Leveraging Genomic Annotations to Uncover Latent Components in Gene Expression Data

Christopher C. Gill
Wolfson College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Hilary Term 2021

Acknowledgements

I would like to thank my supervisor throughout this work, Simon Myers, for supervising both the main part of this work and the previous ancestry project, and for sharing his infectious enthusiasm and passion for science. I would also like to thank Jonathan Marchini, who provided me a route into genetics by supervising a rotation project and setting up and co-supervising the direct-to-consumer ancestry project which followed.

My time at the Department of Statistics would not have been the same without good company in the office and research groups. I am particularly grateful for the many stimulating discussions with Kevin Sharp, Lloyd Elliott, Sile Hu and Sinan Shi.

I thank the Doctoral Training Center for providing me the opportunity to retrain in a new field, the EPSRC for funding the majority of this time, and the Department of Statistics for providing an inviting friendly atmosphere to carry out research.

Lastly, I'm immensely grateful to my wonderful wife Sarka, for her relentless support and encouragement throughout these years.

Abstract

Over the last decade single cell genomics has transformed research in genetics, resolving transcriptional activity at an unprecedented resolution. Over 1000 single cell studies have been published to date and the scale of single cell data is increasing, cell atlases of complex organisms are being curated, and experimental sample sizes are increasing exponentially.

Standard analyses for single cell data include a hard clustering of cells and an investigation of these clusters to identify cell-specific marker genes. This approach is powerful and identifies transcriptomic differences between cells. However, such analyses fail to account for biological heterogeneity present in continuous developmental processes taking place in cells, or to use relevant prior information that may be available to the researcher. In this thesis we investigate incorporating prior information into factor analysis for gene expression analyses, which infers components of genes that are coexpressed or co-repressed across cells, and their expression in each cell. First, we consider structural features of the data, extending an existing tensor decomposition method to four dimensions, facilitating analysis of gene expression data indexed by cell, time, tissue and gene. We demonstrate the relative benefits of the method against a lower dimensional method. Secondly, we develop a widely applicable new prior structure and inference algorithm that allows for incorporating external prior information into factor analysis. The new algorithm automatically selects important annotations and leverages this information to infer biologically meaningful components. Comparing two likelihood models, we evaluate their strengths and weaknesses on simulated data, and the benefit of incorporating prior information into the algorithm. We demonstrate the power of the method to capture meaningful components of expression on a well-studied single cell RNA-seq dataset from cells undergoing spermatogenesis, using transcription factor binding affinities as prior information, and demonstrate the utility of the new prior structure in leveraging such annotations.

Contents

List of Figures

CHAPTER 1

Introduction

1.1 Main Aims

This thesis presents a new factor analysis method and prior structure for analysis of single cell RNA-seq (scRNA-seq) data, while incorporating prior knowledge through gene annotations. This can be viewed as either a first step in a dimension reduction of scRNA-seq data for clustering and downstream analysis, or as a method to determine components of genes that co-vary across cells explaining the variance in the dataset. The method developed here primarily aims to identify components corresponding to co-expressed or co-repressed genes, and the cells they are active in, while leveraging annotations to favour identification of components with interpretable properties. In this first chapter we give general background and context for the problem we are addressing here.

1.2 Background

Information contained in the genome is expressed initially through a process that generates functional molecules such as proteins or certain types of RNA. The first stage in this process is the transcription from DNA to messenger RNA (mRNA). Analysis of the transcriptome is thus a powerful strategy for dissecting the connection between genotype and phenotype of a cell [TLS11], and the measurement of mRNA present in cells is now possible through a number of technologies.

Previously measured in bulk, from tissue samples, since pioneering work in 2009, it has been possible to measure mRNA levels in individual cells, in a process called single cell RNA-seq (scRNA-seq). Since then single cell RNA-seq has become a mainstay of modern genomics, providing a method to interrogate transcriptome-wide dynamics of cells, within and between tissues and throughout developmental pathways. Technological advances have driven an exponential scaling in cell numbers [SVTT18] (see also [SdVBP20], and Figure 1.1) revolutionising single cell transcriptomics. From initially ([TBW$^+$09]) 5 individual murine germ cells, driven by interest in a population of rare cells, the power of single cell transcriptomics has come from technological developments, with the profiling of tens of thousands of cells in parallel being common by 2019. Certain recent datasets show this trend has continued, with some containing over 2 million cells [CSQ$^+$19]. The prevalence of the method in the last decade is evidenced by the over 1000 single cell RNA-seq transcriptomics studies published [SdVBP20].

Single cell RNA-seq is not a single method, but a collection of assays that vary in their strengths and limitations (see [SNL$^+$17] for a discussion). However, all scRNA-seq methods share some common steps. In particular, once transcribed RNA is isolated, it is reverse transcribed to complementary DNA (cDNA) before being amplified by molecular biology methods such as PCR (this combination is known as RT-PCR). The resulting DNA is sequenced allowing the quantification of expression of gene products.

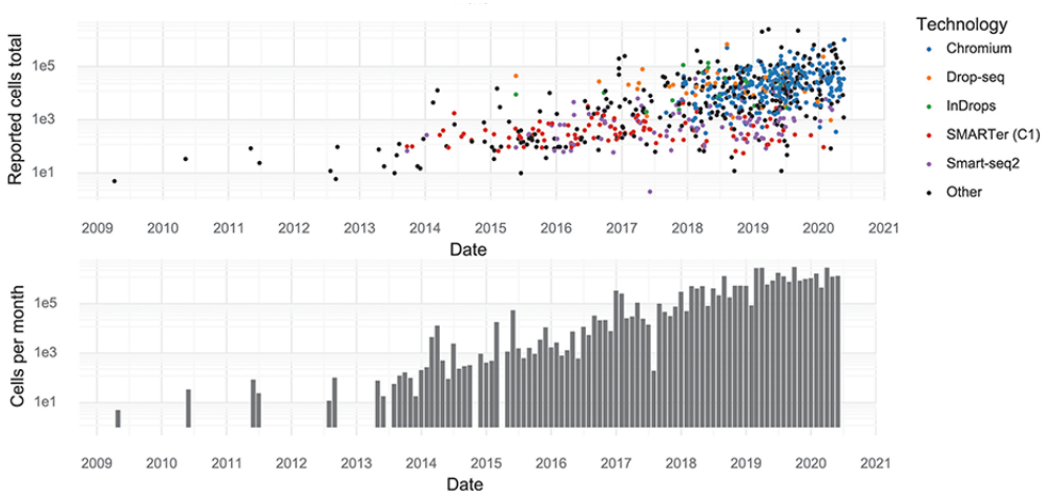

Figure 1.1: **Exponential Scaling of scRNA-seq experiments.** Figure 2 from [SdVBP20].

To enable parallelization over cells, transcripts from each cell are tagged with polyT primer attached to a unique cell barcode, and the majority of scRNA-seq assays now include an additional unique molecular identifier (UMI), enabling pooling of the starting material before reverse transcription and demultiplexing to be carried out later in silico post-amplification. This results in the so-called digital gene expression (DGE) matrix. This combination of UMI and cell barcode uniquely identifies each captured RNA molecule and which cell it originated from resulting in absolute transcript count. A further sample barcode is sometimes used to enable pooling of samples, further increasing the scale of experiments. The precise details vary between protocols but Drop-seq, for example, encapsulates each cell in a droplet containing a microparticle bead covered in UMI/barcode primers, the cells are lysed within each droplet, and the RNA attaches to the primers on the bead [MBS+15].

Cells, considered the fundamental unit of life, have traditionally been classified by morphology. However, the changes in cellular identity and function reflect programmes of transcriptional activity [SVTT18]. Since scRNA-seq provides a (noisy) snapshot of the RNA transcripts in a cell, the power to distinguish and classify cell types lies in the quality of the scRNA-seq assay and the number of cells that can be assayed. This

has led to a number of ambitious projects to classify all cell types in cell atlases of various organisms, notably the Tabula Muris (or Mouse Cell Atlas) [SKN+18] and the Human Cell Atlas [RTL+17, RRSRT17]. These will undoubtedly provide a valuable resource as a baseline of healthy cells in order to investigate the effects of ageing, disease, or response to therapeutic stimulus [AT20].

