde{\psi}_j^{(s,g+1)} - \tilde{\psi}_j^{(s,g)}|, \text{max}_l|\beta_l^{(s,g+1)} - \beta_l^{(s,g)}|, \text{max}_{t,h}|b_{th}^{(s,g+1)} - b_{th}^{(s,g)}|\} < \epsilon_m, \tag{D.4}$$

where b_{th} is an individual element of $\boldsymbol{b}_t$, which is the t-th row of the $\boldsymbol{B}$. The value of ϵ_m was set to 0.001.

D.1.5 Case study: Study information and data processing

In this section, we provide more information about the individual studies contained within the dataset and describe how we set up the data for the case study analyses in Section 4. More complete coding details are provided in the GitHub repository `https://github.com/hheiling/paper_phmmPen_FA`.

The studies used in these analyses are summarized in Supplementary Material Table D.1, which contains the gene expression platform information (RNA-seq vs microarray), the gene set sizes, the sample sizes, the number of events, and the corresponding citations for each of the studies.

Dataset	Platform	Gene Set Size	Sample Size	# Events	Citation
Aguirre	RNA-seq	52576	47	35	Aguirre *et al.* (2018)
CPTAC	RNA-seq	28057	124	46	Cao *et al.* (2021)
Dijk	RNA-seq	49677	90	81	Dijk *et al.* (2020)
Moffitt	Microarray	19749	123	83	Moffitt *et al.* (2015)
PACA_AU	Microarray	47265	63	38	Bailey *et al.* (2016)
Puleo	Microarray	20087	288	181	Puleo *et al.* (2018)
TCGA	RNA-seq	20531	144	75	Raphael *et al.* (2017)

Table D.1: Summaries of PDAC gene expression datasets. Used in case study prediction model of survival.

The subjects used in the analyses were restricted to those with primary pancreatic ductal adenocarinoma samples. Of the total genes listed, 420 of these genes were both part of the 500 member gene list that Moffitt *et al.* (2015) specified as relevant for classifying subjects into the basal vs classical subtypes and common among all of the datasets. (The 500 member gene list is also provided within the aforementioned GitHub repository). We removed the bottom 20% of these 420 genes by rank-transforming the genes and then removing the genes with the lowest average rank.

Because the datasets are a mix of RNA-seq and microarray datasets, we integrated the data together using the data integration rank transformation technique as specified by Rashid *et al.* (2020). This integration technique creates top scoring pairs (TSPs). To illustrate the interpretation of TSPs, let $g_{ki,A}$ and $g_{ki,B}$ be the raw expression of genes A and B in subject i of group k. For each gene pair ($g_{ki,A}$, $g_{ki,B}$), the TSP is an indicator $I(g_{ki,A} > g_{ki,B})$ which specifies which of the two genes has higher expression in the subject. We denote a TSP predictor as "GeneA_GeneB".

To pick the TSPs used in our analyses, we first removed TSPs that had low variation across the samples. If the average value of the TSP was less than 0.10 or greater than 0.90, they were not included in later analyses. Next, we fit single covariate proportional hazard mixed effects models using the **coxme** R package (**?**), where the models contained the TSP, a random intercept, and a random slope for the TSP. From these models, we extracted the estimated log-likelihoods and sorted the TSPs based on these log-likelihoods (highest to lowest).

After sorting the TSPs by their log-likelihoods, we removed TSPs with high correlation. Starting with the TSP with the maximum log-likelihood, we removed all other TSPs with lower (i.e. more negative) log-likelihood values that shared one of the genes in the first TSP. Then we moved to the next TSP with the second highest log-likelihood and repeated this process, continuing in this way for all downstream TSPs. At the end of this procedure, we were left with 168 TSPs.

D.1.6 Case study: Sensitivity analyses

We performed sensitivity analyses on our case study by running the Elastic Net variable selection procedure with alternative values of π—the value that represents the balance between ridge regression and the MCP penalty ($\pi = 0$ represents ridge regression, $\pi = 1$ represents the MCP penalty)—and alternative values for the number of latent common factors r (for the **phmmPen_FA** procedure). We considered values of $\pi = 0.7, 0.8, 0.9, 1.0$.

