Latent Variable Models for Analysing Multidimensional Gene Expression Data

Victoria Hore

Oriel College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

December 22, 2015

Acknowledgements

First, I would like to thank my supervisor, Jonathan Marchini, for all the help he has given me over the past four years. I couldn't have asked for a more supportive and encouraging supervisor, without him this thesis would not exist.

I am very grateful to Kerrin Small, Simon Myers and Mark McCarthy for many fruitful discussions, help interpreting results and getting access to data sets. I thank the Life Sciences Interface Doctoral Training Center for preparing me for the DPhil (insofar as that is possible) and EPSRC for financial support.

My fellow group members have kept me motivated over the last few years with many coffee breaks and interesting discussions. In particular, I am very grateful for Andy's help with linear algebra and Winni's patience with my programming questions.

Finally, I would like to thank my parents for their support and proof-reading. I must also thank my sisters: Katie for proving that it is possible to finish a DPhil, Rosie for always asking how the DPhil is going and Becky for patiently letting me teach her Maths. Last but not least, to Tom for getting me excited about the future and pushing me to finish.

Abstract

Multi-tissue gene expression studies give rise to 3D arrays of data. These experiments make it possible to study the tissue-specific nature of gene regulation and also the relationship between genotypes and higher level traits such as disease status. Analysing these multidimensional data sets is a statistical challenge, as they contain high noise levels and missing data. In this thesis I introduce a new approach for analysing multidimensional gene expression data sets called SPIDER (SParse Integrated DEcomposition for RNA-sequencing). SPIDER is a sparse Bayesian tensor decomposition that models the data as a sum of components (or factors). Each component consists of three vectors of scores or loadings that describe modes of variation across individuals, genes and tissues. Sparsity is induced in the components using a spike and slab prior, allowing for recovery of sparse structure in the data. The decomposition is easily extended to jointly decompose several data types, handle missing data and allow for relatedness between individuals, another common problem in genetics. Inference for the model is performed using variational Bayes.

SPIDER is compared to existing approaches for decomposing multidimensional data via simulations. Results suggest that SPIDER performs comparably to, or better than, existing approaches and particularly well when the underlying signals are very sparse. Additional simulations designed to contain realistic levels of signal and noise suggest that SPIDER has the power to recover gene networks from gene expression data.

I have applied SPIDER to gene expression data measured using RNA-sequencing for 845 individuals in three tissues from the TwinsUK cohort. Estimated components were tested for association with genetic variation genome-wide. Five signals describing gene regulation networks driven by genetic variants are uncovered, building on the current understanding of these pathways. In addition,

components uncovering effects of experimental artefacts and covariates were also recovered from the data.

Contents

1 Introduction **1**

2 Literature review **5**
2.1 Latent variable models for multidimensional data 5
2.1.1 Canonical correlation analysis 7
2.1.2 Inter Battery Factor Analysis 9
2.1.3 Group factor analysis 11
2.1.4 Sparsity 12
2.1.5 Tensor factorisations 14
2.1.6 Identifiability 18
2.2 Gene expression 19
2.2.1 Motivation 19
2.2.2 eQTL mapping 21
2.2.3 Confounding 27
2.3 Discussion 29

3 Methods **31**
3.1 Heuristic description of the method 32
3.2 Notation 33
3.3 Tensor decomposition 33
3.3.1 Model description 33
3.3.2 Priors 34
3.3.3 Full model 37
3.3.4 Overview of variational Bayes 38
3.3.5 Identifiability 44
3.4 Extensions 45
3.4.1 Missing data 45
3.4.2 Related individuals 48
3.4.3 Linked tensor decomposition 50
3.5 Implementation 52
3.5.1 Complexity 53
3.6 Comparison with existing methods 54
3.7 Discussion 56

4 Simulation study **59**
4.1 Method comparisons 59
4.1.1 Comparison of tensor decompositions 60
4.1.2 Comparison of group decompositions 66
4.1.3 Discussion 74

4.2 Trans effect simulations . . . 76
4.2.1 Data simulation . . . 76
4.2.2 The method . . . 81
4.2.3 Post-processing and metrics . . . 81
4.2.4 Discussion . . . 90

5 Results **92**
5.1 Data . . . 92
5.1.1 Data collection . . . 92
5.1.2 Pre-processing . . . 94
5.2 Method . . . 95
5.2.1 Tensor decomposition using SPIDER . . . 95
5.2.2 Post-processing . . . 95
5.2.3 Direct associations . . . 98
5.3 Results . . . 99
5.3.1 Summary of the output . . . 99
5.3.2 Components explaining *trans* effects . . . 102
5.3.3 Highest negative free energy run . . . 114
5.4 Discussion . . . 118

6 Conclusion **120**

Bibliography **122**

A Variational Bayes results and additional updates **135**
A.1 Definitions of probability distributions . . . 135
A.2 Derivation of variational Bayes mean field approximation . . . 136
A.3 Tensor decomposition with spike and slab prior from Mitchell and Beauchamp (1988) . . . 137
A.4 Updates for linked tensor decomposition . . . 140
A.4.1 Full model . . . 140
A.4.2 VB approximation . . . 140
A.4.3 Update for A . . . 140

B Additional simulation results **142**
B.1 Comparison of spike and slab distributions . . . 142
B.2 Extended *trans* effect simulation results . . . 145
B.3 Comparison of priors for the individual scores matrix . . . 147

List of Figures

2.1 Diagram of group factor analysis with component-level sparsity. (Noise terms are not shown.) 11
2.2 Two representations of the PARAFAC decomposition. 15
2.3 Diagram of the Tucker decomposition. (Noise not shown.) . . . 17

3.1 Illustration of the tensor decomposition. (Noise not shown.) . . 34
3.2 Illustration of two missing data scenarios in a multi-tissue gene expression data set. 46
3.3 Illustration of linked tensor decomposition. Data type 2 is a matrix (i.e. $T_2 = 1$), so the context scores are a fixed vector of 1's (represented by hatching). (Noise not shown.) 51

4.1 Example of a data set simulated under the PARAFAC model with $p = 0.1$ (noise not shown). 61
4.2 Root mean squared error for individual scores matrices. Sparsity increases from left to right. Boxplots summarise results for 50 data sets. 67
4.3 Sparse stability index (SSI) for individual scores matrices. Sparsity increases from left to right. Boxplots summarise results for 50 data sets. 67
4.4 ROC curves for recovery of non-zero elements in the loadings matrices. Sparsity increases from left to right. Power and false positive rates for SPIDER were evaluated by thresholding the PIPs at 0.5 (red). A sequence of power and false positive rates for BMTF were obtained by selecting a variety of thresholds for the gene loadings estimates (blue curve). Results for 50 data sets are shown. 67
4.5 Pattern of component activity across data types. 71
4.6 Root mean squared error (RMSE) for recovered individual scores matrices. Boxplots summarise results from across 50 data sets. Noise levels in the simulated data sets increase from left to right. 73
4.7 Sparse stability index (SS1) for recovered individual scores matrices. Boxplots summarise results from across 50 data sets. Noise levels in the simulated data sets increase from left to right. 73

4.8 ROC curves for recovery of non-zero elements in the loadings matrices. Point estimates for SPIDER obtained by thresholding the PIPs at 0.5. No thresholding was required for BGFA as the raw estimates have exact sparsity. Results for CCAGFA were generated using a range of thresholds on the same estimate set, resulting in an ROC curve. The ROC curve for iClusterPlus was generated by running the method multiple times with different tuning parameters. Noise levels in the simulated data sets increase from left to right. The top and bottom rows of plots only differ in their x-axis range. 74
4.9 Illustration of the tensor decomposition, with individual scores vectors being used in a genome-wide scan to identify genetic variants that drive gene networks. 85
4.10 Average correlation between estimated individual scores vectors describing confounding factors and the true confounding. Boxplots summarise results across 50 simulated data sets for each method. 87

5.1 Pattern of missing samples in the data. 93
5.2 Distribution of cluster sizes from clustering components across 10 runs. 99
5.3 Tissue scores matrix for 236 robustly identified components. (A) Each column shows the tissue scores for a component, scaled so that the largest score equals 1. Columns have been arranged to group components with similar tissue patterns. (B) Binary representation of scaled tissue scores (obtained by thresholding scores at 0.5) to highlight the tissue specificity of the components. 100
5.4 Summary of associations between 236 robust components and 11 measured phenotypes. p-values less than 1×10^{-6} are shown in blue and those less than 1×10^{-10} shown in red. Only components with a significant association ($< 1 \times 10^{-6}$) have been plotted (32 in total), and components with similar patterns of association have been placed nearby. 101
5.5 The barplot shows the number of robust components that show a significant association with four batch variables that measure properties of RNA sequencing across the three tissues (A: adipose, L: LCLs, S: skin). The plot shows the numbers of associations between 1×10^{-6} and 1×10^{-10} in blue and less than 1×10^{-10} in red. 102
5.6 Gene ontology p-values for 236 robustly clustered components. x-axis shows the component size (number PIPs > 0.5) and the y-axis shows $-\log_{10}(p)$ for the most strongly associated GO term. The red line indicates a 1% significance threshold. 103

5.7 (A) and (B) show two components recovering the MHC class II regulation pathway. For each component: (Top left) Manhattan plot from a GWAS using individual scores vector as a phenotype. (Top right) Boxplots of individual scores separated into groups according to alleles of the lead GWAS SNP. (Bottom left) Gene loadings vector. (Bottom right) Barplot of component tissue scores. 105
5.8 p-values for marginal associations between genes with non-zero loadings in MHC class II components and lead SNPs associated with the individual scores of these components. Colours represent the three different tissues. Horizontal dashed line indicates a strict Bonferroni threshold for a full *trans* analysis of 9.05×10^{-13}. 107
5.9 MHC class I component. See figure 5.7 for a description of the figure. 109
5.10 Marginal associations between rs289749 and genes with non-zero gene loadings in the MHC class I regulation component. See figure 5.8 for a more detailed description of the figure. . . . 109
5.11 Histone RNA processing component. See figure 5.7 for a description of the figure. 110
5.12 Marginal associations between rs6882516 and genes with non-zero gene loadings in the histone RNA processing component. See figure 5.8 for a more detailed description of the figure. . . . 111
5.13 Type I interferon component. See figure 5.7 for a description of the figure. 113
5.14 Marginal associations between rs6882516 and genes with non-zero gene loadings in the histone RNA processing component. See figure 5.8 for a more detailed description of the figure. . . . 113
5.15 ZNF gene network component. See figure 5.7 for a description of the figure. 115
5.16 Marginal associations between rs12630796 (chromosome 3), rs17611866 (chromosome 16) and genes with non-zero gene loadings in the ZNF gene network component. See figure 5.8 for a more detailed description of the figure. 115

6.1 Illustration of an extension to additionally model *cis* effects. Gene expression and SNP data is jointly decomposed. Tens of thousands of components are fit, with a fixed pattern of zeros for some components (shown in grey) to make the method computationally efficient. 121
6.2 Illustration of a decomposition of gene expression data and gene annotations. A sparse gene loadings matrix is common to both decompositions. 123
6.3 Illustration of a decomposition of a 4D array of gene expression data in multiple tissues at multiple time points. 123

B.1 Histogram of PIPs for different spike and slab priors. 144
B.2 Scatter plots of estimates and true loadings for different spike and slab priors. 145

B.3 (A) Correlations between estimated individual scores vectors (permuted) and the truth, coloured by component type. (B) Posterior means of mixing parameters for T_K, again coloured by component type. 149

Chapter 1

Introduction

With recent advances in experimental techniques, multidimensional genomic data sets are becoming more common. These studies often consist of several different types of data collected for the same set of individuals or samples. Each data type provides additional information about the underlying biological processes and pathways (Hamid et al., 2009). For example, Bell et al. (2011) integrate genetic data, gene expression and methylation levels to identify genetic effects on methylation and gene regulation. High-dimensional data also arises when sets of variables are measured in several tissues, over time or in multiple experimental conditions. Gene expression levels in multiple tissues can be used to understand tissue heterogeneity (e.g. Nica et al. (2011)) and data collected at multiple time points can shed light on ageing or response to an exposure (e.g. Richmond et al. (2014)).

Extracting useful conclusions from these multidimensional data sets is a statistical challenge (Joyce and Palsson, 2006). It is not only the vast size of the data that makes it hard to analyse; biological data can have high noise levels, and contain confounding factors and missing data. Integrating data types becomes harder when data has different formats; genotype data is often categorical (given by allele count) whereas gene expression levels are continuous and many phenotypes are binary (e.g. disease outcome). There are also structural differences between data types, for example; genetic variants are

spatially ordered along chromosomes, time series data also has a natural order and sets of phenotypes may be correlated (Kim, 2015). Furthermore, studies often collect data from related individuals or consist of individuals from subpopulations (e.g. The International HapMap 3 Consortium (2010)).

Often, methods for analysing multidimensional genetic data sets involve several steps; each data type is processed separately then the output is combined in a further analysis along with the remaining data (Reif et al., 2004). Although these approaches can be successful, there are arguments for analysing all the data simultaneously. Each data type or context provides a different view of the underlying biological processes and jointly analysing this data may increase power to uncover interesting signals (Hamid et al., 2009).

Latent variable models (LVMs) are a flexible way of analysing multidimensional data sets. LVMs define a (linear) relationship between a set of latent (unobserved) variables and the set of measured variables (Loehlin, 1998). Examples include probabilistic principal component analysis (Tipping and Bishop, 1999) and factor analysis (Spearman, 1904, 1927) (which assume a Gaussian distribution for the latent variables) and independent component analysis (which assumes a non-Gaussian distribution) (Comon, 1994). LVMs are typically used to reduce the dimensions of a large data set, with the number of latent variables being smaller than the number of measured variables. These methods are also used for clustering, visualisation and uncovering dependencies in the data (Loehlin, 1998).

Many extensions of LVMs to extract structure that is common to multiple data types exist (e.g. Bach and Jordan (2005), Groves et al. (2011), and Virtanen et al. (2012)). By jointly modelling the whole data set, these methods can identify dependencies between variables from different data types. Several approaches enforce sparsity in the latent variables (e.g. Klami et al. (2014), Mo et al. (2013), and Zhao et al. (2014a)); this is necessary if there are fewer observations than variables and also aids interpretation. Furthermore, in many

applications including genomics, the underlying signals in a data set may be naturally sparse (Knowles and Ghahramani, 2011).

Tensor decompositions extract structure from N-dimensional arrays of data (Kolda and Bader, 2009). These approaches are a generalisation of the singular value decomposition to higher-order data. Use of tensor decompositions in genomics are limited (e.g. Li et al. (2013) and Omberg et al. (2007)), but they provide a potentially powerful tool for extracting dependencies in 3D data (Ng et al., 2012).

In this thesis I describe a novel method for extracting signals from multidimensional gene expression data sets. Expression levels are a direct result of transcription and known to be partially heritable (Grundberg et al., 2011). Uncovering the role of genetics in gene regulation is an important step towards understanding the relationship between genotypes and higher level traits.

The approach I take here builds on existing LVMs for analysing multidimensional data. I attempt to address the statistical challenges arising from genomic studies, and extract gene regulation networks driven by genetic variants. With new large scale data sets being collected (e.g. The GTEx Consortium (2015)), these types of approaches will become more necessary in the future. Specifically, the method developed here is a sparse Bayesian PARAFAC decomposition which uses a spike and slab prior to encourage element-wise sparsity in the decomposition. The framework is easily extended to jointly decompose several matrices and tensors, model relatedness between individuals and deal with missing samples.

The second chapter of this thesis consists of a literature review. I start by summarising some of the extensions of LVMs for analysing multidimensional data and also review tensor decompositions. The latter half of the chapter describes the relevant biological background for the thesis. I give a summary of the current methods for identifying expression quantitative trait loci (eQTL) and the statistical challenges involved.

Chapter 3 describes the method I have developed for uncovering structure in multidimensional gene expression data. The chapter starts with a detailed description of the model and inference procedure. Extensions to allow for multiple data types, missing data and related individuals are also given.

Chapter 4 details a range of simulations to compare this new approach to existing methods. I use several different metrics to assess recovery of the latent variables. This chapter also contains a simulation study with data simulated to be as similar as possible to real gene expression data, containing a variety of signals and noise.

In chapter 5 I present results of applying the method to a human study of gene expression data from the TwinsUK cohort (Brown et al., 2014; Buil et al., 2015). The data consists of expression levels in three tissues for a set of female twins. The method recovers several known gene networks.

Finally, a discussion of this work and possible extensions of the method are given in chapter 6.

Chapter 2

Literature review

This chapter starts with a review of latent variable models for uncovering dependencies in multidimensional data sets. The second half of the chapter introduces the biological background for the thesis. More specifically, I describe some of the research being performed to understand the genetic basis of gene regulation.

2.1 Latent variable models for multidimensional data

Latent variable models (LVMs) are a widely used type of unsupervised analysis. Although there are many different types of LVMs, the general idea is to identify a set of latent (hidden) variables which explain structure in the data. These latent variables are usually related to the observed (measured) variables by way of a linear model with optional additive noise.

LVMs can be motivated as a way to reduce the dimension of a data set. Often genomic data sets consist of thousands of variables which may contain a lot of redundancy. LVMs find new representations of the data in terms of a relatively small number of latent variables, while still retaining the majority of the variance in the data. Latent variables are commonly used to understand

structure in a data set, uncover the underlying generative model and visualise data. They can also be used in downstream analyses (Loehlin, 1998).

LVMs have a wide range of applications in genetics. Principal component analysis (PCA) is commonly used to uncover population structure from genotype data (Reich et al., 2008)[1]; factor analysis and independent component analysis (ICA) are used to extract variance due to confounding in omics data (e.g. Stegle et al. (2010) and Teschendorff et al. (2011)). Another application of LVMs is to uncover biological signals, for example, Kong et al. (2009) apply ICA to gene expression levels in samples from a case-control study of Alzheimer's disease. Components with differing importance between the sample groups (case and control) are recovered, and genes with more extreme values within these components implicated in Alzheimer's disease. West (2003) introduces a sparse factor analysis model to identify latent variables distinguishing breast cancer tumour types.

Traditionally, LVMs extract structure from a single matrix of data. However, it is becoming increasingly common (both in genetics and in other fields) for data sets to consist of multiple data types related via a common dimension. In genetics, this common dimension is the set of individuals or samples. A simple approach to analysing these multidimensional data sets is to concatenate data matrices along their common dimension and apply standard LVMs. This solution treats all variables equally which may not always be ideal, for example, different data types may have different signal-to-noise ratios (Groves et al., 2011). Comparisons of concatenation approaches and methods that explicitly allow for multiple data types suggest that the latter group of models are preferable (e.g. Groves et al. (2011), Lian et al. (2015), and Virtanen et al. (2012)).

I now describe some extensions of LVMs which explicitly model multidimensional data. I start with canonical correlation analysis - an approach for

[1]Technically, PCA is not a LVM, but it can be formulated as a probabilistic model.

uncovering dependencies between two data types - and its probabilistic formulation. It should be noted that this is by no means an exhaustive list.

2.1.1 Canonical correlation analysis

Canonical correlation analysis (CCA) is a method for simultaneously analysing two data types (Hardoon et al., 2004). Introduced in 1936 by Hotelling, CCA quantifies the linear relationship between two sets of variables (Hotelling, 1936). Let $Y^{(1)} \in \mathbb{R}^{N \times L_1}$ and $Y^{(2)} \in \mathbb{R}^{N \times L_2}$ be two different types of data measured for a set of N individuals where L_1 and L_2 are the number of variables for data types 1 and 2 respectively. CCA finds pairs of vectors $\mathbf{v}_1$ and $\mathbf{v}_2$, the canonical variates, which maximise the correlation between $Y^{(1)}\mathbf{v}_1$ and $Y^{(2)}\mathbf{v}_2$, the canonical variables. Up to N pairs of canonical variates can be found, with the restriction that a pair is uncorrelated with previously identified pairs.

There are several interpretations of CCA. The canonical variables are linear combinations of the observed variables so in one sense CCA is simply performing multivariate linear regression. CCA can also be interpreted as an extension of PCA, as it finds linear combinations of the two sets of variables to explain covariance in the data. Like PCA, CCA is commonly used for exploratory analysis and data reduction.

In genetics, CCA can be used to identify associations between two sets of variables. For example: Tang and Ferreira (2012) apply CCA to genotype data and a set of leukocyte levels; Naylor et al. (2010) apply CCA to genotype and gene expression data in order to identify expression quantitative trait loci. Soneson et al. (2010) work with a data set containing gene expression and copy number alterations from leukaemia patients to identify features which underlie subtypes of the disease.

Classical CCA requires data sets which contain more observations than variables (i.e. $N > \max(L_1, L_2)$), however this is rarely the case in genetics so sparse implementations of CCA have been developed. These approaches

encourage sparsity in the canonical variates (analogous to performing feature selection in a regression problem) via a penalty term. For example, Witten and Tibshirani (2009) use an L_1 norm. They apply their method to gene expression levels and copy number variation data from patients with lymphoma. The resulting canonical variables are found to be associated with tumour type. Witten and Tibshirani (2009) also present a sparse CCA extended to allow for more than two data types. This method attempts to identify structure common to all data types.

Classical CCA can be interpreted as a probabilistic model (Bach and Jordan, 2005),

$$\begin{aligned} Y^{(1)} &= AX^{(1)} + N(0, \Psi^{(1)}), \\ Y^{(2)} &= AX^{(2)} + N(0, \Psi^{(2)}) \end{aligned} \tag{2.1}$$

where $A \in \mathbb{R}^{N \times C}$, $X^{(d)} \in \mathbb{R}^{C \times L_d}$, $\Psi^{(d)} \in \mathbb{R}^{L_d \times L_d}$ (for $d \in \{1, 2\}$) and $C \leq N$. The rows of the matrices $X^{(1)}$ and $X^{(2)}$ have a similar interpretation to the canonical variates; they describe a transformation into a space where the two data matrices 'look similar'. In classical CCA, this transformation results in correlated vectors, here, shared structure is captured in the matrix A. Noise in the data is modelled using a Gaussian distribution with unrestricted covariance matrix. Structure that underlies both data types is incorporated into the terms $AX^{(1)}$ and $AX^{(2)}$ while structure specific to each data type is modelled in the noise term.

Bach and Jordan (2005) use a maximum likelihood approach to fit (2.1). Klami and Kaski (2006) fit the same model in a Bayesian framework; inference requires inversion of the estimates of the covariance matrices so the method quickly becomes intractable as the number of variables increases.

2.1.2 Inter Battery Factor Analysis

A similar probabilistic model is inter battery factor analysis (IBFA) (Browne, 1979; Tucker, 1958). IBFA extends factor analysis (which models structure within a single matrix) to two data types. Rather than modelling structure that is specific to one data type in the noise term, IBFA models it explicitly as a low rank matrix and restricts the noise term to have a diagonal covariance matrix (2.2)

$$\begin{aligned} Y^{(1)} &= AX^{(1)} + F^{(1)}Z^{(1)} + N(0, \Phi^{(1)}), \\ Y^{(2)} &= AX^{(2)} + F^{(2)}Z^{(2)} + N(0, \Phi^{(2)}). \end{aligned} \tag{2.2}$$

Using naming conventions from factor analysis and PCA, $X^{(d)} \in \mathbb{R}^{C \times L_d}$ and $Z^{(d)} \in \mathbb{R}^{K_d \times L_d}$ are known as the component (or factor) loadings. The terms component and factor are used interchangeably. Each vector of loadings, i.e. a row of a loadings matrix, defines a latent variable as a linear combination of measured variables. $A \in \mathbb{R}^{N \times C}$ and $F^{(d)} \in \mathbb{R}^{N \times K_d}$ are the component scores. Each column of $A \in \mathbb{R}^{N \times C}$ describes structure across individuals that exists in both data types whereas $F^{(d)} \in \mathbb{R}^{N \times K_d}$ describes structure that is specific to data type d. The variable C is the number of components shared across data types and K_1 and K_2 are the number of components unique to data type 1 and 2 respectively. Each component consists of a vector of scores and a vector of loadings. The components should ideally explain all the correlations between the observed variables, resulting in a noise term that has a diagonal covariance matrix.

Inference for IBFA (2.2) is much more computationally efficient than probabilistic CCA (2.1). By concatenating the two data matrices to get $\hat{Y} \in \mathbb{R}^{N \times (L_1 + L_2)}$, the IBFA model can be written as

$$\hat{Y} = \hat{A}\hat{X} + N(0, \hat{\Phi}) \tag{2.3}$$

where

$$\hat{X} = \begin{pmatrix} X^{(1)} & X^{(2)} \\ Z^{(1)} & 0 \\ 0 & Z^{(2)} \end{pmatrix}, \tag{2.4}$$

$\hat{A} \in \mathbb{R}^{N \times (C+K_1+K_2)}$ and $\hat{\Phi}$ is diagonal (Klami et al., 2013). Note that this approach treats the two data types differently so is not equivalent to running standard factor analysis on the concatenated data. Inference for (2.3) consists of estimating $\hat{X}$ given the constrained pattern of zeros. Klami et al. (2013) fit this model in a Bayesian framework and apply the method to genome-wide gene expression and copy number data to uncover genes related to cancer.

Now consider a data set consisting of an arbitrary number of $D \geq 2$ data types; data for the dth type is given by a matrix $Y^{(d)} \in \mathbb{R}^{N \times L_d}$. IBFA can be extended to deal with more data types by estimating the following pattern of zeros in the loadings matrix (Archambeau and Bach, 2009),

$$\hat{X} = \begin{pmatrix} X^{(1)} & X^{(2)} & \cdots & X^{(D)} \\ Z^{(1)} & 0 & \cdots & 0 \\ 0 & Z^{(2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Z^{(D)} \end{pmatrix}. \tag{2.5}$$

This loadings matrix allows for structure that spans all D data matrices and also structure that is specific to each data type. However, this approach is limited; (2.5) does not explicitly allow for structure that underlies an arbitrary subset of the data. A more comprehensive model would specify a pattern of zeros that considers all combinations of subsets. Although feasible for a small number of data types, the number of subsets grows very quickly as D increases.

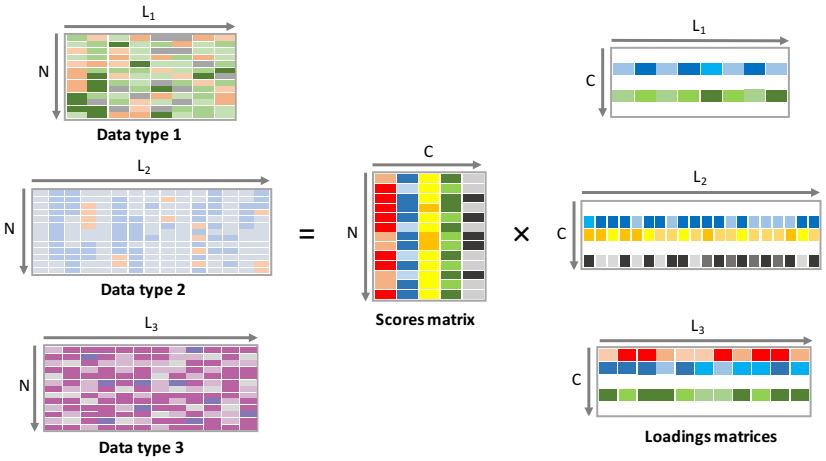

Figure 2.1: Diagram of group factor analysis with component-level sparsity. (Noise terms are not shown.)

2.1.3 Group factor analysis

IBFA uses structural constraints to model variance in a single data type and across multiple data types. An alternative approach is to use regularisation or prior constraints. Consider the group factor analysis model for several data matrices given by,

$$\begin{aligned} Y^{(1)} &= AX^{(1)} + N(0, \phi^{(1)}), \\ Y^{(2)} &= AX^{(2)} + N(0, \phi^{(2)}), \\ &\vdots \\ Y^{(D)} &= AX^{(D)} + N(0, \phi^{(D)}) \end{aligned} \tag{2.6}$$

where A is a scores matrix, $\{X^{(d)}\}_{\forall d}$ are loadings matrices and $\{\phi^{(d)}\}_{\forall d}$ are diagonal covariance matrices[2] (Virtanen et al., 2012). In order to allow for components that are specific to just a subset of the data types, a prior or penalty term is placed on the rows of the loadings matrices which allows for an entire row to shrink to zero (see figure 2.1). In this way the model automatically determines the set of data types in which each component is active.

[2]Note that with non-Gaussianity assumptions on the loadings matrices, this model could be classed as group ICA (Calhoun, 2010).

2.1.4 Sparsity

In addition to accommodating signals that exist in only a subset of the measured data types (i.e. component level sparsity), there are several other arguments for using sparsity in LVMs. Additional element-wise sparsity in the gene loadings matrices performs feature selection, aiding interpretation of the components. Furthermore, sparsity makes sense in the context of biology, where underlying signals in the data are likely to introduce variance into only a small number of the measured variables. Sparsity can also make the decomposition identifiable (see section 2.1.6 for more details). Sparse factor analysis approaches with applications in genetics are fairly common, for example, Carvalho et al. (2009), Engelhardt and Stephens (2010), and West (2003), and extensions of these methods to a sparse group factor analysis include Virtanen et al. (2012) and Zhao et al. (2014a).

Virtanen et al. (2012) fit the model in (2.6) in a Bayesian framework using an automatic relevance determination (ARD) prior (Tipping, 2001). This hierarchical prior takes the form of a Gaussian distribution with a precision parameter that is modelled using a Gamma distribution. As the precision term for a component tends to infinity, all elements in the loadings vector shrink to zero, essentially removing the component. The ARD prior can also generate element-wise sparsity. Integrating out the precision variables results in a marginal distribution equal to a Student-t distribution which has a high density around zero. Another advantage of sparsity is that it provides a way of automatically selecting the model order, i.e. the number of components. If a model is initialised with too many components, then the sparsity prior can remove superfluous components from the model by shrinking them to zero for all data types.

An alternative prior known as a three-parameter Beta is used by Zhao et al. (2014a) (Armagan et al., 2011; Gao et al., 2013). This distribution extends the Beta distribution by adding another parameter to model a wider range of

densities. Zhao et al. (2014a)'s formulation explicitly shrinks globally to remove unwanted components, at a component level to allow for structure that spans only a subset of the input data matrices, and at an element level, resulting in sparse signals. In their paper, Zhao et al. (2014a) apply this approach to genotype data on chromosome 22 and genome-wide gene expression data, initialising the method with 2,000 components. Not surprisingly, given how different the input data is, the majority of estimated components are specific to a single data type. Several dense components specific to the genotype data identify population structure; components spanning both data types tend to be sparse, identifying a small set of SNPs and genes. SNPs and genes identified in the same component tend to be more highly associated when performing univariate tests than expected randomly.

Other sparsity priors include the so-called 'spike and slab' distribution (Goodfellow et al., 2013; Mitchell and Beauchamp, 1988). This distribution, unlike the ARD prior, is discrete. The spike and slab mixture prior consists of a point mass at zero (the 'spike') and a Gaussian distribution (the 'slab'). The spike and slab distribution has a non-zero probability of shrinking an element to exactly zero. This prior was first used in factor analysis by West (2003) for extracting signals from gene expression data consisting of 49 tumour samples and 6,128 genes. Ray et al. (2014) use a spike and slab distribution on both the scores and loadings matrices in a formulation of IBFA for more than 2 data types.

Sparsity can also be applied in a frequentist framework, for example, Mo et al. (2013) using an L_1 penalty (Lasso) on the loadings vectors to get a sparse solution (Tibshirani, 1996).

I now describe some decompositions for three-dimensional data and their relationship to group factor analysis.

2.1.5 Tensor factorisations

Define an Mth-order tensor to be an M-dimensional array (Kolda and Bader, 2009). A first-order tensor is a vector, a 2nd-order tensor is a matrix, and a 3rd-order tensor is a 3-dimensional array etc. An Mth-order tensor can be written as $\mathcal{P} \in \mathcal{R}^{I_1 \times I_2 \times \cdots \times I_M}$ and an element of $\mathcal{P}$ denoted by $\mathcal{P}_{i_1 i_2 \cdots i_M}$.

As with matrices, there is interest in finding alternative representations of tensors in terms of fewer variables. There are several ways to perform tensor decompositions; two common methods are the PARAFAC and Tucker decompositions. Both decompositions are generalisations of the singular value decomposition to higher-order data.

In the remainder of this section, I focus on 3rd-order tensors, although the decompositions described here do extend to higher-order tensors. I also use the term tensor to refer specifically to a 3rd-order tensor unless otherwise specified.

2.1.5.1 PARAFAC decomposition

The PARAFAC (parallel factors) decomposition (also known as the canonical decomposition (CANDECOMP) or CP decomposition), decomposes a tensor into the sum of a finite number of rank-one tensors (Carroll and Chang, 1970; Harshman and Lundy, 1994). A rank-one tensor is simply a tensor that can be written as the outer product ($\circ$) of three vectors. For example, the tensor, $\mathcal{P} \in \mathbb{R}^{I \times J \times K}$ is rank-one if there exist vectors $\mathbf{u} \in \mathbb{R}^I, \mathbf{v} \in \mathbb{R}^J, \mathbf{w} \in \mathbb{R}^K$ such that $\mathcal{P}_{ijk} = u_i v_j w_k$, or $\mathcal{P} = \mathbf{u} \circ \mathbf{v} \circ \mathbf{w}$. The PARAFAC decomposition of $\mathcal{Y} \in \mathbf{R}^{N \times L \times T}$ is given by

$$\mathcal{Y} = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{x}_r \circ \mathbf{b}_r + \mathcal{E} \tag{2.7}$$

where $\mathbf{a}_r \in \mathbb{R}^N$, $\mathbf{x}_r \in \mathbb{R}^L$ and $\mathbf{b}_r \in \mathbb{R}^T$ for $r \in \{1, \cdots, R\}$, and $\mathcal{E} \in \mathbb{R}^{N \times L \times T}$ is a tensor of noise. Figure 2.2(A) illustrates this decomposition. Equation (2.7)

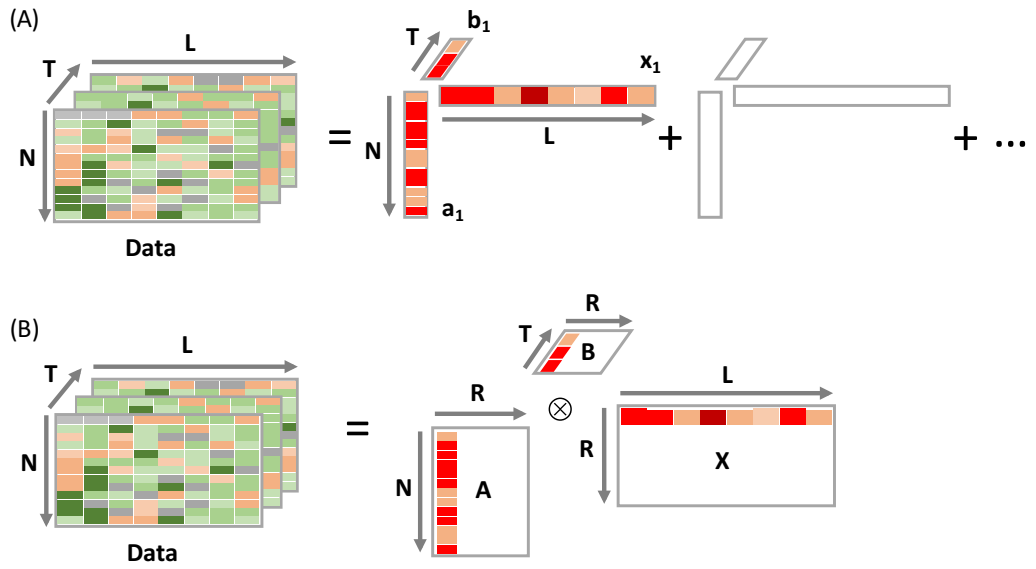

Figure 2.2: Two representations of the PARAFAC decomposition.

can be fit using alternating least squares to minimise the distance between the data and rank-R approximation (Carroll and Chang, 1970; Harshman and Lundy, 1994). This algorithm iteratively updates one matrix at a time, conditional on the current estimates of the other two matrices, resulting in a unique solution under certain conditions.

It is worth pointing out that a probabilistic implementation of the PARAFAC decomposition is a special case of group factor analysis (2.6). Let $A = [\mathbf{a}_1, \dots \mathbf{a}_R]^\top \in \mathbb{R}^{N \times R}$, $B = [\mathbf{b}_1, \dots \mathbf{b}_R]^\top \in \mathbb{R}^{T \times R}$ and $X = [\mathbf{x}_1, \dots \mathbf{x}_R] \in \mathbb{R}^{R \times L}$. Figure 2.2(B) depicts an alternative representation of the PARAFAC model in terms of matrices A, B and X, highlighting the difference between this decomposition and group factor analysis. The PARAFAC decomposition learns a single loadings matrix (X), with contributions from each component, in each slice of the data, differing by up to a scalar multiple (encoded by B). The matrix A has a similar interpretation to the scores matrix from group factor analysis. Group factor analysis can be applied to $\mathcal{Y}^{N \times L \times T}$ by treating the data as T matrices each of dimensions N by L. (Obviously, the split could be made along other dimensions as well.) Group factor analysis learns a separate

loadings matrix for each of the T matrices, making less assumptions than the PARAFAC decomposition, but introducing considerably more parameters.

Sparsity has been employed in implementations of (2.7) by encouraging sparsity in all or one of A, B and X. For example, Allen (2012) uses an L_1 penalty on the underlying components; this approach is applied to gene expression data (at 8932 genes) in 16 tissues for 22 mice. Components uncover differences due to gender and age, and also cluster similar tissues. Padilla and Scott (2015) develop a similar model with arbitrary penalties on the underlying components. Bayesian implementations include that of Zhao et al. (2014a) who place ARD priors on A, B and X. The PARAFAC model is also often used to impute missing data in incomplete tensors (e.g. Zhao et al. (2014a)) and perform non-negative tensor decompositions (e.g. Welling and Weber (2001)). An extension of ICA for 3D data, similar to the PARAFAC model, is given by Beckmann and Smith (2005) where X is assumed to be non-Gaussian.

The PARAFAC framework can be extended to jointly decompose an arbitrary number of tensors with at least one shared dimension[3] (e.g. Groves et al. (2011) and Khan et al. (2014a)). The idea is similar to the way group factor analysis incorporates multiple data types; each tensor is decomposed via (2.7), with a common matrix linking the decompositions across their shared dimension. Groves et al. (2011) apply this model to brain image data (individuals by voxels by modality) for several modalities groups, using a mixture of Gaussians on the loadings matrices. They show that for this application, the linked tensor decomposition outperforms approaches similar to group factor analysis.

[3]Matrices can also be incorporated by treating them as a 3rd-order tensor with the 3rd dimension equal to 1.

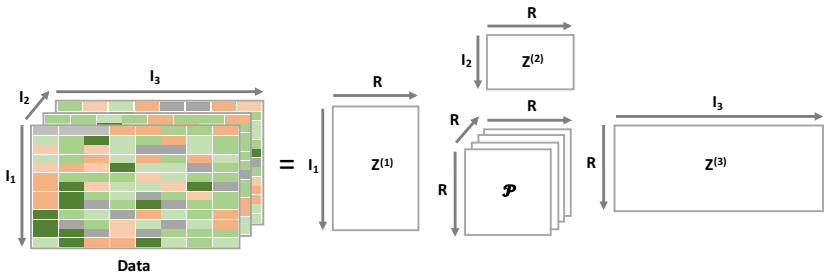

Figure 2.3: Diagram of the Tucker decomposition. (Noise not shown.)