Methods in the field have progressed rapidly, and there are now standardised ways to process and analyse a single cell dataset, with many typical steps automated in software packages such as Seurat [BHS+18], Monocle [TCG+14]. Despite the standardisation, each analysis is unique and most steps vary from one dataset to another. We now describe a few of the typical steps in a single cell RNA-seq analysis.

The so-called digital gene expression (DGE) matrix Y is a matrix of counts, with entries y_{ji} representing the number of mapped reads of gene i from cell j. A typical first step is to perform quality control on cells and genes, filtering cells according to a number of criteria. Although these filters vary from study to study, typical filters will be based on the distributions of the following three covariates:

- the library size (or count depth), that is, the total UMI count in a cell;
- the proportion of mitochondrial genes in a cell;
- the number of genes detected.

The filtrations will typically aim to remove outliers. For example, if the library size and the number of detected genes are particularly large this could be a sign that the row of the matrix corresponds to a doublet, that is, more than one cell was captured in the droplet and tagged with the cell barcode. A high mitochondrial gene proportion could be indicative of cell death, or a cell under stress. Furthermore, [LT19] suggests that a cell with a low library size and low number of genes detected but a high proportion of mitochondrial genes is indicative of a cell whose membrane has broken leaving only RNA in the mitochondria present.

However, this stage of the processing already exhibits context specific choices: it is difficult to set a threshold for these covariates, or their combinations, that is robust to cell-specific processes/context. For example, cells with a high mitochondrial count may be involved in respiratory processes or large library size may be indicative of larger cells.

The DGE matrix Y is large with often over 20000 genes present, and sparse, and it is often reasonable to filter genes which are not expressed in more than a few cells, since the expression of these genes has been missed, possibly due to sampling, so these are not representative of the overall cellular heterogeneity in any quantifiable way.

As described by [LT19], a count in the DGE matrix represents the successful capture of a cell, tagging of the RNA molecule, reverse transcription, and mapping. Therefore, count profiles of identical cells will vary due to the inherent variability of these processes, and normalisation is the stage of the processing employed to remove the effect of this sampling, making cells comparable. A commonly employed method is to scale cells to a constant count depth by dividing by the library size and multiplying by a constant (often referred to as counts per million, or CPM, in the literature), and scaling genes to unit variance, occasionally centering as well to obtain z-scores. Typically a variance stabilising transformation will be applied at this stage. This transformation is primarily applied to account for the mean-variance relationship as many downstream analysis tools assume Gaussian distributed data. Recent work of [Sve20] and [SS20] has shown that, after accounting for the library size, single cell datasets are consistent with a Poisson count model, a property that we make use of in this work.

Following normalisation, one of the first steps in many analyses is to perform some sort of feature selection, selecting highly variable genes, and using these, or principal components derived from them, for dimension reduction and visualisation. The use of non-linear dimension reductions such as t-distributed stochastic neighbourhood

embedding (t-SNE) [vdMH08], and uniform manifold approximation mapping (UMAP) [MHSG18] are extremely common. While UMAP aims to approximate the true topology of the dataset, it is known that t-SNE focuses on capturing local distances between cells rather than global. That is, t-SNE aims to preserve local structure amongst cells, with a trade-off that long-range information is lost [AH18]. Though we have found this to be the case, there is evidence that with careful selection of the parameters defining the details of the algorithm, a t-SNE implementation can capture both local and global structure [KB19]. Both methods are widely used and have some stochastic nature, so researchers benefit most from comparing the projections from several initialisations, or random seeds. We note also that it can be advantageous that UMAP defines a projection through which other data points can be mapped for comparison with the existing dataset, whereas to incorporate out-of-sample data points in a t-SNE dimension reduction requires ad-hoc methods such as those used to visualise velocity vectors suggested in [LMSZ$^+$18].

With the goal of classification of cell types, a typical aim is to perform clustering on the cells, usually by way of clustering on a dimension reduction. This has become a standard analysis pipeline for single cell studies, as the first step to more in depth analyses. Mathematically, this approach requires an unsupervised clustering algorithm, a problem well studied in the machine learning field. Clustering approaches that have been widely used in applications to single cell RNA-seq include k-means, hierarchical clustering, or graph based clustering (also known as community-detection), usually using a k-nearest-neighbours graph derived from a dimension reduction embedding. Since dimension reductions are highly sensitive to batch effects and quality control problems, it is essential that QC steps are stringent and batch effects are removed prior to any dimension reduction or clustering. Moreover, each method for clustering makes assumptions about the clusters in the data, usually in the form of the intra-cluster and inter-cluster metrics used in the algorithm, and the results should be interpreted

in light of this. For example, k-means is fast, but requires a fixed number of clusters to be specified and is stochastic in nature so requires multiple restarts to determine the robustness of the clustering. It also has a bias towards similarly sized clusters and so can miss small clusters. Hierarchical clustering has also been tailored to single cell RNA-seq in a number of applications and published methods ([LTH17, GWP+15, ŽY16, BVW+16]). On the other hand, community detection algorithms on graphs are also well-studied, scale well to millions of samples [BGLL08], and can detect connected clusters (communities) of cells of varied size and structure. This has led in particular to the Louvain community detection algorithm [BGLL08, TWvE19] being included in the popular single cell analysis software Seurat. We note however, that a "hard" clustering of cells, as described in most of the methods above, does little to elucidate identification of cells on a developmental process, and may disguise biological processes that cross the imposed cell-type boundaries, such as transcriptional changes driving a continuous developmental trajectory.

The focus of this thesis is on fitting latent variable models to gene expression data, while incorporating external prior information that may be available to the researcher. That is, we aim to fit some latent, underlying structure, explaining patterns of variability present in the data, including some model for noise or confounding factors. Such models can be used both to extract useful features of the data for further dimension reduction, and to extract meaningful biological insights. Now a widely used approach in genomics [SOAC+18], one of the simplest latent variable models is factor analysis, modelling the variation in a J by I gene expression matrix Y as depending on a number, C, of vectors $x^{(c)}$, of length I such that each cell's gene expression data is modelled as a linear combination of the vectors $x^{(c)}$. In this way, each cell's expression data is modelled as a set of individual weightings, or scores, for each component,

$$y_{ji} = \sum_{c=1}^{C} a_{jc} x_{ci} + \epsilon_{ji}, \tag{1.1}$$

where a_{jc} is the cell score for cell j and component c, x_{ci} is the ith entry of $x^{(c)}$, and ϵ_{ji} is a Gaussian noise term. Thus, we describe this model as

$$Y = AX + \mathcal{N}(0, \Psi), \tag{1.2}$$

where Ψ is a diagonal covariance matrix. The rows of X (the vectors $x^{(c)}$) are the latent factors, or gene loadings vectors, and the columns of the matrix A are the cell weightings, or cell scores vectors. In applications, the matrix A can be used for clustering, pseudo-temporal ordering, or further dimension reduction and visualisation using t-SNE or UMAP. If the cell scores matrix were constrained to have each row a vector of zeros and a single one, then this would be a clustering model. Instead, the cell scores matrix can also be viewed as a 'soft clustering' that allows for continuous differentiation between cells, capturing the varying levels of expression of components of expression captured in the gene loadings matrix. From this viewpoint the cell scores matrix is often referred to as the pattern matrix [SOAC$^+$18]. The loadings matrix contains information pertaining to the sources of variation in the dataset across genes. Each component provides levels of expression of genes which are coexpressed across the dataset, and these can be further analysed for biological insights through, for example, enrichment analysis. For further discussion and references, see [SOAC$^+$18]. Unfortunately, even for two dimensional data sets, a basic factor analysis model lacks identifiability since any orthogonal change of basis preserves both the covariance and the decomposition. In practice, sparsity in the loadings matrix can facilitate rotational identifiability, and otherwise postprocessing techniques are used to identify rotations of the identified factors ([HL94]).