In addition to estimating the number of latent common factors using the Growth Ratio procedure, which estimated a value of $r = 2$, we also fit the model assuming $r = 3$ because the simulations given in the main paper indicated that the Growth Ratio method may underestimate r.

When we set $\pi = 0.9$, which were the main results reported in the paper, the coefficient values and selection results were very consistent for the different values of r used in the analyses. Of the 19 TSPs selected when $\pi = 0.9$ and $r = 2$, 17 were also selected when r was set to 3, and the coefficient values reported were very similar between the two sets of results. When r was set to 3, the algorithm did not select the following TSP covariates: SMPD3_RHOD (log hazard ratio of -0.55) and ZNF165_BMP4 (log hazard ratio of -0.38). Additionally, SMPD3_RHOD was selected to have a random effect across the studies when $r = 3$ instead of CYP2C18_COX6B2 for $r = 2$.

When we set $\pi = 0.8$ and set r to the Growth Ratio estimate of 2, the fixed effects shown in the main paper Figure 1 were also selected, and they reported very similar coefficient values to those presented in Figure 1. The random effect CYP2C18_COX6B2 was selected just as in the main reported results.

If we set $\pi = 0.7$ and again set r to the Growth Ratio estimate of 2, 8 of the 19 TSPs selected in the $\pi = 0.9$ scenario were no longer selected in the $\pi = 0.7$ scenario (see output from code in the GitHub repository `https://github.com/hheiling/paper_phmmPen_FA` for details). The coefficient values for these 8 TSPs were relatively small compared to the 11 TSPs selected in all $\pi = \{0.7, 0.8, 0.9\}$ scenarios, so the results still seemed fairly consistent across the conditions. The $\pi = 0.7$ also selected CYP2C18_COX6B2 as a random effect.

The times to complete the **phmmPen_FA** variable selection procedure for $\pi = \{0.7, 0.8, 0.9, 1.0\}$ was between 2.1 and 2.4 hours when $r = 2$ and between 2.7 and 5.1 hours when $r = 3$.

D.1.7 Software

Software in the form of R code is available through the GitHub repository `https://github.com/hheiling/glmmPen`. Code to run the simulations and the case study analysis is available through the GitHub repository `https://github.com/hheiling/paper_phmmPen_FA`

REFERENCES

Aguirre, A. J., Nowak, J. A., Camarda, N. D., Moffitt, R. A., Ghazani, A. A., Hazar-Rethinam, M., Raghavan, S., Kim, J., Brais, L. K., Ragon, D., *et al.* (2018). Real-time genomic characterization of advanced pancreatic cancer to enable precision medicine. *Cancer discovery*, **8**(9), 1096–1111.

Ahn, S. C. and Horenstein, A. R. (2013). Eigenvalue ratio test for the number of factors. *Econometrica*, **81**(3), 1203–1227.

Archila, F. A. (2020). *mcemGLM: Maximum Likelihood Estimation for Generalized Linear Mixed Models*. R package version 1.1.1.

Austin, P. C. (2017). A tutorial on multilevel survival analysis: methods, models and applications. *International Statistical Review*, **85**(2), 185–203.

Bai, J. and Ng, S. (2002). Determining the number of factors in approximate factor models. *Econometrica*, **70**(1), 191–221.

Bailey, P., Chang, D. K., Nones, K., Johns, A. L., Patch, A.-M., Gingras, M.-C., Miller, D. K., Christ, A. N., Bruxner, T. J., Quinn, M. C., *et al.* (2016). Genomic analyses identify molecular subtypes of pancreatic cancer. *Nature*, **531**(7592), 47–52.

Bates, D., Mächler, M., Bolker, B., and Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, **67**(1), 1–48.

Baynton, M. (1992). Dimensions of control in distance education: A factor analysis. *American Journal of Distance Education*, **6**(2), 17–31.

Becht, E., Giraldo, N. A., Lacroix, L., Buttard, B., Elarouci, N., Petitprez, F., Selves, J., Laurent-Puig, P., Sautes-Fridman, C., Fridman, W. H., and et al. (2016a). Estimating the population abundance of tissue-infiltrating immune and stromal cell populations using gene expression. *Genome Biology*, **17**(218).