2.1.5.2 Tucker decomposition

The Tucker decomposition is an more general alternative to the PARAFAC model. The Tucker decomposition for tensor $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is

$$\mathcal{Y} = \sum_{i=1}^{I_1} \sum_{j=1}^{I_2} \sum_{k=1}^{I_3} p_{ijk}\, \mathbf{z}_i^{(1)} \circ \mathbf{z}_j^{(2)} \circ \mathbf{z}_k^{(3)} + \mathcal{E} \tag{2.8}$$

where $[\mathbf{z}_1^{(s)}, \cdots, \mathbf{z}_{I_s}^{(s)}] \in \mathbb{R}^{I_s \times R}$ for $s \in \{1, 2, 3\}$, $\mathcal{P} \in \mathbb{R}^{R \times R \times R}$ is a core tensor with reduced dimensions and $\mathcal{E}$ is a noise tensor (see figure 2.3). Note that the PARAFAC model is recovered if $\mathcal{P}$ is a diagonal tensor, ie. $p_{ijk} = 1$ for $i = j = k$ and $p_{ijk} = 0$ otherwise.

This decomposition is harder to interpret than the approach taken by PARAFAC. When the matrices $Z^{(1)}$, $Z^{(2)}$ and $Z^{(3)}$ are forced to be orthonormal, the decomposition is known as the higher order singular value decomposition (HOSVD). In this case, the decomposition is essentially finding principal components along each dimension of the tensor (Allen, 2012). It follows that a sparse Tucker decomposition can be obtained if PCA is replaced with sparse PCA (Allen, 2012).

Omberg et al. (2007) use the HOSVD to recover similarities in microarray experiments. In this work, I_1 denotes the number of genes and I_2 and I_3 represent different settings, for example, number of tissues and experimental conditions respectively. The core tensor is interpreted as containing variance patterns that are similar across all experiments.

There are a large number of variations on the Tucker and PARAFAC decompositions, see Kolda and Bader (2009) and links therein for more details. It is worth noting that the Tucker decomposition, unlike a singular value decomposition on a matrix, is not unique. A discussion of the uniqueness of the other approaches discussed in this section is given below.

2.1.6 Identifiability

The question of identifiability becomes important when investigating the output of an LVM. Uniqueness of the solution is crucial if components are to be interpreted as underlying biological signals in the data. Consider a standard factor analysis model for a matrix Y,

$$Y = AX + E, \tag{2.9}$$

where $A \in \mathbf{R}^{N \times C}$ and $X \in \mathbb{R}^{C \times L}$. Latent variables (rows of X) are assumed to be Gaussian, ($X_{c\cdot} \sim \mathcal{N}(0,1)$[4]), as is the noise, ($E_{\cdot l} \sim \mathcal{N}(0, \Lambda)$), with a diagonal covariance Λ. The covariance of Y is given by

$$AA^\top + \Lambda. \tag{2.10}$$

It is clear that the columns and rows of A and X respectively can be permuted and scaled without altering the model fit. This means that the components have no intrinsic scale or ordering. More importantly (2.9) is rotationally invariant. To see this, let F be a $C \times C$ (orthogonal) rotation matrix and let $\hat{A} = AF$ and $\hat{X} = F^\top X$. The matrices $\hat{A}$ and $\hat{X}$ satisfy (2.9) and $\mathrm{Cov}(\hat{A}\hat{X} + E) = (AF)(AF)^\top + \Lambda$ which reduces to (2.10). It follows that $\{\hat{A}, \hat{X}\}$ is also a solution of (2.9). Post-processing steps are often performed to get around this problem of non-uniqueness in factor analysis. For example, a varimax rotation to a sparse solution can be employed to improve interpretability (Kaiser, 1958).

[4] $X_{c\cdot}$ denotes the cth row of X, and $E_{\cdot l}$ denotes the lth column of E.

ICA, unlike factor analysis, is identifiable, at least up to a scaling and permutation (Comon, 1994). ICA assumes that the underlying components have a non-Gaussian distribution. Importantly, non-Gaussian distributions have non-zero nth moments for $n > 2$, meaning that a set of non-Gaussian components are not rotationally invariant. It follows that adding sparsity to the loadings (or scores) of (2.9) makes the decomposition identifiable, as priors encouraging sparsity result in super Gaussian components. This is yet another advantage of using sparsity in factor analysis and group factor analysis models.

Consider the identifiability problem for the PARAFAC decomposition given in (2.7). Kruskal (1977) showed that (under fairly mild conditions), this decomposition is unique (aside from a permutation and scaling). One way to think about the PARAFAC model is that each slice of the tensor is generated via a set of components with the scale of the contributions from each component being different for each slice. Therefore, the data essentially provides multiple pictures of the underlying components, allowing the correct rotation to be recovered uniquely.

2.2 Gene expression

2.2.1 Motivation

The central dogma of biology states that DNA is transcribed to create mRNA which is then manipulated via several different processes to form proteins, the cell's workhorses (Alberts et al., 2002). Save for de novo mutations, genetic material is identical in almost every cell of the body, with current estimates suggesting that there are around 19,000 genes in the human genome (Ezkurdia et al., 2014). It is the unique patterns of expression of these genes that results in different types of cells with different behaviours. The functions of many genes have been evaluated or partially evaluated using knock-out studies, although this information is still incomplete (Mi et al., 2013).

DNA transcription is an involved process requiring complex machinery. In summary, an enzyme called RNA polymerase is recruited to a region upstream of the gene called the promoter and the polymerase then moves along the DNA copying the sequence (Alberts et al., 2002). Regulation of this process is performed in part by proteins called transcription factors (TFs) or TF complexes whose role is to help or hinder the binding of RNA polymerase. A single gene may be transcribed in several different ways, depending on where transcription starts and ends, resulting in different mRNA molecules or transcripts.

Genome-wide association studies (GWAS) perform mass univariate regression to test for associations between genetic variants which are often single nucleotide polymorphisms (SNPs) and high level traits. This approach has successfully identified thousands of genetic variants associated with traits and disease status (Welter et al., 2014). However, these studies often fail to explain the mechanism by which a genetic variant affects a phenotype or alters disease susceptibility (Edwards et al., 2013). Many GWAS hits do not lie in coding regions, or are synonymous, meaning that they do not directly alter a transcript sequence or protein function. Maurano et al. (2012) show that SNPs associated with high level traits and disease status often lie in DNase I hypersensitivity sites, DNA regions to which transcriptional machinery binds. This suggests that genetic variants may play a regulatory role in gene expression.

Gene expression data is an intermediary phenotype, linking genetic material with higher level phenotypes. Expression levels are known to change under different conditions and with age, and to have unique patterns in different tissues (Glass et al., 2013; Grundberg et al., 2011). For example, SNPs implicated in type II diabetes are associated with gene expression levels in adipose and muscle tissues (Below et al., 2011). Furthermore, expression levels are known to change with various environmental exposures (Landi et al., 2008). Gene expression data provides additional information about the context-specific nature of SNP effects, and can help to uncover the mechanisms by which genetics

variants affect high level traits and disease susceptibility. More generally, understanding regulation of gene expression will further the understanding of cellular level processes.

DNA microarrays can be used to measure levels of transcription. First, mRNA molecules are converted into a library of complementary DNA (cDNA), essentially reversing transcription, and tagged with a fluorescent molecule. Then the library is washed across an array of probes where matching sequences hybridise. Levels of mRNA are then calculated by analysing pictures of the fluorescence levels. This technique is limited by the size of the probe set and also suffers from problems of cross hybridisation when probe sequences contain genetic variants (Murphy, 2002).

A recently introduced method, RNA-sequencing, provides a more comprehensive set of gene expression levels. This technique also converts mRNA into a cDNA library which is then passed through a sequencing machine. The result is a set of short sequences called reads which can be mapped back onto a reference genome. Read frequency is used to measure expression levels, either at a gene or transcript level (Ozsolak and Milos, 2011). This method also involves challenges; however RNA-sequencing is believed to be a less noisy, more reliable experimental technique than microarrays (Wang et al., 2009).

In the rest of this section, I review some of the statistical methods being used to understand the genetic basis of gene expression data.

2.2.2 eQTL mapping

Data sets consisting of genetic material and gene, or transcript, expression levels make it possible to study the genetic basis of gene regulation. An expression quantitative trait locus (eQTL) is a sequence variant that is associated with levels of expression of a gene or transcript (Albert and Kruglyak, 2015). Although an eQTL can refer to any sort of sequence variant, it often refers

to a SNP. Searching the genome for eQTLs is known as eQTL mapping and is often performed in a similar way to GWAS, with the expression of a gene being the quantitative phenotype of interest.

2.2.2.1 Types of eQTL

eQTLs can be partitioned into types according to their position in relation to the regulated gene (*local* or *distant* eQTLs), and the functional mechanism that gives rise to the change in expression of the gene (*cis* or *trans* eQTLs) (Sun and Hu, 2013).

The terms *local* and *distant* refer to distance, either physical or genetic, between the eQTL and the regulated gene. Typically, a *local* eQTL is defined as lying within 2MB of the transcription start site of the gene, but this can vary from study to study; anything further away than 2MB is known as a *distant* eQTL (Albert and Kruglyak, 2015).

Cis and *trans* eQTLs distinguish between two different mechanisms by which genetic variants affect gene expression. A *cis* eQTL results in allele-specific expression levels, i.e. the allele on a chromosome determines the expression of its gene copy whereas the allele of the homologous chromosome determines the expression of the opposite gene copy. Often the genetic variant in question lies in the promoter region of a gene and either increases or decreases the binding affinity of a TF, resulting in more or less of the homologous gene being expressed. Heterogeneous individuals and transcript level data are required to identify *cis* eQTLs.

Trans eQTLs do not give rise to allele specific expression. They occur when a SNP (the eQTL) alters a factor in the regulatory mechanism of a gene. Pierce et al. (2014) suggest that *trans* eQTLs are often explained by *cis* mediation. For example, suppose a SNP lies in the promoter region of a TF and is a *cis* eQTL for the TF. A higher concentration of the TF results in higher expression levels of the genes it regulates, so an association between

the SNP and regulated genes should be identifiable.

Another possibility is that a non-synonymous SNP in the TF results in a functional change making it less effective. No *cis* effect would be visible in this case, but the SNP would be correlated with expression levels of downstream genes (Bryois et al., 2014).

Local eQTLs are almost always *cis* eQTLs and *distant* eQTLs are almost always *trans* eQTLs, but there are exceptions (see Ronald et al. (2005) for an example in yeast). Despite this distinction, I will follow the naming conventions used by the majority of the community and use the terms *cis* and *trans* to refer to *local* and *distant* eQTLs respectively.

2.2.2.2 Finding *cis* eQTLs

Identification of *cis* eQTLs often proceeds as follows; for each gene, the set of all SNPs within a specified window of the transcription start site are regressed on the gene expression data. Since the set of SNPs in the window are likely to be in linkage disequilibrium, permutation testing is often used to evaluate a significance threshold. This process has largely been very successful, significant associations between local SNP gene pairs (suggesting a *cis* eQTL) appear to be relatively easy to find, even with small sample sizes.

Results suggest that *cis* acting regulation of gene expression is abundant. In a study of 922 RNA-sequenced samples from whole blood, Battle et al. (2014) found over 75% of the genes tested had a *cis* eQTL. As sample sizes increase, it is expected that *cis* eQTLs will be identified for almost all genes in the genome (Pai et al., 2015). Furthermore, some genes are regulated by several loci showing independent effects on expression levels (Nica et al., 2011). Investigation of the location of putative *cis* eQTLs shows an enrichment of regulatory regions including TF binding sites and DNase1 hypersensitivity sites (Lappalainen et al., 2013). This suggests a process by which a SNP disrupts the regulatory mechanism of a gene, impacting its expression levels.

In addition, SNPs robustly associated with high level traits are more likely to be *cis* eQTLs (Nicolae et al., 2010). However, caution needs to be exercised as both the identified *cis* eQTL and trait associated SNP may be tagging the true causal SNP.

Heritability of gene expression can be evaluated using variance component methods. Grundberg et al. (2011) provide narrow sense (additive heritability) estimates of 0.26, 0.21 and 0.16 for adipose, lymphoblastoid cell lines (LCLs) and skin tissues respectively. Higher estimates were found by Price et al. (2011), 42% in adipose and 63% in whole blood. The variances can be further partitioned to estimate the contribution from *cis* eQTLs in the region of 20% to 40%. These results indicate that at least 60% of the heritability of gene expression is unaccounted for, suggesting that *trans* eQTLs may play a major role in the genetic basis of gene regulation.

2.2.2.3 Finding *trans* eQTLs

Searching for *trans* eQTLs is much more challenging than a *cis* eQTL analysis. A comprehensive *trans* analysis requires mass univariate testing between all pairs of SNPs and genes. This exhaustive approach is not only computationally expensive, but lacks power due to a very large multiple testing burden. For a study of 20,000 genes, a strict Bonferroni corrected significant level of 2.5×10^{-12} is obtained by scaling a genome-wide significance threshold of 5×10^{-8} by the number of genes. This threshold is perhaps overly conservative as it assumes no correlations between genes, however it is not clear how to select the 'correct' threshold for a *trans* analysis.

Using a mass pairwise testing approach, Grundberg et al. (2012) find 639, 557 and 609 *trans* eQTLs in adipose tissue, lymphoblastoid cell lines and skin tissue respectively, in a sample size of 856 (at a FDR of 10%)[5]. Replication rates are low however, with at best 6.4% of findings replicating in another

[5]Note that FDR procedures assume independence between tests which is unlikely to be a valid assumption here due to the correlations between SNPs and gene expression levels.

study. Several recent papers have tried to reduce the multiple testing burden by sub-setting down to a smaller number of SNPs. For example, Westra and Franke (2014) only consider the set of 4,500 SNPs that had previously been found to be GWAS hits; they find and replicate 100 SNPs with associations (in *trans*) with gene expression in peripheral blood. Yao et al. (2015) restrict testing to the set of SNPs associated with cardiovascular traits resulting in a set of just 1,512 SNPs from which 44 *trans* eQTLs are found in whole blood. Bryois et al. (2014) search for *trans* eQTLs in 869 lymphoblastoid cell lines. They theorise that *cis* effects mediate *trans* effects and consider the subset of SNPs already identified as *cis* eQTLs. As expected, this results in an enrichment of associations in *trans* compared to the full *trans* analysis with all SNPs.

Trans analyses are yet to shed much light on regulation in humans. Effect sizes of *trans* eQTLs appear to be much smaller than *cis* eQTLs, which is perhaps expected given the more complex mechanisms involved (Westra and Franke, 2014). In several non-human organisms, hot-spots of regulation whereby a genetic locus regulates multiple transcripts, so-called multi-gene regulators, have been identified (Breitling et al., 2008). There is some evidence of multi-gene regulators in humans (Brynedal et al., 2014), but few have been found (e.g. Small et al. (2011)).

Other approaches to finding *trans* eQTLs are worth describing. A criticism of the standard approach for identifying *trans* eQTLs is that it ignores correlations between genes. Correlations could potentially arise if genes are regulated by the same mechanism or SNP. Jointly modelling expression levels across multiple genes may increase power to find *trans* eQTLs. Scott-Boyer et al. (2012) suggest a hierarchical Bayesian regression model for effects of all SNPs on all genes simultaneously. Sparsity priors are used to encourage many of the effect sizes of SNPs on genes to be zero. The authors show this approach is successful at identifying regulatory hot-spots in rat data, although it is unclear how the approach performs on human data.

Weiser et al. (2014) suggest an alternative method which builds on co-expression and network analysis. A network of co-expressed genes is created by adding sparsity to a correlation matrix. Then, starting with the results of a standard *cis* analysis, they further selectively test for *trans* associations between *cis* eQTLs and nodes on the graph linked to the *cis* gene. A previously known *trans* network involving the gene *IRF7* (Heinig et al., 2010) is recovered using this method, but on a data set substantially smaller than a standard eQTL study.

Rotival et al. (2011) apply ICA to gene expression data to uncover sets of co-regulated genes. ICA also returns a vector of mixing weights for each set of genes describing the relative contribution of the component for each individual. Testing these vectors for association with genetic variants identifies SNPs that may regulate gene networks. Gao et al. (2013) use a similar approach employing sparse factor analysis rather than ICA.

2.2.2.4 Context specific eQTLs

Context specific eQTLs, i.e. eQTLs that act under certain conditions or contexts are of growing interest (Stranger and Raj, 2013). Investigating the specificity of eQTLs can help to understand the mechanisms behind genetic effects on expression levels and potentially aid drug development. Many studies identifying a variety of different context specific eQTLs exist; Dimas et al. (2010) perform eQTL mapping in several cell types, Stranger et al. (2012) look for eQTLs in different populations and Dimas et al. (2012) identify sex specific eQTLs. eQTLs may also be tissue specific, or active in a subset of tissues. For example, to date, the *trans* network involving *KLF14* has only been identified in adipose tissue (Small et al., 2011). Recent large data sets including the TwinsUK cohort (Brown et al., 2014; Buil et al., 2015) and GTEx (The GTEx Consortium, 2015) have made it possible to comprehensively investigate the tissue specificity of eQTLs. GTEx, for example, consists of up to 44 tissues

per individual.

The standard method for eQTL mapping in multiple tissues (or contexts) is to take a two step approach, first performing eQTL mapping in each tissue separately and then comparing the results (Nica et al., 2011). However joint analysis of multiple tissues can increase power to find eQTLs. Flutre et al. (2013) present a multi-tissue approach that tests for associations between a SNP and expression levels of a gene in several tissues, learning an indicator variable that specifies whether an eQTL is active in a particular tissue or not. These models not only output whether an eQTL is active in a particular tissue but also allow for comparisons of effect sizes across tissues.

Studies suggest that many *cis* eQTLs are shared across tissues. An analysis of 9 tissues from the GTEx pilot project find that high numbers of *cis* eQTLs overlap between tissues pairs, with estimates as high as 90% for some pairs of tissues (The GTEx Consortium, 2015). Some tissues are less likely to have shared eQTLs (e.g. blood), but as a general rule, eQTLs appear to be either active in the majority of tissues or in just one or two very similar tissues; 50% of *cis* eQTLs found in this study are active in all 9 tissues.

2.2.3 Confounding

Both microarray and RNA-sequencing data contain confounding, i.e. modes of variation in the data which mask interesting biological signals. Confounding factors are often technical artefacts that arise during experiments. Sequencing batch, experiment date and location are an obvious source of unwanted variation. Sources of confounding specific to RNA-sequencing data include average read length and the GC content of a sample (Hansen et al., 2010; Pickrell et al., 2010).

Confounding factors can also be biological. For example, age is known to correlate with expression levels of hundreds of genes (Glass et al., 2013) and expression levels of some genes are seasonal (Goldinger et al., 2015). Another

source of confounding is population structure as studies commonly contain individuals from across several populations. A naive eQTL study ignoring these sources of variation is likely to detect fewer associations, and may even identify spurious correlations.

Confounding factors will either be measured or unmeasured. Measured factors, (such as age or GC content), can be treated as covariates in eQTL testing. Unmeasured confounding presents more of a problem. Several approaches exist that try to estimate the latent confounding factors, or at least the variance attributed to a set of confounding factors. These approaches can be motivated by looking at the success of PCA for identifying population structure (Novembre et al., 2008).

Many techniques for finding sources of confounding in gene expression data build on factor analysis and independent component analysis. The underlying idea is that these methods extract latent variables that describe global modes of variance in the data, which may describe confounding. Latent variables can then be used as covariates in an eQTL regression.

Perhaps the most widely used method for correcting for confounding is PEER (Stegle et al., 2010). PEER is a probabilistic Bayesian factor analysis model which is able to simultaneously learn unmeasured confounding whilst accounting for known confounding. Stegle et al. (2010) suggest that up to twice the number of *cis* eQTLs can be identified using this approach. PEER factors tend to be highly correlated with known variables describing confounding, although variance attributed to each of these variables is often shared across several factors. Approaches that employ independent component analysis (e.g. Teschendorff et al. (2011)) may be better able to unmix signals from different sources of variation by recovering independent components. However, if the goal is to simply to explain away variance due to confounding then this does not matter.

With all LVM approaches there is a risk that variance due to real biological

signals will be attributed to confounding and removed from the data (Fusi et al., 2012). Some biological signals, such as the effect of age on gene expression, show broad effect patterns similar to batch variables. *Trans* eQTLs may also suffer the same fate and be incorrectly identified as confounding and removed. Goldinger et al. (2013) show that using principal components to correct for confounding in a gene expression study actually reduces the number of *cis* and *trans* eQTLs recovered. Fusi et al. (2012) suggest an approach, PANAMA, that extends methods like PEER to model genes as a mixture of a number of latent variables and a weighted sum of contributions from SNPs. Using the learnt latent variables as covariates in a standard eQTL analysis identified more *cis* and *trans* eQTLs with better replication rates, suggesting that explicit modelling of genetic effects results in improved recovery of confounding factors.

Mostafavi et al. (2013) use prior knowledge to improve confounder recovery and try to reduce the chance that real biological signals are accidentally attributed to unwanted variance. Their model can accommodate factors that introduce both broad and narrow effects of expression. Using information about likely variance patterns from measured factors, they learn a set of unmeasured confounding factors.

2.3 Discussion

In this chapter I have described several LVMs for analysing multidimensional data. LVMs provide a flexible way to uncover structure across several linked matrices or within N-dimensional data arrays. Encouraging sparsity in the estimated latent variables is common (e.g. Knowles and Ghahramani (2011) and West (2003), with applications to gene expression, and Engelhardt and Stephens (2010), with applications to genotype data). Since signals in biological data are expected to only involve a small number of genes; (e.g. transcription factors typically only regulate a couple of hundred genes (Hartemink, 2005)), the sparsity assumption actually reflects beliefs about biological path-

ways.

SNPs regulating multiple genes in *trans* (multi-gene regulators) may also generate sparse signals in gene expression data. LVMs are commonly used in eQTL mapping studies to identify confounding factors. However only a handful of papers have attempted to use LVMs to uncover *trans* eQTLs (Gao et al., 2013; Rotival et al., 2011). Furthermore LVMs have not yet been used for identifying context specific eQTLs. Tensor decompositions provide a way of uncovering shared structure across several data views, making them an interesting tool for identifying *trans* eQTLs in multiple contexts, whilst accounting for confounding factors.

Furthermore, LVMs provide a flexible framework in which to incorporate data types such as chromatin state and methylation levels, other factors involved in gene regulation. The next chapter describes a sparse LVM I have developed for jointly analysing multidimensional genetic data sets.

Chapter 3

Methods

This chapter describes a method for uncovering structure in a three-dimensional gene expression data set. These data sets arise when expression levels for a set of individuals are measured across several contexts. Throughout this chapter I focus on a data set consisting of expression levels in multiple tissues, however this could equally be multiple time points, cell types or conditions; the method can also be applied to other omics data. The approach I propose is a sparse Bayesian implementation of the PARAFAC decomposition. I use variational Bayes for inference with a spike and slab prior to encourage sparsity.

Before formally specifying the method, I give a heuristic description of the ideas behind this approach (section 3.1). Section 3.2 details notation for the remainder of the chapter. Section 3.3 contains a description of the method. I then move on to describe some extensions of the method in section 3.4, including approaches to deal with missing data, related individuals and a linked decomposition for several data types. Section 3.5 details some implementation choices. I finish off the chapter by comparing the method presented here with existing methods (section 3.6) and a discussion (section 3.7).

3.1 Heuristic description of the method

Consider a data set consisting of gene expression levels in multiple tissues for a set of individuals. There will be many biological processes occurring in these tissues, each involving a subset of the genes in the genome. These processes will give rise to variance and correlations in the measured gene expression levels. (I use the term 'process' fairly loosely to describe any sort of biologically driven structure in the data). Some genes may be involved in several processes, other genes may only be involved in one, or none at all. Similarly, a process might be active in all of, or in a subset of, the individuals and tissues. There will also be non-biological modes of variation in the data due to technical artefacts. These will manifest in a similar way to biological signals but may produce broader, more systematic patterns of variation, and are likely to be specific to a single tissue.

Measured expression levels will consist of a mixture of these biological signals, non-biological signals and noise. The method described in this chapter attempts to untangle these underlying signals by decomposing the data into a set of components which describe modes of variance. A component consists of three vectors, an individual scores vector, a tissue scores vector and a gene loadings vector. The gene loadings vector is sparse and describes the contribution from each gene in the component. Similarly, the individual and tissue scores vectors describe the relative degree to which the component is 'on' in each individual and tissue. The method models the data as a linear mixture of the gene loadings vectors, with the scores vectors acting as mixing weights. Ideally, the estimated components should capture all the structure in the data, with each independent signal described by a single component.

3.2 Notation

Capital letters are used to denote matrices. Suppose Y is a matrix with dimensions $I \times J$; the (i, j)th element of Y is denoted y_{ij}. The ith row of Y is a row vector of length J denoted by $\mathbf{y}_{i\cdot}$ and the jth column of Y is a column vector of length I denoted by $\mathbf{y}_j$ or $\mathbf{y}_{\cdot j}$.

Three-dimensional arrays (also referred to as 3-way tensors, or just tensors) are represented using curly letters. For example, $\mathcal{Y} \in \mathbb{R}^{I \times J \times K}$ has dimensions I by J by K; an element of this tensor can be referenced using three indices, e.g. y_{ijk} for $i \in \{1, \ldots, I\}$, $j \in \{1, \ldots, J\}$ and $k \in \{1, \ldots, K\}$. The vector $\mathbf{y}_{lt} \in \mathbb{R}^N$ is used to reference data for gene l in tissue t.

In the following sections, I use c and k as an index over components, n as an index over individuals, l as an index over genes and t as an index over tissues. Limits of summations are ignored if the context is clear.

3.3 Tensor decomposition

3.3.1 Model description

Let $\mathcal{Y} \in \mathbb{R}^{N \times L \times T}$ be a tensor of expression data for N individuals at L genes in T tissues. For now, assume there is no missing data in $\mathcal{Y}$; extensions to allow for missing data are given in section 3.4.1. Expression levels are further assumed to be normalised such that data for each gene in each tissue (i.e. $\mathbf{y}_{lt}$) has zero mean and unit variance.

The data is modelled as a linear combination of C latent variables (called components) and independent Gaussian noise,

$$y_{nlt} = \sum_{c=1}^{C} a_{nc} b_{tc} x_{cl} + \epsilon_{nlt} \tag{3.1}$$

where $X \in \mathbb{R}^{C \times L}$ is a gene loadings matrix; each row of X defines the relative contribution of each measured gene in the component. A is an N by C matrix

of individual scores which describes the individual specific mixing weights for each component. Similarly, the tissue scores matrix $B \in \mathbb{R}^{T \times C}$ contains tissue specific mixing weights. Finally, $\mathcal{E}$ is an $N \times L \times T$ array of independent Gaussian noise,

$$\epsilon_{nlt} \sim \mathcal{N}(\epsilon_{nlt}|0, \lambda_{lt}^{-1}) \tag{3.2}$$

where λ_{lt} is the noise precision for gene l in tissue t. Figure 3.1 shows a diagram of this decomposition with 5 components. A discussion of the validity of the Gaussian noise assumption for RNA-sequencing data is given in section 3.7. Note that (3.1) models the data as a sum of rank one tensors as in the PARAFAC decomposition.

Inference for (3.1) is performed in a Bayesian framework; the next section defines priors for the model, including a sparsity-inducing prior on X.

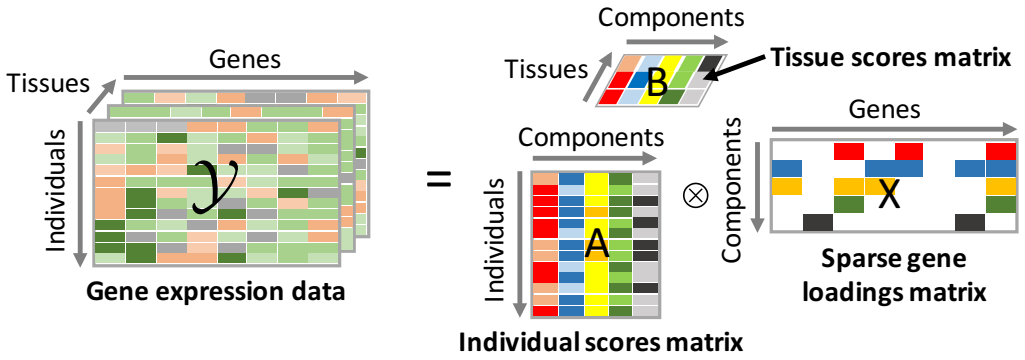

Figure 3.1: Illustration of the tensor decomposition. (Noise not shown.)

3.3.2 Priors

3.3.2.1 Sparsity prior on the gene loadings matrix

Biological processes are likely to only involve a relatively small subset of the total number of genes in the genome, and therefore will be better explained by components with sparse gene loadings vectors. In order to encourage sparsity in the model, a spike and slab (S+S) prior is placed on the rows of X (Lucas

et al., 2006; Mitchell and Beauchamp, 1988). The S+S distribution consists of a mixture of a point mass at zero (the 'spike') and a Gaussian (the 'slab').

The prior on an element of the gene loadings matrix, x_{cl}, is given by

$$P(x_{cl}|p_{cl}, \beta_c) = p_{cl}\mathcal{N}(x_{cl}|0, \beta_c^{-1}) + (1 - p_{cl})\delta_0(x_{cl}) \tag{3.3}$$

where p_{cl} is a mixing weight, β_c is the precision of the 'slab' and δ_0 is a delta function at zero. Note that this S+S formulation is a more general version of the S+S prior used in West (2003), which would be recovered if $p_{cl} = p_c$, i.e. a single mixing parameter for each component. (Inference for (3.1) using the less general S+S prior is give in appendix A.3.) As in Lucas et al. (2006), the more general S+S was found to give lower false positive rates under certain conditions. Simulations showing the different behaviours of the two S+S formulations are given in appendix B.1.

Following convention, a gamma prior is placed on the precision parameters $\beta_c \sim \mathcal{G}(\beta_c|e, f)$ where e and f are hyper-parameters. The prior on p_{cl} is given by another S+S distribution (Lucas et al., 2006),

$$P(p_{cl}|\rho_c) = \rho_c\mathcal{B}(p_{cl}|g, h) + (1 - \rho_c)\delta_0(p_{cl}) \tag{3.4}$$

where ρ_c is a component level mixing parameter. p_{cl} encodes the prior probability of the (c, l)th gene loading being non-zero. If ρ_c takes a value close to 0, the mixture distribution for p_{cl} will be dominated by the delta function at zero and the cth loadings vector will be sparse. On the other hand, if ρ_c is closer to 1, the S+S distribution results in a dense component; this might be more appropriate for modelling confounding factors. To complete the prior on x_{cl}, ρ_c is given a Beta prior with hyperparameters r and z.

Inference for the S+S prior is made easier using a factorisation, $x_{cl} = w_{cl}s_{cl}$

(Titsias and Lázaro-Gredilla, 2011), where

$$P(w_{cl}|\beta_c) = \mathcal{N}(w_{cl}|0, \beta_c^{-1}) \tag{3.5}$$

$$P(s_{cl}|p_{cl}) = \mathcal{B}\text{ernoulli}(s_{cl}|p_{cl}). \tag{3.6}$$

The same approach is used to make inference on p_{cl} tractable, let $p_{cl} = \psi_{cl}\phi_{cl}$, where

$$P(\psi_{cl}) = \mathcal{B}(\psi_{cl}|g, h), \tag{3.7}$$

$$P(\phi_{cl}|\rho_c) = \mathcal{B}\text{ernoulli}(\phi_{cl}|\rho_c). \tag{3.8}$$

3.3.2.2 Priors on the individual and tissue scores matrices

Elements of the individual scores and tissue scores matrices are given standard normal priors,

$$P(a_{nc}) = \mathcal{N}(a_{nc}|0, 1) \tag{3.9}$$

$$P(b_{tc}) = \mathcal{N}(b_{tc}|0, 1). \tag{3.10}$$

This corresponds to a prior belief that individuals and tissues are independent. An extension to explicitly model relatedness between individuals is given in section 3.4.2. Without loss of generality, the variances of the Gaussian distributions can be set to 1 because of the scaling indeterminacy in the model. Scaling factors can be soaked up by the precision terms in the gene loadings matrix.

It is also possible to place spike and slab priors on the individual and tissue scores matrices but that is not something I have investigated here.

3.3.2.3 Prior on noise precision

To complete the model specification, noise precision parameters, λ_{lt}, are given a Gamma prior with hyperparameters u and v,

$$P(\lambda_{lt}) = \mathcal{G}(\lambda_{lt}|u, v). \tag{3.11}$$

Crucially, this model assumes heterogeneous noise, i.e. a different noise precision is learnt for each gene in each tissue. In the context of RNA-sequencing data this model makes sense as some genes may be more prone to sequencing bias or mismapping. An initial version of this model specified homogeneous noise, assuming the same levels of noise across all genes in a single tissue. Like others (e.g. Knowles and Ghahramani (2011)), this was found to be too restrictive.

3.3.3 Full model

The full model is given by,

$$\begin{aligned}
P(\mathcal{Y}|\theta) &= \prod_{l,t} \mathcal{N}_N\Big(\mathbf{y}_{lt}|\sum_c \mathbf{a}_{\cdot c} b_{tc} w_{cl} s_{cl}, \lambda_{lt}^{-1} I_N\Big) \\
P(a_{nc}) &= \mathcal{N}(a_{nc}|0,1) \\
P(b_{tc}) &= \mathcal{N}(b_{tc}|0,1) \\
P(w_{cl}|\beta_c) &= \mathcal{N}\big(w_{cl}|0,\beta_c^{-1}\big) \\
P(s_{cl}|\psi_{cl},\phi_{cl}) &= \mathcal{B}\text{ernoulli}\big(s_{cl}|\psi_{cl}\phi_{cl}\big) \\
P(\beta_c) &= \mathcal{G}(\beta_c|e,f) \\
P(\psi_{cl}) &= \mathcal{B}\text{eta}(\psi_{cl}|g,h) \\
P(\phi_{cl}|\rho_c) &= \mathcal{B}\text{ernoulli}(\psi_{cl}|\rho_c) \\
P(\rho_c) &= \mathcal{B}\text{eta}(\rho_c|r,z) \\
P(\lambda_{lt}) &= \mathcal{G}(\lambda_{lt}|u,v)
\end{aligned} \tag{3.12}$$

where $\theta = (A, B, W, S, \beta, \Psi, \Phi, \rho, \Lambda)$ denotes the set of all parameters and (e, f, g, h, r, z, u, v) is the set of all hyperparameters. The model is fit using an approximate Bayesian method called variational Bayes (Attias, 2000).

3.3.4 Overview of variational Bayes

The goal of Bayesian inference is to evaluate the posterior distribution $P(\theta|\mathcal{Y})$ of the parameters θ given the data $\mathcal{Y}$. In terms of the likelihood $P(\mathcal{Y}|\theta)$, marginal likelihood $P(\mathcal{Y})$ and the priors $P(\theta)$, the posterior is given by,

$$P(\theta|\mathcal{Y}) = \frac{P(\mathcal{Y}|\theta)P(\theta)}{P(\mathcal{Y})}. \tag{3.13}$$

Often an analytic expression for the posterior can not be evaluated. Methods such as MCMC build up a picture of the posterior distribution via sampling. An alternative approach known as variational Bayes (VB) attempts to approximate the true posterior with a simpler form. The approximation (call it $Q(\theta)$) is evaluated to be as close as possible to the true posterior. The Kullback-Lieber (KL) divergence between $Q(\theta)$ and $P(\theta|\mathcal{Y})$ is defined as

$$D_{KL}(Q(\theta)|P(\theta|\mathcal{Y})) = \int \log\left(\frac{Q(\theta)}{P(\theta|\mathcal{Y})}\right) Q(\theta) d\theta = \left\langle \log \frac{Q(\theta)}{P(\theta|\mathcal{Y})} \right\rangle_{Q(\theta)}, \tag{3.14}$$

where $\langle\cdot\rangle_{Q(\theta)}$ denotes an expectation with respect to $Q(\theta)$. The KL divergence is a measure of the similarly between two distributions; it takes values in $[0, \infty)$, with smaller values indicating more similar distributions, and a value of 0 indicating that the distributions are identical. VB aims to find $Q(\theta)$ to minimise $D_{KL}(Q(\theta)|P(\theta|\mathcal{Y}))$.

The log marginal likelihood can be expressed in terms of the KL divergence and another term called the negative free energy $F(Q)$ as follows,

$$\log P(\mathcal{Y}) = \underbrace{\int Q(\theta) \log \frac{P(\mathcal{Y}, \theta)}{Q(\theta)} d\theta}_{:=F(Q)} + D_{KL}(Q(\theta)|P(\theta|\mathcal{Y})). \tag{3.15}$$

The marginal likelihood is independent of Q, so minimising the KL divergence is equivalent to maximising $F(Q)$. Also note that because the KL divergence can not take a negative value, $F(Q)$ is a lower bound to the log marginal likelihood.

The mean field approximation assumes that $Q(\theta) = \prod_i Q(\theta_i)$, i.e. the approximate posterior distribution fully factorises. Under this assumption the following expression for $Q(\theta_i)$ is guaranteed to increase $F(Q)$,

$$Q(\theta_i) = \frac{1}{z}\exp\Big(\big\langle \log P(\theta, \mathcal{Y})\big\rangle_{Q(\theta_{-i})}\Big) \tag{3.16}$$

where θ_{-i} denotes the set of all parameters other than θ_i and z is a normalising constant (see appendix A.2 for the derivation). With an additional constraint that the priors are conjugate, (3.16) simplifies to a set of update rules for the hyperparameters of $Q(\theta_i)$, which depend on moments of $Q(\theta_j)$ for $j \neq i$.

An alternative approach must be used if the priors are not conjugate; in this scenario, a parametric form for the approximate posterior distribution is specified which makes inference tractable. This is known as fixed form VB (Salimans and Knowles, 2013). The parameters of this distribution are then updated in order to maximise the negative free energy.

The VB algorithm consists of iteratively updating parameters (dependent on the current estimates of other parameters) until convergence occurs.

3.3.4.1 Inference via variational Bayes

The majority of the model parameters have conjugate priors and the mean field VB approximation can be used. However, in some cases, fully factorising over parameters may be too strong an assumption. The Gaussian and Bernoulli S+S random variables are highly coupled so dependence is retained in this case. Titsias and Lázaro-Gredilla (2011) show that this approach results in more robust and more accurate estimates. All other parameters with conjugate priors are assumed to fully factorise in the approximate posterior distribution.

Unfortunately the parameters ψ_{cl}, ϕ_{cl} and ρ_c do not have conjugate priors and results from mean field VB can not be applied. Instead, their approximate posterior distributions are specified as point masses.

The approximate posterior distribution $Q(\theta)$ for the model takes the following form,

$$Q(\theta) = \prod_{n,c} Q(a_{nc}) \prod_{t,c} Q(b_{tc}) \prod_{c,l} Q(w_{cl}|s_{cl}) Q(s_{cl}) \prod_{c} Q(\beta_c) \prod_{c,l} \delta_{\psi^*_{cl}}(\psi_{cl}) \prod_{c,l} \delta_{\phi^*_{cl}}(\phi_{cl}) \prod_{c} \delta_{\rho^*_c}(\rho_c) \prod_{l,t} Q(\lambda_{lt}). \tag{3.17}$$

Parameter updates are given below; the VB algorithm iterates through these updates – which are all guaranteed to increase (or at least not decrease) the negative free energy – until convergence. (The algorithm is not guaranteed to converge to the global maximum however, this is discussed more in section 3.3.5).