Beyond matrix factorisation, datasets for gene expression in more dimensions are becoming available, either multi-omics datasets, or gene expression datasets measured in multiple tissues for each sample. When considering a dataset of higher dimensions

one can view it as a series of two dimensional datasets, for example as in Bayesian group factor analysis as described by [VKKK12] to model different data types or contexts while sharing a common sample (cell) scores matrix. However, the results may well reflect the choice of the two-dimensional slicing, each slice could be susceptible to the rotational identifiability issues mentioned above, and it introduces a large number of latent variables. An alternative approach that avoids this pitfall is to extend the factor analysis model to higher dimensions more directly. It can be shown that all tensors are sums of rank one tensors, and it is precisely such a decomposition which we focus on in one of the methods we develop here. The concept of viewing a tensor as a sum of a finite number of rank one tensors appears in the literature in 1927, but became popular in 1970 when introduced as canonical decomposition (CANDECOMP) by [CC70], and Parallel Factor analysis (PARAFAC) by Harshman (see [HL94]). Research into methods aimed at decomposing data tensors to extract underlying explanatory structure has now been ongoing for at least four decades ([KB09]).

One method we build on in this thesis is SDA [HVB+16], which implemented a sparse Bayesian parallel factor analysis model for three-dimensional datasets. We describe this here using the notation and indexing of [HVB+16], developed originally for bulk RNA-seq data, with samples representing individuals. Consider a three-dimensional data tensor $\mathcal{Y} \in \mathbb{R}^{N \times L \times T}$, with measurements for N individuals, T tissues (or contexts, cell-types, time points etc.) and L genes. A PARAFAC decomposition of $\mathcal{Y}$ is described in equation 1.3 and illustrated in Figure 2.1,

$$y_{nlt} = \sum_{c=1}^{C} a_{nc} b_{tc} x_{cl} + \epsilon_{nlt}, \tag{1.3}$$

where C is the number of components, $A \in \mathbb{R}^{N \times C}, B \in \mathbb{R}^{T \times C}, X \in \mathbb{R}^{C \times L}$, and ϵ_{nlt} is

a noise term. This is typically summarised as a sum of rank one tensors, as

$$\mathcal{Y} = \sum_{c=1}^{C} a^{(c)} \circ b^{(c)} \circ x^{(c)} + \epsilon, \tag{1.4}$$

where $\circ$ is an outer tensor product, and $a^{(c)}, b^{(c)}$ are c^{th} columns of A, B respectively and $x^{(c)}$ is the c^{th} row of X. A single component comprises all three of these vectors.

Another approach to three-dimensional tensor decomposition is the Tucker decomposition which includes an extra core tensor $P \in \mathbb{R}^{C \times C \times C}$ where C is the number of components. This approach describes the tensor $\mathcal{Y}$ as a linear combination of tensors of the form $a^{(i)} \circ b^{(j)} \circ x^{(k)}$, with coefficients given by the entries p_{ijk} of P. The Tucker decomposition is more general than a PARAFAC decomposition (recovered by setting $p_{ijk} = 0$ whenever i, j, k are not all equal, and 1 otherwise), but we note that the extra tensor P could introduce many variables that are not required in the PARAFAC decomposition, and may be less easily interpretable in a gene expression context.

A number of different models exist for (parallel, group, or standard) factor analysis of gene expression data, either by applying constraints to the decompositions, or by applying different prior structures to the matrices. For example, in a 2-dimensional setting, by constraining the entries of A and X to be non-negative, one obtains non-negative matrix factorisation (NNMF), which is popular in gene expression analyses, likely because the data is non-negative. More commonly, with the aim of interpreting the components as representing biological processes and the belief that among all genes the number involved in any process will be proportionally low, sparsity is imposed on the loadings matrix. A number of methods exist to facilitate this, often incorporating the spike and slab prior structure. First introduced in [MB88] in the context of variable selection in linear regression, spike and slab priors are widely considered the gold-standard of Bayesian sparsity inducing priors in the literature.

A spike and slab prior structure on x can be described as

$$\mathbb{P}(x) = (1-p)\delta_0(x) + p\mathcal{N}(x|0, \beta^{-1}),$$

for some mixing weight p and precision parameter β. That is, a mixture of a point mass at 0, the *spike* and a more diffuse Gaussian distribution, referred to as the *slab*. This effectively places extra prior density at 0 and inference of small p places most of this prior density at 0, while a large p places the parameter x in the diffuse slab.

There are essentially two versions of this spike and slab in the gene expression factor analysis literature. The group factor analysis method MOFA [AVA+18] utilises a component-wide sparsity parameter p_c, and a component-wide scale in the prior β_c, resulting in a prior as follows:

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

placing conjugate uninformative priors on p_c and β_c. We refer to this prior as the Mitchell and Beauchamp Spike and Slab Prior (MBSS). We note that MOFA is designed as a multi-omics integration model based around group factor analysis but allowing for different data modalities in each view of the data, incorporating Poisson, Bernoulli, and Gaussian likelihood models.

In contrast to this component-wide sparsity, the SDA model implementing parallel, group, and standard factor analysis, follows the approach of Lucas [LCW+06], with a gene-component-wise sparsity inducing prior, placing a Beta-Bernoulli mixture version of the spike and slab prior on p, as follows:

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

where

$$\mathbb{P}(p_{cl}) = (1 - \rho_c)\delta_0(p_{cl}) + \rho_c \mathcal{B}(p_{cl}|e, f)$$

for hyper parameters e, f chosen to reinforce sparsity, and the prior on the component-wise sparsity parameters ρ_c is a beta distribution. We refer to this prior structure as the Lucas Spike and Slab prior (LSS).

Marginalizing over the p_{cl} shows that the LSS prior is identical to setting $p_c = \rho_c \frac{e}{e+f}$ in the MBSS prior (this is also noted in [LCW$^+$06]). However, both [LCW$^+$06] and [Hor15] report that LSS provides an improved false discovery rate in practice in certain scenarios. This may be due to the approximation inherent in the variational inference used in [Hor15], including the imposed posterior independence of certain parameters in the variational inference algorithms, details of specific implementations, or due to the imposed upper bound dictated by choices of e and f on the component-wide sparsity.

The main focus of this thesis is incorporating extra information in latent variable models for gene expression data, with a focus on single cell RNA-seq data in particular. Although SDA was designed for bulk RNA-seq data, the model is very general and still valid for well normalised scRNA-seq data, and can accommodate samples from cells and genes across many contexts (tissues in the SDA model). With the diminishing cost and technological advances driving the field, it is natural to assume that time series data will be available, and we have extended the SDA model into four dimensions, incorporating a time scores matrix. We have investigated the benefits of the four dimensional model, demonstrating that it reliably captures the four-dimensional structure of simulated data, and outperforms the 3D model on a single time slice. However, we find that if the time and context dimensions in the time series data are unfolded into a single context dimension, then the 3D model is sufficient to capture the sample scores and gene loadings as accurately as the 4D model for all but the most sparse datasets. This work is presented in Chapter 2.

Although sparsity inducing prior structures go some way to mitigating the rotational unidentifiability of factor analysis [GBE13], the interpretation of components through their gene loadings requires great care and a significant amount of postprocessing, for example through enrichment analysis. We investigate here an approach to integrating data from other sources into a factor analysis inference algorithm and model. In particular, although there are still many unanswered questions regarding gene expression such as the determination of protein structure from sequence, and how gene expression levels relate to the interaction of genotype and tissue/cell context, much experimental information regarding gene function and transcriptional dynamics has been collected. For example, it is known that certain proteins act as transcription factors, binding to DNA and effectively promoting transcription of certain genes and silencing, or repressing others. Many such transcription factors have been identified and their binding motifs inferred, which is a valuable resource for determining binding affinities near enhancer or promoter regions of genes. This information should ideally inform our analyses of expression data but is currently difficult to incorporate. In this thesis we present a new factor analysis model that integrates prior information in the form of annotation vectors, with a score for each gene in the dataset, into the spike probabilities in a spike and slab factor analysis model. We model the log-odds of the inclusion probability as a linear combination of the annotation vectors and infer which annotations each component depends on as part of the algorithm with immediate biological interpretability.