Becht, E., de Reyniès, A., Giraldo, N. A., Pilati, C., Buttard, B., Lacroix, L., Selves, J., Sautès-Fridman, C., Laurent-Puig, P., and Fridman, W. H. (2016b). Immune and stromal classification of colorectal cancer is associated with molecular subtypes and relevant for precision immunotherapy. *Clinical cancer research*, **22**(16), 4057–4066.

Bondell, H. D., Krishna, A., and Ghosh, S. K. (2010). Joint variable selection for fixed and random effects in linear mixed-effects models. *Biometrics*, **66**(4), 1069–1077.

Booth, J. G. and Hobert, J. P. (1999). Maximizing generalized linear mixed model likelihoods with an automated monte carlo em algorithm. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **61**(1), 265–285.

Bradic, J., Fan, J., and Jiang, J. (2011). Regularization for coxs proportional hazards model with np-dimensionality. *Annals of statistics*, **39**(6), 3092.

Breheny, P. and Huang, J. (2011). Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *Annals of Applied Statistics*, **5**(1), 232–253.

Breheny, P. and Huang, J. (2015). Group descent algorithms for nonconvex penalized linear and logistic regression models with grouped predictors. *Statistics and computing*, **25**(2), 173–187.

Cao, L., Huang, C., Zhou, D. C., Hu, Y., Lih, T. M., Savage, S. R., Krug, K., Clark, D. J., Schnaubelt, M., Chen, L., *et al.* (2021). Proteogenomic characterization of pancreatic ductal adenocarcinoma. *Cell*, **184**(19), 5031–5052.

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., and Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of statistical software*, **76**(1).

Carvalho, C. M., Chang, J., Lucas, J. E., Nevins, J. R., Wang, Q., and West, M. (2008). High-dimensional sparse factor modeling: applications in gene expression genomics. *Journal of the American Statistical Association*, **103**(484), 1438–1456.

Chamberlain, G. and Rothschild, M. (1982). Arbitrage, factor structure, and mean-variance analysis on large asset markets.

Chan, J. C. and Grant, A. L. (2015). Pitfalls of estimating the marginal likelihood using the modified harmonic mean. *Economics Letters*, **131**, 29–33.

Chen, L., Ge, B., Casale, F. P., Vasquez, L., Kwan, T., Garrido-Martín, D., Watt, S., Yan, Y., Kundu, K., Ecker, S., *et al.* (2016). Genetic drivers of epigenetic and transcriptional variation in human immune cells. *Cell*, **167**(5), 1398–1414.

Chen, Z. and Dunson, D. B. (2003). Random effects selection in linear mixed models. *Biometrics*, **59**(4), 762–769.

Chib, S. and Jeliazkov, I. (2001). Marginal likelihood from the metropolis–hastings output. *Journal of the American statistical association*, **96**(453), 270–281.

Clarke, J., Seo, P., and Clarke, B. (2010). Statistical expression deconvolution from mixed tissue samples. *Bioinformatics*, **26**(8), 10431049.

Cortiñas Abrahantes, J. and Burzykowski, T. (2005). A version of the em algorithm for proportional hazard model with random effects. *Biometrical Journal: Journal of Mathematical Methods in Biosciences*, **47**(6), 847–862.

Crooks, J. L., Berger, J. O., and Loredo, T. J. (2007). Posterior-guided importance sampling for calculating marginal likelihoods with application to bayesian exoplanet searches. *Discussion Paper Series of Dept. of Statitical Science.*

Delattre, M., Lavielle, M., Poursat, M.-A., *et al.* (2014). A note on bic in mixed-effects models. *Electronic journal of statistics*, **8**(1), 456–475.

Dijk, F., Veenstra, V. L., Soer, E. C., Dings, M. P., Zhao, L., Halfwerk, J. B., Hooijer, G. K., Damhofer, H., Marzano, M., Steins, A., *et al.* (2020). Unsupervised class discovery in pancreatic ductal adenocarcinoma reveals cell-intrinsic mesenchymal features and high concordance between existing classification systems. *Scientific reports*, **10**(1), 337.