Parameters of the approximate posterior distributions are denoted using an asterisk (*).

Loadings matrix

$$\begin{aligned} Q(w_{cl}|s_{cl}) &= \mathcal{N}\Big(w_{cl}\Big|s_{cl}m^*_{cl}, \big(s_{cl}\sigma^*_{cl} + (1-s_{cl})\langle\beta_c\rangle\big)^{-1}\Big) \\ \sigma^*_{cl} &= \langle\beta_c\rangle + \sum_{n,t}\langle\lambda_{lt}\rangle\langle a^2_{nc}\rangle\langle b^2_{tc}\rangle \\ m^*_{cl} &= \sigma^{*-1}_{cl}\Big(\sum_{n,t}\langle\lambda_{lt}\rangle y_{nlt}\langle a_{nc}\rangle\langle b_{tc}\rangle - \sum_{n,t}\langle\lambda_{lt}\rangle\langle a_{nc}\rangle\langle b_{tc}\rangle\sum_{k\neq c}\langle w_{kl}s_{kl}\rangle\langle a_{nk}\rangle\langle b_{tk}\rangle\Big) \end{aligned} \tag{3.18}$$

$$\begin{aligned} Q(s_{cl}) &= \mathcal{B}\text{ernoulli}(s_{cl}|\gamma^*_{cl}) \\ \gamma^*_{cl} &= \frac{1}{1+e^{-u^*_{cl}}} \\ u^*_{cl} &= \log(\psi^*_{cl}\phi^*_{cl}) - \frac{1}{2}\log\sigma^*_{cl} + \frac{\sigma^*_{cl}}{2}{m^*_{cl}}^2 - \log(1-\psi^*_{cl}\phi^*_{cl}) + \frac{1}{2}\log\langle\beta_c\rangle \end{aligned} \tag{3.19}$$

$$
\begin{aligned}
Q(\beta_c) &= \mathcal{G}(e_c^*, f_c^*) \\
e_c^* &= e + \frac{L}{2} \\
f_c^* &= \Big(\frac{1}{f} + \frac{1}{2}\sum_l \langle w_{cl}^2 \rangle\Big)^{-1}
\end{aligned} \tag{3.20}
$$

Point estimates are obtained for the parameters ψ_{cl}, ϕ_{cl} and ρ_c by directly optimising the negative free energy. The relevant terms of the negative free energy are given by $\tilde{F}$.

$$
\begin{aligned}
\tilde{F} &:= \sum_{c,l} \big\langle \log P(s_{cl}|\psi_{cl}, \phi_{cl}) \big\rangle + \sum_{c,l} \big\langle \log P(\psi_{cl}) \big\rangle + \sum_{c,l} \big\langle \log P(\phi_{cl}|\rho_c) \big\rangle + \sum_{c} \big\langle \log P(\rho_c) \big\rangle \\
&= \sum_{c,l} \big(\langle s_{cl} \rangle \log(\psi_{cl}^* \phi_{cl}^*) + \langle 1 - s_{cl} \rangle \log(1 - \psi_{cl}^* \phi_{cl}^*) \big) \\
&\quad + \sum_{c,l} \big((g-1) \log \psi_{cl}^* + (h-1) \log(1 - \psi_{cl}^*) \big) \\
&\quad + \sum_{c,l} \big(\phi_{cl}^* \log \rho_c^* + (1 - \phi_{cl}^*) \log(1 - \rho_c^*) \big) \\
&\quad + \sum_{c} \big((r-1) \log \rho_c^* + (z-1) \log(1 - \rho_c^*) \big)
\end{aligned} \tag{3.21}
$$

The equation $\frac{\delta F}{\delta \rho_c} = 0$ has a closed form solution so ρ_c^* is found as follows,

$$
\rho_c^* = \frac{\sum_l \phi_{cl}^* + r - 1}{L + r + z - 2} \tag{3.22}
$$

Newton's method is used to optimise $\tilde{F}$ for $(\psi_{cl}^*, \phi_{cl}^*)$. The optimisation problem is

$$
(\psi_{cl}^*, \phi_{cl}^*) = \text{argmax}_{(\psi_{cl}, \phi_{cl})} \tilde{F} \tag{3.23}
$$

The gradient and Hessian matrix of $\tilde{F}$ are given by

$$\mathbf{g} = \begin{pmatrix} \frac{\langle s_{cl}\rangle}{\psi_{cl}} - \frac{\langle 1-s_{cl}\rangle\phi_{cl}}{1-\psi_{cl}\phi_{cl}} + \frac{g-1}{\psi_{cl}} - \frac{h-1}{1-\psi_{cl}} \\ \frac{\langle s_{cl}\rangle}{\phi_{cl}} - \frac{\langle 1-s_{cl}\rangle\psi_{cl}}{1-\psi_{cl}\phi_{cl}} + \log\rho_c - \log(1-\rho_c) \end{pmatrix} \tag{3.24}$$

$$H = \begin{pmatrix} -\frac{\langle s_{cl}\rangle}{\psi_{cl}^2} - \frac{\langle 1-s_{cl}\rangle\phi_{cl}^2}{(1-\psi_{cl}\phi_{cl})^2} - \frac{qr-1}{\psi_{cl}^2} - \frac{q(1-r)-1}{(1-\psi_{cl})^2} & -\frac{\langle 1-s_{cl}\rangle}{(1-\psi_{cl}\phi_{cl})^2} \\ -\frac{\langle 1-s_{cl}\rangle}{(1-\psi_{cl}\phi_{cl})^2} & -\frac{\langle s_{cl}\rangle}{\phi_{cl}^2} - \frac{\langle 1-s_{cl}\rangle\psi_{cl}^2}{(1-\psi_{cl}\phi_{cl})^2} \end{pmatrix} \tag{3.25}$$

and (ψ_{cl}, ϕ_{cl}) is updated as follows,

$$\begin{pmatrix} \psi_{cl}^{i+1} \\ \phi_{cl}^{i+1} \end{pmatrix} = \begin{pmatrix} \psi_{cl}^{i} \\ \phi_{cl}^{i} \end{pmatrix} - \alpha H^{i-1}\mathbf{g}^i \tag{3.26}$$

where α is a step-size determined using a backtracking line search, i.e. $\alpha = 1$ initially, then it is reduced until $\tilde{F}^{i+1} > \tilde{F}^i$ is satisfied.

Update for a_{nc}

$$\begin{aligned} Q(a_{nc}) &= \mathcal{N}(a_{nc}|\mu_{nc}^*, \omega_{nc}^{*-1}) \\ \omega_{nc}^* &= 1 + \sum_{l,t}\langle\lambda_{lt}\rangle\langle b_{tc}^2\rangle\langle w_{cl}^2 s_{cl}^2\rangle \\ \mu_{nc}^* &= \omega_{nc}^{*-1}\Big(\sum_{l,t}\langle\lambda_{lt}\rangle y_{nlt}\langle b_{tc}\rangle\langle w_{cl}s_{cl}\rangle - \sum_{l,t}\langle\lambda_{lt}\rangle\langle b_{tc}\rangle\langle w_{cl}s_{cl}\rangle\sum_{k\neq c}\langle a_{nk}\rangle\langle b_{tk}\rangle\langle w_{kl}s_{kl}\rangle\Big) \end{aligned} \tag{3.27}$$

Update for b_{tc}

$$\begin{aligned} Q(b_{tc}) &= \mathcal{N}(b_{tc}|\nu_{tc}^*, \tau_{tc}^{*-1}) \\ \tau_{tc}^* &= 1 + \sum_{n,l}\langle\lambda_{lt}\rangle\langle a_{nc}^2\rangle\langle w_{cl}^2 s_{cl}^2\rangle \\ \nu_{tc}^* &= \tau_{tc}^{*-1}\Big(\sum_{n,l}\langle\lambda_{lt}\rangle y_{nlt}\langle a_{nc}\rangle\langle w_{cl}s_{cl}\rangle - \sum_{n,l}\langle\lambda_{lt}\rangle\langle a_{nc}\rangle\langle w_{cl}s_{cl}\rangle\sum_{k\neq c}\langle a_{nk}\rangle\langle b_{tk}\rangle\langle w_{kl}s_{kl}\rangle\Big) \end{aligned} \tag{3.28}$$

Update for λ_{lt}

$$Q(\lambda_{lt}) = \mathcal{G}(\lambda_{lt}|u^*_{lt}, v^*_{lt}) \tag{3.29}$$

$$u^*_{lt} = u + \frac{N}{2}$$

$$v^*_{lt} = \left(\frac{1}{v} + \frac{1}{2}\sum_n \left\langle \left(y_{nlt} - \sum_c a_{nc} b_{tc} w_{cl} s_{cl}\right)^2 \right\rangle\right)^{-1} \tag{3.30}$$

Negative free energy: The negative free energy is a lower bound of the marginal likelihood. The updates given above are guaranteed to increase the free energy (3.31).

$$
\begin{aligned}
F(Q) = &-\frac{NLT}{2}\log 2\pi + \frac{N}{2}\sum_{l,t}\langle \log \lambda_{lt}\rangle - \frac{1}{2}\sum_{n,l,t}\langle\lambda_{lt}\rangle\big\langle (y_{nlt} - \Sigma_c a_{nc} b_{tc} w_{cl} s_{cl})^2\big\rangle \\
&- \frac{1}{2}\sum_{n,c}\langle a_{nc}^2\rangle - \frac{1}{2}\sum_{n,c}\log|\omega_{cl}^*| + \frac{NC}{2} \\
&- \frac{1}{2}\sum_{t,c}\langle b_{tc}^2\rangle - \frac{1}{2}\sum_{t,c}\log|\nu_{tc}| + \frac{TC}{2} \\
&+ \frac{L}{2}\sum_{c}\big\langle \log\beta_c\big\rangle + \frac{CL}{2} - \frac{1}{2}\sum_{c,l}\langle\beta_c\rangle\big\langle w_{cl}^2\big\rangle \\
&- \frac{1}{2}\sum_{c,l}\gamma_{cl}^*\log\sigma_{cl}^* + \frac{1}{2}\sum_{c,l}(1-\gamma_{cl}^*)\log\langle\beta_c\rangle \\
&\sum_c\Bigg(- \log\Gamma(e) - e\log f + (e-1)(\psi(e_c^*) + \log\hat{f}_c) - \frac{e_c^* f_c^*}{f} \\
&+ e_c^* + \log f_c^* + \log\Gamma(e_c^*) - (e_c^*-1)\psi(e_c^*)\Bigg) \\
&+ \sum_{c,l}\Bigg(\langle s_{cl}\rangle\langle\log\psi_{cl}\phi_{cl}\rangle + (1-\langle s_{cl}\rangle)\langle\log\big(1-\psi_{cl}\phi_{cl}\big)\rangle \\
&- \langle s_{cl}\rangle\log\langle s_{cl}\rangle - (1-\langle s_{cl}\rangle)\log(1-\langle s_{cl}\rangle)\Bigg) \\
&+ \sum_{c,l}\Big((g-1)\log\psi_{cl}^* + (h-1)\log(1-\psi_{cl}^*)\Big) \\
&+ \sum_{cl}\Big(\phi_{cl}^*\log\rho_c^* + (1-\phi_{cl}^*)\log(1-\rho_c^*)\Big) \\
&+ \sum_{cl}\Big((r-1)\log\phi_{cl}^* + (z-1)\log(1-\phi_{cl}^*)\Big) \\
&+ \sum_{lt}\Bigg(- \log\Gamma(u) - u\log v + (u-1)(\psi(u_{lt}^*) + \log v_{lt}^*) - \frac{u_{lt}^* v_{lt}^*}{v} \\
&+ u_{lt}^* + \log v_{lt}^* + \log\Gamma(u_{lt}^*) - (u_{lt}^*-1)\psi(u_{lt}^*)\Bigg) \qquad (3.31)
\end{aligned}
$$

3.3.5 Identifiability

The model in (3.1) is partially identifiable. The sign and scale of the gene loadings, individual scores and tissue scores vectors are not fully determined by the model. Furthermore, a permutation of the components does not affect the model fit. The method is not rotationally invariant however. This follows

from identifiability results for the PARAFAC model and also from the sparsity assumptions on the gene loadings matrix (see section 2.1.6). In practice, identifiability issues do arise. It is hard to tell the extent of this problem, but if components are dense, they are likely to suffer from lack of identifiability, especially if they are active in only one tissue.

The bigger problem affecting the uniqueness of the solution stems from the nature of the VB algorithm. VB is a deterministic method that iteratively updates parameters to increase an objective function, i.e. it hill climbs. The state space for this method is complex and the VB algorithm is bound to get stuck in local optima. Section 4.2.3.1 contains an approach that tries to (at least partially) circumvent this problem.

3.4 Extensions

The Bayesian inference procedures used here provide a very flexible framework on which to extend the method. This section describes some extensions to handle missing data and account for related individuals. The section also contains a model which jointly analyses several omics data sets simultaneously, following the approach detailed in Groves et al. (2011).

3.4.1 Missing data

In this section I describe two extensions to the model which allow for missing data. Missing data is common in genetic studies, often arising due to experimental artefacts. There is a special case of missingness that is prevalent in multidimensional studies: missing samples, where the term 'sample' refers to data for an individual in one tissue (see figure 3.2). Missing samples arise if data collection is not complete or samples are removed during quality control. Similar situations arise when data is collected across several data types. Missing samples result in missing rows in the data. The default method for dealing

with missing samples is to remove individuals without complete data, however this can reduce the size of the data set considerably.

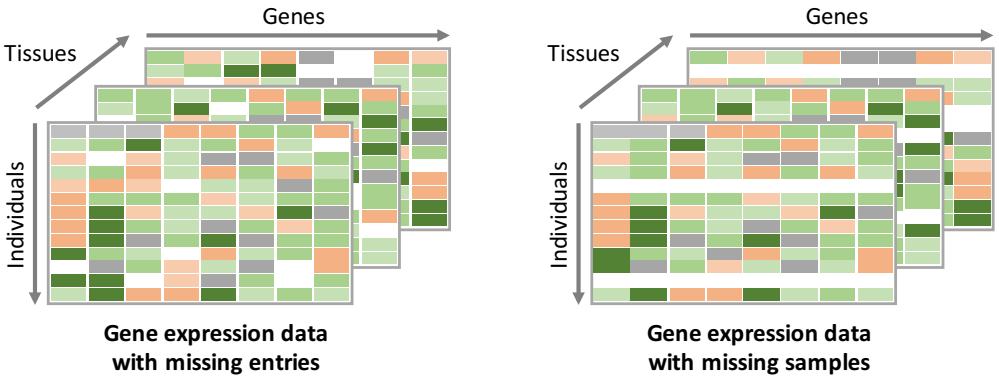

Figure 3.2: Illustration of two missing data scenarios in a multi-tissue gene expression data set.

In the VB framework, missing entries in the data tensor can be treated as another parameter in the model, and their posterior distribution learnt. Not only does this mean that no data has to be removed, but this solution also results in estimates of the missing data. I also describe a method for handling the special case of missing samples.

3.4.1.1 Method 1: Learning a posterior distribution

Denote an element in $\mathcal{Y} \in \mathbb{R}^{N \times L \times T}$ by y^m_{nlt} if it is missing, and y^o_{nlt} if it is observed. Missing data entries are treated as another parameter in the model with a Gaussian prior,

$$P(y^m_{nlt}|\theta) = \mathcal{N}\left(y^m_{nlt}|\Sigma_c a_{nc} b_{tc} w_{cl} s_{cl}, \lambda^{-1}_{lt}\right), \tag{3.32}$$

where θ is the set of all model parameters (not including the missing data entries). Using the model likelihood and priors given in section 3.3.3, an approximate posterior distribution (denoted by $Q(y^m_{nlt})$) can be learnt for the

missing entries,

$$Q(y^m_{nlt}) = \mathcal{N}\big(y^m_{nlt} | \sum_c \langle a_{nc}\rangle \langle b_{tc}\rangle \langle w_{cl} s_{cl}\rangle, \langle \lambda^{-1}_{lt}\rangle\big) \tag{3.33}$$

In this extended model, the updates for the model parameters (θ) are very similar to the set of updates given in section 3.3.4.1. However, they need to be altered to reflect the uncertainty in the estimates of the missing data points. Suppose the data entry given by indices $\{n, l, t\}$ is missing, then occurrences of y_{nlt} are replaced by the posterior mean $\langle y^m_{nlt}\rangle$, and y^2_{nlt} replaced by the posterior 2nd moment $\langle (y^m_{nlt})^2\rangle = \langle y^m_{nlt}\rangle^2 + \langle \lambda^{-1}_{lt}\rangle$. As before, model parameters are sequentially updated, with an additional update for the missing entries at each iteration. The expression for the negative free energy also needs to be updated to reflect uncertainty in $Q(y^m_{nlt})$.

The posterior mean, $\langle y_{nlt}\rangle$ can be used for data imputation. However, it is worth pointing out (as with much of the rest of the model) that it is hard to know whether the assumptions made here are valid. Missing data in genetics is often not 'missing completely at random'. Systematic sequencing artefacts and missing samples result in missingness patterns that are not random and it is unclear how this will affect the posterior estimates of the missing entries.

3.4.1.2 Method 2: Handling missing samples

The approach described here is an alternative way of handling missing samples that does not involve imputation. Instead, missing samples are removed from the model likelihood.

Missing samples correspond to missing vectors within the data tensor; for example, if data for individual n in tissue t is missing, then y_{nlt} will be missing for all l. Let $\mathcal{J}$ be a binary indicator matrix of dimensions N by T, where $\mathcal{J}_{nt} = 1$ if data for individual n in tissue t exists and $\mathcal{J}_{nt} = 0$ otherwise. Using

observed data only, the likelihood can be written as (Chan et al., 2002),

$$P(\mathcal{Y}|\theta) = \prod_{n,l,t} \mathcal{N}\Big(y_{nlt}|\sum_{c} a_{nc}b_{tc}w_{cl}s_{cl}, \lambda_{lt}^{-1}\Big)^{\mathcal{J}_{nt}}. \tag{3.34}$$

With this likelihood, together with the priors defined previously (section 3.3.3), a set of VB updates can be derived which are similar to those given in section 3.3.4.1. The only difference being that the indicator matrix $\mathcal{J}$ needs to be added into any expression with a sum over n or t. This removes the contribution from the missing data entries in the updates.

3.4.2 Related individuals

As it stands, the tensor decomposition ignores any relatedness between individuals. However, genetic studies often contain closely related individuals by design, or include distantly related individuals by chance (if recruitment occurs within a small geographical area). A kinship matrix $K \in \mathbf{R}^{N\times N}$ can be used to summarise the genetic relatedness between individuals where an element of K, $k_{ij} \in [0, 1]$, is a measure of the relatedness between individuals i and j (Speed and Balding, 2014). The kinship matrix may be calculated from pedigree information or estimated from genotype data. Expression data from related individuals is likely to be correlated due to shared genetic material and environment. Explicitly modelling these correlations may lead to better signal recovery in gene expression studies. This is certainly the case in genome wide association studies (Korte et al., 2012).

Consider the following prior for the individual scores matrix,

$$P(A) = \prod_{c} \mathcal{N}_N\big(\mathbf{a}_c|0, \alpha_c K + (1-\alpha_c)I_N\big) \tag{3.35}$$

where α_c is a parameter in the set $[0, 1]$. The columns of A (individual scores vectors) now have a multivariate normal prior with covariance given by a mixture of the kinship matrix and an identity matrix. The mixture parameter, α_c

is given an uninformative beta prior, $P(\alpha_c) = \mathcal{B}\text{eta}(\alpha_c|1,1)$. Importantly, a different mixing parameter is learnt for each component.

Gene expression data consists of both genetic signals (e.g. a *trans* network or the genetic basis of ageing) and non-genetic signals (e.g. batch effects or environmental signals). The idea behind this prior is that it can accommodate both types of signals. If α_c is close to 1 then the prior covariance matrix for $\mathbf{a}_c$ is approximately K. This imposes structure on the individual scores vectors such that related individuals have more similar scores, resulting in a 'genetic' component. On the other hand, if α_c is close to 0 then the prior has no genetic basis and the original prior for $\mathbf{a}_c$ is recovered. It is important to note that α_c can not be interpreted as a heritability estimate for the cth component, however it does provide some information about the component's heritability relative to the other components.

Implementing this extension involves a change to the update for A and the addition of an update for α_c; the remaining parameter updates do not change. $\mathbf{a}_c$ and α_c are assumed to be independent in the approximate posterior distribution and the (approximate) posterior distribution of α_c is taken to be a delta function at α_c^*.

The update for A under the new prior becomes,

$$
\begin{aligned}
Q(\mathbf{a}_c) &= \mathcal{N}_N(\mathbf{a}_c|\boldsymbol{\mu}_c^*, \Omega_c^{*-1}) \\
\Omega_c^* &= \Big(\alpha_c^* K + (1-\alpha_c^*) I_N\Big)^{-1} + \Big(\sum_{l,t} \langle\lambda_{lt}\rangle \langle b_{tc}^2\rangle \langle w_{cl}^2 s_{cl}^2\rangle\Big) I_N \\
\boldsymbol{\mu}_c^* &= \Omega_c^{*-1}\Big(\sum_{l,t} \langle\lambda_{lt}\rangle \mathbf{y}_{lt} \langle b_{tc}\rangle \langle w_{cl} s_{cl}\rangle - \sum_{l,t} \langle\lambda_{lt}\rangle \langle b_{tc}\rangle \langle w_{cl} s_{cl}\rangle \sum_{k\neq c} \langle\mathbf{a}_k\rangle \langle b_{tk}\rangle \langle w_{kl} s_{kl}\rangle\Big).
\end{aligned}
\tag{3.36}
$$

A naive implementation of $\boldsymbol{\mu}_c^*$ has complexity $\mathcal{O}(N^3)$, as an N by N matrix needs to be inverted at each iteration. An efficient formulation using the eigendecomposition of K avoids this cost. Let $K = QDQ^t$ be the eigendecomposition of K where Q is an orthonormal matrix of eigenvectors and D

is a diagonal matrix with eigenvalues on the diagonal. The expression for $\boldsymbol{\mu}_c^*$ requires Ω_c^{*-1}, which can be rewritten as

$$\Omega_c^{*-1} = Q\Big(((1-\alpha_c)I_N + \alpha_c D)^{-1} + \sum_{l,t}\langle\lambda_{lt}\rangle\langle b_{tc}^2\rangle\langle w_{cl}^2 s_{cl}^2\rangle I_N\Big)^{-1} Q^t \tag{3.37}$$

which only requires the inversion of a diagonal matrix. Using (3.37) reduces the complexity of the expression for $\boldsymbol{\mu}_c^*$ to $\mathcal{O}(N^2)$ per iteration, with a one-off cost of $\mathcal{O}(N^3)$ to calculate Q and D. Note that (3.37) is no longer valid when missing samples have been removed from the model likelihood[1].

Point estimates, α_c^*, are evaluated to optimise the negative free energy,

$$\alpha_c^* \leftarrow \alpha_c^* + \Delta\sum_n(-1+D_{nn})\Big(-\frac{1}{1-\alpha_c+\alpha_c D_{nn}} + \frac{\big(Q(\boldsymbol{\mu}_c^*\boldsymbol{\mu}_c^{*t}+\Omega_c^{*-1})\big)_{nn}}{(1-\alpha_c+\alpha_c D_{nn})^2}\Big) \tag{3.38}$$

where $\Delta = 0.0001$ is the step size.

3.4.3 Linked tensor decomposition

Consider a study consisting of D types of omics data for a set of N individuals. Let each data set d be represented by the tensor, $\mathcal{Y}^{(d)} \in \mathbb{R}^{N\times L_d\times T_d}$ where L_d is the number of variables measured for data type d and T_d is the number of contexts (or conditions) in which these variables were measured. If data for type d is collected in only a single context then $T_d = 1$. Importantly, all tensors are linked by their shared first dimension, N.

The data is modelled as follows (Groves et al., 2011),

$$y_{nlt}^{(d)} = \sum_c a_{nc} b_{tc}^{(d)} x_{cl}^{(d)} + \epsilon_{nlt}^{(d)} \qquad \text{for } d \in \{1,\cdots,D\} \tag{3.39}$$

where $A \in \mathbb{R}^{N\times C}$ is the individual scores matrix (shared across all data types),

[1]This is because the second term in the expression for Ω_c^* depends on n (via a binary indicator matrix), i.e it is a diagonal matrix rather than a multiple of the identity matrix.

$B^{(d)} \in \mathbb{R}^{T_d \times C}$ is a context specific scores matrix for data type d and $X^{(d)} \in \mathbb{R}^{C \times L_d}$ is a loadings matrix for data type d. A noise tensor for each data type is given by $\mathcal{E}^{(d)} \in \mathbb{R}^{N \times L_d \times T_d}$. Figure 3.3 shows a diagram of this decomposition. Each data tensor is decomposed using (3.1), with the constraint that a single individual scores matrix is common across all data types. In practice, if $T_d = 1$ for a data type d, then $B^{(d)}$ has dimensions 1 by C and is fixed to a vector of ones during inference (shown as hatching in figure 3.3).

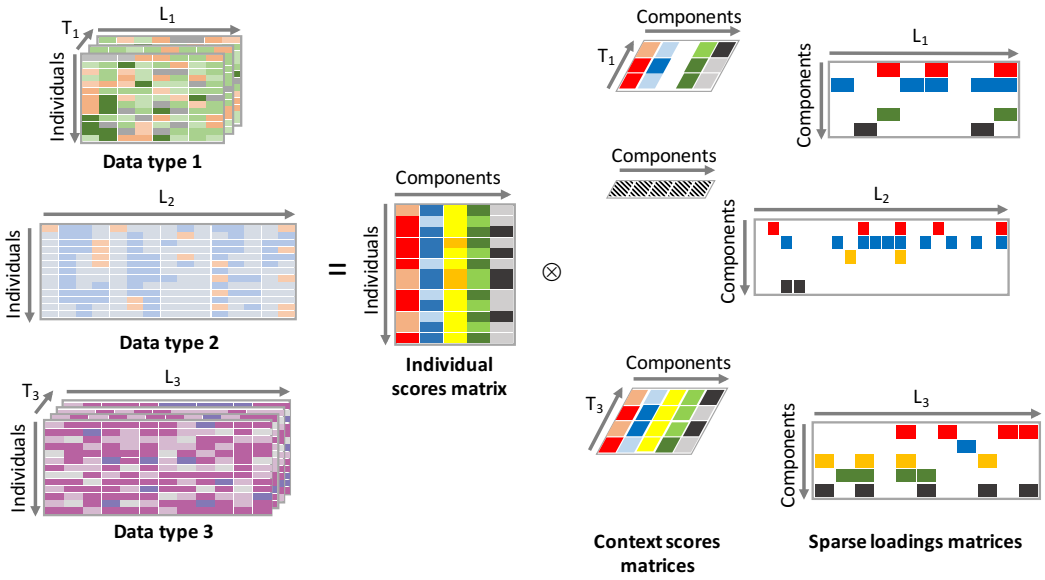

Figure 3.3: Illustration of linked tensor decomposition. Data type 2 is a matrix (i.e. $T_2 = 1$), so the context scores are a fixed vector of 1's (represented by hatching). (Noise not shown.)

The full model for (3.39) is given in appendix A.4. Again, spike and slab priors are used for the loadings matrices to encourage sparsity. Updates for the loadings and context scores matrices for a data type d are effectively identical to the (single) tensor decomposition already considered. Importantly, updates for $X^{(d)}$ and $B^{(d)}$ do not depend on $X^{(d')}$ and $B^{(d')}$ for any $d' \neq d$. The update for A is dependent on all other current parameter estimates. One way to think about this update is that it averages over the estimates for A that one would get if performing separate decompositions for each data type. (In reality this is not quite the case because the prior also needs to be considered.) The update for A is given in appendix A.4.

It is important to note that (3.39) can model a variety of different types of underlying structure in the data. Components can be shrunk to zero for a particular data type allowing for the model to capture signals that exist in an arbitrary subset of the data. For example, in figure 3.3, the yellow component is active in only data types 2 and 3.

This linked tensor decomposition is a generalisation of several models. In particular, the tensor decomposition in section 3.3.3 is recovered if $D = 1$. Furthermore, if $T_1 = 1$, then the model collapses to sparse factor analysis. A group factor analysis model (2.6) is recovered if $T_d = 1$ for all d.

3.5 Implementation

An implementation of the linked tensor decomposition is coded in `C++` using a matrix library called Eigen[2]. This approach is referred to as SPIDER (sparse integrated decompostions for RNA-seq) in the remainder of this thesis. As previously mentioned, the single tensor decomposition described in section 3.3.3 is recovered for the case $D = 1$.

Uninformative priors are placed on the precision of the noise ($\lambda_{lt}^{(d)}$) and 'slab' precision ($\beta_c^{(d)}$) by setting $u = 10^{-6}$, $v = 10^6$, $e = 10^{-6}$ and $f = 10^6$. A flat prior is used for the component sparsity parameters ($\rho_c^{(d)}$) with $r = z = 1$. Finally, the Beta prior on ϕ_{cl} is specified with parameters[3] $g = h = 0$. This prior was found to produce the best results on both simulated and real data.

The majority of the model parameters are initialised by randomly drawing values from the prior. Elements in $S^{(d)}$, $\Psi^{(d)}$ and $\Phi^{(d)}$ are initialised to 0.5.

Determining the number of components C (i.e. model selection) for these types of models is not trivial. One way to pick C involves running the method

[2] `http://eigen.tuxfamily.org`

[3] This choice of hyperparameters results in a Beta distribution with half the probability mass at 0 and the other half at 1. Previous implementations of this prior suggest hyperparameters which place most of the probability mass near 1 (Carvalho et al., 2008; Lucas et al., 2006). These implementations use samplings methods to evaluate a posterior distribution (Gibbs and MCMC). When VB is used for inference, it seems to be important that the Beta part of the mixture distribution can be estimated as zero as well.

multiple times with different values for C, and selecting the run with the largest value of the negative free energy. (The negative free energy is a lower bound of the log marginal likelihood and therefore a proxy for model fit.) In practice, if SPIDER is initialised with too many components, it will shrink entire gene loadings vectors to zero to remove superfluous components. This behaviour is also seen when an ARD prior is used. Initial experiments are required to evaluate the optimal number of components to initialise SPIDER with; too many components will slow down the algorithm, but too few may result in signals being missed. SPIDER is run so that a small number components are consistently shrunk to zero.

Convergence of the VB algorithm is tracked using change in the posterior mean of $S^{(d)}$. The algorithm is terminated once the number of elements in $\langle S^{(d)} \rangle$ crossing a threshold of 0.5 per iteration drops to less than 1, or if the algorithm reaches 3,000 iterations.

The model output is an approximate posterior distribution over the matrices A, $B^{(d)}$, $X^{(d)}$ and $S^{(d)}$. The posterior mean of these parameters is used for further analysis. (VB is known to underestimate variances so the posterior distributions themselves are not investigated (Jaakkola and Jordan, 2000).) $\langle S_{cl}^{(d)} \rangle$ can be interpreted as the (approximate) posterior probability that variable l from data type d contributes to component c, or the posterior inclusion probability (PIP).

3.5.1 Complexity

The linked tensor decomposition (3.1) is linear in the size of the input data, N, L_d and T_d and quadratic in the number of components. In genetic studies, the number of variables $\sum_d L_d$ is often considerably larger than the other two dimensions, so the majority of the computational time is spent updating the loadings matrices. Updates for each element of a loadings matrix are dependent on the current parameter estimates from within the same column, but not on

the rest of the matrix. It is therefore perfectly valid in the VB framework to update elements in a row in parallel. OpenMP[4] is used for this parallelisation. Another speed-up can be obtained by suspending the updates of components that have been shrunk to zero. The number of $\langle S_{cl}^{(d)} \rangle < 0.05$ within each loadings vector is tracked at each iteration. If $\langle S_{cl}^{(d)} \rangle < 0.05$ for all l, then component c is determined to be 'not active' in data type d, and parameters for that component are no longer updated.

The extension to explicitly model relatedness increases the complexity from linear to quadratic in N. However, for large $\sum_d L_d$ this is not limiting. Combining this extension with the extension from section 3.4.1.2 to deal with missing samples further increases the complexity to $\mathcal{O}(N^3)$, which is not feasible for most genetic data sets.

Table 3.1 detail some timings of running SPIDER on a 3D aray of data with a variety of different dimensions.

Dimensions of input data $N \times L \times T$	**Number of components** C	**Time**	**Notes**
$200 \times 500 \times 3$	16	25 seconds	Timings for simulated data from section 4.1.1.
$700 \times 2,500 \times 3$	100	43 minutes	Timings for simulated data from section 4.2.
$845 \times 18,409 \times 3$	1,000	20 hours*	Timings for real data analysis from chapter 5.

Table 3.1: Timings for SPIDER on different sizes data sets. Timings are given for a single run of the tensor decomposition (3.1). Convergence was determined using the criteria in section 3.5. An asterisk (*) indicates that SPIDER was run with parallelisation.

3.6 Comparison with existing methods

This section describes the similarities and differences between SPIDER and existing methods in the literature. Motivation for this work came from Groves

[4] `http://www.openmp.org`

et al. (2011). Groves et al. (2011) implement a linked tensor decomposition to jointly analyse brain image data. Signals underlying brain image data are likely to be very different than those found in gene expression data. Sparse structure is not expected in brain image data, Groves et al. (2011) use a mixture of Gaussians prior on their equivalent of a (gene) loadings matrix, which does not tend to produce sparse components. The approach described in this thesis adds sparsity to the model from Groves et al. (2011). Since starting this work, several similar models (described below) have been published.

SPIDER implements a sparse Bayesian PARAFAC decomposition (Carroll and Chang, 1970; Harshman and Lundy, 1994). Existing sparse implementations of PARAFAC tend to use penalties or priors on one or all of the matrices A, B and X. Non-Bayesian implementations include Allen (2012) who place L_1 penalties on A, B and X. Padilla and Scott (2015) develop a model which allows for arbitrary penalties, for example, the fused lasso penalty introduces smoothness in the components.

Bayesian implementations of PARAFAC tend to use an ARD prior to regularise, for example, Zhao et al. (2014a) use this prior on all component vectors and perform inference using VB. As far as I know, an implementation of the PARAFAC model with a S+S prior to encourage element-wise sparsity is new to this work. Khan et al. (2014b) use a S+S prior to obtain component level shrinkage alongside an ARD prior for element-wise sparsity. This ARD prior is more general than the prior from Zhao et al. (2014a) as a precision variable is learnt for each element in the loadings matrix (rather than each row). A simulation study comparing the Khan et al. (2014b) model with SPIDER is given in chapter 4.

Use of a S+S distribution in a matrix decomposition is not novel however. S+S distributions have been used to encourage sparsity in factor analysis, e.g. West (2003) and Lucas et al. (2006). In addition, Ray et al. (2014) use a S+S prior in a model similar to group factor analysis, with inference performed

via Gibbs sampling. A variational Bayes implementation of the S+S prior from Lucas et al. (2006), the formulation used here, has not been previously published.

As already mentioned, the linked tensor framework follows work done by Groves et al. (2011). Groves et al. (2011) also fit their decomposition using VB, but use a mixture of Gaussians as a prior on the loadings matrices, so the estimated components tend not to be sparse. Khan et al. (2014b) also propose a similar framework and extend the linked decomposition to allow for arbitrarily linked tensors and matrices. Similar non-Bayesian linked decompositions include Yokota et al. (2012) and Ermiş et al. (2013).

The work presented here on missing data follows Chan et al. (2002) and has been used previously in tensor decompositions (e.g Zhao et al. (2014a)). Extensions to explicitly model relatedness are novel. Table 3.2 contains a summary of a variety of features of SPIDER and some similar approaches, showing how SPIDER fits into the existing literature.

3.7 Discussion

In this chapter I have suggested a new method for analysing multidimensional gene expression data sets. The approach uses a tensor decomposition to uncover latent components describing modes of variance in the data. A flexible prior is used to encourage sparsity in the latent components, allowing for both sparse and dense signals to be recovered. A tissue scores matrix is estimated to describe the activity of each component within each tissue. I have also described extensions which deal with various challenges that arise when analysing genetic data: missing data, related individuals and additional data types.

It is important to discuss the assumptions made in this method and consider to what extend these assumptions are valid, and whether violations of the assumptions matter. I assume a Gaussian noise model; for raw expression data, this is probably far from the truth. However, various normalisation

techniques are commonly applied to gene expression data to make the data look more 'normal'. Another assumption that I make is that the posterior distribution factorises almost fully. It is very hard to say what the effect of this is. This could be circumnavigated by using an exact Bayesian approach like MCMC, but this would likely come with large computational costs.

A final assumption that I want to raise is whether the PARAFAC decomposition is the correct model for gene expression data. The PARAFAC model assumes that signals that are active across several tissues involve the same set of genes. It is possible that the more relaxed group factor analysis model better reflects the nature of signals in expression data. Group factor analysis requires estimation of more parameters however, so this is a trade off. In addition, both decompositions assume linearity, and it is unclear how linear biological pathways actually are.

Probably the greatest challenge of using latent variable models is in the interpretation of the output. In the next section, I describe some simulations in which I use SPIDER to uncover *trans* eQTLs in data from multiple tissues. A variety of post-processing steps are taken to interpret the components and evaluate whether they are describing variation driven by genetics. Chapter 5 describes results of applying this method to a multi-tissue gene expression data set.

	Sparsity	Inference	Joint decomposition of matrices	Tensor decomposition	Joint decomposition of tensors	*Model selection	**Noise model	Missing data	Related individuals
Klami et al. (2014) (CCAGFA)	ARD prior	Variational Bayes	✓	✗	✗	Via ARD prior	Heterogeneous	✗	✗
Zhao et al. (2014b) (BGFA)	Three parameter Beta prior	Variational expectation maximisation	✓	✗	✗	Global shrinkage defined in prior	Heterogeneous	✗	✗
Ray et al. (2014)	Hierarchical Beta-Bernoulli construction of S+S prior	Gibbs sampling	✓	✗	✗	Beta-Bernoulli process	Homogeneous	✗	✗
Zhao et al. (2014a)	ARD priors on all scores and loadings matrices	Variational Bayes	✗	✓	✗	Via ARD priors	Homogeneous	✓Missing elements	✗
Groves et al. (2011)	✗	Variational Bayes	✓	✓	✓	Via an ARD prior on the individual scores matrix	Homogeneous	✗	✗
Khan et al. (2014b) (BMTF)	ARD prior for element-wise sparsity	Gibbs sampling	✓	✓	✓	Component-wise S+S prior	Homogeneous	✗	✗
SPIDER	S+S prior for element-wise sparsity	Variational Bayes	✓	✓	✓	Result of the S+S prior	Heterogeneous	✓Missing samples	✓Kinship-informed prior on the individual scores vectors.

Table 3.2: Comparison of SPIDER with existing methods. (This table contains only a subset of the multidimensional decompositions in the literature.) *Model selection refers to how the number of components is chosen. **A noise model is said to be homogeneous if the same noise precision is assumed for all genes, and heterogeneous if a different noise precision is specified for each gene.