The only other model in the literature capable of incorporating prior annotations that we are aware of is f-scLVM [BPM$^+$17], designed for use with pathway annotations such as those available from the REACTOME [JTGV$^+$05] database. The f-scLVM model also uses a spike and slab sparsity inducing prior structure, incorporating a component for each annotation, with spike inclusion probability approximately determined by the gene inclusion in the corresponding pathway subject to some

specified false negative rate and false positive rate. While this is a useful model, we note that restricting each component to be informed by exactly one annotation is limiting; in the event that more than one annotation informs inclusion in a single underlying component of the data this model will need to infer multiple correlated components, each explaining less variance than a single component that would depend on combinations of annotations. Moreover, the annotations incorporated are binary vectors, devoid of any level of uncertainty as is available for annotations such as transcription factor binding affinities.

The model and inference algorithm, presented in Chapter 3, defines a new prior structure for data integration in factor analysis, and has numerous other scenarios it could be applied to. Due to the prevalence of Gaussian models currently in use [LT19], and the recent results suggesting that (conditional on library size), scRNA-seq data is Poisson distributed, we have implemented both a Poisson model which incorporates library size, and a Gaussian model, and compare and contrast on both Gaussian and count data in a simulation study. We focus our application of the model on a rich scRNA-seq dataset that has been previously studied [JWR$^+$19]. This large dataset is from a Drop-seq protocol, involving approximately 20000 cells from testis from several mouse lines, including mutant mice. Since several of the mutant mouse strains are chosen specifically such that their cells arrest at certain stages of spermatogenesis, this dataset is enriched for early meiotic cells. We identify many components identifying known biological function and which capture a continuous developmental trajectory across cells when cell scores are ordered chronologically. Moreover, we demonstrate that a post-meiotic switch in gene expression which occurs during spermatogenesis is consistent with signatures of transcriptional regulation inferred as part of the algorithm.

CHAPTER 2

Sparse Bayesian Four Dimensional Tensor Decomposition for Gene Expression Data

Disease etiology may be better understood through the study of gene expression in four dimensional (4D) experiments that consist of measurements on multiple individuals, genes, tissues and under multiple conditions or through time. For example, a single cell dataset [CLT+20] has recently been studied in connection with COVID-19 severity with multiple individuals, with gene expression measurements in several tissues and several time points after onset of symptoms. We note however that such datasets have only appeared after the completion of this part of the work presented in this thesis. Motivated by the possibility of such datasets, we developed a sparse Bayesian four dimensional tensor decomposition algorithm. It is possible to restrict our four-dimensional implementation to recover the three-dimensional model from [HVB+16]. In this chapter we describe the model, inference algorithm and a simulation study, based on that from [HVB+16], in which we aim to determine the extent to which the

fourth dimension in the model aids analyses. We illustrate the utility of the method using simulated datasets, and show that when 4D data is available our method shows improved performance when compared to using a 3D method on a single slice of the 4D dataset. We also compare the results of the 4D method to that of the 3D method on a suitable unfolding of the data, demonstrating that the 3D and 4D methods perform similarly on their common dimensions, while the 4D method accurately recovers the additional structure in the data. Software implementing our approach is available at `https://github.com/marchinilab/SDA4D`

2.1 The Model

The model is an extension of parallel factor analysis CANDECOMP/PARAFAC (CP) to four dimensions with a sparse loadings matrix. This is a generalisation of the SDA model used in [Hor15] and [HVB+16] to four dimensions. Specifically, for a tensor $\mathcal{Y} \in \mathbb{R}^{N \times L \times M \times T}$, we consider the decomposition

$$y_{nlmt} = \sum_{c=1}^{C} a_{nc} b_{tc} d_{mc} x_{cl} + \epsilon_{nlmt}, \tag{2.1}$$

where C is the number of components, $A \in \mathbb{R}^{N \times C}, B \in \mathbb{R}^{T \times C}, D \in \mathbb{R}^{M \times C}$, $X \in \mathbb{R}^{C \times L}$ and ϵ_{nlmt} is Gaussian noise. Figure 2.1 is an illustration of the model. The motivation is to consider a gene expression dataset of N individuals, T tissues, M time points, across L genes and we model the noise as having variance constant across individuals and time, with $\mathbb{P}(\epsilon_{nlmt}) = \mathcal{N}(\epsilon_{nlmt}|0, \lambda_{lt}^{-1})$, where λ_{lt} is the precision, indexed by the tissue and gene.

Parallel factor tensor decompositions are invariant under permutation of the components. That is, if P is a permutation matrix and A, B, D, X are a PARAFAC decomposition of a tensor $\mathcal{Y}$, then AP, BP, DP, PX are also a PARAFAC decomposition of $\mathcal{Y}$. Furthermore, sign and scale are non-identifiable in a PARAFAC

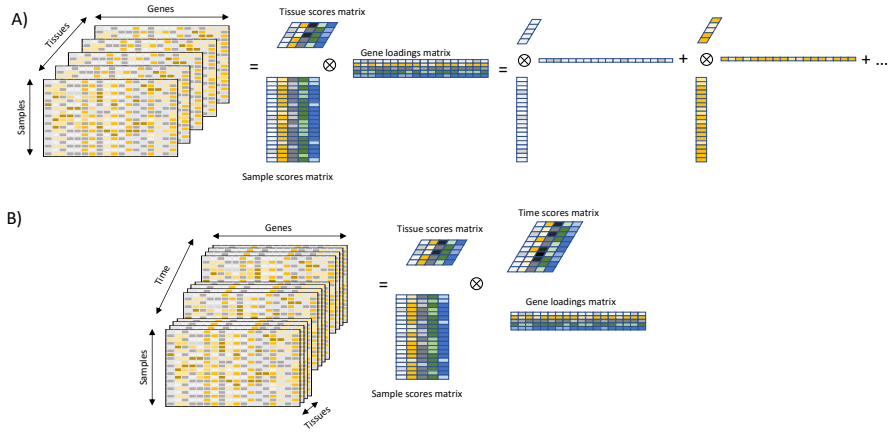

Figure 2.1: A) Diagram of a PARAFAC decomposition of a three dimensional tensor. B) Diagram of the four-dimensional PARAFAC model.

decomposition. We take into account these limitations when evaluating the results of the method in simulations. We follow the approach of [Hor15] and with these limitations in mind, we use standard normal distributions for priors of a_{nc}, b_{tc}, d_{mc}. We place a conjugate prior on λ_{lt}, a Gamma distribution with parameters u, v.

Amongst a full spectrum of genes in any dataset, the number involved in any particular biological process will be small, so we aim to impose sparsity on the gene loadings vectors, using a 'spike-and-slab' prior, with a large density at zero, and remaining density spread as a normal distribution with mean zero and some precision. Assuming uniform precision across the component, the prior on x_{cl} is defined as follows:

$$\mathbb{P}(x_{cl}) = p_{cl}\mathcal{N}(x_{cl}|0, \beta_c^{-1}) + (1 - p_{cl})\delta_0(x_{cl}), \tag{2.2}$$

where β_c is the precision of the 'slab', and p_{cl} is a weight indicating whether the gene is active in that component. Following [TLG11] we facilitate inference with such a distribution by expressing x_{cl} as a product of a Gaussian distributed random variable w_{cl} and a Bernoulli distributed random variable s_{cl}. That is, we write $x_{cl} = w_{cl}s_{cl}$

where

$$\mathbb{P}(w_{cl}) = \mathcal{N}(w_{cl}|0, \beta_c^{-1}),$$
$$\mathbb{P}(s_{cl}) = \mathcal{B}\text{ernoulli}(s_{cl}|p_{cl}).$$

We apply a conjugate prior to β_c, a Gamma distribution with parameters e, f. The mixing weight p_{cl} is also given a spike and slab prior, this time involving a Beta distribution:

$$\mathbb{P}(p_{cl}) = \rho_c \mathcal{B}(p_{cl}|g, h) + (1 - \rho_c)\delta_0(p_{cl}). \tag{2.3}$$

Such a hierarchical structure on the 'spike' was first introduced in [LCW+06], demonstrating that it reduced the false discovery rate in certain scenarios. Since p_{cl} is the likelihood that gene l is 'on' in component c (that is, $x_{cl} \neq 0$), the value of ρ_c determines the sparsity of the component. If ρ_c takes a value close to 1 then p_{cl} will take values from the Beta distribution, resulting in a dense component, whereas, if ρ_c is close to zero then p_{cl} has density highest at 0 and will result in a sparse component. We typically initialise with g, h close to zero resulting in a Beta distribution with peaks at zero and one, which further imposes sparsity. To make inference more straightforward, we write $p_{cl} = \phi_{cl}\psi_{cl}$, where

$$\mathbb{P}(\phi_{cl}|\rho_c) = \mathcal{B}\text{ernoulli}(\phi_{cl}|\rho_c);$$
$$\mathbb{P}(\psi_{cl}) = \mathcal{B}(\psi_{cl}|g, h);$$

The prior for ρ_c is a beta distribution with parameters r, z.