Donohue, M., Overholser, R., Xu, R., and Vaida, F. (2011). Conditional akaike information under generalized linear and proportional hazards mixed models. *Biometrika*, **98**(3), 685–700.

Eddelbuettel, D. and François, R. (2011). Rcpp: Seamless R and C++ integration. *Journal of Statistical Software*, **40**(8), 1–18.

Eddelbuettel, D. and Sanderson, C. (2014). Rcpparmadillo: Accelerating r with high-performance c++ linear algebra. *Computational Statistics and Data Analysis*, **71**, 1054–1063.

Fama, E. F. and French, K. R. (1992). The cross-section of expected stock returns. *the Journal of Finance*, **47**(2), 427–465.

Fan, J. and Li, R. (2002). Variable selection for cox's proportional hazards model and frailty model. *The Annals of Statistics*, **30**(1), 74–99.

Fan, J., Fan, Y., and Lv, J. (2008). High dimensional covariance matrix estimation using a factor model. *Journal of Econometrics*, **147**(1), 186–197.

Fan, J., Liao, Y., and Mincheva, M. (2013). Large covariance estimation by thresholding principal orthogonal complements. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **75**(4), 603–680.

Fan, Y. and Li, R. (2012). Variable selection in linear mixed effects models. *Annals of statistics*, **40**(4), 2043.

Fava, J. L. and Velicer, W. F. (1992). An empirical comparison of factor, image, component, and scale scores. *Multivariate Behavioral Research*, **27**(3), 301–322.

Fitzmaurice, G. M., Laird, N. M., and Ware, J. H. (2012). *Applied longitudinal analysis*, volume 998, chapter 14. John Wiley & Sons, 2nd edition.

Fourment, M., Magee, A. F., Whidden, C., Bilge, A., Matsen IV, F. A., and Minin, V. N. (2020). 19 dubious ways to compute the marginal likelihood of a phylogenetic tree topology. *Systematic biology*, **69**(2), 209–220.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, **33**(1), 1–22.

Friedman, M. (1982). Piecewise exponential models for survival data with covariates. *The Annals of Statistics*, **10**(1), 101–113.

Garcia, R. I., Ibrahim, J. G., and Zhu, H. (2010). Variable selection for regression models with missing data. *Statistica Sinica*, **20**(1), 149.

Givens, G. H. and Hoeting, J. A. (2012). *Computational statistics*, volume 703, chapter 7. John Wiley & Sons, 2nd edition.

Gong, T. and Szustakowski, J. D. (2013). Deconrnaseq: a statistical framework for deconvolution of heterogeneous tissue samples based on mrna-seq data. *Bioinformatics*, **29**(8), 10831085.

Gorsuch, R. L. (2014). *Factor analysis: Classic edition*. Routledge.

Gosink, M. M., Petrie, H. T., and Tsinoremas, N. F. (2007). Electronically subtracting expression patterns from a mixed cell population. *Bioinformatics*, **23**(24), 33283334.

Groll, A. (2017). Package glmmlasso.

Guadagnoli, E. (1984). The relationship of sample size to the stability of component patterns: A simulation study.

Gurka, M. J., Edwards, L. J., and Muller, K. E. (2011). Avoiding bias in mixed model inference for fixed effects. *Statistics in medicine*, **30**(22), 2696–2707.

Hadfield, J. D. (2010). Mcmc methods for multi-response generalized linear mixed models: The MCMCglmm R package. *Journal of Statistical Software*, **33**(2), 1–22.

Hayashi, A., Fan, J., Chen, R., Ho, Y.-j., Makohon-Moore, A. P., Lecomte, N., Zhong, Y., Hong, J., Huang, J., Sakamoto, H., *et al.* (2020). A unifying paradigm for transcriptional heterogeneity and squamous features in pancreatic ductal adenocarcinoma. *Nature Cancer*, **1**(1), 59–74.