Chapter 4

Simulation study

In the previous chapter, I described a new method (SPIDER) for extracting sparse signals from a multidimensional data set. This chapter starts with some simulations designed to compare SPIDER with existing approaches. These simulations aim to evaluate the differences between various latent variable models and their sensitivity to different sparsity and noise levels. In the second half of the chapter, I use simulated data to evaluate whether SPIDER has the power to find *trans* effects in multidimensional gene expression data.

4.1 Method comparisons

This section starts with a comparison between SPIDER and another sparse tensor decomposition, BMFT. These approaches differ in the way they perform inference and encourage sparsity, but assume the same underlying model for the data. Data is simulated assuming a PARAFAC model with sparse components, and a variety of post-processing techniques are used to test component recovery. Since there are a limited number of sparse tensor decompositions with implementations available, I also compare the group factor analysis version of SPIDER with similar methods. Again these models only differ in their implementation.

4.1.1 Comparison of tensor decompositions

4.1.1.1 Method descriptions

Bayesian Matrix Tensor Factorisation (BMTF) is a linked decomposition of an arbitrary number of matrices and tensors (Khan and Kaski, 2014; Khan et al., 2014b). This approach is perhaps the most similar to the method described in this thesis. These simulations consider a specific case of BMTF where the input data, $\mathcal{Y}$, is a single 3D array. BMTF performs a PARAFAC decomposition, with a prior encouraging sparsity on the loadings matrix given by,

$$\begin{aligned} x_{cl} &\sim h_c \delta_0 + (1 - h_c)\mathcal{N}(x_{cl}|0, \alpha_{cl}^{-1}), \\ h_c &\sim \mathcal{B}\text{ernoulli}(\pi_c). \end{aligned} \tag{4.1}$$

The spike and slab distribution in (4.1) has a mixing parameter, $h_c \in \{0, 1\}$, which determines the activity of each component. If $h_c = 0$ then the whole component is shrunk to zero; if $h_c = 1$, then the prior on the loadings vector reduces to an element-wise ARD prior. (In comparison, SPIDER uses a spike and slab distribution to obtain element-wise sparsity.) BMTF places a Beta hyperprior on π_c and a Gamma prior on the variance parameters α_{cl}. Standard normal priors are specified for the individual and tissue scores matrices, and a homogeneous Gaussian noise model is assumed.

Inference is performed using Gibbs sampling. Code for BMTF written in `R` can be downloaded from

`http://research.cs.aalto.fi/pml/software/bmtf/`.

4.1.1.2 Data simulation

Data was simulated under the PARAFAC model,

$$y_{nlt} = \sum_{c=1}^{C=8} a_{nc} b_{tc} x_{cl} + \epsilon_{nlt} \tag{4.2}$$

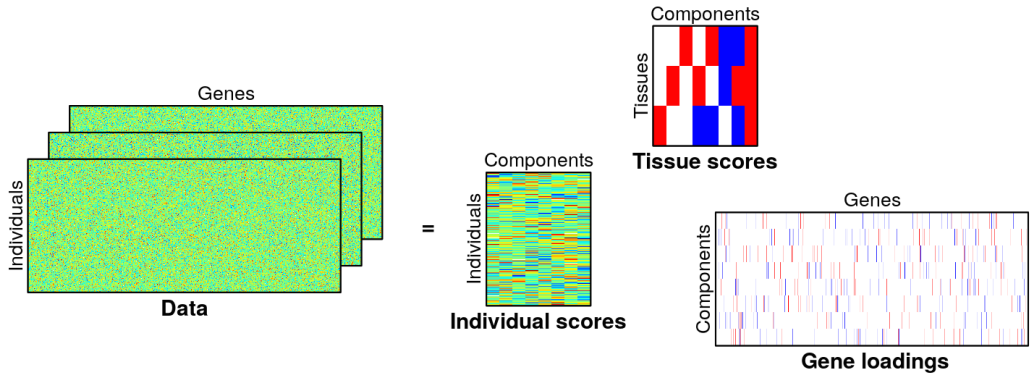

Figure 4.1: Example of a data set simulated under the PARAFAC model with $p = 0.1$ (noise not shown).

with $C = 8$ components and dimensions $N = 200$ individuals, $L = 500$ genes and $T = 3$ tissues. Three components were simulated to be active in a single tissue, a further three were active in 2 tissues and the remaining two components were active in all tissues. If a component c was active in tissue t then b_{tc} was randomly sampled from $\{-1, 1\}$, otherwise, b_{tc} was set to zero. An example data set showing the pattern of zeros in B is given in figure 4.1.

The gene loadings vectors (rows of X) were simulated to be sparse, with an element set to zero (with probability $1-p$) or drawn from $\mathcal{N}(0, 1)$ (with probability p). The parameter p determines the fraction of non-zero elements in X. This choice of distribution for the gene loading vectors does favour SPIDER, as this is exactly the spike and slab that SPIDER fits. However, it is also an obvious choice for simulating sparse vectors (e.g. Zhao et al. (2014b)). The individual scores matrix A was drawn from $\mathcal{N}(0, 1)$. Finally, a homogeneous noise model was used, with ϵ_{nlt} drawn from $\mathcal{N}(0, 10)$.

Data sets with increasing levels of sparsity were simulated with 50 data sets generated for each value of $p \in \{0.5, 0.4, 0.3, 0.2, 0.1\}$.

4.1.1.3 Post-processing and metrics

For method $\in \{\text{SPIDER}, \text{BMTF}\}$, denote the estimated individual scores matrix, tissue scores matrix and gene loadings matrix by A^{method}, B^{method} and

X^{method} respectively. The true set of scores and loadings matrices are given by A^{truth}, B^{truth} and X^{truth}.

Number of estimated components: Both methods can automatically shrink components to zero to estimate the true number of underlying components, i.e. perform model selection. Both methods were initialised with 16 components and the number of estimated components recorded to compare performance.

A set of 8 estimated components are required for the following post-processing steps. If more than 8 components were estimated, extra components were removed to leave the set most correlated with the true individual scores. If fewer than 8 components were estimated, then additional components consisting of all zeros were used to make up the difference.

Permutation indeterminacy: Both models have a scaling and permutation indeterminacy. This means that the estimated components will not necessarily be in the same order as the true components, and a direct comparison can not be made. An exhaustive search was performed to find the permutation of the estimated components which best matched the truth. The optimal permutation was selected to maximise the average (absolute) correlation between the true and estimated individual scores vectors. Once the optimal permutation was recovered, the signs of the estimated components were flipped (if necessary) so that correlations were positive. Correlations involving zero components were taken to be 0.

Root mean squared error (RMSE): Root mean squared error (RMSE) was used to evaluate similarity between the true and estimated individual scores vectors. In addition to the optimal permutation, RMSE requires that vectors be on the same scale. Scaling was performed so that the estimated and true scores vectors both had unit variance. RMSE between the true and

estimated scores matrices (after a permutation and scaling) was defined as

$$\text{RMSE} = \sqrt{\text{mean}\big((A^{\text{truth}} - A^{\text{method}})^2\big)}. \tag{4.3}$$

Sparse stability index (SSI): If the set of estimated components is very poor, it can be hard to find the best permutation. The sparse stability index (SSI) is invariant to scaling and permutation (Gao et al., 2013).

Let $\Sigma \in \mathbb{R}^{C\times C}$ be a matrix such that Σ_{ck} is the absolute correlation between the cth true individual scores vector and the kth estimated individual scores vector. Additionally let $\mathbf{s}^r \in \mathbb{R}^C$ and $\mathbf{s}^c \in \mathbb{R}^C$ be row and column means of Σ respectively. The sparse stability index is defined as

$$\begin{aligned}\text{SSI} = \frac{1}{2C}\sum_{i=1}^{C}\left[\max\big(\Sigma_{i\cdot}\big) - \frac{\sum_{j=1}^{C}\mathbb{1}\big(\Sigma_{i,j} > \mathbf{s}_i^r\big)\Sigma_{ij}}{C-1}\right] \\ + \frac{1}{2C}\sum_{j=1}^{C}\left[\max\big(\Sigma_{\cdot j}\big) - \frac{\sum_{i=1}^{C}\mathbb{1}\big(\Sigma_{i,j} > \mathbf{s}_i^c\big)\Sigma_{ij}}{C-1}\right],\end{aligned} \tag{4.4}$$

where $\mathbb{1}(\cdot)$ is an indicator function; it takes a value of 1 if the condition in the brackets is true and a value of zero otherwise.

Equation (4.4) penalises cases in which there is more than one large number in each row or column of Σ. This occurs if one of the true components is split across two components in the estimated set. The metric also penalises the case where there are no large numbers in a row or column of Σ. This arises if the estimated set of components misses one of the true components, or, if the estimated set contains a component that does not exist in the true components. A higher SSI implies a better correspondence between the two input matrices.

Receiver operating characteristic (ROC curve) Finally, the two methods were compared by evaluating recovery of the correct set of non-zero elements in the gene loadings matrix. Neither SPIDER not BMTF produce exact sparsity. For SPIDER, posterior inclusion probabilities (PIPs) were used to

threshold the loadings matrix, X^{SPIDER}, to create a set of genes with non-zero loadings. PIPs tended to be at the extremes of the set $[0, 1]$, i.e. either close to 0 or close to 1. A threshold of 0.5 was used, but any threshold in (0.1, 0.9) would give very similar results. It is less clear how to threshold the estimates for BMTF, as different thresholds on X^{BMTF} give rise to very different levels of sparsity. Rather then selecting an arbitrary threshold, a wide range of thresholds were tried, varying between the extreme cases of all zeros to no zeros in X^{BMTF}. For each threshold, power and false positive rates (FPR) were calculated, and plotted to create an ROC curve. Power and FPR point estimates for SPIDER were also evaluated. (Thresholded loadings matrices are denoted using a hat, e.g. $\hat{X}^{\text{method}}$.)

Power and false positive rates (FPR) were calculated as follows,

$$\text{Power} = \frac{\sum_{c=1}^{C}\sum_{l=1}^{3L}\mathbb{1}\big(\hat{X}_{cl}^{\text{method}} \neq 0 \text{ and } X_{cl}^{\text{truth}} \neq 0\big)}{\sum_{c=1}^{C}\sum_{l=1}^{3L}\mathbb{1}\big(X_{cl}^{\text{truth}} \neq 0\big)}, \tag{4.5}$$

$$\text{FPR} = \frac{\sum_{c=1}^{C}\sum_{l=1}^{3L}\mathbb{1}\big(\hat{X}_{cl}^{\text{method}} \neq 0 \text{ and } X_{cl}^{\text{truth}} = 0\big)}{\sum_{c=1}^{C}\sum_{l=1}^{3L}\mathbb{1}\big(X_{cl}^{\text{truth}} \neq 0\big)}. \tag{4.6}$$

A summary of the metrics used is given in table 4.1.

Metric	**Description**
Number of components estimated	Evaluates model selection.
Root mean squared error (RMSE) for individual scores matrices	Measures the absolute difference between true and estimated matrices. Requires permutation and scaling of estimated components.
Sparse stability index (SSI) for individual scores matrices	Measure of how similar two component sets are. Invariant to permutations and scalings.
ROC curve (power and false positive rate)	Evaluates recovery of sparsity in loadings matrices. Requires a permutation of the estimated components.

Table 4.1: Summary of metrics used for method comparison.

4.1.1.4 Run settings

Both methods were initialised with 16 components, double the true number of components. BMTF was run using the default settings and the posterior mean used as a point estimate. SPIDER was run using settings given in section 3.5. SPIDER was run 10 times and the set of estimates with the highest negative free energy selected.

4.1.1.5 Results

	p	**Number of estimated components**										
		1	2	3	4	5	6	7	8	9	10	11
SPIDER	0.5								21	19	8	2
	0.4								38	9	3	
	0.3								47	3		
	0.2								46	4		
	0.1					5	28	13	4			
BMTF	0.5								50			
	0.4			1	1	2	1	3	42			
	0.3			9	10	4	6	6	15			
	0.2		1	36	9	3	1					
	0.1	3	8	32	7							

Table 4.2: Frequency table showing the number of component recovered by SPIDER and BMTF across 50 data sets; the true number of components is 8. p determines the sparsity of the true components; sparsity increases as p decreases.

Table 4.2 is a frequency table showing the number of components estimated by each method across 50 data sets. The behaviour of the two methods changes as the sparsity levels increase. When the simulated data set contains dense components ($p=0.5$), SPIDER often overestimates the number of components. This is presumably a unwanted effect of adding an extra level of hierarchy into the spike and slab, making it harder to remove components. The performance of SPIDER improves as the sparsity increases however, with the correct number of component being estimated more often. For high sparsity ($p=0.1$), the

performance starts to decrease again with too few components being estimated. This is unsurprising as the signal-to-noise ratios are decreasing with increasing sparsity. BMTF accurately estimates the number of components 100% of the time when the components are dense ($p = 0.5$). However, for sparser components, this approach tends to underestimate the number of components. This may be a result of the component-level spike and slab removing components too aggressively.

It is important to note that the additional components SPIDER estimated were removed to optimise RMSE between the true and estimated individual scores vectors, so the following results are biased towards SPIDER.

Figure 4.2 shows the RMSE for estimated sets of individual scores. Each boxplot summarises results from across 50 data sets. As the sparsity increases, the performance of both models decay, with SPIDER performing better than BMTF at high sparsity levels. The multi-modal distribution appearing in some of the boxplots is caused by underestimation of the number of components. BMTF performs less well at higher sparsity levels because of this. The SSI shows a similar pattern of behaviour to RMSE (see figure 4.3).

The power and FPRs for recovery of non-zero elements in the estimated loadings matrices are shown in figure 4.4. The spike and slab distribution used in SPIDER appears to perform better at feature selection than the ARD prior used in BMTF. This is perhaps unsurprising as the spike and slab distribution is discrete, whereas the ARD prior only puts a high density near zero. However, the disparity in performance is probably mainly due to the different number of components being estimated by the two methods. The next section further compares the spike and slab approach with the ARD prior.

4.1.2 Comparison of group decompositions

In this section, I compare the group factor analysis version of SPIDER with three other methods from the literature, BGFA (Zhao et al., 2014b), CCAGFA

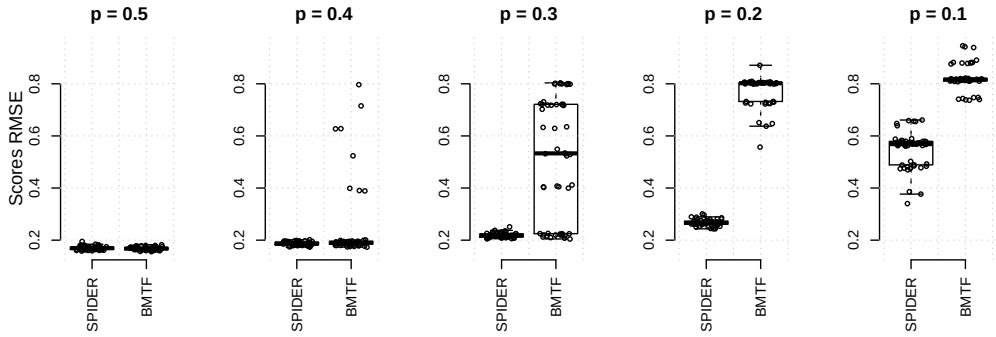

Figure 4.2: Root mean squared error for individual scores matrices. Sparsity increases from left to right. Boxplots summarise results for 50 data sets.

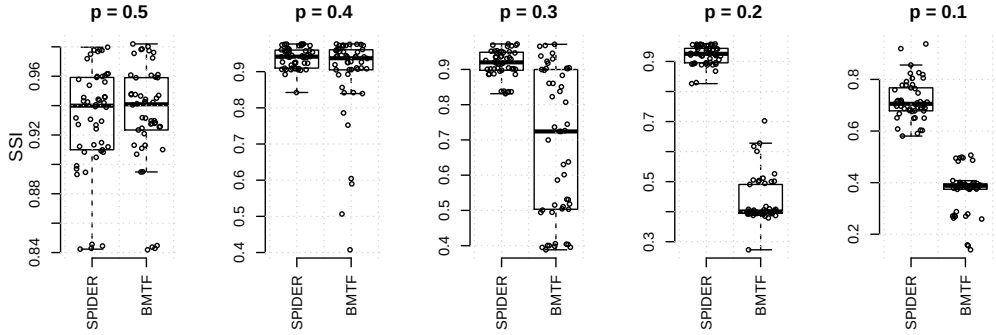

Figure 4.3: Sparse stability index (SSI) for individual scores matrices. Sparsity increases from left to right. Boxplots summarise results for 50 data sets.

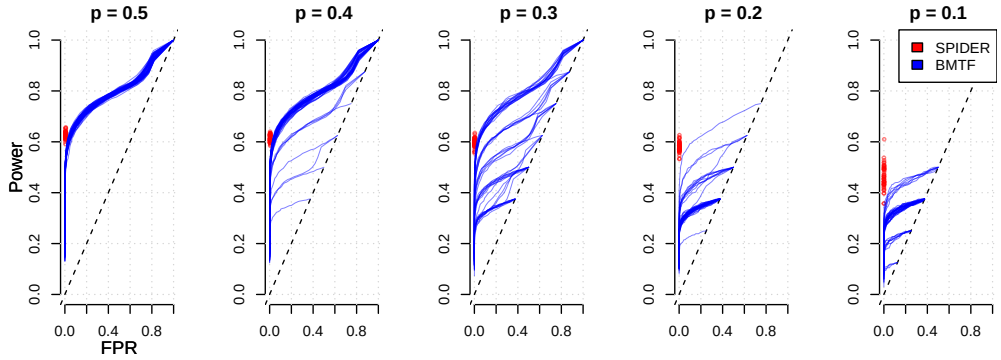

Figure 4.4: ROC curves for recovery of non-zero elements in the loadings matrices. Sparsity increases from left to right. Power and false positive rates for SPIDER were evaluated by thresholding the PIPs at 0.5 (red). A sequence of power and false positive rates for BMTF were obtained by selecting a variety of thresholds for the gene loadings estimates (blue curve). Results for 50 data sets are shown.

(Klami et al., 2014), and iClusterPlus (Mo et al., 2013). All four methods are based on group factor analysis,

$$y_{nl}^{(d)} = \sum_{c=1}^{C} a_{nc} x_{cl}^{(d)} + \epsilon_{nl}^{(d)} \tag{4.7}$$

for data $Y^{(d)} \in \mathbb{R}^{N \times L_d}$. Equation (4.7) decomposes several linked matrices to uncover individual and shared structure. The methods differ in the assumptions they make on the loadings matrices, noise and the inference scheme.

4.1.2.1 Method descriptions

BGFA: Bayesian Group Factor Analysis (BGFA) (Zhao et al., 2014b) employs a three-parameter beta prior to encourage sparsity in the loadings matrices (Armagan et al., 2011; Gao et al., 2013). Briefly, this distribution extends the Beta distribution, adding another parameter to allow it to model a wider range of densities. The BGFA formulation explicitly shrinks globally, at a factor level and at an element level in the loadings matrix,

$$\begin{aligned} x_{cl}^{(d)} &\sim \mathcal{N}(0, \theta_{cl}^{(d)}), \\ \theta_{cl}^{(d)} &\sim \pi^{(d)} \mathcal{G}(g, \delta_{cl}^{(d)}) + (1 - \pi^{(d)}) \delta(\theta_{cl}), \\ \pi^{(d)} &\sim \mathcal{B}\text{eta}(1, 1). \end{aligned} \tag{4.8}$$

$\pi^{(d)}$ defines global shrinkage, if $\theta_{cl}^{(d)}$ takes a very small value, then $x_{cl}^{(d)}$ will also be small. Heteroscedastic noise is assumed in this model.

Inference for this model has two steps. First, a Gibbs sampler is run to find a good set of initial estimates. These are then used as input for a variational expectation maximization algorithm which finds maximum a posteriori estimates. The code is available for download from
`http://beehive.cs.princeton.edu/software/`.

CCAGFA: CCAGFA is an R package[1] that implements several different models (canonical correlation analysis and group factor analysis) described in Klami et al. (2013, 2014) and Virtanen et al. (2011, 2012). The method used here is group factor analysis from (Klami et al., 2014), referred to as CCAGFA from now on.

CCAGFA places an ARD prior on the loadings matrices to encourage sparsity,

$$x_{cl}^{(d)} \sim \mathcal{N}(0, (\alpha_c^{(d)})^{-1}) \tag{4.9}$$

A noise model, $E^{(d)} \sim \mathcal{N}(e^{(d)}|0, (\tau^{(d)})^{-1})$, with a different precision variable for each data type is employed. Inference is performed using variational Bayes. Note that CCAGFA is similar to BMFT with the use of an ARD prior, although CCAGFA learns one precision variable per component, not per element. Also, CCAGFA does not use a spike and slab prior to switch off components, however components can be shrunk to zero if the component precision grows very large.

iClusterPlus: iClusterPlus is a method for joint clustering of multidimensional data (Mo et al., 2013; Shen et al., 2009, 2013). The approach performs several generalised linear regressions, with latent variables $\mathbf{z}_n$ common to all regressions. The framework explicitly allows for binary, categorical and continuous input data types by modelling the data with Binomial, multinomial, and Gaussian random variables respectively. When the input data is continuous, as in these simulations, the model is,

$$y_{nl}^{(d)} \sim \mathcal{N}(y_{nl}^{(d)}|\mathbf{z}_n \boldsymbol{\beta}_l^{(d)}, (\sigma_l^{(d)})^2), \tag{4.10}$$

$$\mathbf{z}_n \sim \mathcal{N}(0, 1), \tag{4.11}$$

[1] `http://cran.r-project.org/web/packages/CCAGFA/`

assuming the data has zero mean. A lasso penalty is placed on the loadings vectors, with tuning parameters γ, (γ can also be data type specific), to give the following penalised likelihood,

$$\max_{\boldsymbol{\beta}_l^{(d)}} \; l(y_{nlt}, \mathbf{z}_n; \boldsymbol{\beta}_l) - \sum_d \gamma ||\boldsymbol{\beta}_l^{(d)}||_1. \tag{4.12}$$

Penalised likelihood estimation is performed using a Monte-Carlo Newton-Raphson algorithm. iClusterPlus can be thought of as a non-Bayesian version of group factor analysis. Regression coefficients $\boldsymbol{\beta}_l$ and latent variables $\mathbf{z}_n$, have a similar interpretation to the variable loadings and individual scores in (4.7) respectively. iClusterPlus is available as an R package[2].

Table 4.3 summarises the differences between these methods.

Model	**Sparsity**	**Inference**
SPIDER	Spike and slab prior	Variational Bayes
CCAGFA	ARD prior	Variational Bayes
BGFA	Three parameter Beta prior	Gibbs sampling then variational expectation maximisation
iClusterPlus	Lasso penalty	Monte-Carlo Newton-Raphson

Table 4.3: Summary of the main features of the methods being compared.

4.1.2.2 Data simulation

Data was simulated for $D = 3$ data types and $N = 200$ individuals; each data type consisted of L=500 variables. Data was generated under the group factor analysis model as a linear combination of $C = 8$ underlying components and additive noise as in (4.7). Note that the three data types could represent three tissues, and $L = 500$ variables could be expression levels for 500 genes.

Of the 8 components simulated, three were active in just one data type, a further three were active in two of the data types and the remaining two

[2] `https://www.bioconductor.org/packages/release/bioc/html/iClusterPlus.html`

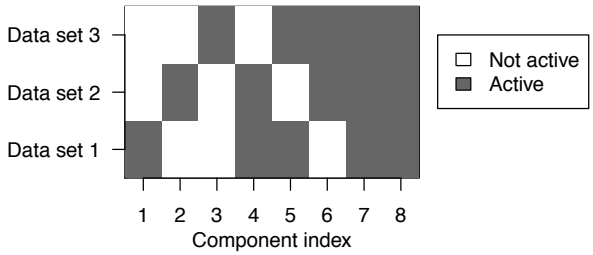

Figure 4.5: Pattern of component activity across data types.

were active in all three data types. (A component is said to be 'active' in a data type if it contributes to the variance in that data matrix.) Figure 4.5 summarises the component activity patterns. If a component was not active in a particular data type, then the relevant row of the loadings matrix was set to zero. For components which were active, their loadings vectors were sparse, with 90% of the elements equal to 0. The non-zero elements were drawn from $\mathcal{N}(0, 1)$.

Elements in the individual scores matrix A were also drawn from $\mathcal{N}(0, 1)$. Finally, a homoscedastic noise model with precision λ was used, i.e. $\epsilon_{nl}^{(d)} \sim \mathcal{N}(0, \lambda^{-1})$ for $d \in \{1, 2, 3\}$. In order to evaluate the performance of the methods at different signal-to-noise levels, data was simulated with three different values of λ, (0.1, 0.2 and 0.3).

For each of the three noise levels, 50 data sets were simulated. Signal-to-noise ratios for each data set were calculated as follows,

$$SNR = \frac{\sum_d \text{trace}\big((AX^{(d)})(AX^{(d)})^t\big)}{\sum_d \text{trace}(E^{(d)}E^{(d)t})}. \tag{4.13}$$

Values of $\lambda = 0.3, 0.2$ and 0.1 corresponded to signal-to-noise ratios (averaged over 50 data sets) of 0.15, 0.1 and 0.05 respectively.

4.1.2.3 Run settings

All methods can estimate the number of underlying components in the data, however for simplicity, the models were initialised using the true number of

components. If fewer than 8 components were estimated, then components consisting of zeros were added to make up the difference.

Default parameter settings for BGFA, CCAGFA and iClusterPlus were used. The parameter setting described in section 3.5 were used for SPIDER. SPIDER, BGFA and CCAGFA were all run 10 times and the set of estimates that resulted in the largest value of the negative free energy selected. Posterior means were used as point estimates for SPIDER and CCAGFA. BGFA evaluates a maximum a posteriori estimate.

The iClusterPlus algorithm requires selection of a tuning parameter (γ) which determines the sparsity of the resulting components. Rather than selecting a single value for γ, iClusterPlus was run 20 times with different values for γ in $[0, 1]$. This range covered the extreme cases of complete sparsity to very dense components. As this procedure was slow, results for iClusterPlus were only performed for one noise level, $\lambda = 0.1$.

4.1.2.4 Post-processing and metrics

Estimated and true loadings matrices were concatenated to create matrices of dimensions C by $3L$, then the post-processing steps described in section 4.1.1.3 were used. Sparse estimates for SPIDER were obtained by thresholding PIPs at 0.5. BGFA generated sparse loadings so no thresholding was required. CCAGFA does not give exact sparsity so a sequence of thresholds was used to create ROC curves (as for BMTF in the previous simulations). iClusterPlus gives sparse estimates; power and FPRs were calculated for each value of γ and plotted to create an ROC curve. For the RMSE and SSI statistics, the value of γ that resulted in a FPR of 0.01 is reported for iClusterPlus.

4.1.2.5 Results

Figures 4.6 and 4.7 show RMSE and SSI for the estimated individual scores matrices at 3 different noise levels. The multi-modal distribution seen in some

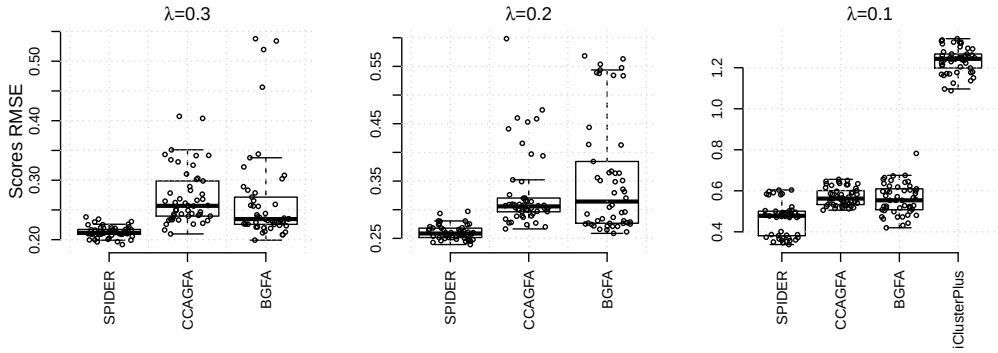

Figure 4.6: Root mean squared error (RMSE) for recovered individual scores matrices. Boxplots summarise results from across 50 data sets. Noise levels in the simulated data sets increase from left to right.

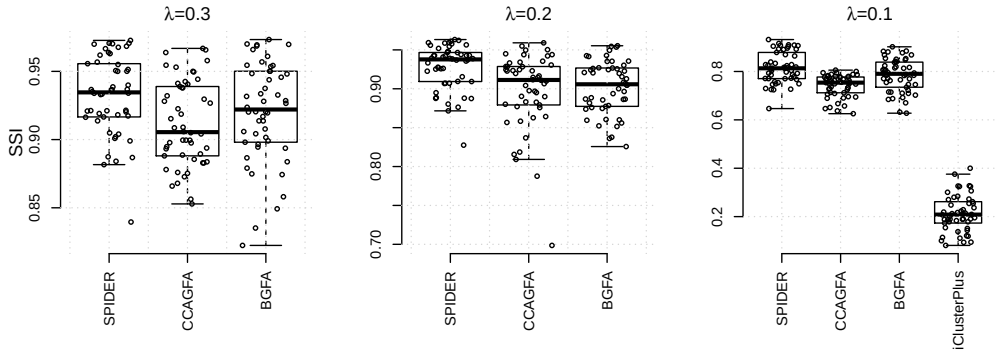

Figure 4.7: Sparse stability index (SS1) for recovered individual scores matrices. Boxplots summarise results from across 50 data sets. Noise levels in the simulated data sets increase from left to right.

of these boxplots is again due to an incorrect number of components being recovered. RMSE and SSI degrade as the noise levels increase, with SPIDER slightly outperforming BGFA and CCAGFA. iClusterPlus does not recover the individual scores matrix as well as the other approaches. However, it should be noted that the tuning parameter was selected to optimise metrics on the loadings matrices rather then individual scores matrix.

Power and FPRs for the recovery of non-zero elements in the loadings matrices are shown in the ROC curves in figure 4.8. SPIDER, CCAGFA and BGFA perform similarly. BGFA appears to shrink aggressively, resulting in a very low FPRs at the expense of power. SPIDER on the other hand

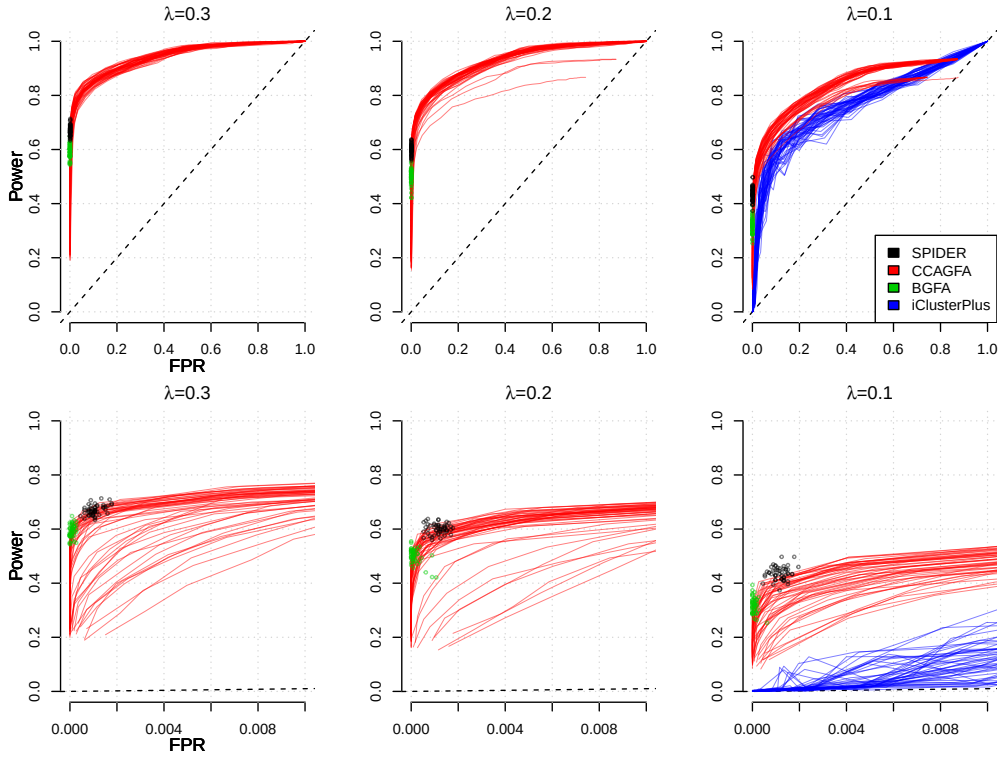

Figure 4.8: ROC curves for recovery of non-zero elements in the loadings matrices. Point estimates for SPIDER obtained by thresholding the PIPs at 0.5. No thresholding was required for BGFA as the raw estimates have exact sparsity. Results for CCAGFA were generated using a range of thresholds on the same estimate set, resulting in an ROC curve. The ROC curve for iClusterPlus was generated by running the method multiple times with different tuning parameters. Noise levels in the simulated data sets increase from left to right. The top and bottom rows of plots only differ in their x-axis range.

shrinks slightly less strictly resulting in more power but higher FPRs compared to BGFA. Performance of CCAGFA depends heavily on the threshold, but performs well at some thresholds. iClusterPlus does not recover sparsity in the loading matrices as well as the other methods.

4.1.3 Discussion

This simulation study suggests that SPIDER performs comparably to, or better than, existing approaches in most scenarios, especially when signals are sparse.

In a comparison of sparse tensor decompositions, SPIDER struggled to cor-

rectly estimate the number of underlying components, overestimating the true number when the components were dense. BMTF on the other hand correctly estimated the number of components in these dense cases. This is a definite advantage of the component-level spike and slab prior. For data containing sparse components however, this component-level shrinkage appeared to be too aggressive, and resulted in the poorer performance of BMTF across several metrics. In applications such as gene expression data, sparse components recovering sparse signals are likely to be the most interesting. These simulations suggest that the element-wise spike and slab prior is better at recovering sparse components, and that SPIDER may be more appropriate for analysing gene expression experiments.

The spike and slab prior appears to recover element-wise sparsity in the data better than approaches using the ARD prior. The ARD prior struggles to shrink elements very close to zero. In addition, the spike and slab prior estimates PIPs, which is a nice way to introduce exact sparsity. SPIDER slightly outperforms BGFA in some scenarios. The variational expectation maximisation algorithm of BGFA, which estimates the posterior mode only, may damage performance because it can not take uncertainty into account.

It is important to note that the methods compared here are just a subset of the sparse tensor decomposition and group factor analysis implementations in the literature. In addition, the simulation settings used here probably favoured SPIDER. A more comprehensive set of simulations with a wider range of distributions for the sparse loadings vectors should be investigated. For example, the loadings vectors could be simulated using a 'spike' and uniform distribution. Additional metrics could also be employed to further compare different methods, for example, Khan et al. (2014b) present simulations with performance evaluated using predictive accuracy.

4.2 Trans effect simulations

This section describes a simulation study to evaluate whether SPIDER has the power to find *trans* effects in gene expression data. The simulated data consisted of genotypes and gene expression data in several tissues, containing a variety of signals including *trans* effects, *cis* effects and confounding factors. The tensor decomposition version of SPIDER was compared to individual matrix decompositions which analysed data for each tissue independently. In addition, to test robustness to missing data, the tensor decomposition was run with a subset of the samples hidden.

4.2.1 Data simulation

Simulated data consisted of genotype and gene expression data for $N = 700$ related individuals (150 monozygotic twin pairs, 150 dizygotic twin pairs and 100 singletons). Gene expression data was simulated at $L = 2,500$ genes in $T = 3$ tissues and contained both non-genetic signals (noise and confounding factors) and genetic signals (*cis* and *trans* effects). It was assumed that each gene contained only one SNP; this simplified case is equivalent to assuming that there is at most one *cis* eQTL for each gene.

4.2.1.1 Genotypes

Of the $L = 2,500$ SNPs simulated, $C_{\text{cis}} = 500$ (20%) were randomly selected to be *cis* eQTLs. A *cis* eQTL partially determined the expression of its nearby gene. A subset of the *cis* eQTLs were additionally assumed to be *trans* eQTLs. *Trans* eQTLs were not only associated with a nearby gene (via a *cis* effect) but also multiple other genes, creating a *trans* network.

Let $G \in \mathbb{R}^{N \times L}$ be the matrix of simulated genotype data. Genotypes were simulated under the Hardy-Weinberg equilibrium with a minor allele frequency (MAF) drawn uniformly from $[0.05, 0.5]$ (unless the SNP was also a *trans* eQTL in which case MAF $= 0.3$). Monozygotic twins shared all their

genetic material and dizygotic twins shared half of their genetic material. All SNPs were sampled independently.

4.2.1.2 Gene expression

The simulated gene expression data ($\mathcal{Y}^{N\times L\times T}$) consisted of noise, confounding factors, *cis* and *trans* effects. The vector of expression levels for gene l in tissue t is denoted $\mathbf{y}_{lt}$. The data was simulated in stages, noise, confounding factors and *cis* effects were generated initially. These three signals were then combined, and *trans* effects incorporated.

Noise: The following model was used to simulate noise,

$$\mathbf{y}_{lt}^{\text{noise}} \sim \mathcal{N}(0, \lambda_{lt}^{-1}),\ \sqrt{\lambda_{lt}^{-1}} \sim \mathcal{G}(100, 0.01). \tag{4.14}$$

The standard deviation of the noise $\sqrt{\lambda_{lt}^{-1}}$ was drawn from a Gamma distribution with mean 1 and low variance. Although this model does define heteroscedastic noise, there is little variation in the noise precision across genes and tissues. (Note that λ_{lt} is the noise precision for gene l in tissue t.)

Confounding factors: $C_{\text{cf}} = 10$ independent confounding factors were simulated as follows,

$$\mathbf{y}_{lt}^{\text{cf}} = \sum_{c=1}^{C_{\text{cf}}=10} \mathbf{a}_c b_{tc} x_{cl} \quad \text{for} \quad l \in \{1, \ldots, L\},\ t \in \{1, 2, 3\}$$

$$\mathbf{a}_c \sim \mathcal{N}_N(\mathbf{a}_c|0, I_N),\ |b_{tc}| \sim \mathcal{G}(100, 0.01),\ x_{cl} \sim 0.5\mathcal{N}(x_{cl}|0, 0.1) + 0.5\delta_0(x_{cl}) \tag{4.15}$$

where the sign of b_{tc} was randomly selected. Note that the confounding factors were simulated under the PARAFAC model with a sparsity level of 50%.

***Cis* effects:** Let $\mathbf{g}_l \in \mathbb{R}^N$ be the vector of simulated genotypes for SNP l. Supposing that SNP l was a *cis* eQTL, then its contribution to the expression

of gene l was given by

$$\mathbf{y}_{lt}^{\text{cis}} = \hat{\alpha}_l \tilde{\alpha}_t \mathbf{g}_l, \tag{4.16}$$

where

$$\begin{cases} |\hat{\alpha}_l| = \phi_l^{\text{cis}} \\ \tilde{\alpha}_t = 1 \end{cases} \quad \text{if SNP } l \text{ also acted as a } \textit{trans} \text{ eQTL}$$
$$\begin{cases} |\hat{\alpha}_l| \sim \mathcal{G}(4, 0.1) \\ \tilde{\alpha}_t \sim \mathcal{B}\text{ernoulli}(0.5) \end{cases} \quad \text{otherwise.} \tag{4.17}$$

$\hat{\alpha}_l$ can be thought of as an effect size (the effect direction is random) and $\tilde{\alpha}_t$ as a binary value indicating whether the *cis* effect was active in tissue t. A different effect size was used depending on whether the eQTL was also a *trans* eQTL or not; this is discussed more later. A *cis* effect was active in a particular tissue with probability 0.5, (unless the eQTL was also a *trans* eQTL, in which case it was active in every tissue, although it did not necessarily target downstream genes in every tissue). $\mathbf{y}_{lt}^{\text{cis}}$ was set to zero if SNP l was not a *cis* eQTL.