By setting $M = 1$, fixing $\langle d_{mc} \rangle = 1$ for all c, and neglecting to update D, we regain a three-dimensional model that is essentially equivalent to the method detailed in [HVB+16], though we note that since our method was implemented independently exact details may differ.

We summarise the model as follows for a data tensor $\mathcal{Y} = (y_{nlmt})_{n,l,m,t}$:

The Likelihood

$$\mathbb{P}(\mathcal{Y}|\Theta) = \prod_{n,l,m,t} \mathcal{N}\left(y_{nlmt}|\sum_{c=1}^{C} a_{nc}b_{tc}d_{mc}w_{cl}s_{cl}, \lambda_{lt}^{-1}\right).$$

Prior distributions

$$\begin{aligned}
\mathbb{P}(\lambda_{lt}) &= \mathcal{G}(\lambda_{lt}|u, v);\\
\mathbb{P}(a_{nc}) &= \mathcal{N}(a_{nc}|0, 1);\\
\mathbb{P}(b_{tc}) &= \mathcal{N}(b_{tc}|0, 1);\\
\mathbb{P}(d_{mc}) &= \mathcal{N}(d_{mc}|0, 1);\\
\mathbb{P}(w_{cl}|\beta_c) &= \mathcal{N}(w_{cl}|0, \beta_c^{-1});\\
\mathbb{P}(s_{cl}|\phi_{cl}, \psi_{cl}) &= \mathcal{B}\text{ernoulli}(s_{cl}|\psi_{cl}\phi_{cl});\\
\mathbb{P}(\beta_c) &= \mathcal{G}(\beta_c|e, f);\\
\mathbb{P}(\phi_{cl}|\rho_c) &= \mathcal{B}\text{ernoulli}(\phi_{cl}|\rho_c);\\
\mathbb{P}(\psi_{cl}) &= \mathcal{B}(\psi_{cl}|g, h);\\
\mathbb{P}(\rho_c) &= \mathcal{B}(\rho_c|r, z).
\end{aligned}$$

2.1.1 Model fitting

Due to the complex nature of this model, exact inference is not possible, so we have implemented a variational Bayes (VB), approximate inference, approach. We defer details and references on variational inference to Appendix A.1.2.

We follow the approach of [TLG11] for inference with a spike and slab prior, and we find it beneficial to couple w_{cl}, and s_{cl} in the parameter partition as they together parametrize the spike and slab distribution. We partition the parameters as follows:

$$\{a_{nc}\}, \{b_{tc}\}, \{d_{mc}\}, \{\phi_{cl}\}, \{\psi_{cl}\}, \{\rho_c\}, \{\beta_c\}, \{\lambda_{lt}\}, \{w_{cl}, s_{cl}\}.$$

Deriving the updates as described in Section A.1.2, this results in the following VB updates, where we denote $\mathbb{E}_Q(\cdot)$ by $\langle\cdot\rangle$:

Individual scores matrix:

$$\begin{aligned}
Q^*(a_{nc}) &= \mathcal{N}(a_{nc}|\mu^*_{nc}, \omega^{*-1}_{nc}) \\
\omega^*_{nc} &= 1 + \sum_{l,m,t} \langle\lambda_{lt}\rangle\langle b^2_{tc}\rangle\langle d^2_{mc}\rangle\langle w^2_{cl}s^2_{cl}\rangle, \\
\mu^*_{nc} &= \frac{1}{\omega^*_{nc}} \left(\sum_{l,m,t} \langle\lambda_{lt}\rangle\langle b_{tc}\rangle\langle d_{mc}\rangle\langle w_{cl}s_{cl}\rangle \left(y_{nlmt} - \sum_{k\neq c} \langle a_{nk}\rangle\langle b_{tk}\rangle\langle d_{mk}\rangle\langle w_{kl}s_{kl}\rangle \right) \right).
\end{aligned}$$

Tissue scores matrix:

$$\begin{aligned}
Q^*(b_{tc}) &= \mathcal{N}(b_{tc}|\nu^*_{tc}, \tau^{*-1}_{tc}), \\
\tau^*_{tc} &= 1 + \sum_{l,m,n} \langle\lambda_{lt}\rangle\langle a^2_{nc}\rangle\langle d^2_{mc}\rangle\langle w^2_{cl}s^2_{cl}\rangle, \\
\nu^*_{tc} &= \frac{1}{\tau^*_{tc}} \left(\sum_{n,l,m} \langle\lambda_{lt}\rangle\langle a_{nc}\rangle\langle d_{mc}\rangle\langle w_{cl}s_{cl}\rangle \left(y_{nlmt} - \sum_{k\neq c} \langle a_{nk}\rangle\langle b_{tk}\rangle\langle d_{mk}\rangle\langle w_{kl}s_{kl}\rangle \right) \right).
\end{aligned}$$

Time scores matrix:

$$
\begin{aligned}
Q^*(d_{mc}) &= \mathcal{N}(d_{mc}|\alpha^*_{mc}, \beta^{*-1}_{mc}), \\
\beta^*_{mc} &= 1 + \sum_{n,l,t} \langle \lambda_{lt} \rangle \langle a^2_{nc} \rangle \langle b^2_{tc} \rangle \langle w^2_{cl} s^2_{cl} \rangle, \\
\alpha^*_{mc} &= \frac{1}{\beta^*_{mc}} \left(\sum_{n,l,t} \langle \lambda_{lt} \rangle \langle a_{nc} \rangle \langle b_{tc} \rangle \langle w_{cl} s_{cl} \rangle \left(y_{nlmt} - \sum_{k \neq c} \langle a_{nk} \rangle \langle b_{tk} \rangle \langle d_{mk} \rangle \langle w_{kl} s_{kl} \rangle \right) \right).
\end{aligned}
$$

Noise precision:

$$
\begin{aligned}
Q^*(\lambda_{lt}) &= \mathcal{G}(\lambda_{lt}|u^*_{lt}, v^*_{lt}) \\
u^*_{lt} &= u + \frac{NM}{2}, \\
v^*_{lt} &= \left(\frac{1}{v} + \frac{1}{2} \sum_{n,m} \left\langle \left(y_{nlmt} - \sum_{c=1}^{C} a_{nc} b_{tc} d_{mc} w_{cl} s_{cl} \right)^2 \right\rangle \right)^{-1}.
\end{aligned}
$$

Gene loadings matrix:

$$
\begin{aligned}
Q^*(\beta_c) &= \mathcal{G}(\beta_c|e_c^*, f_c^*) \\
e_c^* &= e + \frac{L}{2}, \\
f_c^* &= \left(\frac{1}{f} + \frac{1}{2}\sum_l \langle w_{cl}^2\rangle\right)^{-1} \\
Q^*(w_{cl}|s_{cl}) &= \mathcal{N}\left(w_{cl}|s_{cl}m_{cl}^*, (s_{cl}\sigma_{cl}^* + (1-s_{cl})\langle\beta_c\rangle)^{-1}\right) \\
\sigma_{cl}^* &= \langle\beta_c\rangle + \sum_{m,n,t}\langle\lambda_{lt}\rangle\langle a_{nc}^2\rangle\langle b_{tc}^2\rangle\langle d_{mc}^2\rangle \\
m_{cl}^* &= \frac{1}{\sigma_{cl}^*}\left(\sum_{m,n,t}\langle\lambda_{lt}\rangle\langle a_{nc}\rangle\langle b_{tc}\rangle\langle d_{mc}\rangle\left(y_{nlmt} - \sum_{k\neq c}\langle a_{nk}\rangle\langle b_{tk}\rangle\langle d_{mk}\rangle\langle w_{kl}s_{kl}\rangle\right)\right) \\
Q^*(s_{cl}) &= \mathcal{B}\text{ernoulli}\left(s_{cl}|\gamma_{cl}^*\right) \\
\gamma_{cl}^* &= \frac{1}{1+e^{-u_{cl}^*}} \\
u_{cl}^* &= -\frac{1}{2}\log\sigma_{cl}^* + \frac{1}{2}(m_{cl}^*)^2\sigma_{cl}^* + \log(\phi_{cl}^*\psi_{cl}^*) + \frac{1}{2}\log\langle\beta_c\rangle - \log(1-\phi_{cl}^*\psi_{cl}^*)
\end{aligned}
$$