Heavens, A., Fantaye, Y., Mootoovaloo, A., Eggers, H., Hosenie, Z., Kroon, S., and Sellentin, E. (2017). Marginal likelihoods from monte carlo markov chains. *arXiv preprint arXiv:1704.03472*.

Heiling, H., Rashid, N., Li, Q., and Ibrahim, J. (2023a). *glmmPen: High Dimensional Penalized Generalized Linear Mixed Models (pGLMM)*. R package version 1.5.3.4.

Heiling, H. M., Rashid, N. U., Li, Q., Peng, X. L., Yeh, J. J., and Ibrahim, J. G. (2023b). Efficient computation of high-dimensional penalized generalized linear mixed models by latent factor modeling of the random effects.

Heiling, H. M., Rashid, N. U., Li, Q., and Ibrahim, J. G. (2023c). glmmpen: High dimensional penalized generalized linear mixed models.

Hoffman, M. D. and Gelman, A. (2014). The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res.*, **15**(1), 1593–1623.

Holford, T. R. (1980). The analysis of rates and of survivorship using log-linear models. *Biometrics*, pages 299–305.

Houseman, E. A., Accomando, W. P., Koestler, D. C., Christensen, B. C., Marsit, C. J., Nelson, H. H., Wiencke, J. K., and Kelsey, K. T. (2012). Dna methylation arrays as surrogate measures of cell mixture distribution. *BMC bioinformatics*, **13**(1), 1–16.

Houseman, E. A., Molitor, J., and Marsit, C. J. (2014). Reference-free cell mixture adjustments in analysis of dna methylation data. *Bioinformatics*, **30**(10), 1431–1439.

Ibrahim, J. G., Zhu, H., Garcia, R. I., and Guo, R. (2011). Fixed and random effects selection in mixed effects models. *Biometrics*, **67**(2), 495–503.

Jacobson, N. S., Dobson, K. S., Truax, P. A., Addis, M. E., Koerner, K., Gollan, J. K., Gortner, E., and Prince, S. E. (1996). A component analysis of cognitive-behavioral treatment for depression. *Journal of consulting and clinical psychology*, **64**(2), 295.

Jin, C., Chen, M., Lin, D., and Sun, W. (2020). Cell type aware analysis of rna-seq data (carseq) reveals difference and similarities of the molecular mechanisms of schizophrenia and autism. *bioRxiv*.

Kane, M. J., Emerson, J., and Weston, S. (2013). Scalable strategies for computing with massive data. *Journal of Statistical Software*, **55**(14), 1–19.

Kapetanios, G. (2010). A testing procedure for determining the number of factors in approximate factor models with large datasets. *Journal of Business & Economic Statistics*, **28**(3), 397–409.

Khorana, A. A., Mangu, P. B., Berlin, J., Engebretson, A., Hong, T. S., Maitra, A., Mohile, S. G., Mumber, M., Schulick, R., Shapiro, M., *et al.* (2016). Potentially curable pancreatic cancer: American society of clinical oncology clinical practice guideline. *Journal of Clinical Oncology*, **34**(21), 2541–2556.

Kleinman, K., Lazarus, R., and Platt, R. (2004). A generalized linear mixed models approach for detecting incident clusters of disease in small areas, with an application to biological terrorism. *American Journal of Epidemiology*, **159**(3), 217–224.

Kolde, R. (2019). *pheatmap: Pretty Heatmaps*. R package version 1.0.12.

Laird, N. and Olivier, D. (1981). Covariance analysis of censored survival data using log-linear analysis techniques. *Journal of the American Statistical Association*, **76**(374), 231–240.

Lebrigand, K., Bergenstråhle, J., Thrane, K., Mollbrink, A., Barbry, P., Waldmann, R., and Lundeberg, J. (2020). The spatial landscape of gene expression isoforms in tissue sections. *bioRxiv*.

Lee, K. E., Kim, Y., and Xu, R. (2014). Bayesian variable selection under the proportional hazards mixed-effects model. *Computational statistics & data analysis*, **75**, 53–65.

Leek, J. T. and Storey, J. D. (2007). Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS genetics*, **3**(9), e161.