Combining noise, confounding factors and *c*is effects: Simulated noise, confounding factors and *cis* effects were combined additively to get a temporary set of expression levels for a gene l in tissue t,

$$\mathbf{y}_{lt}^{\text{tmp}} = \mathbf{y}_{lt}^{\text{noise}} + \mathbf{y}_{lt}^{\text{cf}} + \mathbf{y}_{lt}^{\text{cis}}. \tag{4.18}$$

***T*rans effects:** Finally, *trans* effects were simulated. *Trans* associations between SNPs and distant genes were created as follows; the *trans* SNP regulated a nearby gene (via a *cis* effect), it was further assumed that this gene was a transcription factor (abbreviated as TF) and regulated multiple downstream genes (called target genes). Importantly, the *trans* eQTL was only indirectly

associated with the target genes.

Data sets were simulated to contain 20 *trans* effects. The number of target genes in the *trans* networks (M_{trans}) varied, and target genes were selected at random. (For simplicity, a gene acting as a TF in a *trans* effect could not additionally be involved in another *trans* effect, however any of the other genes – including those regulated by a regular *cis* eQTL – could be regulated by any number of TFs.) *Trans* effects were active in one, two or three tissues. If the *trans* effect was active in several tissues, the same network of genes was regulated in each tissue.

Let l be a gene, and S_l be the set of TFs that regulate it. *Trans* effects were simulated as follows,

$$\begin{aligned} \mathbf{y}_{lt}^{\text{trans}} &= \sum_{j \in S_l} \hat{\beta}_{lj} \tilde{\beta}_{tj} \mathbf{y}_{jt}^{\text{tmp}} \\ |\hat{\beta}_{lj}| &\sim \mathcal{G}(\psi^{trans}, 0.02) \end{aligned} \tag{4.19}$$

where $\hat{\beta}_l$ was the relative effect of TF j on gene l (with a random effect direction). $\hat{\beta}_{tj}$ was equal to 1 if the TF j was active in tissue t and 0 otherwise. In order to investigate a variety of scenarios, of the 20 *trans* effects simulated, 12 were active in just one tissue, 4 were active in 2 tissues and the remaining 4 were active in all tissues. If l was not a target gene for any of the 20 TFs (i.e. S_l was the empty set) then $\mathbf{y}_{lt}^{\text{trans}} = 0$.

Selecting parameters for the *trans* effects: Three parameters determine the strength of a simulated *trans* effect: the effect of the *cis* eQTL on the TF (ϕ^{cis}); the effect of the TF on the target genes (determined by ψ^{trans}) and the number of target genes (M_{trans}). In order to make the data as realistic as possible, these parameters were selected so that signal strengths were similar to those seen in real data sets. The real *trans* signal used as a reference was the *KLF14 trans* signal (Small et al., 2011).

For half of the *trans* effects, ϕ^{cis} and ψ^{trans} were selected to match the effect

sizes seen in the *KLF14 trans* signal. The remaining 10 *trans* effects were weaker, for these signals, values of the parameters were halved. *Trans* effects had either 150 or 75 target genes (6% or 3% of all genes). Table 1 gives a summary of the *trans* effects simulated.

Index	ϕ^{cis}	ψ^{trans}	M_{trans}
1-5*	0.6	20	150
6-10**	0.3	10	150
11-15*	0.6	20	75
16-20**	0.3	10	75

Table 4.4: Summary of the parameters used to simulate *trans* effects. *Trans* effects were grouped into sets of 5 according to their signal strength (ϕ^{cis} and ψ^{trans}) and number of target genes (M_{trans}). $*$ indicates a signal strength that matches the *KLF14 trans* signal and $**$ indicates a weaker signal. Within the groups they were further split according to their activity in different tissues; 3 were active in just one tissue, 1 was active in 2 tissues and 1 was active in all three tissues.

Combining all of the data: Finally, the contribution from the *trans* effects was incorporated to create a final set of simulated expression levels,

$$\mathbf{y}_{nl}^{\text{final}} = \mathbf{y}_{nl}^{\text{tmp}} + \mathbf{y}_{nl}^{\text{trans}}. \tag{4.20}$$

A summary of the simulation parameters are given in table 4.5. A total of 50 data sets were simulated.

Parameter	Value	Description
T	3	Number of tissues
N	700	Number of individuals
L	2,500	Number of genes
C_{cf}	10	Number of confounding factors
C_{cis}	500	Number of *cis* effects
C_{trans}	20	Number of *trans* effects

Table 4.5: Simulation parameters.

4.2.2 The method

Several different versions of SPIDER were run, see summary in table 4.6. The tensor decomposition was run with a Gaussian prior on the individual scores matrix (denoted T_G) and also a kinship-informed prior on the individual scores matrix (T_K) (see details in section 3.4.2). In addition, individual matrix decompositions of each tissues were performed using SPIDER with $T = 1$; this approach is an implementation of sparse factor analysis.

Performance in the presence of missing samples was also tested. Up to 2 samples were removed for each individual at random such that only 75% of the data remained. Two versions of SPIDER were run in this scenario (i) the extension to deal with missing samples given in section 3.4.1.2, which essentially ignores the missing data, and (ii) a naive approach that removes any individual with missing samples, then runs a tensor decomposition on the remaining data. On average, only 350 individuals had complete data. For both of these methods, a Gaussian prior was used for the individual scores matrix. These two approaches, (i) and (ii), are denoted T_G^i and T_G^r respectively (i for ignore and r for remove).

All methods were run with the initial number of components set to 100; this was a sufficient number for all the models to recover the *trans* effects and confounding factors. It was not expected that the methods pick up all the *cis* eQTLs. Parameter settings are given in section 3.5.

All methods were run 10 times with random initialisations.

4.2.3 Post-processing and metrics

The aim of these simulations was to investigate recovery of the underlying signals in the data. Several metrics were utilised to evaluate recovery of the confounding factors and *trans* effects. For the *trans* effects, both recovery of the causal SNP and also the set of target genes was evaluated.

Although variational Bayes is a deterministic algorithm, there is no guar-

	Method	Description
Complete data	T_G	Tensor decomposition with a Gaussian prior on the individual scores matrix.
	T_K	Tensor decomposition with a mixture of the Kinship matrix and identity as the prior on the individual scores matrix (see section 3.4.2).
	M_G	Matrix decompositions on data for each tissue separately. Gaussian prior on the individual scores matrices.
Missing data	T_G^i	Missing samples are ignored in the model likelihood (see section 3.4.1.2).
	T_G^r	Individuals with any missing data were removed leaving a set of individuals with complete data.

Table 4.6: Different versions of SPIDER run on simulated data.

antee that different initialisations will result in the same set of component estimates. The negative free energy can be used to select the 'best' run. An alternative approach suggested here attempts to combine results from across multiple runs. The idea is to average similar components from across multiple runs of the method to get a set of 'robust' components. Hierarchical clustering was used to group similar components.

4.2.3.1 Clustering

Hierarchical clustering was performed as follows; initially all estimated components (from across 10 runs) were placed in their own cluster, then at each step the two clusters deemed most similar were combined. The `R` function `hclust` was used for clustering, with a similarity metric given by the absolute correlation between individual scores vectors (the dissimilarity metric used by `hclust` was 1 - the absolute correlation). The algorithm was terminated when similarity between any two clusters dropped below a threshold of 0.5. Only clusters containing 5 or more components were used in further analysis.

Once components had been clustered, an 'averaged component' was cre-

ated for each cluster. An averaged individual scores vector was obtained by averaging (via the mean) the set of individual scores vectors in the cluster (after normalisation to unit variance and zero mean, and flipping the sign of the components to ensure they were all positively correlated). An averaged gene loadings vector was calculated in the same way. Averaged tissue scores were also calculated like this, but not normalised to have unit variance. The PIP vectors did not require scaling or sign flipping; they were averaged using the median. Finally, elements in the averaged gene loadings vectors were set to zero if their corresponding (averaged) PIPs were below 0.5.

The following performance metrics were applied to the component estimates from the 'best' run based on the negative free energy, and the averaged components from the clustering approach.

4.2.3.2 Confounding factors

To assess whether the models recovered the confounding factors, a search was performed to find the set of estimated components that best explained the true confounding. This was performed by maximising the absolute correlation between the true confounding factors, and the estimated individual component scores (via a greedy algorithm), resulting in a set of 10 estimated components that looked most like the true confounding factors. Recovery was then evaluated by calculating the average absolute correlation between these estimated individual scores vectors and the truth.

4.2.3.3 GWAS

The gene loadings vector for a component defines a set of genes (i.e. genes with PIPs > 0.5). The corresponding individual scores vector identifies the extent to which the component is on or off in each individual. For example, if the gene set consisted of correlated genes whose expression was associated with insulin production, then the individual scores vector might be a proxy

variable for insulin levels. Similarly, if the gene set consisted of genes regulated by a TF, then the scores vector might reflect variation in the expression of the TF. Further, if the TF was itself regulated by a SNP, then the scores vector may cluster individuals according to the allele count at that SNP.

Each individual scores vector can be treated as a phenotype in a genome-wide scan for association. (This idea was independently discovered, but a very similar approach is taken in Rotival et al. (2011), where independent component analysis is performed on a matrix of expression levels.) SNPs significantly associated with an individual scores vector have an interpretation as driving variation in the gene set identified by the component. In the first example given above, a significant SNP might suggest a genetic basis for insulin production. In the other example, a significant SNP might be an eQTL for the TF.

In these simulations, if an individual scores vector was significantly associated with a SNP simulated to be a *trans* eQTL, then the *trans* signal was said to be recovered by the component. A p-value threshold of 2×10^{-7} was used to determine significance. GWAS were performed using a mixed model (Zhou and Stephens, 2014).

As an aside, it is worth comparing this GWAS approach using individual scores vectors to an exhaustive search for *trans* eQTLs. Looking for *trans* eQTLs requires millions of univariate tests between all SNP-gene pairs whereas here, tests are performed between each SNP-component pair. If the number of components is smaller than the number of genes, fewer tests are required, and the multiple testing burden is relieved somewhat.

The fraction of each type of *trans* effect recovered (across 50 simulated data sets) was recorded. Results for the matrix decomposition were averaged across the three tissues, taking into account the fact that the signal may not have existed in all three data types.

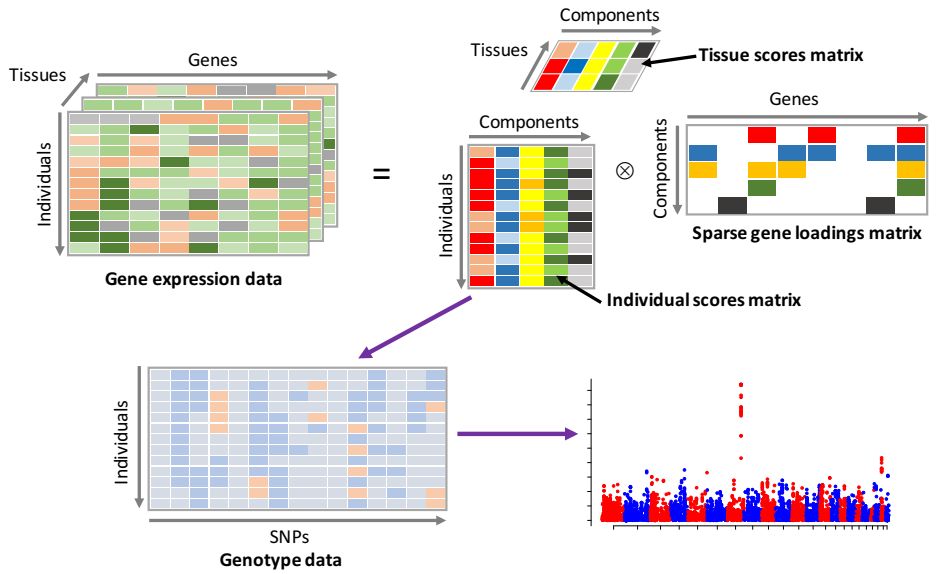

Figure 4.9: Illustration of the tensor decomposition, with individual scores vectors being used in a genome-wide scan to identify genetic variants that drive gene networks.

4.2.3.4 Power to detect regulated genes

Once a component had been identified as describing a *trans* effect (based on a GWAS hit at a *trans* eQTL), the component was further investigated to evaluate whether the correct set of target genes had been recovered. Power and false positive rates (FPR) were used to compare the genes in the component, (those with PIPs>0.5), with the true set of target genes. For a *trans* effect with vector of PIPs given by $\hat{\mathbf{s}}$, power and FPR were defined as

$$\text{Power} = \frac{\sum_l \mathbb{I}(\tilde{\mathbf{s}}_l = 1, \hat{\mathbf{s}}_l > 0.5)}{\sum_l \mathbb{I}(\tilde{\mathbf{s}}_l = 1)} \tag{4.21}$$

$$\text{FPR} = \frac{\sum_l \mathbb{I}(\tilde{\mathbf{s}}_l = 0, \hat{\mathbf{s}}_l > 0.5)}{\sum_l \mathbb{I}(\tilde{\mathbf{s}}_l = 0)} \tag{4.22}$$

where $\tilde{\mathbf{s}}$ is a binary vector of length L such that $\tilde{\mathbf{s}}_l = 1$ if gene l is a target gene in the *trans* signal and $\tilde{\mathbf{s}}_l = 0$ otherwise. ($\mathbb{I}$ is an indicator function.) Results for the matrix decompositions were averaged across tissues, taking into account the number of tissues each *trans* effect was active in.

4.2.3.5 Combining factors

In some situations, a single *trans* effect, active in multiple tissues, was modelled by several components. This occurs because – although the simulated *trans* effects act on the same set of target genes in each tissue – the contribution to the expression of the target genes varies. It is easy to see why this happens by considering the expression of the TF in each tissue. Expression of the TF depends on the genotype (an effect which is shared across tissues), and two tissue independent effects; confounding factors and noise. When these latter two effects are dominant, the expression of the TF is largely uncorrelated across tissues and its effect on the target genes in each tissue will differ. The model then treats *trans* effect as a different signal in each tissue, resulting in the signal being recovered in several components. When this happened, (i.e. when several components had individual scores vectors significantly associated with the same SNP), the components were combined by averaging the scores and loadings vectors.

4.2.3.6 Results

Figure 4.10 shows the average correlation between the true and estimated component scores recovering confounding factors. The boxplots summarise results from across 50 data sets. As expected, a joint analysis via a tensor decomposition (T_G and T_K) outperforms an analysis of each tissue separately (M_G). Even when data is missing (T_G^i and T_G^r), confounding factor recovery is good.

Table 4.7 summarises results of *trans* effect recovery for *trans* signals with signal strength half that of the *KLF14* signal. The recovery of *trans* effects with signal strengths similar to the *KLF14* signal is almost perfect (see appendix B.2). Table 4.7 gives the fraction of each type of *trans* effect recovered (over 50 data sets). *Trans* effects in single tissues were harder to recover than *trans* effects in multiple tissues, and for these signals, performance of the tensor and

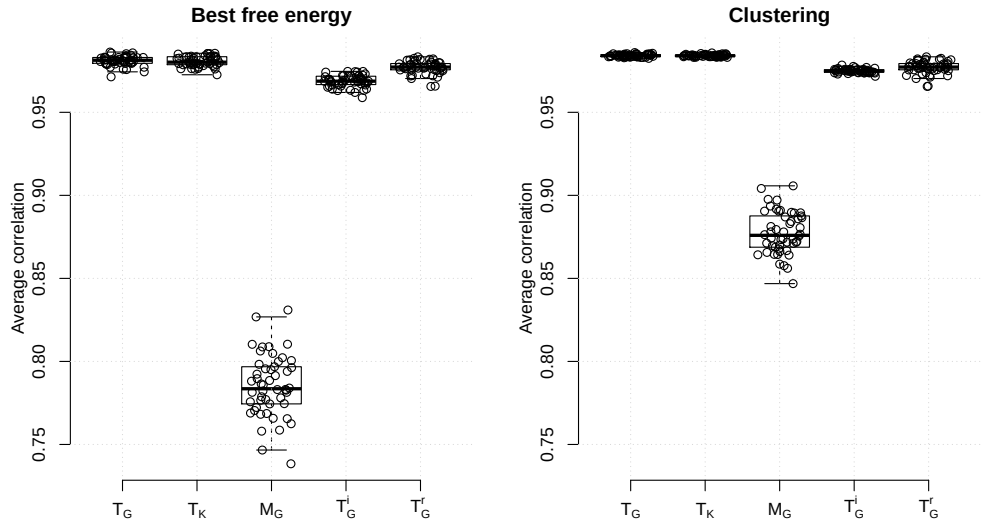

Figure 4.10: Average correlation between estimated individual scores vectors describing confounding factors and the true confounding. Boxplots summarise results across 50 simulated data sets for each method.

matrix approaches are comparable. However, for the *trans* effects active in two or three tissues, the tensor approaches performed considerably better than the matrix decomposition. This is likely a result of the tensor decomposition pooling information from across multiple tissues, and also better explaining confounding.

Results for the tensor decomposition with different priors on the individual scores matrix (T_G and T_K) show no obvious difference (table 4.7). With high levels of noise and confounding in the data, it is unclear how heritable the *trans* signals actually are. More simulations comparing these two priors are given in appendix B.3. In these simulations, the mixture prior recovers the underlying signals in the data only slightly better than the Gaussian prior, showing that the Gaussian prior is surprisingly flexible. Table 4.7 also shows that the clustering approach to combine results across multiple runs slightly outperforms the results for the best free energy run.

Conditional on *trans* effects being recovered by the models, the power to find the target genes involved is given in table 4.8. Again, only results for

M_{trans}	#tiss	Best free energy			Clustering		
		T_G	T_K	M_G	T_G	T_K	M_G
150	1	0.40	0.39	0.37	0.39	0.39	0.37
	2	0.82	0.82	0.36	0.82	0.84	0.39
	3	0.92	0.94	0.35	0.96	0.96	0.34
75	1	0.34	0.31	0.41	0.43	0.44	0.42
	2	0.70	0.78	0.39	0.80	0.76	0.39
	3	0.90	0.94	0.41	0.98	0.94	0.43

Table 4.7: Fraction of *trans* effects recovered. Results averaged across 50 data sets with no missing samples. Three approaches are compared; T_G and T_K perform tensor decompositions with a Gaussian prior and Kinship-informed prior on the individual scores matrix respectively, and M_G performs matrix decompositions on data for each tissue separately, with a Gaussian prior on the individual scores matrices. Only results for *trans* signals of strength half that of the *KLF14* *trans* signal are shown. The best result in each row is highlighted in red.

M_{trans}	#tiss	Best free energy			Clustering		
		T_G	T_K	M_G	T_G	T_K	M_G
150	1	0.84(0.04)	0.85(0.04)	0.68(0.18)	0.85(0.03)	0.85(0.03)	0.66(0.15)
	2	0.84(0.03)	0.83(0.07)	0.71(0.15)	0.84(0.04)	0.84(0.04)	0.59(0.22)
	3	0.83(0.06)	0.85(0.04)	0.64(0.19)	0.85(0.04)	0.85(0.04)	0.62(0.20)
75	1	0.78(0.10)	0.77(0.11)	0.71(0.13)	0.77(0.06)	0.77(0.10)	0.70(0.12)
	2	0.80(0.10)	0.80(0.09)	0.67(0.16)	0.80(0.06)	0.80(0.07)	0.68(0.17)
	3	0.82(0.05)	0.82(0.06)	0.71(0.13)	0.83(0.05)	0.83(0.05)	0.69(0.13)

Table 4.8: Power to find target genes in *trans* effects, conditional on the *trans* eQTL being recovered. Results averaged across 50 data sets with no missing samples. Three approaches are compared; T_G and T_K perform tensor decompositions with a Gaussian prior and Kinship-informed prior on the individual scores matrix respectively, and M_G performs matrix decompositions on data for each tissue separately, with a Gaussian prior on the individual scores matrices. Only results for *trans* signals of strength half that of the *KLF14* *trans* signal are shown. The best result in each row is highlighted in red.

M_{trans}	#tiss	Clustering	
		T_G^i	T_G^r
150	1	0.21	0.05
	2	0.58	0.28
	3	0.78	0.54
75	1	0.20	0.01
	2	0.56	0.14
	3	0.80	0.38

Table 4.9: Fraction of *trans* effects recovered. Results averaged across 50 data sets in which 25% of samples were missing. Two approaches to dealing with missingness are compared, T_G^i ignore the missing samples in the data likelihood (see section 3.4.1.2) and T_G^r retains only individuals with complete data. Only results for *trans* signals of strength half that of the *KLF14 trans* signal are reported.

signals with half the signal strength of the *KLF14 trans* signal are presented. Power to recover target genes is consistently high, and appears fairly independent of the number of target genes (M_{trans}) and activity of the *trans* effect across tissues. The tensor decomposition results in a higher power to find *trans* effects compared to the matrix decompositions. Clustering does not appear to have any benefits over the best free energy run in terms of power. False positive rates were consistently below 0.5% for all methods (data not shown).

The performance of SPIDER when the data contained missing samples was also investigated. T_G^i uses information from all individuals, irrespective of whether they have missing samples, whereas T_G^r removes individuals with any missing data before running the tensor decomposition. (On average, after removing individuals with any missingness, only 350 individuals remained.) Not surprisingly, T_G^i consistently outperforms T_G^r, recovering more *trans* effects, with a higher power to find the target genes (see tables 4.9 and 4.10). In fact, the power to find target genes using T_G^i is comparable to the matrix decomposition approach M_G, on the complete data set. These results show the benefits of a method that can deal with missing samples. Again, false positive rates for these methods are very low ($< 0.5\%$).

The complete set of results for these simulations is given in appendix B.2.

M_{trans}	#tiss	Clustering	
		T_G^i	T_G^r
150	1	0.78(0.04)	0.56(0.05)
	2	0.69(0.13)	0.60(0.07)
	3	0.71(0.07)	0.61(0.05)
75	1	0.68(0.09)	0.35(0.05)
	2	0.67(0.11)	0.38(0.10)
	3	0.68(0.11)	0.47(0.08)

Table 4.10: Power to find target genes in *trans* effects, conditional on the *trans* eQTL being recovered. Results averaged across 50 data sets in which 25% of samples were missing. Two approaches to dealing with missingness are compared, T_G^i ignores the missing samples in the data likelihood (see section 3.4.1.2) and T_G^r retains only individuals with complete data. Only results for *trans* signals of strength half that of the *KLF14 trans* signal are reported.

4.2.4 Discussion

This simulation study was designed to evaluated whether tensor decompositions have the power to recover *trans* effects in gene expression data. Simulated *trans* effects with signal strengths equal to, and smaller than, known *trans* signals were reliably recovered using SPIDER. The tensor decomposition, which jointly analyses data from across several tissues, was shown to outperform a matrix decomposition that analysed data in one tissue at a time. Even with 25% of the samples missing, SPIDER recovered many of the underlying *trans* effects. A clustering approach that combined component estimates from across multiple runs of the method boosted performance compared to using the run with the highest negative free energy.

Often, a single *trans* signal, active in several tissues, was recovered in multiple components. This is due to high levels of confounding and noise, resulting in TF expression levels being very different across tissues. This suggests that the tensor decomposition assumes an incorrect structure for the data. A more general model, group factor analysis, would allow for each recovered signal to consist of a different set of genes. For the data simulated here, a loadings matrix for each tissue is not necessary however, instead, a model with a single

shared loadings matrix but a different individual scores matrix for each tissue might be more appropriate. This would model *trans* signals with a different patterns of TF expression across different tissues, but the same set of target genes. There are undoubtedly differences between real *trans* signals and the signals simulated here however, and it is not obvious which model would best fit the real data.

Tensor decompositions provide several advantages over existing methods for identifying *trans* eQTLs. They allow confounding factors and biological signals to be recovered jointly, decreasing the risk that interesting signals are accidentally attributed to unwanted variation. The tissue specificity of signals in the data can also be recovered. Furthermore, data reduction using latent variable models, and then performing GWAS, reduces multiple testing burdens considerably. In the next chapter, SPIDER is used to recover *trans* signals, including the *KLF14 trans* signal, in a real multi-tissue gene expression study,.

Chapter 5

Results

In the previous chapter I described some simulations evaluating whether SPIDER could recover *trans* effects from noisy gene expression data. Simulated *trans* effects consisted of a network of genes regulated by a SNP. In order to identify the causal SNP, I treated the estimated individual scores vectors as phenotypes in a genome-wide scan for association.

In this chapter I present an analysis of real gene expression data in multiple tissues. I use a similar post-processing GWAS step to find potential SNPs that drive structure in the component estimates. I also investigate the set of estimated components by testing for association with measured phenotypes (e.g. age, BMI) and experimental variables, and perform gene ontology analysis on the gene networks identified.

5.1 Data

5.1.1 Data collection

The data analysed in this chapter consists of gene expression levels in multiple tissues for a set of 845 female twins from the TwinsUK cohort. Of the 845 study participants, 294 are monozygotic twins, 454 are dizygotic twins and 97 are singletons. Gene expression was measured using RNA-sequencing in

three tissues, (subcutaneous) adipose, lymphoblastoid cell lines (LCLs)[1] and (relatively photo-protected) skin (Brown et al., 2014; Buil et al., 2015). Data for all three tissues exist for the majority of individuals. Blood expression levels also exist for a subset of the individuals but due to high levels of missing data, this tissue was removed from the analysis. Experiments were performed using the Illumina TruSeq sample preparation kit and sequenced on a HiSeq2000 machine. Reads were mapped on to the GRCh37 reference genome using BWA v0.5.9 (Li and Durbin, 2009). Only uniquely mapping reads were retained. Reads per kilobase per million (RPKMs) were used for further analysis. Some samples are missing but otherwise the data is complete.

Genotyping was performed using a combination of the HumanHap300, HumanHap610Q, 1M-Duo and 1.2MDuo Illumina arrays. Samples were imputed using the 1,000 Genomes Project Phase I reference panel (interim, data freeze 10 November 2010) (The 1000 Genomes Project Consortium, 2012) using IMPUTE2 (Howie et al., 2009). Genotypes are available on only 795 of the 845 individuals. Figure 5.1 shows the patterns of missingness across the three tissues and genotypes. 578 individuals have expression data in all three tissues and genetic data.

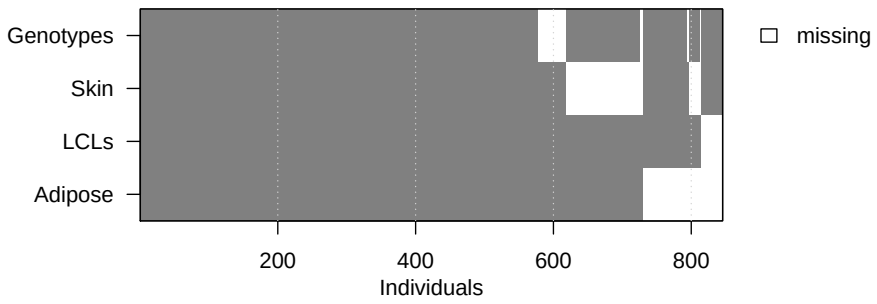

Figure 5.1: Pattern of missing samples in the data.

In addition to genotype and gene expression data, 11 concurrently measured phenotypes and 4 sequencing variables are available. The 11 phenotypes

[1]LCLs are created by infecting B-lymphocyte cells with the Epstein-Barr virus.

are: age, BMI, weight, height, total cholesterol, HDL cholesterol, LDL cholesterol (calc), total triglycerides, adiponectin, insulin and glucose. There is some missing data in these measurements. Variables derived from sequencing experiments include: (i) mode of the insert size, (ii) GC content of the reads, (iii) sequencing date and (iv) primer index. Small (unavoidable) differences in library preparation can result in sequencing biases and these four variables have previously been identified as giving rise to sample-specific variation (Hansen et al., 2010; Pickrell et al., 2010).

5.1.2 Pre-processing

Several pre-processing steps were performed on the expression (RPKM) data. The following steps are a fairly standard approach to normalisation of gene expression data (e.g. The GTEx Consortium (2015)). Genes with more than 20% zeros in all three tissues were removed. Genes on the Y chromosome and mitochondrial DNA were also removed leaving 18,409 genes. The following normalisation steps were then performed for each tissue independently: (i) expression levels were quantile normalised across samples so that each sample had the same distribution and (ii) data for each gene was rank transformed onto a standard normal distribution. No pre-processing to remove technical artefacts or other covariates was performed. Genes were annotated using the GENCODE v10 annotation (Harrow et al., 2012).

Genotypes were filtered to remove SNPs with a minor allele frequency below 1% and IMPUTE info value less than 0.8. This resulted in a set of 6,200,000 SNPs. A kinship matrix was calculated from the genotype data using GEMMA (Zhou and Stephens, 2014),

$$K = \frac{1}{L_G} G G^\top,$$

where G is the matrix of mean-centred genotypes, with each column represent-

ing a SNP, and L_G is the total number of SNPs.

5.2 Method

5.2.1 Tensor decomposition using SPIDER

The expression data across three tissues can be represented by a 3D array of dimensions $845 \times 18,409 \times 3$. SPIDER, the approach introduced in chapter 3, was applied to the data 10 times with a different initialisation for each run. The number of components was initially set to 1,000. This number was selected via trial and error to ensure that there were always superfluous components; 1,000 appeared to be sufficient as the method consistently removed around 50 components. Each run of SPIDER took around 20 hours.

Missing samples were handled by ignoring their contribution in the model likelihood using the method described in section 3.4.1.2. Due to computational reasons, a standard normal prior on the elements of the individual scores matrix was used (rather than a prior that incorporates the kinship matrix). Simulations given in section 4.2 and appendix B.3 suggest that although a kinship matrix improves component recovery, it only makes incremental differences and the standard normal prior is surprisingly flexible. The priors and convergence criteria are described in section 3.5. Posterior expectations were used as point estimates for the individuals and tissue scores matrices, gene loadings matrix and posterior inclusion probabilities (PIPs).

5.2.2 Post-processing

Following the approach suggested in section 4.2.3.1, component estimates from across multiple runs were combined via clustering to create a set of 'robust' components. Components were clustered based on correlations between individual scores vectors, using a correlation threshold of 0.6 to terminate the process. This value was selected in order to make the clustering conservative,

only very similar components were combined into clusters. Clusters containing 5 or more components were then averaged and used in further post-processing steps. Results for the run with the highest negative free energy are also presented.

The following post-processing steps can be applied to clustered components or components from the highest negative free energy run. PIPs were thresholded at 0.5 to create a set of genes with non-zero loadings. In reality, PIPs are unlikely to be calibrated, i.e. a PIP of 0.5 does not mean that a gene is active in the component with probability one half. However it was rare that PIPs were not close to one or zero meaning that any threshold in $[0.1, 0.9]$ gave a very similar gene set. Gene loadings with PIPs < 0.5 were set to zero. The vast majority of these points had mean loadings very close to zero anyway.

5.2.2.1 Classifying factors according to tissue specificity

The tissue scores matrix determines the tissue-specificity of each component. A tissue score close to zero suggests a component is not active in that particular tissue. In practice, the distribution of tissue scores was trimodal, clearly distinguishing between a cluster of values close to zero and clusters away from zero (with positive or negative values). Tissues scores below a hand-selected threshold were set to zero.

This thresholding approach may seem unrigorous. However component activities were only used for summarising and visualising data, and not downstream analyses.

5.2.2.2 GWAS

For each component, the individual scores vector was used to perform a genome-wide scan for association. The scores vectors were first subset down to the set of 795 individuals for which imputed genotype data is available,

then rank-transformed to $\mathcal{N}(0,1)$[2]. GEMMA was used to perform testing via a mixed model to account for relatedness (Zhou and Stephens, 2014). p-values were obtained using a likelihood ratio test. A stringent Bonferroni correction of 1×10^{-10} was used to determine significance, obtained by scaling a genome-wide significance level of 5×10^{-8} to correct for the number of components tested.

5.2.2.3 Association with phenotypes and sequencing variables

To further interpret components, individual scores vectors were tested for association with phenotypes and sequencing variables. A mixed model was used to perform univariate association testing between a scores vector and phenotype, with the phenotype as the dependent variable (Zhou and Stephens, 2014)[3]. For the associations with age, only one member of each twin pair (selected randomly) was used. The categorical batch variables (date and primer index) were dealt with by creating binary vectors (one for each category) and individually testing these as a fixed effect in a mixed model. A p-value cut-off of 1×10^{-6} was used to determine significance for these associations.

5.2.2.4 Gene ontology analysis

While the individual scores matrix uncovers structure across individuals, the loadings matrix identifies the set of genes driving this structure. To test whether these gene sets identified genes involved in the same mechanism or function, a gene ontology analysis was performed.

Gene ontology (GO) terms are a set of annotations based on the current knowledge of gene functions (Ashburner et al., 2000). GO terms are split into

[2]Transforming the individual scores vectors reduced the number of false associations. Occasionally, estimated components identified an individual who appeared to be an outlier. The individual scores for such a component consisted of one very extreme value with the remaining scores close to zero. Genome-wide scans for these components resulted in unusual patterns of low p-values identifying SNPs with low MAF. These spurious associations disappeared after transforming the scores vectors.

[3]GEMMA was used with the `-notsnp` setting

three categories, biological process, molecular function and cellular component, and range from being very specific to very general. Genes can be annotated with several GO terms. Gene ontology analysis evaluates whether a particular set of genes is enriched for a GO term in comparison to a background gene set. Gene ontology analysis was carried out using the R package topGO (Alexa et al., 2006). topGO uses Fisher's exact test to get a p-value for enrichment based on the expected and observed number of genes with a GO term. Of the 18,409 genes used in this analysis, 13,965 have GO annotations.

To get a significance level for this analysis 10,000 sets of genes of random sizes were sampled, and an enrichment analysis performed for each set. For each gene set, the smallest resulting p-value was used to create a null distribution. Based on the null, a significant level of 1% was estimated to be 1×10^{-6} for any sized gene set.

5.2.3 Direct associations

A standard *trans* analysis for this data set would require tests for association between all SNP-gene pairs in three tissues. Scaling a genome-wide significance level of 5×10^{-8} by the number of genes and tissues would result in a strict Bonferroni significance threshold of $\frac{5 \times 10^{-8}}{18{,}409 \times 3} = 9.1 \times 10^{-13}$. (This threshold assumes no correlations between genes so may be too conservative.) The approach suggested here, using the individual scores vectors, considerably cut down the number of tests performed, and also moves away from this marginal approach. However it is also interesting to investigate marginal associations between the genes identified by a component and the SNP associated with that component's individual scores vector.

Marginal associations were performed via a linear mixed model (Zhou and Stephens, 2014) using the normalised expression levels in each tissue separately. In order to account for unmeasured confounding factors, PEER was fit to each tissue's expression data with 15 factors, and these factors used as covariates

in the mixed model (Stegle et al., 2010). Although it is possible for methods such as PEER to explain away *trans* effects (Fusi et al., 2012), in this analysis the use of 15 PEER factors improved results. Adding in additional covariates (age, BMI, GC content and insert size mode) made no further improvements.

5.3 Results

5.3.1 Summary of the output

The median number of components estimated per run was 942 (min: 935, max: 947). The set of all components (9,416 in total) was used for clustering; cluster sizes are shown in figure 5.2. Many components do not cluster, but this is not surprising as the clustering criterion was fairly strict. A small number of clusters have size > 10 which suggest component splitting, i.e. when one signal was described by two components within the same run. 236 clusters had a size of 5 or more; these clusters were averaged to get a set of 236 (robust) components which were used in further analysis.

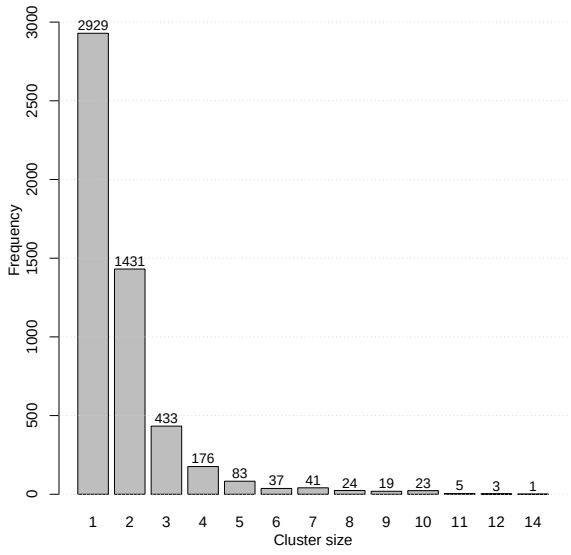

Figure 5.2: Distribution of cluster sizes from clustering components across 10 runs.

The tissue activity scores for these 236 components are shown in figure 5.3.

A threshold of 0.5 was used to determine specificity. The majority of components are active in only one tissue (57 in adipose, 74 in LCLs and 70 in skin). An additional 14 components are active in adipose and skin only, one component is active in LCLs and skin. No components are specific to just adipose and LCLs. It is expected that LCLs would be the 'odd-tissue-out' because of the way cell lines are generated, and the lack of symmetry in these numbers reflects this. The remaining 20 components are active in all three tissues.

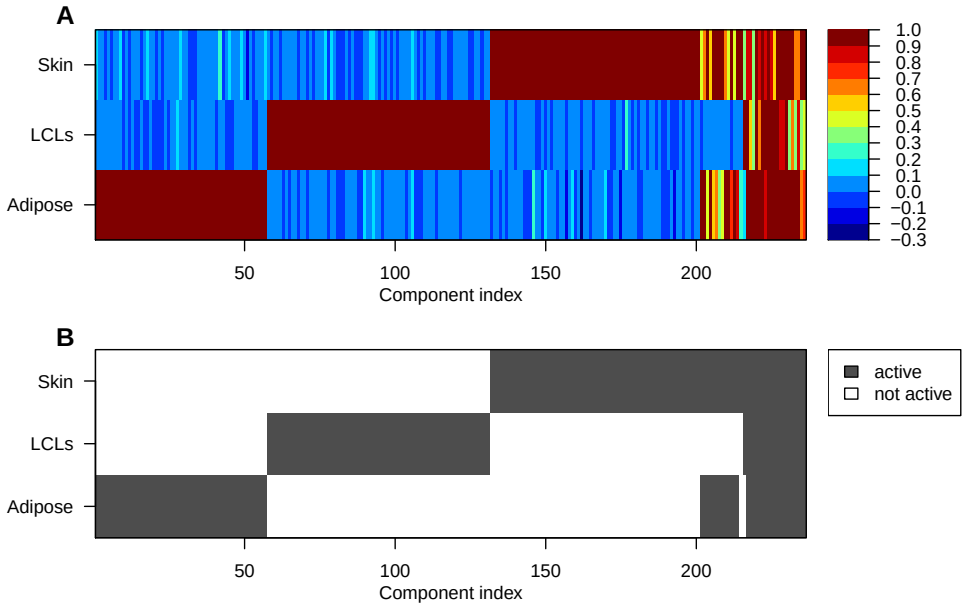

Figure 5.3: Tissue scores matrix for 236 robustly identified components. (A) Each column shows the tissue scores for a component, scaled so that the largest score equals 1. Columns have been arranged to group components with similar tissue patterns. (B) Binary representation of scaled tissue scores (obtained by thresholding scores at 0.5) to highlight the tissue specificity of the components.