Following the same procedure for the variables $\rho_c, \psi_{cl}, \phi_{cl}$ does not result in a closed form distribution. In this case we maximise the ELBO, denoted $F(Q)$, with respect to point estimates of these parameters. The corresponding component of the ELBO dependent on these parameters is $\tilde{F}(Q)$, defined as follows:

$$
\begin{aligned}
\tilde{F}(Q) &= \mathbb{E}_Q\left(\sum_c \log\mathbb{P}(\rho_c) + \sum_{c,l}\left(\log\mathbb{P}(s_{cl}|\psi_{cl},\phi_{cl}) + \log\mathbb{P}(\psi_{cl}) + \log\mathbb{P}(\phi_{cl}|\rho_c)\right)\right) \\
&= \sum_c\left((r-1)\log(\rho_c^*) + (z-1)\log(1-\rho_c^*)\right) \\
&\quad + \sum_{c,l}\left(\langle s_{cl}\rangle\log(\phi_{cl}^*\psi_{cl}^*) + (1-\langle s_{cl}\rangle)\log(1-\phi_{cl}^*\psi_{cl}^*)\right) \\
&\quad + \sum_{c,l}\left((g-1)\log(\psi_{cl}^*) + (h-1)\log(1-\psi_{cl}^*)\right) \\
&\quad + \sum_{c,l}\left(\phi_{cl}^*\log\rho_c^* + (1-\phi_{cl}^*)\log(1-\rho_c^*)\right).
\end{aligned}
$$

There is a unique solution to $\frac{\partial \tilde{F}}{\partial \rho_c} = 0$, which gives the update for ρ_c as

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

For the remaining parameters ϕ_{cl}, ψ_{cl} we optimise $F(Q)$ by maximising $\tilde{F}(Q)$ using Newton's method in two dimensions. The gradient vector is

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

The Hessian matrix, H, is calculated as

$$\begin{pmatrix} \frac{\partial^2 \tilde{F}}{\partial \phi_{cl}^2} & \frac{\partial^2 \tilde{F}}{\partial \phi_{cl} \partial \psi_{cl}} \\ \frac{\partial^2 \tilde{F}}{\partial \phi_{cl} \partial \psi_{cl}} & \frac{\partial^2 \tilde{F}}{\partial \psi_{cl}^2} \end{pmatrix} = \begin{pmatrix} -\frac{\langle s_{cl} \rangle}{\phi_{cl}^2} - \frac{\psi_{cl}^2(1-\langle s_{cl} \rangle)}{(1-\phi_{cl}^*\psi_{cl}^*)^2} & -\frac{(1-\langle s_{cl} \rangle)}{(1-\phi_{cl}\psi_{cl})^2} \\ -\frac{(1-\langle s_{cl} \rangle)}{(1-\phi_{cl}\psi_{cl})^2} & -\frac{\langle s_{cl} \rangle}{\psi_{cl}^2} - \frac{\phi_{cl}^2(1-\langle s_{cl} \rangle)}{(1-\phi_{cl}^*\psi_{cl}^*)^2} - \frac{(g-1)}{\psi_{cl}^2} - \frac{(h-1)}{(1-\psi_{cl})^2} \end{pmatrix}$$

The update for (ϕ_{cl}, ψ_{cl}) is then

$$\begin{pmatrix} \phi_{cl}^{i+1} \\ \psi_{cl}^{i+1} \end{pmatrix} = \begin{pmatrix} \phi_{cl}^{i} \\ \psi_{cl}^{i} \end{pmatrix} - \alpha (H^i)^{-1} \nabla \tilde{F}^i,$$

where α is chosen by a backtracking line search to ensure the step increases $\tilde{F}(Q)$.

Each of these updates is guaranteed to increase the ELBO,

$$F(Q) = \mathbb{E}_Q \left(\log \left(\frac{\mathbb{P}(\mathcal{Y}|\theta)\mathbb{P}(\theta)}{Q(\theta)} \right) \right).$$

The ELBO (or negative free energy) for this model is as follows:

$$
\begin{aligned}
F(Q) = \text{constant } &+ \frac{1}{2}\sum_{l,m,n,t}\left(\langle\log(\lambda_{lt})\rangle - \langle\lambda_{lt}\rangle\left\langle\left(y_{nlmt} - \sum_c a_{nc}b_{tc}d_{mc}w_{cl}s_{cl}\right)^2\right\rangle\right) \\
&- \frac{1}{2}\sum_{n,c}\langle a_{nc}^2\rangle - \frac{1}{2}\sum_{n,c}\log|\omega_{nc}^*| \\
&- \frac{1}{2}\sum_{t,c}\langle b_{tc}^2\rangle - \frac{1}{2}\sum_{t,c}\log|\tau_{tc}^*| \\
&- \frac{1}{2}\sum_{m,c}\langle d_{mc}^2\rangle - \frac{1}{2}\sum_{m,c}\log|\beta_{mc}^*| \\
&+ \frac{L}{2}\sum_c\langle\log\beta_c\rangle - \frac{1}{2}\sum_{c,l}\langle\beta_c\rangle\langle w_{cl}^2\rangle \\
&- \frac{1}{2}\sum_{c,l}\langle s_{cl}\rangle\log(\sigma_{cl}^*) - \frac{1}{2}\sum_{c,l}(1-\langle s_{cl}\rangle)\log(\langle\beta_c\rangle) \\
&+ \sum_c\left((r-1)\log(\rho_c^*) + (z-1)\log(1-\rho_c^*)\right) \\
&+ \sum_{c,l}\left((g-1)\log(\psi_{cl}^*) + (h-1)\log(1-\psi_{cl}^*)\right) \\
&+ \sum_{c,l}\left(\phi_{cl}^*\log(\rho_c^*) + (1-\phi_{cl}^*)\log(1-\rho_c^*)\right) \\
&+ \sum_{c,l}\left(\langle s_{cl}\rangle\log(\phi_{cl}^*\psi_{cl}^*) + (1-\langle s_{cl}\rangle)\log(1-\phi_{cl}^*\psi_{cl}^*)\right) \\
&-(1-\langle s_{cl}\rangle)\log(1-\langle s_{cl}\rangle) - \langle s_{cl}\rangle\log\langle s_{cl}\rangle) \\
&+ \sum_c\left(e\log f_c^* + (e-e_c^*)\psi(e_c^*) + e_c^* - \frac{1}{f}e_c^*f_c^* + \log\Gamma(e_c^*)\right) \\
&+ \sum_{l,t}\left(u\log v_{lt}^* + (u-u_{lt}^*)\psi(u_{lt}^*) + u_{lt}^* - \frac{1}{v}u_{lt}^*v_{lt}^* + \log\Gamma(u_{lt}^*)\right)
\end{aligned}
$$

Here ψ is the digamma function.

2.2 A Simulation Study

We carried out a simulation study to evaluate the benefits of the 4D model, applied to 4D data, versus the 3D model applied to simulated data that replicates the simulations from [HVB$^+$16]. These 3D datasets are effectively a single time point of a 4D dataset. We also compare the results from the 4D algorithm to the results of the 3D model applied to a 3D unfolding of the 4D dataset.