Li, B., Severson, E., Pignon, J.-C., Zhao, H., Li, T., Novak, J., Jiang, P., Shen, H., Aster, J. C., Rodig, S., *et al.* (2016). Comprehensive analyses of tumor immunity: implications for cancer immunotherapy. *Genome Biology*, **17**(1), 174.

Li, Z. and Wu, H. (2019). Toast: improving reference-free cell composition estimation by cross-cell type differential analysis. *Genome biology*, **20**(1), 190.

Lorah, J. and Womack, A. (2019). Value of sample size for computation of the bayesian information criterion (bic) in multilevel modeling. *Behavior research methods*, **51**(1), 440–450.

Love, M. I., Huber, W., and Anders, S. (2014). Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome biology*, **15**(12), 550.

Lu, P., Nakorchevskiy, A., and Marcotte, E. M. (2003). Expression deconvolution: A reinterpretation of dna microarray data reveals dynamic changes in cell populations. *Proceedings of the National Academy of Sciences*, **100**(18), 1037010375.

Lusa, L., McShane, L. M., Reid, J. F., De Cecco, L., Ambrogi, F., Biganzoli, E., Gariboldi, M., and Pierotti, M. A. (2007). Challenges in projecting clustering results across gene expression–profiling datasets. *JNCI: Journal of the National Cancer Institute*, **99**(22), 1715–1723.

Ma, S., Ogino, S., Parsana, P., Nishihara, R., Qian, Z., Shen, J., Mima, K., Masugi, Y., Cao, Y., Nowak, J. A., *et al.* (2018). Continuity of transcriptomes among colorectal cancer subtypes based on meta-analysis. *Genome biology*, **19**(1), 142.

Maier, M. J. (2014). Dirichletreg: Dirichlet regression for compositional data in r. *Research Report Series / Department of Statistics and Mathematics*, **125**.

Maynard, K. R., Collado-Torres, L., Weber, L. M., Uytingco, C., Barry, B. K., Williams, S. R., Catallini, J. L., Tran, M. N., Besich, Z., Tippani, M., Chew, J., Yin, Y., Kleinman, J. E., Hyde, T. M., Rao, N., Hicks, S. C., Martinowich, K., and Jaffe, A. E. (2020). Transcriptome-scale spatial gene expression in the human dorsolateral prefrontal cortex. *bioRxiv*.

Moffitt, R. A., Marayati, R., Flate, E. L., Volmar, K. E., Loeza, S. G. H., Hoadley, K. A., Rashid, N. U., Williams, L. A., Eaton, S. C., Chung, A. H., *et al.* (2015). Virtual microdissection identifies distinct tumor-and stroma-specific subtypes of pancreatic ductal adenocarcinoma. *Nature genetics*, **47**(10), 1168.

Nazarov, P. V., Wienecke-Baldacchino, A. K., Zinovyev, A., Czerwińska, U., Muller, A., Nashan, D., Dittmar, G., Azuaje, F., and Kreis, S. (2019). Deconvolution of transcriptomes and mirnomes by independent component analysis provides insights into biological processes and clinical outcomes of melanoma patients. *BMC medical genomics*, **12**(1), 1–17.

Newman, A. M., Liu, C. L., Green, M. R., Gentles, A. J., Feng, W., Xu, Y., Hoang, C. D., Diehn, M., and Alizadeh, A. A. (2015). Robust enumeration of cell subsets from tissue expression profiles. *Nature Methods*, **12**(5), 453457.

Newman, A. M., Steen, C. B., Liu, C. L., Gentles, A. J., Chaudhuri, A. A., Scherer, F., Khodadoust, M. S., Esfahani, M. S., Luca, B. A., Steiner, D., *et al.* (2019). Determining cell type abundance and expression from bulk tissues with digital cytometry. *Nature biotechnology*, **37**(7), 773–782.

Onatski, A. (2010). Determining the number of factors from empirical distribution of eigenvalues. *The Review of Economics and Statistics*, **92**(4), 1004–1016.

Pajor, A. (2017). Estimating the marginal likelihood using the arithmetic mean identity. *Bayesian Analysis*, **12**(1), 261–287.