Of the 236 components, 26 have individual scores vectors that are significantly associated with SNPs (using a significance threshold of 1×10^{-10}). These components tend to be very sparse; the median number of genes with non-zero loadings is 17.5 (min: 3, max: 160). For 20 of these components, a cluster of genes near the significant SNP have extreme gene loadings, with the remainder of genes identified having loadings close to zero, suggesting these components are recovering *cis* effects. The remaining 6 components show more interesting gene loadings patterns, with large loadings identified on several chromosomes.

These gene loading patterns look more consistent with a *trans* effect. A closer look at these 6 components is given in section 5.3.2.

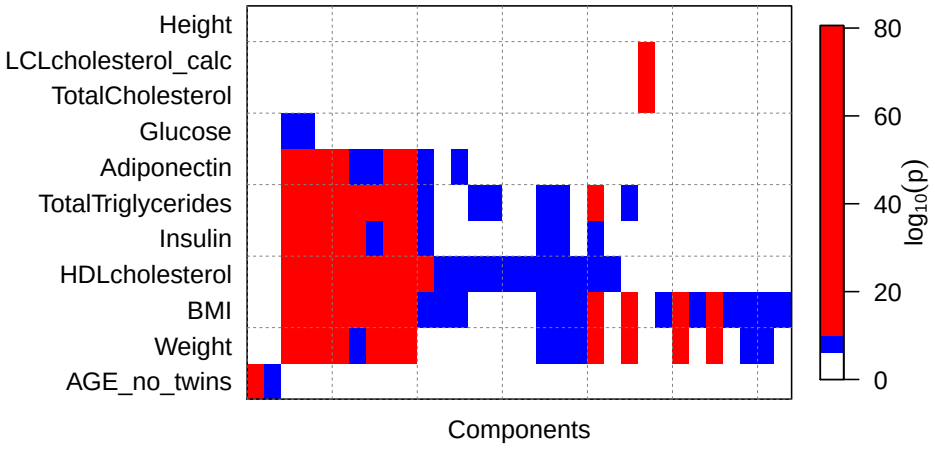

Figure 5.4: Summary of associations between 236 robust components and 11 measured phenotypes. p-values less than 1×10^{-6} are shown in blue and those less than 1×10^{-10} shown in red. Only components with a significant association ($< 1 \times 10^{-6}$) have been plotted (32 in total), and components with similar patterns of association have been placed nearby.

Many components are associated with phenotypes. Figure 5.4 shows the set of components with significant associations with at least one phenotype (p-value $< 1 \times 10^{-6}$). Components and phenotypes with similar association patterns have been plotted nearby. Metabolic-related phenotypes tend to cluster and are associated with many components. These components tend to be exclusively in adipose tissue, which is perhaps not surprising; it is known that expression of many genes in adipose tissue are associated with BMI (Buil et al., 2015). Two components are associated with age, these components are only active in LCLs. Figure 5.5 shows the number of components associated with various sequencing variables.

Figure 5.6 shows the results of a GO analysis for each component. Many components are enriched for GO terms (with p-values $< 1 \times 10^{-6}$) suggesting the components are identifying sets of similar genes. However, these results

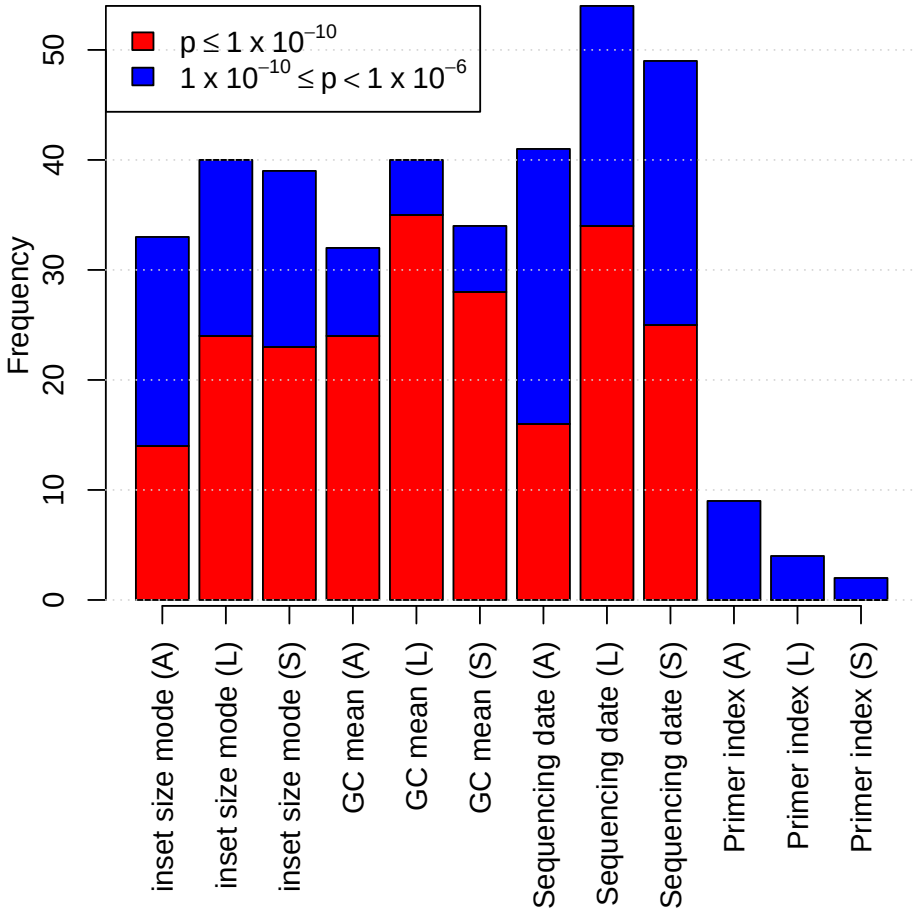

Figure 5.5: The barplot shows the number of robust components that show a significant association with four batch variables that measure properties of RNA sequencing across the three tissues (A: adipose, L: LCLs, S: skin). The plot shows the numbers of associations between 1×10^{-6} and 1×10^{-10} in blue and less than 1×10^{-10} in red.

should be viewed with some caution, especially when the gene set is large. In addition, some of the significant GO terms are very generic.

A summary of the number of components associated with SNPs, phenotypes, sequencing variables and enriched for GO terms is given in table 5.1.

5.3.2 Components explaining *trans* effects

This section looks more closely at 6 of the components significantly associated with a SNP and possibly uncovering a *trans* association. Of these 6 com-

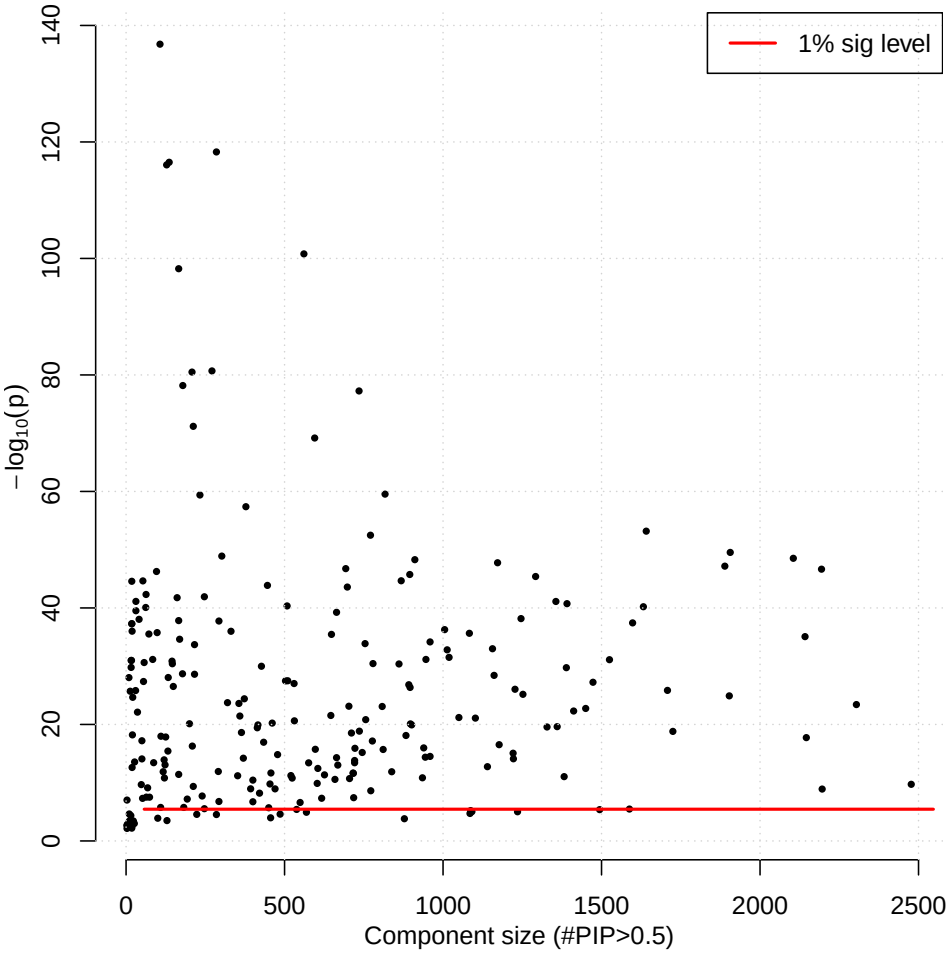

Figure 5.6: Gene ontology p-values for 236 robustly clustered components. x-axis shows the component size (number PIPs > 0.5) and the y-axis shows $-\log_{10}(p)$ for the most strongly associated GO term. The red line indicates a 1% significance threshold.

ponents, one is active in adipose and skin, 3 are active in LCLs only and the remaining 2 are active in all three tissues. Interestingly, all of these components show mainly unidirectional gene loadings, suggesting recovery of a directional effect on gene expression.

5.3.2.1 Regulation of MHC class II genes

Two of the 6 components have similar gene loadings vectors (figure 5.7). One component is active in adipose and skin, and the other in LCLs. A genome-wide scan for association using these components' individual scores as phe-

	Tissue activation pattern							Row totals
	A	L	S	AL	AS	LS	ALS	
Number of components	57	74	70	0	14	1	20	236
SNP (1×10^{-10}) *cis*	1	1	0	0	0	0	18	20
SNP (1×10^{-10}) *trans*	0	3	0	0	1	0	2	6
Phenotype (1×10^{-6})	21	0	8	0	3	0	0	32
Sequencing (1×10^{-6})	37	53	39	0	2	0	0	131
GO term (1×10^{-6})	49	68	63	0	14	1	5	200

Table 5.1: Summary of 236 components obtained when clustering results across 10 runs. Components are categorised according to which set of tissues they are active in (A: adipose, L: LCLs, S: skin) using a threshold of 0.5 on the tissue scores matrix. The first row gives the number of components with each activation pattern; subsequent rows summarise the number of components associated with SNPs, phenotypes, sequencing variables and enriched for GO terms. Components associated with SNPs are further categorised according to whether they describe *cis* or *trans* effects via a visual inspection. Significant levels are given in brackets.

notypes uncovers a cluster of significant SNPs on chromosome 16. The lead SNPs for the two components are rs9924520 (p-value = 1.33×10^{-23}, MAF = 0.247) and rs7194862 (p-value = 1.74×10^{-14}, MAF = 0.282). These SNPs are in strong LD ($r^2 = 0.82$). rs9924520 is an intronic variant in the gene *CIITA* and rs7194862 lies upstream of *CIITA*. *CIITA* has a non-zero loading (i.e. PIP > 0.5) in both components, and is a known transactivator of the major histocompatibility complex (MHC) class II molecules (Reith et al., 2005). Clusters of MHC class II genes on chromosome 6 (shown in purple in figure 5.7) are also identified in both components.

The MHC molecules are divided into two classes, (class I and II), dependent on their structure and function. Complexes of these molecules are found on the membranes of many cells in the body and play an important role in immune response. Their job is to detect pathogen fragments, and display these fragments to another class of immune cells, T-cells (Janeway et al., 2001).

The MHC class II genes are highly regulated via a complex process involving many elements. Upstream of the MHC class II genes lies a highly conserved

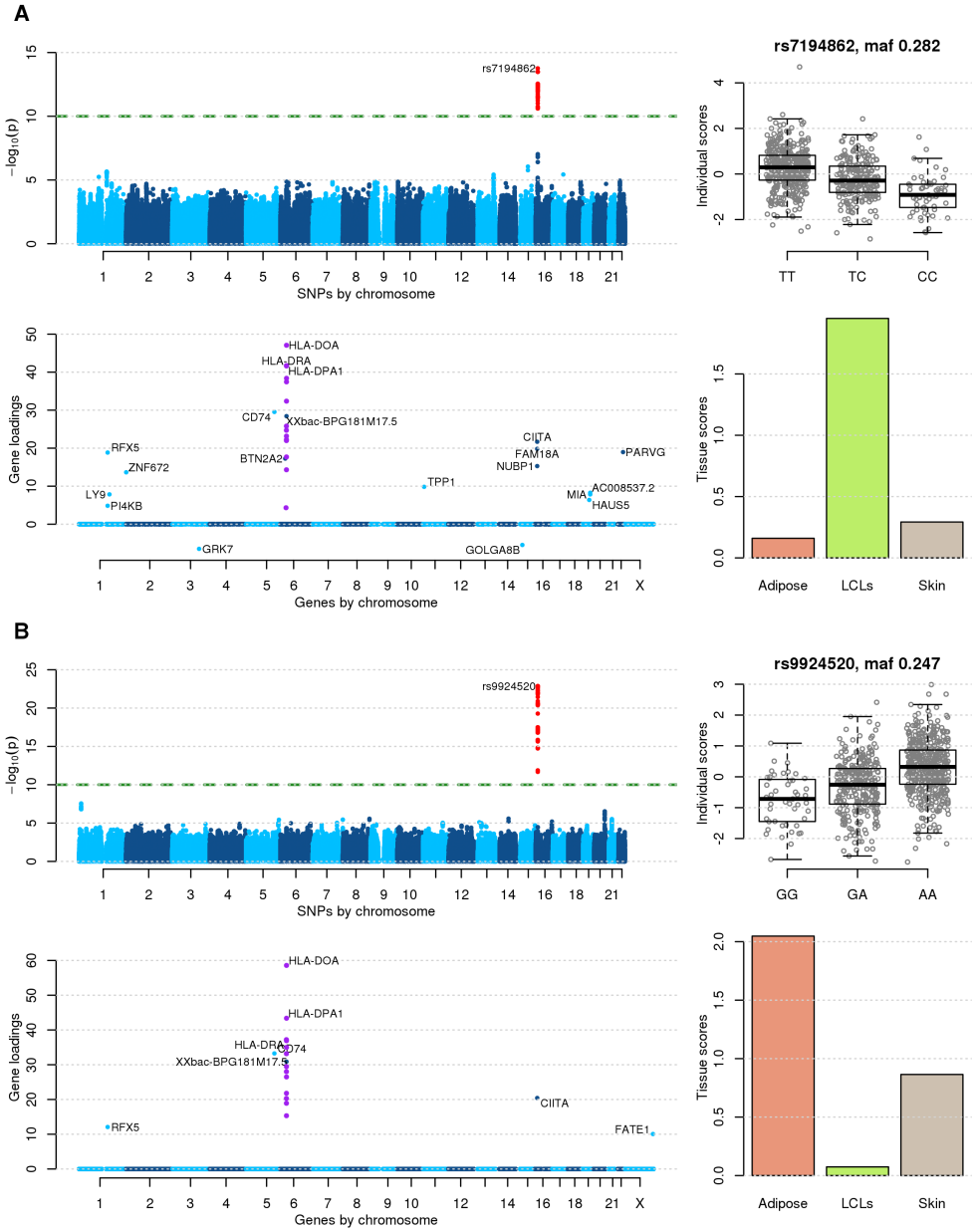

Figure 5.7: (A) and (B) show two components recovering the MHC class II regulation pathway. For each component: (Top left) Manhattan plot from a GWAS using individual scores vector as a phenotype. (Top right) Boxplots of individual scores separated into groups according to alleles of the lead GWAS SNP. (Bottom left) Gene loadings vector. (Bottom right) Barplot of component tissue scores.

sequence called an SXY module. A complex of several proteins including three RFX proteins (RFX5, RFXANK and RFXAP), CREB1, ATF1 and members of the nuclear factor Y family bind to this module, creating what is known as the MHC class II enhanceosome (Wong et al., 2014). (The gene on chromosome 1 which codes for RFX5 has a non-zero loading in both components.) It is via interactions between this complex and transactivator CIITA that the MHC class II genes are regulated.

CIITA is also known to regulate genes outside of the MHC class II family including *CD74* on chromosome 5 (Krawczyk et al., 2008). This gene (which has non-zero loadings in both components) associates with MHC class II molecules and is involved in regulation of antigen presentation.

The component active in LCLs also identifies a handful of other genes with non-zero gene loadings. Several of these genes are known targets of CIITA in B cells, monocytes and dendritic cells (*RFX5*, *TPP1*, *ZNF672*, *PARVG* and *HAUS5*) (Krawczyk et al., 2008; Wong et al., 2014). Of the remaining genes in the component, some have links to the immune system; *LY9* is an immunomodulatory receptor and *BTN2A2* is a member of the immunoglobulin gene superfamily.

It is unclear why two components are identified for this pathway rather than one. The individual scores vectors for the two components are not highly correlated (cor = 0.154). Expression patterns of *CIITA* (the gene driving this component) are different across tissues (cor $<$ 0.268). Random variation affecting the expression levels of *CIITA* may be propagating to other genes in the network resulting in two components rather than one. This behaviour was seen in simulations.

Marginal associations between genes in these components and the lead SNPs provides further evidence of a *trans* effect. Figure 5.8 shows results of associations between rs992450 and rs7194862 and the genes involved in these components across all three tissues. A strict Bonferroni threshold of

9.05×10^{-13} is indicated on the plot. At this threshold, several genes show associations with the two SNPs and several more genes are close to significance. Interestingly, *RFX5* would not have been recovered using this analysis, despite its involvement in the pathway.

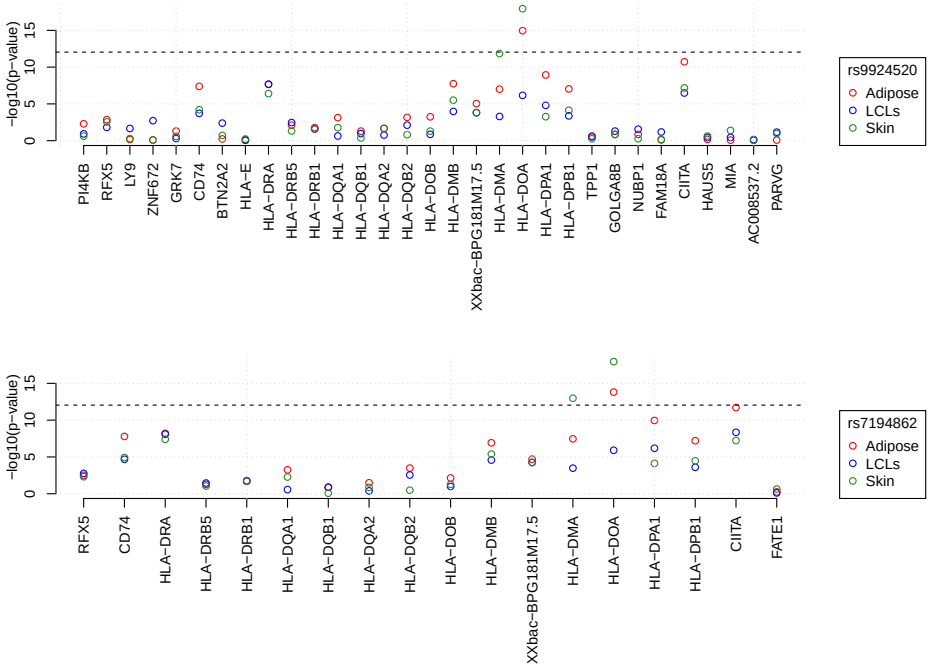

Figure 5.8: p-values for marginal associations between genes with non-zero loadings in MHC class II components and lead SNPs associated with the individual scores of these components. Colours represent the three different tissues. Horizontal dashed line indicates a strict Bonferroni threshold for a full *trans* analysis of 9.05×10^{-13}.

5.3.2.2 Regulation of MHC class I genes

Figure 5.9 shows a sparse component identifying a handful of genes scattered across the genome and a cluster of MHC class I genes (*HLA-A*, *HLA-B*, *HLA-C*, *HLA-E*, *HLA-F*) on chromosome 6. The component is active in all three tissues. The individual scores are significantly associated with a cluster of SNPs on chromosome 16. The lead SNP, rs289749 (p-value $= 1.34 \times 10^{-11}$, MAF $= 0.3$), lies in an intron of the *NLRC5* gene which is a transactivator of the MHC class I genes, and has a non-zero loading in the component (Meissner

et al., 2010). (*NLRC5* is also known as *CITA*.) The mechanisms of this process are unknown, but it is believed that the NLRC5 protein associates with a subunit of a transcription factor complex acting on the MHC class I genes (Meissner et al., 2012).

The component in figure 5.9 identifies several other genes involved in the MHC class I antigen presentation pathway (Kobayashi and Elsen, 2012). These include *B2M* which is part of the MHC class I complex, subunits of the proteasome complex (*PSMB9*, *PSMB8*) and transporters (*TAP1*, *TAP2*). Additionally, *NLRC5* is known to induce expression of *B2M*, *TAP1* and *PSMB9* (Meissner et al., 2010). Also in the component are the Butyrophilin genes, *BTN2A1*, *BTN2A2*, *BTN3A1*, *BTN3A2* and *BTN3A3*, members of the immunoglobulin superfamily.

Figure 5.10 shows results of marginal associations between lead SNP rs289749 and the genes involved in this component. *NLRC5* is highly associated with rs289749 in skin, and several other genes also show marginal associations with rs289749 in skin. Strikingly, there is no signal in the marginal associations in adipose or LCLs, which contradicts the pattern of tissue scores for this component. One explanation is that SPIDER does a better job at modelling confounding compared to PEER, allowing the signal in all three tissues to be recovered. It is also possible that SPIDER incorrectly identified the component activity across the tissues.

5.3.2.3 Regulation of histone RNA processing

Figure 5.11 shows a component active in LCLs only which is significantly associated with rs6882516 on chromosome 5 (p-value = 2.39×10^{-15}, MAF = 0.206). This SNP lies in the $3'$ untranslated region of *LSM11*, a gene with a non-zero loading in the component. The component also identifies 23 histone genes in the 6p21 cluster with non-zero loadings, as well as *HIST2H2BE* on 1q21, *HIST3H2A* on 1q42, *H2AFX* on 11q23 and *HIST4H4* on 12q12.

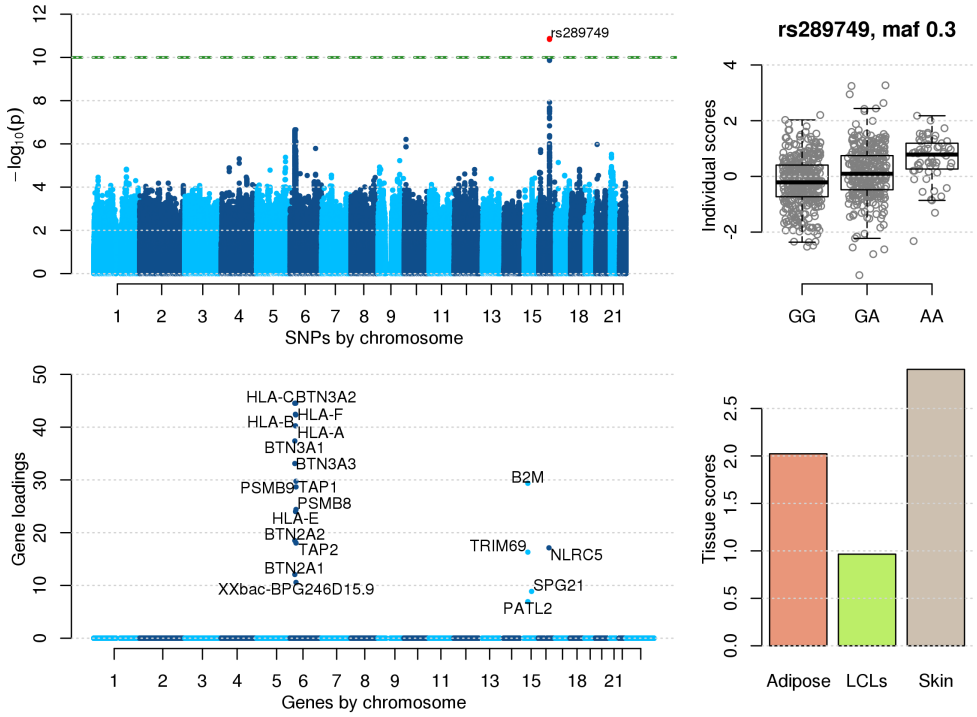

Figure 5.9: MHC class I component. See figure 5.7 for a description of the figure.

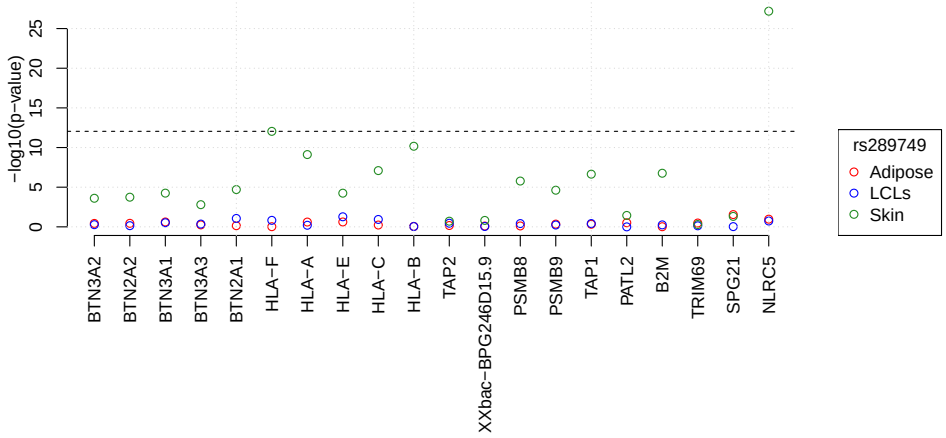

Figure 5.10: Marginal associations between rs289749 and genes with non-zero gene loadings in the MHC class I regulation component. See figure 5.8 for a more detailed description of the figure.

Factors involved in the transcriptional regulation and post-transcriptional modifications of histone genes organise into a small number of subnuclear foci. A key player in activation of histone gene expression in the G1/S phase of the cell cycle is p220$^{\text{NPAT}}$. The p220$^{\text{NPAT}}$ complex associates with histone gene

clusters at 1q21 and 6p22 in a study of human embryonic stem cells (Ghule et al., 2008). Additionally, LSM11, LSM10 and protein complex U7 snRNP associate with the same clusters. U7 snRNP protein complex also contains the LSM11 protein (Pillai et al., 2003). snRNPs catalyse a post transcriptional modification called splicing whereby introns are removed from pre-mRNA. U7 snRNP specifically acts on the 3′ end of histone mRNA molecules with the N-terminus of LSM11 thought to play a crucial role (Pillai et al., 2003).

Marginal associations for this component are consistent with the signal being active in only LCLs (figure 5.12). rs6882516 is marginally associated with *LSM11* and several histone genes, although only the *cis* effect on *LSM11* passes a strict Bonferroni correction for a full *trans* analysis.

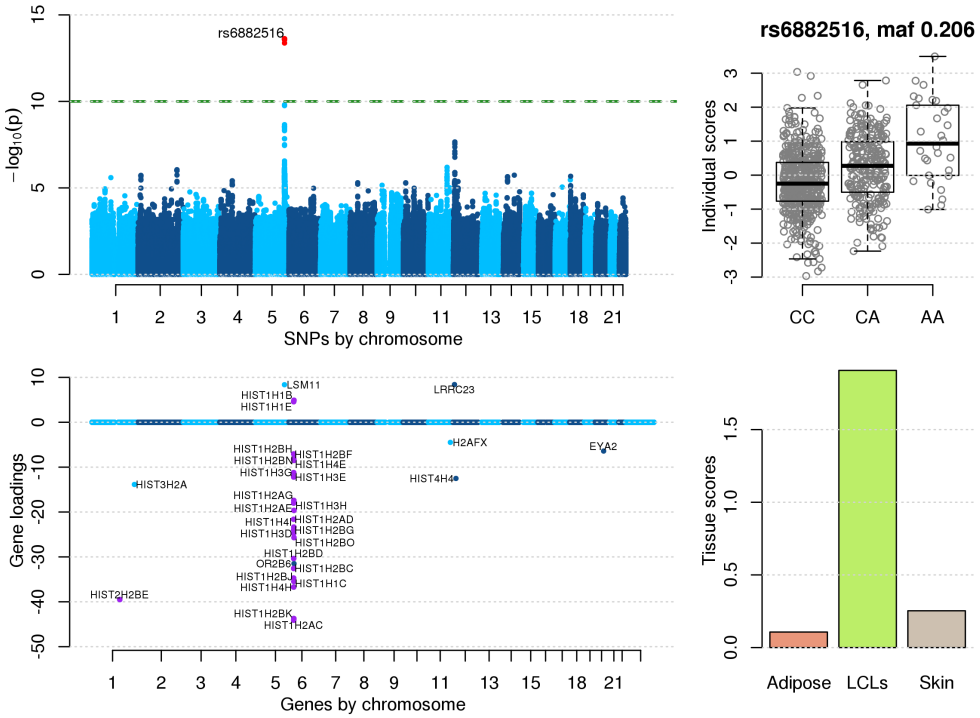

Figure 5.11: Histone RNA processing component. See figure 5.7 for a description of the figure.

5.3.2.4 Type I interferon signals

Figure 5.13 shows a component with a set of 160 genes with PIPs > 0.5 scattered across the genome. GO analysis for this component shows enrichment

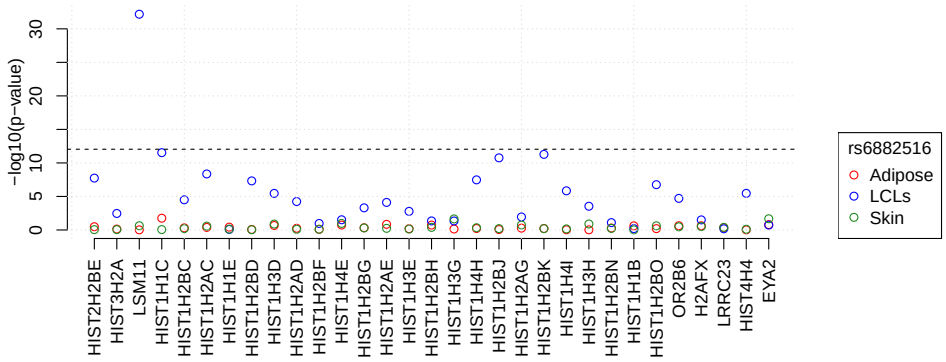

Figure 5.12: Marginal associations between rs6882516 and genes with non-zero gene loadings in the histone RNA processing component. See figure 5.8 for a more detailed description of the figure.

for the terms 'defense response to virus' and 'response to type I interferon' with p-values of 1.73×10^{-42} and 1.23×10^{-38} respectively. A total of 70 genes in this analysis are annotated with the 'response to type I interferon' GO term of which 28 have a PIP > 0.5 in this component. The tissue scores indicate that this component is only active in LCLs.

Interferons are small proteins produced when a cell is under attack, they activate expression of hundreds of interferon-stimulated genes (ISGs). Genes annotated with the 'response to type I interferon' GO term in the component include: all four of the 2′-5′ oligoadenylate synthetase gene family (*OAS1*, *OAS2*, *OAS3*, *OASL*), known to be actively induced by interferons (Kakuta et al., 2002); interferon γ-inducible protein genes (*IFI6*, *IFI44L*, *IFI16*, *IFIH1*, *IFIT1*, *IFIT3*, *IFIT5*, *IFIT2*, *IFITM1*, *IFITM2*, *IFI35*) and the *MX1* and *MX2* genes, which are also related to interferon signaling. Additionally, the component identifies *STAT1*, *STAT2* and *IRF9* which form a complex that plays a role in the transcription of ISGs (Fink and Grandvaux, 2013). The type I interferon genes themselves are not identified by the component.

The individual scores for this component are significantly associated with a SNP on chromosome 22 (rs2401506, p-value$= 9.82 \times 10^{-16}$, MAF $= 0.358$),

which lies 5 kb upstream of *USP18* (a gene with a non-zero loading). *USP18* (ubiquitin specific peptidase 18), itself a interferon stimulated gene, encodes for a protein USP18 (also known as UBP43) which has several functions. It is mainly known for its role in removing ISG15 (also identified by the component) from protein complexes via a process called deISGylation (Malakhov et al., 2003). There is evidence to suggest that USP18 is also involved in regulation of interferon pathways; USP18 may negatively regulate the JAK-STAT signalling pathway (via a process independent of deISGylation) (Malakhov et al., 2003; Malakhova et al., 2006). Cells become less sensitive to type I and III interferon signalling and it is believed that the extent of this desensitisation is controlled by USP18 (François-Newton et al., 2011).

Immune response is a complex process likely to involve multiple regulating factors and feedback loops so it is hard to determine whether *USP18* is the major driving factor here. *Trans* eQTLs in *IRF7* (Heinig et al., 2010), *KAT8* and *TRAPPC9* (Gao et al., 2013) involving networks of interferon stimulated genes have previously been identified providing further evidence of the complexity of these pathways.

Figure 5.14 shows marginal associations for this component. There is a marked enrichment of low p-values in LCLs but not the other tissues, reflecting the tissue specificity of this component.

5.3.2.5 Zinc finger network

Figure 5.15 shows a component whose gene loadings identify a cluster of genes on chromosome 19, the vast majority of which are in the ZNF gene family. ZNF genes encode for a group of transcription factors that contain zinc finger domains, allowing for binding to DNA (and other molecules). About 800 genes encode for proteins with zinc finger domains, many of which lie on chromosome 19 (Urrutia, 2003).

Performing a GWAS using the individual scores vector for this component

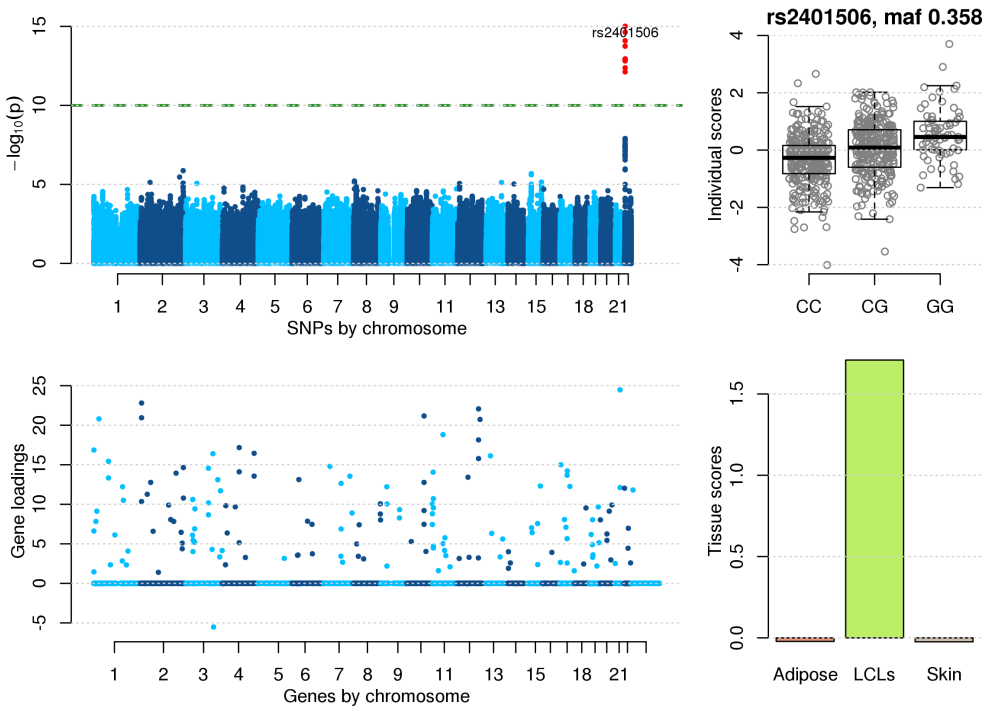

Figure 5.13: Type I interferon component. See figure 5.7 for a description of the figure.

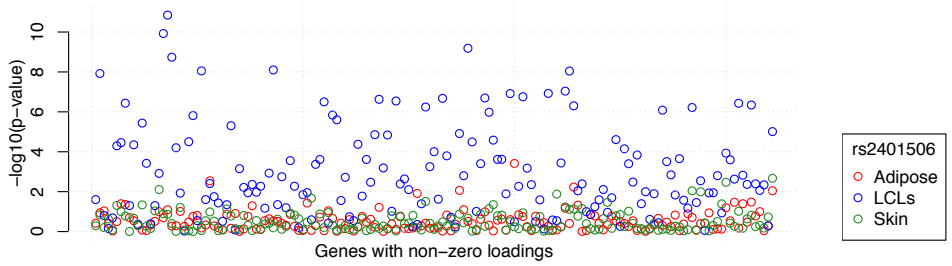

Figure 5.14: Marginal associations between rs6882516 and genes with non-zero gene loadings in the histone RNA processing component. See figure 5.8 for a more detailed description of the figure.

uncovers significantly associated SNPs on chromosome 3 and chromosome 16. The lead SNP on chromosome 3, rs12630796 (p-value $= 5.10 \times 10^{-17}$, MAF $=$ 0.487) in an intronic variant in the *SENP7* gene. (*SENP7* has a zero loading in the component.) A SNP in high LD with this SNP, (rs13320918, p-value $=$ 7.34×10^{-15}, MAF $= 0.377$) has previously been shown to be a microRNA QTL for miR-1270, which is located on chromosome 19p12 in a zinc finger cluster.

Another study links 4 other intronic SNPs in *SENP7* (rs2553419, rs2682386, rs9859077, rs2141180) (in high LD with each other and with rs13320918) with *cis* acting regulation of *SENP7* in CD4 and CD8 lymphocytes (Lemire et al., 2015) and *trans* acting regulation of three ZNF genes on chromosome 19 (*ZNF154*, *ZNF274* and *ZNF814*; all identified in the component). They further found these SNPs to be methylation QTLs for CpG sites in regions around the same genes.

The lead SNP on chromosome 16 (rs17611866, p-value = 5.40×10^{-21}, MAF = 0.251) is a mis-sense variant in *ZNF75A*, and one of 6 ZNF genes in a local cluster. Neighbouring genes *ZNF263* and *TIGD7* have non-zero gene loadings. It is important to be aware that genes in the ZNF gene family have similar genetic sequences. Despite only using uniquely mapped reads to calculate RPKM levels, it is possible that this component might be partially uncovering a mis-mapping issue.

Direct associations for this component are given in figure 5.16. These results provide additional evidence for associations between SNPs on chromosome 3 and 16 and ZNF genes on chromosome 19.