We simulate 3D data for $N = 200$ individuals, across $L = 500$ genes, and $T = 3$ tissues with $C = 8$ underlying components. The data is generated by specifying matrices A, B, and X and adding noise. We generate an Individual Scores matrix A by sampling each entry from $N(0, 1)$. B is the tissues matrix, and we give this a strong signal that differs strongly over tissues. Specifically, we simulate 8 components, across three tissues such that three components are active in just one tissue, three are active in two tissues and two are active in three tissues. The non-zero entries of B are sampled from $\{-1, 1\}$. The gene loadings matrix X is simulated by specifying a level of sparsity, chosen by fixing the occupancy p . Each entry of X is non-zero with probability p, in which case it is sampled from $N(0, 1)$. We consider occupancy and sparsity to be equivalent, defining sparsity as $1 - p$. Finally, each data point is generated as

$$y_{nlmt} = \sum_{c=1}^{C} a_{nc} b_{tc} d_{mc} x_{cl} + \epsilon_{nlmt}, \tag{2.4}$$

where ϵ_{nlmt} is sampled from $N(0, 10)$.

The 4D data was simulated in a similar manner, for $N = 200$ individuals across $L = 500$ genes, in $T = 3$ tissues, and across $M = 16$ time points. We simulated an underlying set of $C = 8$ components. A, B and X were generated as above, and we

generated D deterministically, according to the following formulae (for $m = 1, \ldots, 16$)

$$d_{m1} = \sin\left(\frac{m\pi}{3}\right), \quad d_{m2} = \sin\left(\frac{m\pi}{4}\right), \quad d_{m3} = \sin\left(\frac{m\pi}{8}\right), \quad d_{m4} = \sin\left(\frac{m\pi}{11}\right),$$
$$d_{m5} = \cos\left(\frac{m\pi}{3}\right), \quad d_{m6} = \cos\left(\frac{m\pi}{4}\right), \quad d_{m7} = \cos\left(\frac{m\pi}{8}\right), \quad d_{m8} = \cos\left(\frac{m\pi}{11}\right).$$

For each four dimensional tensor $Y^{(1)}$ of dimension $N \times L \times M \times T$ we also unfold this into a three dimensional tensor $Y^{(2)}$ of dimension $N \times L \times MT$. We consider the individuals and genes to be fixed, and the tissue and time dimensions to capture context, which is how one might interpret the so-called tissue scores matrix in the three dimensional tensor. This results in an unfolding as $Y^{(2)}_{n,l,m+(t-1)*M} = Y^{(1)}_{nlmt}$. In this way, the unfolding determines a one-to-one correspondence between entries in the two tensors.

We varied the level of sparsity, such that $p \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$, and at each level of sparsity we simulated 50 datasets. Variational inference is deterministic, and guaranteed to converge to a local maximum of the negative free energy, but can be sensitive to initialisation. For this reason we ran the method ten times on each dataset and consider only the run which achieves the largest ELBO at termination. Following a similar approach to that taken in [Hor15], for the purposes of the simulations we stop the algorithm as soon as the average rate of posterior inclusion probabilities (PIPs) $\langle s_{cl} \rangle$ passing the threshold of 0.5 across 10-iterations drops below 1. In all inspected cases, the ELBO had also converged at this point. The algorithm can shrink components to zero, automatically selecting the number of non-zero components. We initialised with double the true number of components in all cases, and fix the hyperparameters as follows: $e = 10^{-6}, f = 10^{6}, g = 0, h = 0$, $u = 10^{-6}, v = 10^{6}, r = 1, z = 1$ resulting in uninformative priors on the precision of the noise and 'slab', a flat prior on the component sparsity parameter ρ_c, and a beta with point masses at 0 and 1 for ψ_{cl}.

The algorithm scales linearly in N, L, M, T and quadratically in the number of

(N, L, M, T, C)	Time for 200 iterations (s)	Time to convergence (s)
$(200, 500, 16, 3, 8)$	131	8
$(400, 1000, 32, 6, 16)$	1891	611

Table 2.1: **Example running times.** To demonstrate timing, we simulated two datasets with occupancy $p = 0.4$, simulated with 8 components each, and with dimensions (N, L, M, T) as given in the first column, identically to the simulated datasets used in the main text except extending the underlying tissue scores matrix B beyond $T = 3$ to $T = 6$ by drawing from a standard normal distribution. On a desktop computer running `R` version 3.5.2, we ran the method on each dataset twice, for 200 iterations, and to the convergence criterion described in the main text.

components C. Example running times are given in Table 2.1. Figure 2.2(A) shows the simulated tissue and temporal loadings, together with the estimates of these loadings when our algorithm was applied to a single simulated dataset, with occupancy $p = 0.4$.

2.2.1 Evaluating performance

We compared performance of the models by evaluating how well the method estimates the number of components and how well the individual scores and the gene loadings were recovered.

Identifying estimated components

To evaluate the estimated number of non-zero components, we consider the posterior means $\langle s_{cl} \rangle, \langle a_{nc} \rangle, \langle b_{tc} \rangle, \langle d_{mc} \rangle$. The PIPs $\langle s_{cl} \rangle$ are thresholded such that an entry $\langle s_{cl} \rangle \geq 0.5$ indicates $x_{cl} \neq 0$ and gene l is active in component c. The posterior mean values $\langle s_{cl} \rangle$ tend to be either close to zero, or to one, so although a threshold of 0.5 was used, any value between 0.1 and 0.9 gave very similar results.

Having identified the components that were shrunk to zero, those columns of A, B, D were removed. Computing the remaining postprocessing and metrics requires the correct number of components in all cases. If the number of estimated components

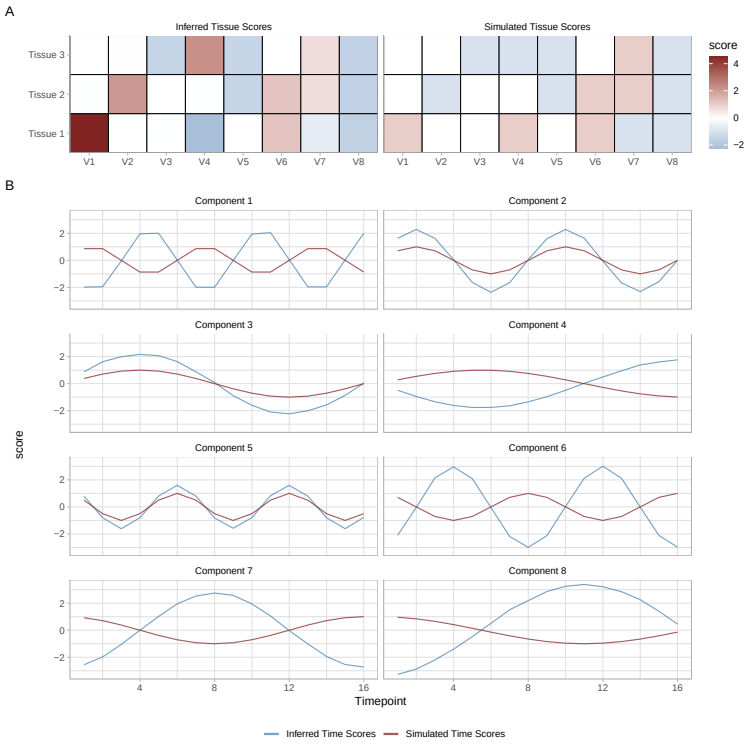

Figure 2.2: **Example simulated and recovered tissue and time scores matrices.** (A) The recovered and simulated tissue scores matrices. Component alignment was carried out with the individual scores matrix as described in the main text. Up to scale and sign (which are undetermined in the model), the tissue scores matrix is recovered well, and we note the scale is common on any given column reflecting the component-independence of scale in the model. (B) The corresponding recovered and simulated time scores for each component. Up to scale and sign these are recovered almost perfectly.

was less than the true number, then additional zero components were added to X, and columns of zeros added to A, B, D. If more than 8 components were estimated then the components were selected to maximise the absolute correlation of the individual scores vectors with the true individual scores vectors.

The estimated individual, tissue, and time scores, and gene loadings matrices produced by this method are denoted $\hat{A}, \hat{B}, \hat{D}$ and $\hat{X}$ respectively.

Individual scores vectors

Since there is a permutation and scale indeterminacy in the model we perform a search to determine the optimal permutation of components. That is, we permute

the estimated components to maximise the average absolute correlation between the individual scores vectors, and change the sign of the vector in order to make the resulting correlations positive.