Paquet, E. R. and Hallett, M. T. (2015). Absolute assignment of breast cancer intrinsic molecular subtype. *Journal of the National Cancer Institute*, **107**(1), dju357.

Parikh, N., Boyd, S., *et al.* (2014). Proximal algorithms. *Foundations and trends® in Optimization*, **1**(3), 127–239.

Parikshak, N. N., Swarup, V., Belgard, T. G., Irimia, M., Ramaswami, G., Gandal, M. J., Hartl, C., Leppa, V., Ubieta, L. d. l. T., Huang, J., Lowe, J. K., Blencowe, B. J., Horvath, S., and Geschwind, D. H. (2016). Genome-wide changes in lncRNA, splicing, and regional gene expression patterns in autism. *Nature*, **540**(7633), 423–427.

Patil, P. and Parmigiani, G. (2018). Training replicable predictors in multiple studies. *Proceedings of the National Academy of Sciences*, **115**(11), 2578–2583.

Pinheiro, J., Bates, D., DebRoy, S., Sarkar, D., Heisterkamp, S., and Van Willigen, B. (2017). Package nlme. *Linear and nonlinear mixed effects models, version*, **3**(1).

Puleo, F., Nicolle, R., Blum, Y., Cros, J., Marisa, L., Demetter, P., Quertinmont, E., Svrcek, M., Elarouci, N., Iovanna, J., *et al.* (2018). Stratification of pancreatic ductal adenocarcinomas based on tumor and microenvironment features. *Gastroenterology*, **155**(6), 1999–2013.

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Raphael, B. J., Hruban, R. H., Aguirre, A. J., Moffitt, R. A., Yeh, J. J., Stewart, C., Robertson, A. G., Cherniack, A. D., Gupta, M., Getz, G., *et al.* (2017). Integrated genomic characterization of pancreatic ductal adenocarcinoma. *Cancer cell*, **32**(2), 185–203.

Rashid, N., Sun, W., and Ibrahim, J. G. (2014). Some statistical strategies for dae-seq data analysis: variable selection and modeling dependencies among observations. *Journal of the American Statistical Association*, **109**(505), 78–94.

Rashid, N. U., Li, Q., Yeh, J. J., and Ibrahim, J. G. (2020). Modeling between-study heterogeneity for improved replicability in gene signature selection and clinical prediction. *Journal of the American Statistical Association*, **115**(531), 1125–1138.

Riester, M., Wei, W., Waldron, L., Culhane, A. C., Trippa, L., Oliva, E., Kim, S.-h., Michor, F., Huttenhower, C., Parmigiani, G., *et al.* (2014). Risk prediction for late-stage ovarian cancer by meta-analysis of 1525 patient samples. *JNCI: Journal of the National Cancer Institute*, **106**(5).

Ripatti, S. and Palmgren, J. (2000). Estimation of multivariate frailty models using penalized partial likelihood. *Biometrics*, **56**(4), 1016–1022.

Roberts, G. O. and Rosenthal, J. S. (2009). Examples of adaptive mcmc. *Journal of Computational and Graphical Statistics*, **18**(2), 349–367.

Rodriguez, G. (2010). Survival models. *Quantile*, (5), 1–27.

SAS Institute Inc. (2008). *SAS/STAT Software, Version 9.2*. Cary, NC.

Schelldorfer, J., Meier, L., and Bühlmann, P. (2014). Glmmlasso: an algorithm for high-dimensional generalized linear mixed models using l1-penalization. *Journal of Computational and Graphical Statistics*, **23**(2), 460–477.

Schmidt-Catran, A. W. and Fairbrother, M. (2016). The random effects in multilevel models: Getting them wrong and getting them right. *European Sociological Review*, **32**(1), 23–38.

Shen-Orr, S. S., Tibshirani, R., Khatri, P., Bodian, D. L., Staedtler, F., Perry, N. M., Hastie, T., Sarwal, M. M., Davis, M. M., Butte, A. J., and et al. (2010). Cell type-specific gene expression differences in complex tissues. *Nature Methods*, **7**(4), 287289.

Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. (2011). Regularization paths for cox's proportional hazards model via coordinate descent. *Journal of Statistical Software*, **39**(5), 1–13.

Snook, S. C. and Gorsuch, R. L. (1989). Component analysis versus common factor analysis: A monte carlo study. *Psychological bulletin*, **106**(1), 148.

Sotiriou, C. and Piccart, M. J. (2007). Taking gene-expression profiling to the clinic: when will molecular signatures become relevant to patient care? *Nature reviews cancer*, **7**(7), 545–553.

Sun, W., Liu, Y., Crowley, J. J., Chen, T.-H., Zhou, H., Chu, H., Huang, S., Kuan, P.-F., Li, Y., Miller, D., *et al.* (2015). Isodot detects differential rna-isoform expression/usage with respect to a categorical or continuous covariate with high sensitivity and specificity. *Journal of the American Statistical Association*, **110**(511), 975–986.

Swisher, E. M., Taniguchi, T., and Karlan, B. Y. (2012). Molecular scores to predict ovarian cancer outcomes: a worthy goal, but not ready for prime time.

Szyszkowicz, M. (2006). Use of generalized linear mixed models to examine the association between air pollution and health outcomes. *International journal of occupational medicine and environmental health*, **19**(4), 224–227.

Terry M. Therneau and Patricia M. Grambsch (2000). *Modeling Survival Data: Extending the Cox Model*. Springer, New York.

Teschendorff, A. E. and Zheng, S. C. (2017). Cell-type deconvolution in epigenome-wide association studies: a review and recommendations. *Epigenomics*, **9**(5), 757–768.

Therneau, T. M. (2021). *A Package for Survival Analysis in R*. R package version 3.2-11.

Thompson, J. A., Fielding, K. L., Davey, C., Aiken, A. M., Hargreaves, J. R., and Hayes, R. J. (2017). Bias and inference from misspecified mixed-effect models in stepped wedge trial analysis. *Statistics in medicine*, **36**(23), 3670–3682.

Tibshirani, R. (1997). The lasso method for variable selection in the cox model. *Statistics in medicine*, **16**(4), 385–395.

Tran, M.-N., Nguyen, N., Nott, D., and Kohn, R. (2020). Bayesian deep net glm and glmm. *Journal of Computational and Graphical Statistics*, **29**(1), 97–113.

Vaida, F. and Xu, R. (2000). Proportional hazards model with random effects. *Statistics in medicine*, **19**(24), 3309–3324.

Velicer, W. F. and Jackson, D. N. (1990). Component analysis versus common factor analysis: Some issues in selecting an appropriate procedure. *Multivariate behavioral research*, **25**(1), 1–28.

Waldron, L., Haibe-Kains, B., Culhane, A. C., Riester, M., Ding, J., Wang, X. V., Ahmadifar, M., Tyekucheva, S., Bernau, C., Risch, T., *et al.* (2014). Comparative meta-analysis of prognostic gene signatures for late-stage ovarian cancer. *JNCI: Journal of the National Cancer Institute*, **106**(5).

Wang, E. T., Sandberg, R., Luo, S., Khrebtukova, I., Zhang, L., Mayr, C., Kingsmore, S. F., Schroth, G. P., and Burge, C. B. (2008). Alternative isoform regulation in human tissue transcriptomes. *Nature*, **46**(7221), 470476.

Wang, N., Gong, T., Clarke, R., Chen, L., Shih, I.-M., Zhang, Z., Levine, D. A., Xuan, J., and Wang, Y. (2014). Undo: a bioconductor r package for unsupervised deconvolution of mixed gene expressions in tumor samples. *Bioinformatics*, **31**(1), 137139.

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

Xu, R., Vaida, F., and Harrington, D. P. (2009). Using profile likelihood for semiparametric model selection with application to proportional hazards mixed models. *Statistica Sinica*, **19**(2), 819.

Zhong, Y., Wan, Y.-W., Pang, K., Chow, L. M., and Liu, Z. (2013). Digital sorting of complex tissues for cell type-specific gene expression profiles. *BMC Bioinformatics*, **14**(1), 89.