5.3.3 Highest negative free energy run

A summary of the component estimates for the run with the highest negative free energy is given in table 5.2. A total of 944 components were estimated. The majority of these components are specific to a single tissue (188 in adipose, 273 in LCLs and 203 in skin). Of the components that are active in 2 tissues, the majority are shared between adipose and skin. 101 components are active in all three tissues. Performing a GWAS with the individual scores vectors identified a set of 51 components associated with a SNP at a significance threshold of 1×10^{-10}. A visual inspection of these components identified 39 which appeared to be *cis* effects, with the remaining 12 being possible *trans* effects. Five out of 12 of these components showed similar loadings

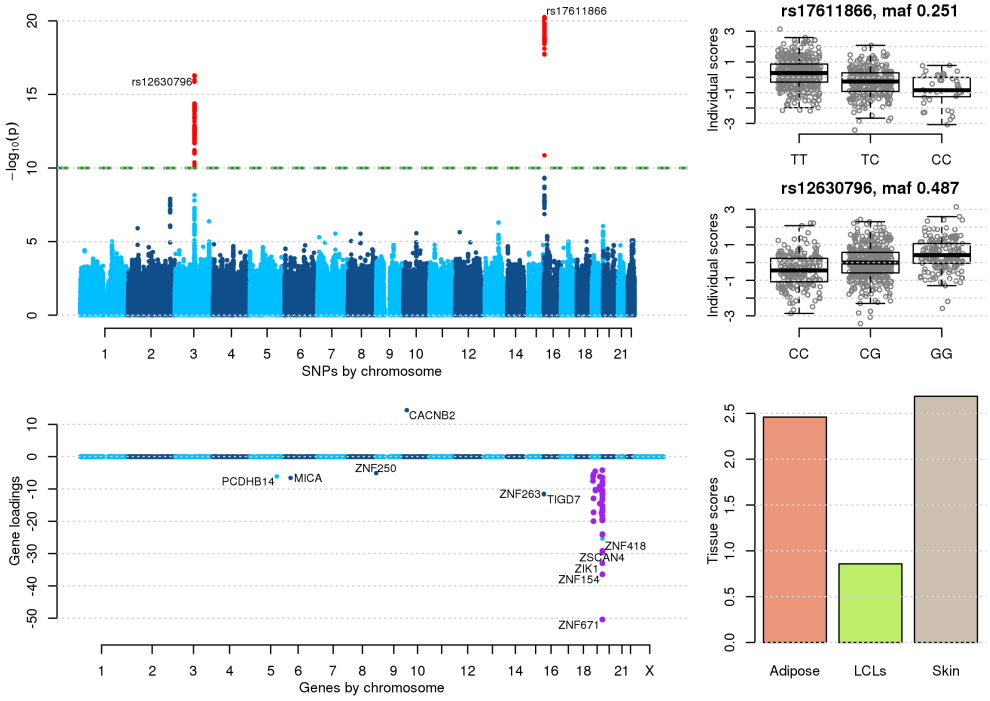

Figure 5.15: ZNF gene network component. See figure 5.7 for a description of the figure.

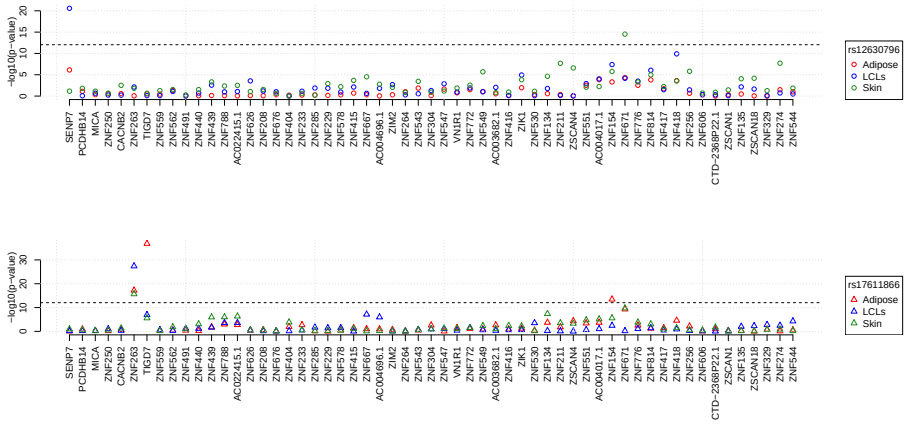

Figure 5.16: Marginal associations between rs12630796 (chromosome 3), rs17611866 (chromosome 16) and genes with non-zero gene loadings in the ZNF gene network component. See figure 5.8 for a more detailed description of the figure.

patterns to the putative *trans* signals described above (a further component which did not quite reach the significance threshold was similar to the MHC class I component.) Many components are also associated with phenotypes, sequencing variables and enriched for GO terms.

Four components were significantly associated with a cluster of SNPs on chromosome 7. These SNPs have previously been identified as driving a network of genes in *trans* via a transcription factor, KLF14.

	Tissue activation pattern							Row totals
	A	L	S	AL	AS	LS	ALS	
# Components	188	273	203	24	140	15	101	944
SNP (1×10^{-10}) *cis*	0	1	0	1	1	0	36	39
SNP (1×10^{-10}) *trans*	3	2	0	0	0	1	6	12
Phenotype (1×10^{-6})	52	0	21	0	17	0	1	91
Sequencing (1×10^{-6})	88	121	103	6	45	5	16	384
GO term (1×10^{-6})	145	219	144	11	91	9	36	655

Table 5.2: Summary of 944 components obtained from run with best free energy. See table 5.1 for table description.

5.3.3.1 KLF14

KLF14 (Kruppel-like factor 14) is a transcription factor encoded by the *KLF14* gene on chromosome 7. Just upstream of the KLF14 gene is a group of highly correlated SNPs (including rs4731702) that act as a *cis* eQTL for KLF14. This group of SNPs are also associated with type 2 diabetes (Voight et al., 2010) and HDL cholesterol (Teslovich et al., 2010) (in females only) at genome-wide significance. Small et al. (2011) performed univariate testing between rs4731702 and expression levels of over 24,000 transcripts. The results show a significant enrichment of low p-values suggesting a broad impact on regulation. KLF14 appears to be the linking factor, acting to regulate multiple genes across the genome. A total of 385 genes has been identified as regulated in *trans* by rs4737102 in the TwinsUK cohort (Small et al. *unpublished*). Many genes implicated in this network are associated with metabolic phenotypes. Interestingly, this effect has only been identified in adipose tissue from females, and only when the effect allele is maternally inherited. (This *trans* network was used to estimates signal strengths in the simulations given in section 4.2.)

Four components are associated at genome-wide significance with SNPs near *KLF14* on chromosome 7 (p-values in the range $(1.48 \times 10^{-12}, 2.72 \times 10^{-16})$). The lead SNPs for these components are in LD ($r^2 > 0.993$) and include rs4731702. All components are dense with 681, 720, 749 and 1429 non-zero gene loadings, and all are specific to adipose tissue, reflecting the known specificity of the *KLF14 trans* effect. Of the 385 genes previously identified as involved in this signal, 359 are being used in the current analysis. Table 5.3 shows the actual and expected overlap between gene sets from the 4 components and these 359 genes. For all the components, an enrichment was seen.

Component index	# PIPs > 0.5	overlap	fraction	expected fraction
161	681	32	0.089	0.037
166	720	40	0.111	0.039
387	749	38	0.106	0.041
412	1429	80	0.223	0.078

Table 5.3: Actual and expected overlap between genes identified in components associated with SNPs near *KLF14* and genes previously identified as involved in the *KLF14 trans* signal. Component size is defined as the number of PIPs > 0.5. Expected fraction is evaluated as the component size divided by 18,409 times 359.

It is not clear why this signal is found across multiple components; the component scores vectors for these components are not highly correlated (pairwise correlations < 0.21). It is possible that the regulation of genes by *KLF14* is part of a larger pathway involving multiple factors and feedback loops and it is this larger process that is being recovered. Furthermore, it is not clear why these components do not cluster well, but this could again be explained by the complexity of the underlying processes.

5.4 Discussion

In this chapter I have described results of applying SPIDER to a real multi-tissue gene expression data set. A clustering approach was used to combine component estimates from across 10 runs to get a set of robust components for downstream analysis. 26 components were associated with a SNP and 6 of these components were identified as representing 5 putative *trans* effects. Evidence from existing literature and marginal associations back up these findings.

There are several advantages of using tensor decompositions for analysing gene expression data. First, the decomposition pools information from across all genes and all tissues, possibly gaining power to find signals in the data. Second, it simultaneously models a variety of different signals in the data, including confounding. Finally, it provides a way to reduce the multiple testing burden of a traditional *trans* analysis; rather then testing for association between all SNP-gene pairs, tests are only required for SNP-component pairs.

A comprehensive comparison of SPIDER with other *trans* analysis techniques has not been performed. The limited set of marginal associations performed here suggest that SPIDER recovers a more complete picture of the underlying processes. However it is hard to quantify this, the Bonferroni significance threshold used here is probably too strict as genes tend to be correlated.

There are downsides to the tensor decomposition approach. Interpretation of the components is tricky. It is not obvious that a SNP associated with a component's individual scores vector would be associated with all the genes identified by the component. It is further hard to evaluate false discovery rates for a methods such as this. It is also important to note that some of the results from SPIDER were inconsistent, most worryingly the MHC class I regulation component tissue activities. However, as an exploratory approach for finding possible *trans* eQTLs, the approach has many benefits.

The group factor analysis version of SPIDER could be run to investigate

whether this model for the data is more appropriate. Additionally, a joint decomposition of the gene expression and genotype data (and phenotypes) would be interesting and also avoid some of the post-processing steps. The set of genotypes would have to be reduced to make this computationally feasible.

Chapter 6

Conclusion

This thesis suggests a new approach called SPIDER for analysing multidimensional gene expression data. The approach builds on existing latent variable models to extract sparse structure from 3D arrays (tensors), with extensions to allow for decompositions of linked matrices and tensors. The novelty of this work is in the use of a spike and slab prior within a tensor decomposition to encourage sparsity. In addition, the approach can handle missing data and related individuals, common problems in genetic studies. Simulations given in chapter 4 suggest a joint analysis of expression data via a tensor decomposition has the power to recover *trans* effects.

Chapter 5 contained an analysis of a real multi-tissue gene expression data set. The aim of this analysis was two fold; to validate the tensor decomposition approach and to recover novel biological signals in the data. Several *trans* effects were detected in the data, with evidence from the literature backing up these results. In many cases, a fuller picture of the underlying pathways were recovered using SPIDER, compared to an approach using marginal associations. SPIDER may be a useful exploratory tool for uncovering *trans* networks in future multi-tissue studies such as GTEx (The GTEx Consortium, 2015).

There are several future directions for this work which involve extending the model. SPIDER appears to preferentially extract signals involving several genes, e.g. *trans* effects. However ideally, an analysis of gene expression data

would simultaneously uncover both *cis* as well as *trans* effects. Incorporating genotypes into the analysis via a linked matrix-tensor decomposition may help uncover more genetic signals in the gene expression data, including *cis* effects. However, since most genes are believed to have a *cis* eQTL, this approach would require fitting of tens of thousands of components. The obvious challenge with this extension is computational, as SPIDER is quadratic with respect to the number of components. Components describing *cis* effects should be very sparse however, with the only non-zero elements in the loadings vectors near the causal SNP and regulated gene. Using this structure, the algorithm complexity could be reduced by constraining some of the components to take specific (fixed) patterns of zeros in the loadings matrices, and only updating a small local window in these components. Figure 6.1 illustrates this extension.

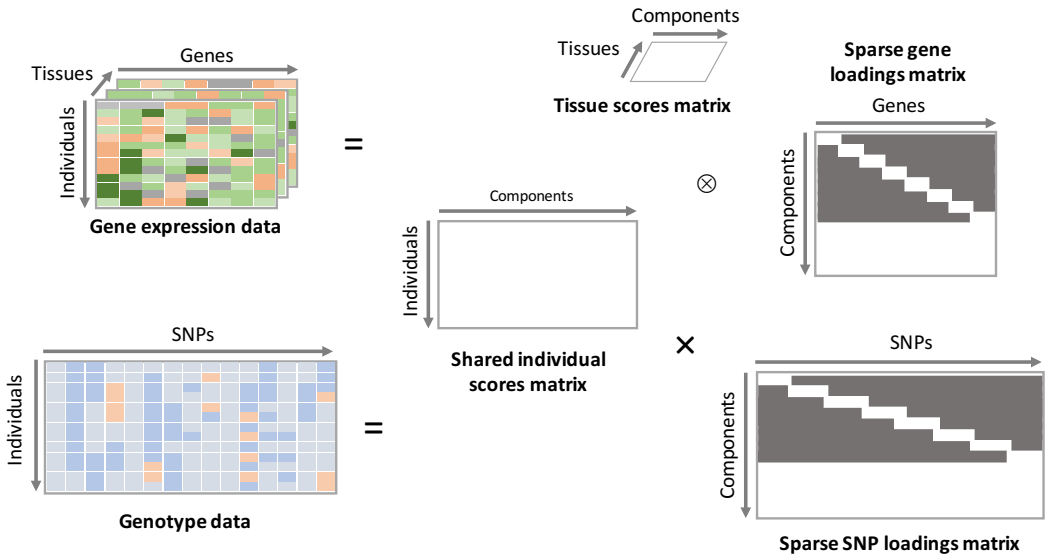

Figure 6.1: Illustration of an extension to additionally model *cis* effects. Gene expression and SNP data is jointly decomposed. Tens of thousands of components are fit, with a fixed pattern of zeros for some components (shown in grey) to make the method computationally efficient.

Another extension allows for analysis of data with a mixture of data types, including binary, categorical and continuous data. In this thesis, I only consider continuous data, however SPIDER could be extended to explicitly allow for a variety of different data formats. Matrix and tensor decompositions of binary data tend to use a logistic likelihood function (e.g. Nickel and Tresp

(2013)). This likelihood function makes the variational Bayes framework described here intractable, but additional approximations can be used to make inference possible (Seeger and Bouchard, 2012). Furthermore, a multinomial distribution could be used for SNP data to model allele count. A model incorporating decompositions with these different likelihoods could be linked via a shared individual scores matrix as in group factor analysis. This is a similar idea to iClusterPlus (Mo et al., 2013), but in a Bayesian framework.

Throughout this thesis the data sets I have considered conformed to a particular structure. Data tensors and matrices were required to be linked via a single shared dimension, individuals. However there are interesting genetic scenarios in which this might not be the case. Gene annotations can be formatted into a binary matrix with dimensions given by annotation and gene. An integrated analysis of gene expression and gene annotations would involve matrices with a shared second dimension, genes. Figure 6.2 suggests a joint decomposition of the expression and annotation data where the gene loadings matrix, rather than a scores matrix, is shared. Components in this decomposition would link annotations with sets of genes, potentially using the annotations to inform recovery of a gene network in an unsupervised manner. This approach has previously been suggested by Khan et al. (2014b).

SPIDER can also be extended to allow for four-dimensional data, for example, gene expression data in multiple tissues at multiple time points (figure 6.3). Here, an additional time scores matrix is learnt to estimate component activity across time.

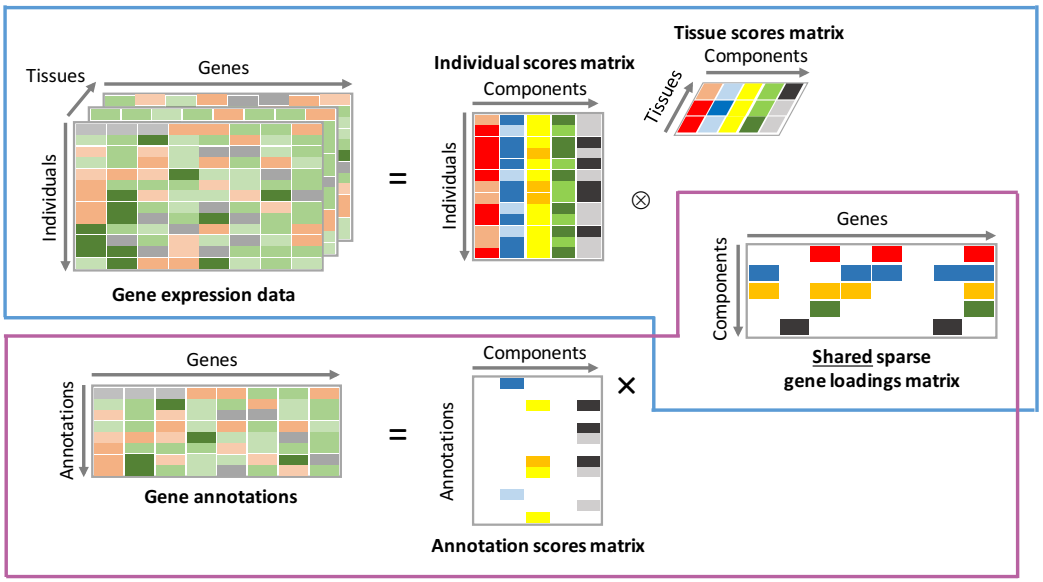

Figure 6.2: Illustration of a decomposition of gene expression data and gene annotations. A sparse gene loadings matrix is common to both decompositions.

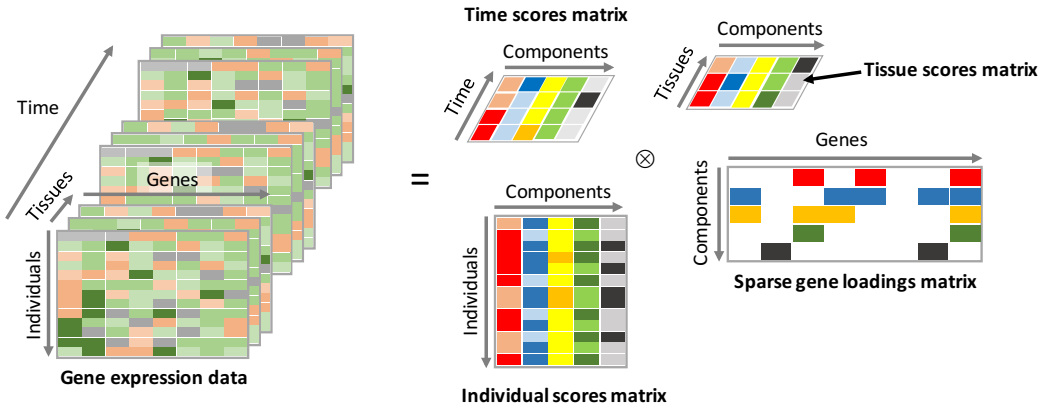

Figure 6.3: Illustration of a decomposition of a 4D array of gene expression data in multiple tissues at multiple time points.

Bibliography

F. W. Albert and L. Kruglyak (2015). "The role of regulatory variation in complex traits and disease". *Nature Reviews Genetics* 16.4, pp. 197–212.

B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, et al. (2002). *Molecular Biology of the Cell, Fourth Edition*. 4th ed. Garland Science.

A. Alexa, J. Rahnenführer, and T. Lengauer (2006). "Improved scoring of functional groups from gene expression data by decorrelating GO graph structure." *Bioinformatics* 22.13, pp. 1600–7.

G. I. Allen (2012). "Sparse Higher-Order Principal Components Analysis". *International Conference on Artificial Intelligence and Statistics*.

C. Archambeau and F. R. Bach (2009). "Sparse probabilistic projections". *Advances in neural information processing systems* 21, pp. 73–80.

A. Armagan, D. B. Dunson, and M. Clyde (2011). "Generalized Beta Mixture of Gaussians". *Advances in Neural Information Processing Systems* 24, pp. 523–531.

M. Ashburner, C. Ball, J. Blake, D. Botstein, H. Butler, et al. (2000). "Gene ontology: tool for the unification of biology." *Nature genetics* 25.1, pp. 25–29.

H. Attias (2000). "A variational Bayesian framework for graphical models". *Advances in neural information processing systems*.

F. R. Bach and M. I. Jordan (2005). "A Probabilistic Interpretation of Canonical Correlation Analysis".

A. Battle, S. Mostafavi, X. Zhu, J. B. Potash, M. M. Weissman, et al. (2014). "Characterizing the genetic basis of transcriptome diversity through RNA-sequencing of 922 individuals." *Genome research* 24.1, pp. 14–24.

C. Beckmann and S. Smith (2005). "Tensorial extensions of independent component analysis for multisubject FMRI analysis". *Neuroimage* 25, pp. 294–311.

J. T. Bell, A. A. Pai, J. K. Pickrell, D. J. Gaffney, R. Pique-Regi, et al. (2011). "DNA methylation patterns associate with genetic and gene expression variation in HapMap cell lines." *Genome biology* 12.1, R10.

J. E. Below, E. R. Gamazon, J. V. Morrison, A. Konkashbaev, A. Pluzhnikov, et al. (2011). "Genome-wide association and meta-analysis in populations from Starr County, Texas, and Mexico City identify type 2 diabetes sus-

ceptibility loci and enrichment for expression quantitative trait loci in top signals". *Diabetologia* 54.8, pp. 2047–2055.

R. Breitling, Y. Li, B. M. Tesson, J. Fu, C. Wu, et al. (2008). "Genetical Genomics: Spotlight on QTL Hotspots". *PLoS Genetics* 4.10, e1000232.

A. A. Brown, A. Buil, A. Viñuela, T. Lappalainen, H.-F. Zheng, et al. (2014). "Genetic interactions affecting human gene expression identified by variance association mapping." *eLife* 3.2012, e01381.

M. W. Browne (1979). "The maximum-likelihood solution in inter-battery factor analysis". *British Journal of Mathematical and Statistical Psychology* 32.1, pp. 75–86.

B. Brynedal, T. Raj, B. E. Stranger, R. Bjornson, and B. M. Neale (2014). "Cross-phenotype meta-analysis reveals large-scale trans-eQTLs mediating patterns of transcriptional co-regulation". *arXiv:1402.1728*.

J. Bryois, A. Buil, D. M. Evans, J. P. Kemp, S. B. Montgomery, et al. (2014). "Cis and trans effects of human genomic variants on gene expression." *PLoS genetics* 10.7, e1004461.

A. Buil, A. A. Brown, T. Lappalainen, A. Viñuela, M. N. Davies, et al. (2015). "Gene-gene and gene-environment interactions detected by transcriptome sequence analysis in twins". *Nature Genetics* 47.1, pp. 88–91.

V. D. Calhoun (2010). "A review of group ICA for fMRI data and ICA for joint inference of imaging genetic and ERP data". *Neuroimage* 45.1, S163–S172.

J. D. Carroll and J.-J. Chang (1970). "Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition". *Psychometrika* 35.3, pp. 283–319.

C. M. Carvalho, J. Chang, J. E. Lucas, J. R. Nevins, Q. Wang, et al. (2008). "High-Dimensional Sparse Factor Modeling: Applications in Gene Expression Genomics." *Journal of the American Statistical Association* 103.484, pp. 1438–1456.

C. M. Carvalho, N. G. Polson, and J. G. Scott (2009). "Handling Sparsity via the Horseshoe". *Journal of Machine Learning Research WCP* 5.73-80, pp. 73–80.

K. Chan, T. Lee, and T. Sejnowski (2002). "Handling missing data with variational Bayesian learning of ICA". *Advances in Neural Information Processing Systems*.

P. Comon (1994). "Independent component analysis, A new concept?" *Signal Processing* 36.3, pp. 287–314.

A. S. Dimas, S. Deutsch, B. E. Stranger, B Stephen, C. Borel, et al. (2010). "Common regulatory variation impacts gene expression in a cell type dependent manner". 325.5945, pp. 1246–1250.

A. S. Dimas, A. A. Nica, S. Montgomery, B. Stranger, T. Raj, et al. (2012). "Sex-biased genetic effects on gene regulation in humans". *Genome research* 22.12, pp. 2368–2375.

S. Edwards, J. Beesley, J. French, and A. Dunning (2013). "Beyond GWASs: Illuminating the Dark Road from Association to Function". *The American Journal of Human Genetics* 93.5, pp. 779–797.

B. E. Engelhardt and M. Stephens (2010). "Analysis of population structure: a unifying framework and novel methods based on sparse factor analysis." *PLoS genetics* 6.9, e1001117.

B. Ermiş, E. Acar, and A. T. Cemgil (2013). "Link prediction in heterogeneous data via generalized coupled tensor factorization". *Data Mining and Knowledge Discovery* 29.1, pp. 203–236.

I. Ezkurdia, D. Juan, J. M. Rodriguez, A. Frankish, M. Diekhans, et al. (2014). "Multiple evidence strands suggest that there may be as few as 19 000 human protein-coding genes." *Human molecular genetics* 23.22, pp. 5866–5878.

K. Fink and N. Grandvaux (2013). "STAT2 and IRF9: Beyond ISGF3." *Jak-Stat* 2.4, e27521.

T. Flutre, X. Wen, J. Pritchard, and M. Stephens (2013). "A statistical framework for joint eQTL analysis in multiple tissues." *PLoS genetics* 9.5, e1003486.

V. François-Newton, G. Magno de Freitas Almeida, B. Payelle-Brogard, D. Monneron, L. Pichard-Garcia, et al. (2011). "USP18-Based Negative Feedback Control Is Induced by Type I and Type III Interferons and Specifically Inactivates Interferon α Response". *PLoS ONE* 6.7, e22200.

N. Fusi, O. Stegle, and N. D. Lawrence (2012). "Joint modelling of confounding factors and prominent genetic regulators provides increased accuracy in genetical genomics studies". *PLoS Computational Biology* 8.1, e1002330.

C. Gao, C. D. Brown, and B. E. Engelhardt (2013). "A latent factor model with a mixture of sparse and dense factors to model gene expression data with confounding effects". *arXiv:1310.4792v1*, pp. 1–28.

P. N. Ghule, Z. Dominski, X.-C. Yang, W. F. Marzluff, K. A. Becker, et al. (2008). "Staged assembly of histone gene expression machinery at subnuclear foci in the abbreviated cell cycle of human embryonic stem cells." *Proceedings of the National Academy of Sciences of the United States of America* 105.44, pp. 16964–9.

D. Glass, A. Viñuela, M. N. Davies, A. Ramasamy, L. Parts, et al. (2013). "Gene expression changes with age in skin, adipose tissue, blood and brain." *Genome biology* 14.7, R75.

A. Goldinger, A. K. Henders, A. F. McRae, N. G. Martin, G. Gibson, et al. (2013). "Genetic and nongenetic variation revealed for the principal components of human gene expression". *Genetics* 195.3, pp. 1117–1128.

A. Goldinger, K. Shakhbazov, A. K. Henders, A. F. McRae, G. W. Montgomery, et al. (2015). "Seasonal Effects on Gene Expression". *Plos One* 10.5, e0126995.

I. J. Goodfellow, A. Courville, and Y. Bengio (2013). "Scaling up spike-and-slab models for unsupervised feature learning". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8, pp. 1902–1914.

A. R. Groves, C. F. Beckmann, S. M. Smith, and M. W. Woolrich (2011). "NeuroImage Linked independent component analysis for multimodal data fusion". *NeuroImage* 54.3, pp. 2198–2217.

E. Grundberg, V. Adoue, T. Kwan, B. Ge, Q. L. Duan, et al. (2011). "Global Analysis of the Impact of Environmental Perturbation on cis-Regulation of Gene Expression". *PLoS Genetics* 7.1, e1001279.

E. Grundberg, K. S. Small, A. K. Hedman, A. C. Nica, A. Buil, et al. (2012). "Mapping cis- and trans-regulatory effects across multiple tissues in twins". *Nature Genetics* 44.10, pp. 1084–1089.

J. S. Hamid, P. Hu, N. M. Roslin, V. Ling, C. M. T. Greenwood, et al. (2009). "Data integration in genetics and genomics: methods and challenges." *Human genomics and proteomics* 1.

K. D. Hansen, S. E. Brenner, and S. Dudoit (2010). "Biases in Illumina transcriptome sequencing caused by random hexamer priming". *Nucleic Acids Research* 38.12, e131–e131.

D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor (2004). "Canonical correlation analysis: An overview with application to learning methods". *Neural Computation* 2664, pp. 2639–2664.

J. Harrow, A. Frankish, J. M. Gonzalez, E. Tapanari, M. Diekhans, et al. (2012). "GENCODE: The reference human genome annotation for the ENCODE project". *Genome Research* 22.9, pp. 1760–1774.

R. A. Harshman and M. E. Lundy (1994). "PARAFAC: Parallel factor analysis". *Computational Statistics & Data Analysis* 18.1, pp. 39–72.

A. J. Hartemink (2005). "Reverse engineering gene regulatory networks". *Nature Biotechnology* 23.5, pp. 554–555.

M. Heinig, E. Petretto, C. Wallace, L. Bottolo, M. Rotival, et al. (2010). "A trans-acting locus regulates an anti-viral expression network and type 1 diabetes risk." *Nature* 467.7314, pp. 460–464.

H. Hotelling (1936). "Relations between two sets of variates". *Biometrika* 28.3, pp. 321–377.

B. N. Howie, P. Donnelly, and J. Marchini (2009). "A flexible and accurate genotype imputation method for the next generation of genome-wide association studies". *PLoS Genetics* 5.6, e1000529.

T. S. Jaakkola and M. I. Jordan (2000). "Bayesian parameter estimation via variational methods". *Statistics And Computing* 10.1, pp. 25–37.

C. A. Janeway, P. Travers, M. J. Walport, and M. J. Shlomchik (2001). *Immunobiology: the immune system in health and disease*. Vol. 2 London: Churchill Livingstone London.

A. R. Joyce and B. Palsson (2006). "The model organism as a system: integrating 'omics' data sets." *Nature reviews. Molecular cell biology* 7.3, pp. 198–210.

H. F. Kaiser (1958). "The varimax criterion for analytic rotation in factor analysis." *Psychometrika* 23, pp. 187–200.

S. Kakuta, S. Shibata, and Y. Iwakura (2002). "Genomic structure of the mouse 2',5'-oligoadenylate synthetase gene family." *Journal of interferon & cytokine research : the official journal of the International Society for Interferon and Cytokine Research* 22.9, pp. 981–993.

S. A. Khan, S. Virtanen, O. P. Kallioniemi, K. Wennerberg, A. Poso, et al. (2014a). "Identification of structural features in chemicals associated with cancer drug response: a systematic data-driven analysis". *Bioinformatics* 30.17, pp. i497–i504.

S. A. Khan and S. Kaski (2014). "Bayesian Multi-View Tensor Factorization". *Machine Learning and Knowledge Discovery in Databases, ECML PKDD*, pp. 656–671.

S. A. Khan, E. Leppäaho, and S. Kaski (2014b). "Multi-tensor factorization". *arXiv:1412.4679v1*, p. 22.

D. Kim (2015). "Methods of integrating data to uncover genotype – phenotype interactions". *Nature Publishing Group* 16.2, pp. 85–97.

A. Klami and S. Kaski (2006). "Generative models that discover dependencies between data sets". *Proceedings of Machine Learning for Signal Processing*, pp. 123–128.

A. Klami, S Virtanen, and S Kaski (2013). "Bayesian canonical correlation analysis". *The Journal of Machine Learning Research* 14, pp. 965–1003.

A. Klami, S. Virtanen, E. Leppäaho, and S. Kaski (2014). "Group Factor Analysis". *Neural Networks and Learning Systems, IEEE Transactions* 26.9.

D. Knowles and Z. Ghahramani (2011). "Nonparametric Bayesian sparse factor models with application to gene expression modeling". *The Annals of Applied Statistics* 5.2B, pp. 1534–1552.

K. S. Kobayashi and P. J. van den Elsen (2012). "NLRC5: a key regulator of MHC class I-dependent immune responses." *Nature reviews. Immunology* 12.12, pp. 813–20.

T. G. Kolda and B. W. Bader (2009). "Tensor Decompositions and Applications". *SIAM review* 51.3, pp. 455–500.

W. Kong, X. Mou, Q. Liu, and Z. Chen (2009). "Independent component analysis of Alzheimer's DNA microarray gene expression data". *Molecular Neurodegeneration* 4.1, pp. 1–14.

A. Korte, B. J. Vilhjálmsson, V. Segura, A. Platt, Q. Long, et al. (2012). "A mixed-model approach for genome-wide association studies of correlated traits in structured populations." *Nature genetics* 44.9, pp. 1066–71.

M Krawczyk, Q Seguin-Estevez, E Leimgruber, P Sperisen, C Schmid, et al. (2008). "Identification of CIITA regulated genetic module dedicated for antigen presentation". *PLoS Genet* 4.4, e1000058.

J. B. Kruskal (1977). "Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics". *Linear Algebra and its Applications* 18.2, pp. 95–138.

M. T. Landi, T. Dracheva, M. Rotunno, J. D. Figueroa, H. Liu, et al. (2008). "Gene expression signature of cigarette smoking and its role in lung adenocarcinoma development and survival." *PloS one* 3.2, e1651.

T. Lappalainen, M. Sammeth, M. R. Friedländer, P. A. C. 't Hoen, J. Monlong, et al. (2013). "Transcriptome and genome sequencing uncovers functional variation in humans." *Nature* 501.7468, pp. 506–11.

M. Lemire, S. H. Zaidi, M. Ban, B. Ge, D. Aïssi, et al. (2015). "Long-range epigenetic regulation is conferred by genetic variation located at thousands of independent loci". *Nature Communications* 6.

H. Li and R. Durbin (2009). "Fast and accurate short read alignment with Burrows-Wheeler transform". *Bioinformatics* 25.14, pp. 1754–1760.

X. Li, Y. Ye, M. Ng, and Q. Wu (2013). "MultiFacTV: module detection from higher-order time series biological data." *BMC genomics* 14.Suppl 4, S2.

W. Lian, P. Rai, E. Salazar, and L. Carin (2015). "Integrating Features and Similarities : Flexible Models for Heterogeneous Multiview Data". *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

J. C. Loehlin (1998). *Latent variable models: An introduction to factor, path, and structural analysis* . Lawrence Erlbaum Associates Publishers.

J. Lucas, C. Carvalho, Q. Wang, A. Bild, J. Nevins, et al. (2006). "Sparse statistical modelling in gene expression genomics". *Bayesian Inference for Gene Expression and Proteomics* 1.

M. P. Malakhov, K. I. Kim, O. A. Malakhova, B. S. Jacobs, E. C. Borden, et al. (2003). "High-throughput Immunoblotting ubiquitin-like protein isg15 modifies key regulators of signal transduction". *Journal of Biological Chemistry* 278.19, pp. 16608–16613.

O. A. Malakhova, K. I. I. Kim, J.-K. Luo, W. Zou, K. G. S. Kumar, et al. (2006). "UBP43 is a novel regulator of interferon signaling independent of its ISG15 isopeptidase activity". *The EMBO Journal* 25.11, pp. 2358–2367.

M. T. Maurano, R. Humbert, E. Rynes, R. E. Thurman, H. Wang, et al. (2012). "Systematic Localization of Common Disease-Associated Variation in Regulatory DNA". *Science* 337.6099, pp. 1190–1195.

T. B. Meissner, A. Li, A. Biswas, K.-H. Lee, Y.-J. Liu, et al. (2010). "NLR family member NLRC5 is a transcriptional regulator of MHC class I genes." *Proceedings of the National Academy of Sciences of the United States of America* 107.31, pp. 13794–13799.

T. B. Meissner, Y.-J. Liu, K.-H. Lee, A. Li, A. Biswas, et al. (2012). "NLRC5 cooperates with the RFX transcription factor complex to induce MHC class I gene expression." *Journal of immunology* 188.10, pp. 4951–4958.

H. Mi, A. Muruganujan, J. T. Casagrande, and P. D. Thomas (2013). "Large-scale gene function analysis with the PANTHER classification system." *Nature protocols* 8.8, pp. 1551–66.

T. Mitchell and J. Beauchamp (1988). "Bayesian Variable Selection in Linear Regression". *Journal of the American Statistical Association* 83.404, pp. 1023–1032.

Q. Mo, S. Wang, V. E. Seshan, A. B. Olshen, N. Schultz, et al. (2013). "Pattern discovery and cancer gene identification in integrated cancer genomic data." *Proceedings of the National Academy of Sciences of the United States of America* 110.11, pp. 4245–50.

S. Mostafavi, A. Battle, X. Zhu, A. E. Urban, D. Levinson, et al. (2013). "Normalizing RNA-sequencing data by modeling hidden covariates with prior knowledge." *PloS one* 8.7, e68141.

D. Murphy (2002). "Gene Expression Studies Using Microarrays: Principles, Problems, and Prospects". *Advan Physiol Educ* 26.4, pp. 256–270.

M. Naylor, X. Lin, S. Weiss, B. Raby, and C. Lange (2010). "Using canonical correlation analysis to discover genetic regulatory variants". *PloS one* 5.5, e10395.

J Ng, L Barrett, A. Wong, and D. Kuh (2012). "The role of longitudinal cohort studies in epigenetic epidemiology: challenges and opportunities". *Genome Biology* 13.6, p. 246.

A. C. Nica, L. Parts, D. Glass, J. Nisbet, A. Barrett, et al. (2011). "The architecture of gene regulatory variation across multiple human tissues: the MuTHER study." *PLoS genetics* 7.2, e1002003.

M. Nickel and V. Tresp (2013). "Logistic Tensor Factorization for Multi-Relational Data". *arXiv:1306.2084v1*.

D. L. Nicolae, E. Gamazon, W. Zhang, S. Duan, M. E. Dolan, et al. (2010). "Trait-associated SNPs are more likely to be eQTLs: annotation to enhance discovery from GWAS." *PLoS genetics* 6.4, e1000888.

J. Novembre, T. Johnson, K. Bryc, Z. Kutalik, A. R. Boyko, et al. (2008). "Genes mirror geography within Europe." *Nature* 456.7218, pp. 98–101.

L. Omberg, G. H. Golub, and O. Alter (2007). "A tensor higher-order singular value decomposition for integrative analysis of DNA microarray data from

different studies". *Proceedings of the National Academy of Sciences* 104.47, pp. 18371–18376.

F. Ozsolak and P. M. Milos (2011). "RNA sequencing: advances, challenges and opportunities." *Nature reviews. Genetics* 12.2, pp. 87–98.

O. H. M. Padilla and J. G. Scott (2015). "Tensor decomposition with generalized lasso penalties". *arXiv:1502.06930v1*.

A. A. Pai, J. K. Pritchard, and Y. Gilad (2015). "The Genetic and Mechanistic Basis for Variation in Gene Regulation". 11.1.

J. K. Pickrell, J. C. Marioni, A. A. Pai, J. F. Degner, B. E. Engelhardt, et al. (2010). "Understanding mechanisms underlying human gene expression variation with RNA sequencing". *Nature* 464.7289, pp. 768–772.

B. L. Pierce, L. Tong, L. S. Chen, R. Rahaman, M. Argos, et al. (2014). "Mediation Analysis Demonstrates That Trans-eQTLs Are Often Explained by Cis-Mediation: A Genome-Wide Analysis among 1,800 South Asians." *PLoS genetics* 10.12, e1004818.

R. S. Pillai, M. Grimmler, G. Meister, C. L. Will, R. Lührmann, et al. (2003). "Unique Sm core structure of U7 snRNPs: Assembly by a specialized SMN complex and the role of a new component, Lsm11, in histone RNA processing". *Genes and Development* 17.18, pp. 2321–2333.

A. L. Price, A. Helgason, G. Thorleifsson, S. A. McCarroll, A. Kong, et al. (2011). "Single-Tissue and Cross-Tissue Heritability of Gene Expression Via Identity-by-Descent in Related or Unrelated Individuals". *PLoS Genetics* 7.2, e1001317.

P. Ray, L. Zheng, J. Lucas, and L. Carin (2014). "Bayesian joint analysis of heterogeneous genomics data." *Bioinformatics* 30.10, pp. 1370–6.

D. Reich, A. L. Price, and N. Patterson (2008). "Principal component analysis of genetic data". *Nature genetics* 40.5, pp. 491–492.

D. M. Reif, B. C. White, and J. H. Moore (2004). "Integrated analysis of genetic, genomic and proteomic data." *Expert review of proteomics*, pp. 67–75.

W. Reith, S. LeibundGut-Landmann, and J.-M. Waldburger (2005). "Regulation of MHC class II gene expression by the class II transactivator." *Nature reviews. Immunology* 5.10, pp. 793–806.

R. C. Richmond, A. J. Simpkin, G. Woodward, T. R. Gaunt, O. Lyttleton, et al. (2014). "Prenatal exposure to maternal smoking and offspring DNA methylation across the lifecourse: findings from the Avon Longitudinal Study of Parents and Children (ALSPAC)". *Human Molecular Genetics* ddu739.

J. Ronald, R. B. Brem, J. Whittle, and L. Kruglyak (2005). "Local Regulatory Variation in Saccharomyces cerevisiae". *PLoS Genetics* 1.2, e25.

M. Rotival, T. Zeller, P. S. Wild, S. Maouche, S. Szymczak, et al. (2011). "Integrating Genome-Wide Genetic Variations and Monocyte Expression Data Reveals Trans-Regulated Gene Modules in Humans". *PLoS Genetics* 7.12, e1002367.

T. Salimans and D. A. Knowles (2013). "Fixed-form variational posterior approximation through stochastic linear regression". *Bayesian Analysis* 8.4, pp. 837–882.

M. P. Scott-Boyer, G. C. Imholte, A. Tayeb, A. Labbe, C. F. Deschepper, et al. (2012). "An Integrated Hierarchical Bayesian Model for Multivariate eQTL Mapping". *Statistical Applications in Genetics and Molecular Biology* 11.4.

M. Seeger and G. Bouchard (2012). "Fast Variational Bayesian Inference for Non-Conjugate Matrix Factorization Models". *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics* No. EPFL-C.

R. Shen, A. B. Olshen, and M. Ladanyi (2009). "Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis." *Bioinformatics* 25.22, pp. 2906–12.

R. Shen, S. Wang, and Q. Mo (2013). "Sparse integrative clustering of multiple omics data sets". *The Annals of Applied Statistics* 7.1, pp. 269–294.

K. S. Small, A. K. Hedman, E. Grundberg, A. C. Nica, G. Thorleifsson, et al. (2011). "Identification of an imprinted master trans regulator at the KLF14 locus related to multiple metabolic phenotypes." *Nature genetics* 43.6, pp. 561–564.

C. Soneson, H. Lilljebjörn, T. Fioretos, and M. Fontes (2010). "Integrative analysis of gene expression and copy number alterations using canonical correlation analysis." *BMC bioinformatics* 11.1, p. 191.

C. Spearman (1904). "General Intelligence, Objectively Determined and Measured". *The American Journal of Psychology* 15.2, pp. 201–292.

C. Spearman (1927). "The abilities of man".

D. Speed and D. J. Balding (2014). "Relatedness in the post-genomic era: is it still useful?" *Nature Reviews Genetics* 16.1, pp. 33–44.

O. Stegle, L. Parts, R. Durbin, and J. Winn (2010). "A Bayesian framework to account for complex non-genetic factors in gene expression levels greatly increases power in eQTL studies." *PLoS computational biology* 6.5, e1000770.

B. E. Stranger and T. Raj (2013). "Genetics of human gene expression." *Current opinion in genetics & development* 23, pp. 627–34.

B. E. Stranger, S. B. Montgomery, A. S. Dimas, L. Parts, O. Stegle, et al. (2012). "Patterns of cis regulatory variation in diverse human populations." *PLoS genetics* 8.4, e1002639–e1002639.

W. Sun and Y. Hu (2013). "eQTL Mapping Using RNA-seq Data". *Statistics in Biosciences* 5.1, pp. 198–219.

C. S. Tang and M. a. R. Ferreira (2012). "A gene-based test of association using canonical correlation analysis." *Bioinformatics* 28.6, pp. 845–50.

A. E. Teschendorff, J. Zhuang, and M. Widschwendter (2011). "Independent surrogate variable analysis to deconvolve confounding factors in large-scale microarray profiling studies." *Bioinformatics* 27.11, pp. 1496–505.

T. M. Teslovich, K. Musunuru, A. V. Smith, A. C. Edmondson, I. M. Stylianou, et al. (2010). "Biological, clinical and population relevance of 95 loci for blood lipids". *Nature* 466.7307, pp. 707–713.

The 1000 Genomes Project Consortium (2012). "An integrated map of genetic variation from 1,092 human genomes". *Nature* 491.7422, pp. 56–65.

The GTEx Consortium (2015). "The Genotype-Tissue Expression (GTEx) pilot analysis: Multitissue gene regulation in humans". *Science* 348.6235, pp. 648–660.

The International HapMap 3 Consortium (2010). "Integrating common and rare genetic variation in diverse human populations." *Nature* 467.7311, pp. 52–58.

R. Tibshirani (1996). "Regression shrinkage and selection via the lasso". *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1, pp. 267–288.

M. Tipping and C. Bishop (1999). "Probabilistic principal component analysis". *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61.3, pp. 611–622.

M. E. Tipping (2001). "Sparse Bayesian Learning and the Relevance Vector Machine". *Journal of Machine Learning Research*, pp. 211–244.

M. Titsias and M. Lázaro-Gredilla (2011). "Spike and slab variational inference for multi-task and multiple kernel learning". *Neural Information Processing Systems*, pp. 1–9.

L. R. Tucker (1958). "An inter-battery method of factor analysis". *Psychometrika* 23.2, pp. 111–136.

R. Urrutia (2003). "KRAB-containing zinc-finger repressor proteins." *Genome biology* 4.10, p. 231.

S. Virtanen, A. Klami, and S. Kaski (2011). "Bayesian CCA via group sparsity". *Proceedings of the 28th International Conference on Machine Learning*, pp. 2339–2347.

S. Virtanen, A. Klami, S. A. Khan, and S. Kaski (2012). "Bayesian group factor analysis". *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics*, pp. 1269–1277.

B. F. Voight, L. J. Scott, V. Steinthorsdottir, A. P. Morris, C. Dina, et al. (2010). "Twelve type 2 diabetes susceptibility loci identified through large-scale association analysis." *Nature genetics* 42.7, pp. 579–589.

Z. Wang, M. Gerstein, and M. Snyder (2009). "RNA-Seq: a revolutionary tool for transcriptomics." *Nature reviews. Genetics* 10.1, pp. 57–63.

M. Weiser, S. Mukherjee, and T. S. Furey (2014). "Novel Distal eQTL Analysis Demonstrates Effect of Population Genetic Architecture on Detecting and Interpreting Associations". *Genetics* 198.3, pp. 879–893.

M. Welling and M. Weber (2001). "Positive tensor factorization". *Pattern Recognition Letters* 22.12, pp. 1255–1261.

D. Welter, J. MacArthur, J. Morales, T. Burdett, P. Hall, et al. (2014). "The NHGRI GWAS Catalog, a curated resource of SNP-trait associations". *Nucleic Acids Research* 42, pp. 1001–1006.

M. West (2003). "Bayesian factor regression models in the "large p, small n" paradigm". *Bayesian Statistics* 7, pp. 733–742.

H.-J. Westra and L. Franke (2014). "From genome to function by studying eQTLs." *Biochimica et biophysica acta* 1842.10, pp. 1896–1902.

D. M. Witten and R. J. Tibshirani (2009). "Extensions of Sparse Canonical Correlation Analysis with Applications to Genomic Data". *Statistical applications in genetics and molecular biology* 8.1.

D. Wong, W. Lee, P. Humburg, S. Makino, E. Lau, et al. (2014). "Genomic mapping of the MHC transactivator CIITA using an integrated ChIP-seq and genetical genomics approach". *Genome Biology* 15, pp. 1–15.

C. Yao, B. Chen, R. Joehanes, B. Otlu, X. Zhang, et al. (2015). "Integromic Analysis of Genetic Variation and Gene Expression Identifies Networks for Cardiovascular Disease Phenotypes". *Circulation* 131.6, pp. 536–549.

T. Yokota, A. Cichocki, and Y. Yamashita (2012). *Linked PARAFAC/CP tensor decomposition and its fast implementation for multi-block tensor analysis*, 84–91 Springer Berlin Heidelberg.

Q. Zhao, L. Zhang, and A. Cichocki (2014a). "Bayesian CP Factorization of Incomplete Tensors with Automatic Rank Determination". *arXiv:1401.6497v2*.

S. Zhao, C. Gao, S. Mukherjee, and B. E. Engelhardt (2014b). "Bayesian group latent factor analysis with structured sparse priors". *arXiv:1441.2698v1*.

X. Zhou and M. Stephens (2014). "Efficient multivariate linear mixed model algorithms for genome-wide association studies." *Nature methods* 11.4, pp. 407–9.

Appendix A

Variational Bayes results and additional updates

A.1 Definitions of probability distributions

Normal with mean μ and precision λ, $\mathcal{N}(x|\mu, \lambda^{-1})$

$$f(x|\mu, \lambda) = \sqrt{\frac{\lambda}{2\pi}} e^{-\frac{\lambda}{2}(x-\mu)^2}$$

Multivariate Normal with mean $\boldsymbol{\mu}$ and precision matrix Σ, $\text{MVN}(\mathbf{x}|\boldsymbol{\mu}, \Sigma^{-1})$

$$f(\mathbf{x}|\boldsymbol{\mu}, \Sigma) = \sqrt{\frac{|\Sigma|}{(2\pi)^n}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \Sigma (\mathbf{x}-\boldsymbol{\mu})}$$

Gamma with shape k and scale θ, $\mathcal{G}(x|k, \theta)$

$$f(x|k, \theta) = \frac{1}{\Gamma(k)\theta^k} x^{k-1} e^{-\frac{x}{\theta}}$$

Beta with shape parameters $a > 0$ and $b > 0$, $\mathcal{B}(x|a, b)$

$$f(x|a, b) = \frac{1}{B(a, b)} x^{a-1}(1-x)^{b-1}$$
$$\text{where} \quad B(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$$

Bernoulli with parameter $p \in (0, 1)$

$$f(x|p) = p^x(1-p)^{1-x}$$
$$f(x=0|p) = p,\ f(x=1|p) = 1-p$$

A.2 Derivation of variational Bayes mean field approximation

This section contains a derivation of the update rules for the mean field VB approximation and uses the notation defined in section 3.3.4. Assume the approximate posterior distribution fully factorises,

$$Q(\theta) = \prod_i Q(\theta_i).$$

For an arbitrary i, let $Q(\theta) = Q(\theta_{-i})Q(\theta_i)$ where θ_{-i} denotes the set of all parameters except θ_i. The negative free energy can be rewritten as

$$\begin{aligned} F\big(Q(\Theta)\big) &= \left\langle \log \frac{P(\theta, \mathcal{Y})}{Q(\theta_{-i})Q(\theta_i)} \right\rangle_{Q(\theta_{-i})Q(\theta_i)} \\ &= \big\langle \log P(\theta, \mathcal{Y}) \big\rangle_{Q(\theta_{-i})Q(\theta_i)} - \big\langle \log Q(\theta_i) \big\rangle_{Q(\theta_{-i})Q(\theta_i)} - \big\langle \log Q(\theta_{-i}) \big\rangle_{Q(\theta_{-i})Q(\theta_i)} \\ &= \big\langle \log P(\theta, \mathcal{Y}) \big\rangle_{Q(\theta_{-i})Q(\theta_i)} - \big\langle \log Q(\theta_i) \big\rangle_{Q(\theta_i)} - \sum_{j \neq i} \big\langle \log Q(\theta_j) \big\rangle_{Q(\theta_j)}. \end{aligned}$$

Define the distribution $\tilde{Q}(\theta_i)$ as

$$\tilde{Q}(\theta_i) = \frac{1}{z} \exp\Big(\big\langle \log P(\theta, \mathcal{Y}) \big\rangle_{Q(\Theta_{-i})} \Big),$$

or equivalently,

$$\log z + \log \tilde{Q}(\theta_i) = \big\langle \log P(\theta, \mathcal{Y}) \big\rangle_{Q(\theta_{-i})}.$$

where z is a normalising constant. In terms of $\tilde{Q}(\theta_i)$, the negative free energy becomes,

$$\begin{aligned} F\big(Q(\theta)\big) &= \big\langle \log z + \log \tilde{Q}(\theta_i) \big\rangle_{Q(\theta_i)} - \big\langle \log Q(\theta_i) \big\rangle_{Q(\theta_i)} - \sum_{j \neq i} \big\langle \log Q(\theta_j) \big\rangle_{Q(\theta_j)} \\ &= -\left\langle \log \frac{Q(\theta_i)}{\tilde{Q}(\theta_i)} \right\rangle_{Q(\theta_i)} + \log z - \sum_{j \neq i} \big\langle \log Q(\theta_j) \big\rangle_{Q(\theta_j)} \\ &= -D_{KL}\big(Q(\theta_i) | \tilde{Q}(\theta_i)\big) + \log z - \sum_{i \neq j} \big\langle \log Q(\theta_j) \big\rangle_{Q(\theta_j)} \end{aligned}$$

It is now obvious that maximising $F\big(Q(\theta)\big)$ with respect to $Q(\theta_i)$ is equivalent to minimising $D_{KL}\big(Q(\theta_i)|\tilde{Q}(\theta_i)\big)$, and this occurs when $Q(\theta_i) = \tilde{Q}(\theta_i)$.

A.3 Tensor decomposition with spike and slab prior from Mitchell and Beauchamp (1988)

This section details the updates for a formulation of SPIDER with a less general spike and slab prior on the gene loadings matrix. Specifically, the prior on x_{cl} is given by (Mitchell and Beauchamp, 1988),

$$P(x_{cl}|p_c) = p_c \mathcal{N}(x_{cl}|0, \beta_c^{-1}) + (1 - p_c)\delta_0(x_{cl}) \tag{A.1}$$

$$P(p_c) = \mathcal{B}\text{ernoulli}(p_c|g, h) \tag{A.2}$$

$$P(\beta_c) = \mathcal{G}(\beta_c|e, f). \tag{A.3}$$

Note that a single mixing parameter p_c is learnt for each component. A factorisation of the spike and slab distribution into the product of a Gaussian and

Bernoulli random variable is used, $x_{cl} = w_{cl}s_{cl}$. The full model is given by,

$$
\begin{aligned}
P(\mathcal{Y}|\theta) &= \prod_{l,t} \mathcal{N}_N\Big(\mathbf{y}_{lt}|\sum_c \mathbf{a}_{\cdot c} b_{tc} w_{cl} s_{cl}, \lambda_{lt}^{-1} I_N\Big) \\
P(a_{nc}) &= \mathcal{N}(a_{nc}|0,1) \\
P(b_{tc}) &= \mathcal{N}(b_{tc}|0,1) \\
P(w_{cl}|\beta_c) &= \mathcal{N}\big(w_{cl}|0,\beta_c^{-1}\big) \\
P(s_{cl}|p_c) &= \mathcal{B}\text{ernoulli}\big(s_{cl}|p_c\big) \\
P(\beta_c) &= \mathcal{G}(\beta_c|e,f) \\
P(p_c) &= \mathcal{B}\text{eta}(p_c|g,h) \\
P(\lambda_{lt}) &= \mathcal{G}(\lambda_{lt}|u,v)
\end{aligned} \tag{A.4}
$$

where $\theta = (A, B, W, S, \beta, P, \Lambda)$ denotes the set of all parameters. The approximate posterior distribution $Q(\theta)$ for (A.4) takes the following form,

$$
\begin{aligned}
Q(\theta) = \prod_{n,c} Q(a_{nc}) \prod_{t,c} Q(b_{tc}) \prod_{c,l} Q(w_{cl}|s_{cl}) Q(s_{cl}) \prod_c Q(\beta_c) \\
\prod_c Q(p_c) \prod_{l,t} Q(\lambda_{lt}).
\end{aligned} \tag{A.5}
$$

Updates for a_{nc}, b_{tc} and λ_{lt} are identical to those given in section 3.3.4.1. The remaining updates for (A.4) are

$$
\begin{aligned}
Q(w_{cl}|s_{cl}) &= \mathcal{N}\Big(w_{cl}\Big|s_{cl}m^*_{cl}, \big(s_{cl}\sigma^*_{cl} + (1-s_{cl})\langle\beta_c\rangle\big)^{-1}\Big) \\
\sigma^*_{cl} &= \langle\beta_c\rangle + \sum_{n,t}\langle\lambda_{lt}\rangle\langle a^2_{nc}\rangle\langle b^2_{tc}\rangle \\
m^*_{cl} &= \sigma^{*-1}_{cl}\Big(\sum_{n,t}\langle\lambda_{lt}\rangle y_{nlt}\langle a_{nc}\rangle\langle b_{tc}\rangle - \sum_{n,t}\langle\lambda_{lt}\rangle\langle a_{nc}\rangle\langle b_{tc}\rangle\sum_{k\neq c}\langle w_{kl}s_{kl}\rangle\langle a_{nk}\rangle\langle b_{tk}\rangle\Big)
\end{aligned}
\tag{A.6}
$$

$$
\begin{aligned}
Q(s_{cl}) &= \mathcal{B}\text{ernoulli}(s_{cl}|\gamma^*_{cl}) \\
\gamma^*_{cl} &= \frac{1}{1+e^{-u^*_{cl}}} \\
u^*_{cl} &= \log\langle p_c\rangle - \frac{1}{2}\log\sigma^*_{cl} + \frac{\sigma^*_{cl}}{2}{m^*_{cl}}^2 - \log(1-\langle p_c\rangle) + \frac{1}{2}\log\langle\beta_c\rangle \\
Q(\beta_c) &= \mathcal{G}(e^*_c, f^*_c) \\
e^*_c &= e + \frac{L}{2} \\
f^*_c &= \Big(\frac{1}{f} + \frac{1}{2}\sum_l\langle w^2_{cl}\rangle\Big)^{-1} \\
Q(p_c) &= \mathcal{G}(g^*_c, h^*_c) \\
g^*_c &= g + \sum_l\langle s_{cl}\rangle \\
h^*_c &= h + L - \sum_l\langle s_{cl}\rangle
\end{aligned}
$$

A.4 Updates for linked tensor decomposition

A.4.1 Full model

$$
\begin{aligned}
P(\{\mathcal{Y}^{(1)}, \cdots, \mathcal{Y}^{(D)}\}|\theta) &= \prod_{d,l,t} \mathcal{N}_N\Big(\mathbf{y}_{lt}^{(d)} | \textstyle\sum_c \mathbf{a}_{\cdot c} b_{tc}^{(d)} w_{cl}^{(d)} s_{cl}^{(d)}, (\lambda_{lt}^{(d)})^{-1} I_N\Big) \\
P(a_{nc}) &= \mathcal{N}(a_{nc}|0,1) \\
P(b_{tc}^{(d)}) &= \mathcal{N}(b_{tc}^{(d)}|0,1) \\
P(w_{cl}^{(d)}|\beta_c^{(d)}) &= \mathcal{N}\big(w_{cl}^{(d)}|0, (\beta_c^{(d)})^{-1}\big) \\
P(s_{cl}^{(d)}|\psi_{cl}^{(d)}, \phi_{cl}^{(d)}) &= \mathcal{B}\text{ernoulli}\big(s_{cl}^{(d)}|\psi_{cl}^{(d)}\phi_{cl}^{(d)}\big) \\
P(\beta_c^{(d)}) &= \mathcal{G}(\beta_c^{(d)}|e,f) \\
P(\psi_{cl}^{(d)}) &= \mathcal{B}\text{eta}(\psi_{cl}^{(d)}|g,h) \\
P(\phi_{cl}^{(d)}|\rho_c^{(d)}) &= \mathcal{B}\text{ernoulli}(\psi_{cl}^{(d)}|\rho_c^{(d)}) \\
P(\rho_c^{(d)}) &= \mathcal{B}\text{eta}(\rho_c^{(d)}|r,z) \\
P(\lambda_{lt}^{(d)}) &= \mathcal{G}(\lambda_{lt}^{(d)}|u,v)
\end{aligned} \tag{A.7}
$$

A.4.2 VB approximation

$$
\begin{aligned}
Q(\theta) = &\prod_{n,c} Q(a_{nc}) \prod_{d,t,c} Q(b_{tc}^{(d)}) \prod_{d,c,l} Q(w_{cl}^{(d)}|s_{cl}^{(d)}) Q(s_{cl}^{(d)}) \prod_{d,c} Q(\beta_c^{(d)}) \\
&\prod_{d,c,l} \delta_{\psi_{cl}^{(d)*}}(\psi_{cl}^{(d)}) \prod_{d,c,l} \delta_{\phi_{cl}^{(d)*}}(\phi_{cl}^{(d)}) \prod_{d,c} \delta_{\rho_c^{(d)*}}(\rho_c^{(d)}) \prod_{d,l,t} Q(\lambda_{lt}^{(d)})
\end{aligned} \tag{A.8}
$$

A.4.3 Update for A

Updates for all parameters other than A are identical to the updates for the single tensor decomposition so are not repeated here.

$$
\begin{aligned}
Q(a_{nc}) &= \mathcal{N}(a_{nc}|\mu^*_{nc}, (\omega^*_{nc})^{-1}) \\
\omega^*_{nc} &= 1 + \sum_{d,l,t} \langle \lambda^{(d)}_{lt} \rangle \langle (b^{(d)}_{tc})^2 \rangle \langle (w^{(d)}_{cl})^2 (s^{(d)}_{cl})^2 \rangle \\
\mu^*_{nc} &= \omega^{*-1}_{nc} \Big(\sum_{d,l,t} \langle \lambda^{(d)}_{lt} \rangle y^{(d)}_{nlt} \langle b^{(d)}_{tc} \rangle \langle w^{(d)}_{cl} s^{(d)}_{cl} \rangle - \sum_{d,l,t} \langle \lambda^{(d)}_{lt} \rangle \langle b^{(d)}_{tc} \rangle \langle w^{(d)}_{cl} s^{(d)}_{cl} \rangle \sum_{k \neq c} \langle a_{nk} \rangle \langle b^{(d)}_{tk} \rangle \langle w^{(d)}_{kl} s^{(d)}_{kl} \rangle \Big)
\end{aligned}
$$

Appendix B

Additional simulation results

B.1 Comparison of spike and slab distributions

These simulations are designed to show the behaviour of two formulations of a spike and slab (S+S) prior on the loadings matrix in a latent variable model. Details of the two S+S priors are given below, the two priors follow the approaches given in Mitchell and Beauchamp (1988) and Lucas et al. (2006) and I will refer to them as S+S(Mitchell) and S+S(Lucas) respectively.

S+S(Mitchell)

$$x_{cl} \sim p_c \mathcal{N}(0, \beta_c^{-1}) + (1 - p_c)\delta_0(x_{cl})$$
$$p_c \sim \mathcal{B}\text{eta}(1, 1)$$
$$\beta_c \sim \mathcal{G}(\beta_c | e, f)$$

S+S(Lucas)

$$
\begin{aligned}
x_{cl} &\sim p_{cl}\mathcal{N}(0, \beta_c^{-1}) + (1 - p_{cl})\delta_0(x_{cl}) \\
p_{cl} &\sim \rho_c\mathcal{B}\text{eta}(g, h) + (1 - \rho_c)\delta_0(p_{cl}) \\
\rho_c &\sim \mathcal{B}\text{eta}(1, 1) \\
\beta_c &\sim \mathcal{G}(\beta_c|e, f)
\end{aligned}
$$

Data simulation: The data matrix, $Y \in \mathbb{R}^{N\times L}$, for $N = 50$ and $L = 500$ was simulated under a sparse factor analysis model, $Y = AX + E$, with $C = 8$ components. $A \in \mathbb{R}^{N\times C}$ was dawn from $\mathcal{N}(0, 1)$. $X \in \mathbb{R}^{C\times L}$ was simulated to be sparse, with 10% of the elements in X randomly drawn from $\mathcal{N}(0, 1)$, and the remaining elements set to zero. Finally, noise was also drawn from a Gaussian distribution, $E \in \mathbb{R}^{N\times L} \sim \mathcal{N}(0, 10)$. SPIDER was run on the data matrix with the two different S+S priors. Results from a single data set showing typical behaviours are presented.

Results: Figure B.1 show histograms of the estimated PIPs for the two spike and slab distributions. The PIPs for S+S(Lucas) are considerably cleaner with values either very close to 0 or to 1. The PIPs for S+S(Mitchell) on the other hand are scattered across the interval [0,1]. In particular, the mode is near 0.15, with very few PIPs close to zero. A more extreme version of this behaviour was seen when running a tensor decomposition with the S+S(Mitchell) prior on real gene expression; the model was failing to estimate any PIPs below a threshold. The reason for this is unclear, but it seems that the element-wise inclusion probabilities, S_{cl}, and component-level sparsity parameter, p_c, are highly coupled in the S+S(Mitchell) model. As a result, if p_c is large, all values of S_{cl} are constrained from shrinking to 0. The additional level of hierarchy used by S+S(Lucas) weakens the link between element-wise sparsity and component-wise sparsity, avoiding this behaviour.

Figure B.2 shows scatter plots of the estimated loadings against the true

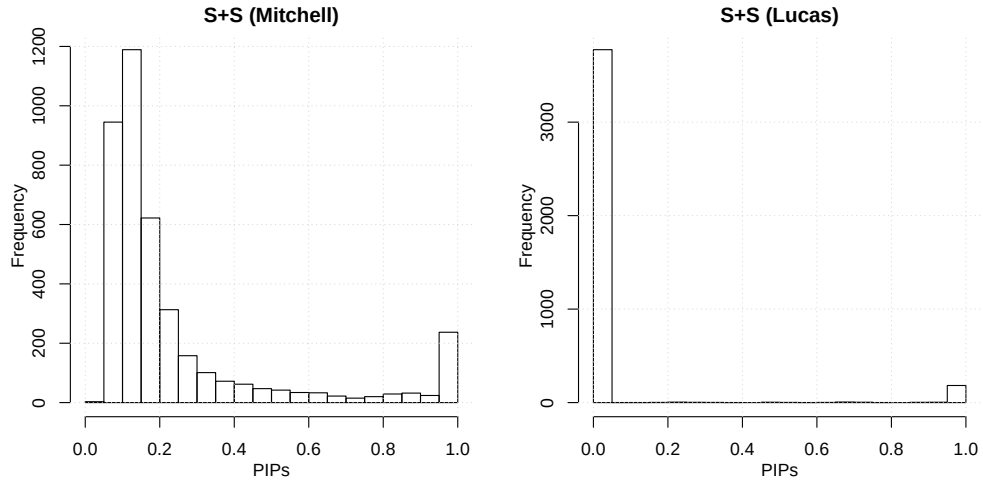

Figure B.1: Histogram of PIPs for different spike and slab priors.

loadings. Points on the vertical line are false negatives and points on the horizontal line are false positives. S+S(Mitchell) generates more false positives than S+S(Lucas), but has a higher true positive rate. Lucas et al. (2006) also report that S+S(Mitchell) generates a surprising number of false positives in a regression model with inference performed using MCMC.

The simulation parameters used here were chosen specifically to create a situation in which the two S+S priors showed different results. It is likely that the optimal choice of S+S prior is very dependent on the situation, and there are situations in which S+S(Lucas) outperforms S+S(Mitchell) and vica versa. The decision to pursue S+S(Lucas) was made because it produced more realistic looking signals on the TwinsUK multi-tissue gene expression data.

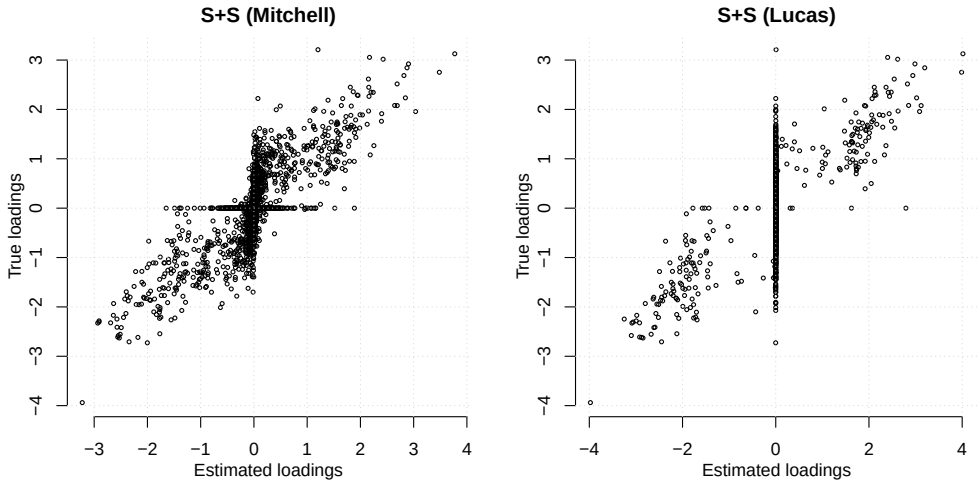

Figure B.2: Scatter plots of estimates and true loadings for different spike and slab priors.

B.2 Extended *trans* effect simulation results

M_{trans}	#tiss	Best free energy					Clustering				
		T_G	T_K	M_G	T_G^i	T_K^r	T_G	T_K	M_G	T_G^i	T_K^r
150	1	1.00	1.00	1.00	0.99	0.97	1.00	1.00	1.00	0.99	0.97
	2	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	3	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
75	1	1.00	1.00	1.00	1.00	0.94	1.00	1.00	1.00	1.00	0.94
	2	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	3	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
150	1	0.40	0.39	0.37	0.20	0.05	0.39	0.39	0.37	0.21	0.05
	2	0.82	0.82	0.36	0.38	0.20	0.82	0.84	0.39	0.58	0.28
	3	0.92	0.94	0.35	0.54	0.46	0.96	0.96	0.34	0.78	0.54
75	1	0.34	0.31	0.41	0.16	0.01	0.43	0.44	0.42	0.20	0.01
	2	0.70	0.78	0.39	0.42	0.12	0.80	0.76	0.39	0.56	0.14
	3	0.90	0.94	0.41	0.54	0.24	0.98	0.94	0.43	0.80	0.38

Table B.1: Fraction of simulated *trans* effects recovered. Results averaged across 50 data sets. Signals with effect sizes matching the *KLF14 trans* signal are indicated by * and signals with effect sizes half that of the *KLF14 trans* signal indicated by **. See section 4.2 for more details.

M_{trans}	#tiss	Best free energy					Clustering				
		T_G	T_K	M_G	T_G^i	T_K^r	T_G	T_K	M_G	T_G^i	T_K^r
150*	1	1.00(0.00)	1.00(0.01)	0.99(0.02)	1.00(0.01)	0.98(0.04)	0.99(0.06)	1.00(0.00)	0.99(0.06)	1.00(0.00)	0.99(0.01)
	2	1.00(0.00)	1.00(0.00)	0.99(0.09)	1.00(0.00)	0.99(0.03)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	0.99(0.01)
	3	1.00(0.00)	1.00(0.00)	0.99(0.08)	1.00(0.00)	0.99(0.03)	1.00(0.00)	1.00(0.00)	0.98(0.11)	1.00(0.00)	0.99(0.01)
75*	1	1.00(0.00)	1.00(0.00)	1.00(0.01)	1.00(0.01)	0.98(0.01)	1.00(0.00)	1.00(0.00)	0.99(0.05)	1.00(0.01)	0.98(0.01)
	2	1.00(0.00)	1.00(0.00)	1.00(0.01)	1.00(0.00)	0.99(0.01)	1.00(0.00)	1.00(0.00)	1.00(0.00)	1.00(0.00)	0.99(0.01)
	3	1.00(0.00)	1.00(0.00)	1.00(0.01)	1.00(0.01)	0.99(0.01)	1.00(0.00)	1.00(0.00)	1.00(0.01)	1.00(0.01)	0.99(0.01)
150**	1	0.84(0.04)	0.85(0.04)	0.68(0.18)	0.78(0.04)	0.61(0.08)	0.85(0.03)	0.85(0.03)	0.66(0.15)	0.78(0.04)	0.56(0.05)
	2	0.84(0.03)	0.83(0.07)	0.71(0.15)	0.71(0.12)	0.63(0.06)	0.84(0.04)	0.84(0.04)	0.59(0.22)	0.69(0.13)	0.60(0.07)
	3	0.83(0.06)	0.85(0.04)	0.64(0.19)	0.70(0.13)	0.60(0.06)	0.85(0.04)	0.85(0.04)	0.62(0.20)	0.71(0.07)	0.61(0.05)
75**	1	0.78(0.10)	0.77(0.11)	0.71(0.13)	0.70(0.11)	0.41(0.08)	0.77(0.06)	0.77(0.10)	0.70(0.12)	0.68(0.09)	0.35(0.05)
	2	0.80(0.10)	0.80(0.09)	0.67(0.16)	0.69(0.10)	0.37(0.10)	0.80(0.06)	0.80(0.07)	0.68(0.17)	0.67(0.11)	0.38(0.10)
	3	0.82(0.05)	0.82(0.06)	0.71(0.13)	0.68(0.10)	0.49(0.11)	0.83(0.05)	0.83(0.05)	0.69(0.13)	0.68(0.11)	0.47(0.08)

Table B.2: Power (1 s.d) to recover target genes in simulated *trans* effects, conditional on a *trans* effect being identified. Results averaged across 50 data sets. Signals with effect sizes matching the *KLF14 trans* signal are indicated by * and signals with effect sizes half that of the *KLF14 trans* signal indicated by **. See section 4.2 for more details.

B.3 Comparison of priors for the individual scores matrix

This section describes a simulation comparing two different priors on the individual scores matrix A. The vanilla prior on the A assumes $a_{nc} \sim \mathcal{N}(0,1)$ (section 3.1), and the method with this prior is referred to as T_G. A prior using information about the relatedness of the individuals is given by $\mathbf{a}_c \sim \mathcal{N}_N(0, \alpha_c K + (1-\alpha_c) I_N)$ where K is a kinship matrix and $\alpha_c \sim \mathcal{B}\text{eta}(1,1)$ is a component-specific mixing parameter (see section 3.4.2 for more details). This approach is referred to as T_K.

Data simulation and performance metrics: Data was simulated for 100 pairs of monozygotic twins, i.e. $N = 200$ individuals. A factor analysis model, $Y = AX + E$, was assumed for the data with $C = 6$ underlying component and $L = 500$ variables. The kinship matrix $K \in \mathbb{R}^{N \times N}$ was defined such that $K_{ij} = 1$ if $i = j$ or if individuals i and j were members of the same twin set, otherwise, $K = 0$. $X \in \mathbb{R}^{C \times L}$ was simulated to be sparse, with 10% of the elements in X drawn randomly from $\mathcal{N}(0,1)$, and the remaining elements set to zero. Noise was also drawn from a Gaussian distribution, $E \in \mathbb{R}^{N \times L} \sim \mathcal{N}(0, \lambda^{-1})$ for $\lambda \in \{0.3, 0.2, 0.1\}$. Individual scores vectors were simulated using a multivariate normal with a variety of different covariance matrices so that the data consisted of 2 heritable components, 2 partially heritable components and 2 components with no genetic basis. Table B.3 summarises the distributions used for the simulated individual scores vectors.

Both T_G and T_K were initialised with 6 components, and parameter settings from section 3.5 used. 50 data sets were simulated for each noise level, $\lambda \in \{0.3, 0.2, 0.1\}$. Permutations to match the true and estimated components were performed as in section 4.1.1.3. For each component type (heritable, partially heritable and not heritable), the number of recovered components was recorded, as in some cases, whole components were shrunk to zero. For

the components that were recovered, the correlations between the true and estimated individual scores vectors were used to compare the two methods.

Simulation distribution	Component type
$\mathcal{N}_N(0, K)$	Heritable
$\mathcal{N}_N(0, \frac{1}{2}K + \frac{1}{2}I_N)$	Partially heritable
$\mathcal{N}_N(0, I_N)$	Not heritable

Table B.3: Simulation distributions for individual scores vectors. Each data set contained 2 of each type of component.

Results: The fraction of each component type, (heritable, partially heritable and not heritable), recovered across 50 data sets with $\lambda = 0.1$ is shown in table B.4. The kinship-informed method T_K recovers more components, with a bias towards the heritable components. For noise levels, $\lambda = 0.3$ and 0.2, all components were recovered.

Figure B.3(A) shows the correlation between simulated and estimated individual scores vectors for the two methods. Each point on one of these plots is the correlation between an estimated individual scores vector and the truth in a single data set. Points are coloured according to component type. The results for the two models are very similar, indicating that the Gaussian prior is in fact very flexible. T_K, the model that uses the kinship matrix, does perform better however, especially for the components which are genetic (blue points).

Figure B.3(B) shows the posterior point estimates for the mixing parameters α_c for T_K. The method does well at estimating α_c near the true value (1 for the heritable components, $\frac{1}{2}$ for the partially heritable components and 0 otherwise). As the noise increases ($\lambda = 0.3$ to $\lambda = 0.1$), the estimates decay.

	Heritable	Partially heritable	Not heritable
T_G	0.70	0.80	0.71
T_K	0.91	0.85	0.72

Table B.4: Fraction of components recovered across 50 data sets. Results are grouped by component type. Only results for high noise levels ($\lambda = 0.1$) are shown as recovery was perfect for the other noise levels.

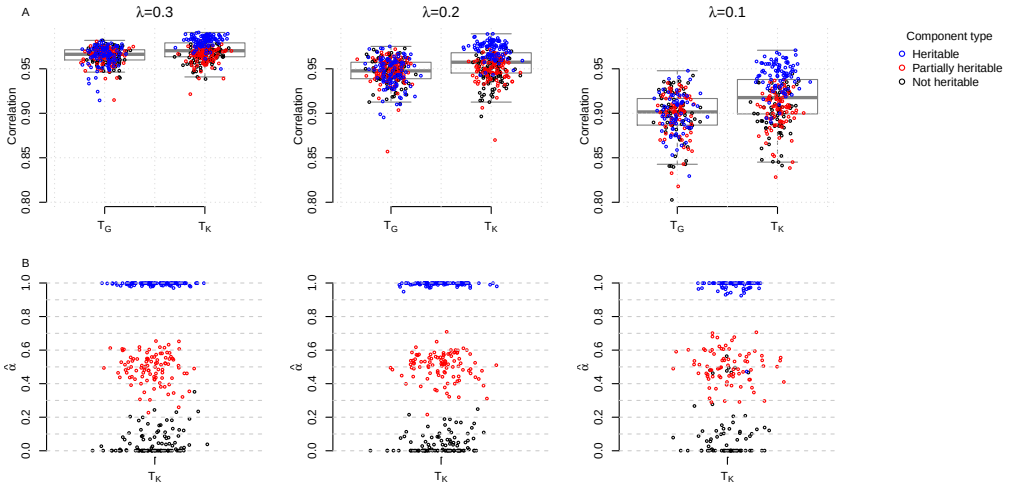

Figure B.3: (A) Correlations between estimated individual scores vectors (permuted) and the truth, coloured by component type. (B) Posterior means of mixing parameters for T_K, again coloured by component type.