We compare the estimated individual scores vectors with the true individual scores vectors by root mean squared error. Since root mean squared error (2.5) is affected by scaling, each of the estimated and true individual scores vectors are scaled to have unit variance where the component is non-zero.

$$RMSE(\hat{A}, A) = \sqrt{\frac{1}{NC}\sum_{n,c}(\hat{a}_{nc} - a_{nc})^2} \tag{2.5}$$

With a poor set of estimates, the optimal permutation may not be very well-defined (that is, it may not be significantly better than all others). In view of this, we use the sparse stability index as introduced in [GBE13]. Let Σ be the $C \times C$ matrix with i, j entry the absolute correlation of the ith column of $\hat{A}$ and the jth column of A. With this definition, the sparse stability index is defined as

$$\begin{aligned} SSI =& \frac{1}{2C}\sum_{i=1}^{C}\left(\max_j \Sigma_{ij} - \frac{\sum_{j=1}^{C}\mathbb{1}(\Sigma_{ij} > \mathrm{s}_i^r)\Sigma_{ij}}{C-1}\right) \\ &+ \frac{1}{2C}\sum_{j=1}^{C}\left(\max_i \Sigma_{ij} - \frac{\sum_{i=1}^{C}\mathbb{1}(\Sigma_{ij} > \mathrm{s}_j^c)\Sigma_{ij}}{C-1}\right), \end{aligned}$$

where $\mathrm{s}^r, \mathrm{s}^c$ are the vectors of row and column means of Σ respectively, and $\mathbb{1}(\cdot)$ denotes an indicator function. This is invariant to scaling and permutation.

The SSI penalises cases where more than one entry in a row or column is close to one, or if a row or column contains no large entries. The former might happen if a component is split into two or more estimated components, and the latter if a component is missed. A high SSI indicates a good match between A and $\hat{A}$.

Gene loadings vectors

Finally, we consider recovery of the set of non-zero gene loadings, by varying a threshold of inclusion on the posterior inclusion probabilities, calculating the false positive rate and true positive rate of the method and summarizing this as a receiver operating characteristic curve. The false positive rate (FPR) is calculated as in equation (2.6), and true positive rate (TPR) as in equation (2.7), where X is the simulated gene loadings matrix, and $\hat{X}$ is the inferred gene loadings matrix, with columns aligned using the individual scores matrix. We define $\hat{X}_{cl} \neq 0$ precisely when the posterior inclusion probability exceeds a defined threshold.

$$\text{FPR} = \frac{\sum_{c=1}^{C}\sum_{l=1}^{L}\mathbb{1}(\hat{X}_{cl} \neq 0 \text{ and } X_{cl} = 0)}{\sum_{c=1}^{C}\sum_{l=1}^{L}\mathbb{1}(X_{cl} = 0)}. \tag{2.6}$$

$$\text{TPR} = \frac{\sum_{c=1}^{C}\sum_{l=1}^{L}\mathbb{1}(\hat{X}_{cl} \neq 0 \text{ and } X_{cl} \neq 0)}{\sum_{c=1}^{C}\sum_{l=1}^{L}\mathbb{1}(X_{cl} \neq 0)}. \tag{2.7}$$

2.3 Results

Figure 2.3 shows the results of the 3D and 4D models at estimating the number of components for each of the 5 levels of sparsity and each of the data types. We find that for occupancy $p \geq 0.2$, the algorithm tends to overestimate the number of components in both 3D and 4D datasets, but this improves as p decreases (as the gene loadings matrix becomes increasingly sparse).

The number of components estimated in the 4D simulations typically showed less variability than the 3D simulation, and for each sparsity level, was typically closer to the true value. The 3D model on the unfolded dataset performs qualitatively similarly to the 4D model, with both typically overestimating the number of components for occupancy ≥ 0.2, when components are often split into two or more, and tending to underestimate the number of components for very sparse $p = 0.1$.

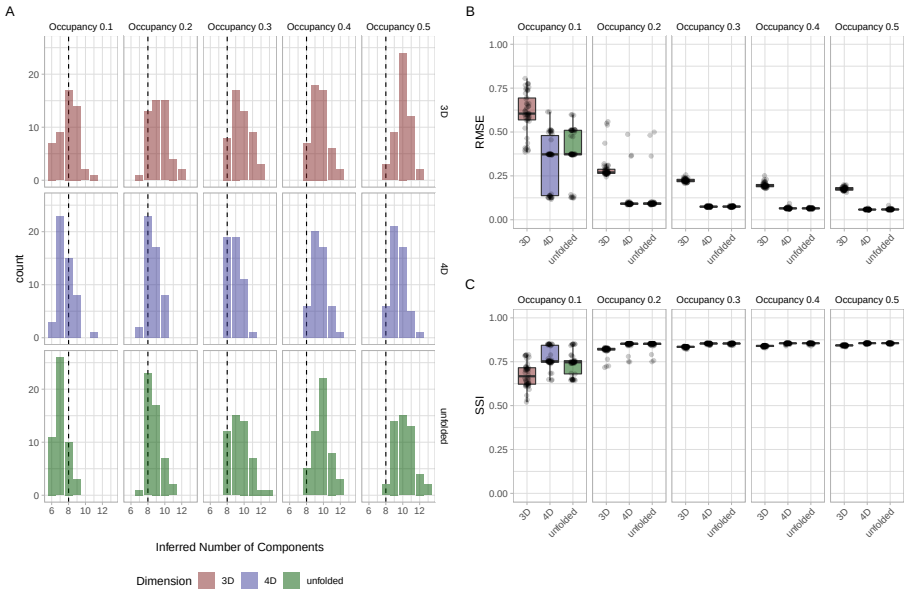

Figure 2.3: A) Comparison of estimated number of components. The dashed line indicates the true value of 8 underlying components. B) RMSE of the simulated and inferred cell scores after alignment as described in the main text. C) SSI of the inferred and simulated cell scores after alignment.

Figure 2.3(B,C) shows the RMSE and SSI metrics between estimated and simulated individual scores matrices for each sparsity level. We note that for 8 components, the SSI metric is bounded above by $\frac{6}{7}$. Both statistics appear stable and tightly distributed for occupancy between $p = 0.2$ and $p = 0.5$. It is noticeable that performance slightly worsens as the simulations become more sparse (low p), but the performance is particularly less predictable at $p = 0.1$. The RMSE achieved by the four-dimensional algorithm is generally lower than that achieved by the three-dimensional algorithm and the SSI is generally higher in the four dimensional results than those of the three-dimensional results, highlighting the improved performance achieved by incorporating time-series data. However, the 3D model applied to the unfolded datasets perform very similarly to the 4D model in all but the occupancy 0.1 datasets, where the $4D$ model outperforms the unfolded results. This is an important observation as it may be the most realistic cases since relatively few genes are involved in most biological processes and we note also that many inferred components in the scRNA-seq data

analyses presented in Chapter 5 have occupancy less than 0.1.

We note also that in order to calculate the statistics we use here, we have discarded estimated components since the algorithm often inferred more than the 8 simulated components. It is perfectly possible for this method to split a component into two or more inferred components. This would not be detected in the statistics used here and could lead to decreased sparse stability index, or increased root mean squared error.

We also compare the false-positive-rate and true-positive-rate achieved by each method across sparsity levels in Figure 2.4. In all cases, the false positive rate is very conservative. These results show a similar pattern to the other statistics, a decrease in performance as sparsity increases. The four-dimensional algorithm again seems comparable to the three-dimensional results from the unfolded data, and clearly outperforms the three-dimensional algorithm on a smaller dataset, with increased true positive rate and lower false positive rate across the majority of simulations (in fact all simulations for $p = 0.3, 0.4$, and $p = 0.5$).

We note also that the time and tissue scores matrices have been recovered very well in all scenarios. An example of the tissue scores matrix as simulated and as recovered is shown in Figure 2.2(A). For the same dataset and run, we plot the recovered time scores in Figure 2.2(B). In both plots the scale and sign indeterminacy of the model is clear, but subject to this consideration, the recovery is excellent. These components were aligned with the simulated components by the individual scores matrix, following the method described earlier in the text. Applying this same method and recording the average absolute correlation achieved across the columns of the estimated and simulated time scores matrices, we found exceptionally clear recovery of the time scores matrix (Figure 2.5). We consider this to be a relatively complex time scores matrix, with several columns highly correlated, but it was recovered extremely well.

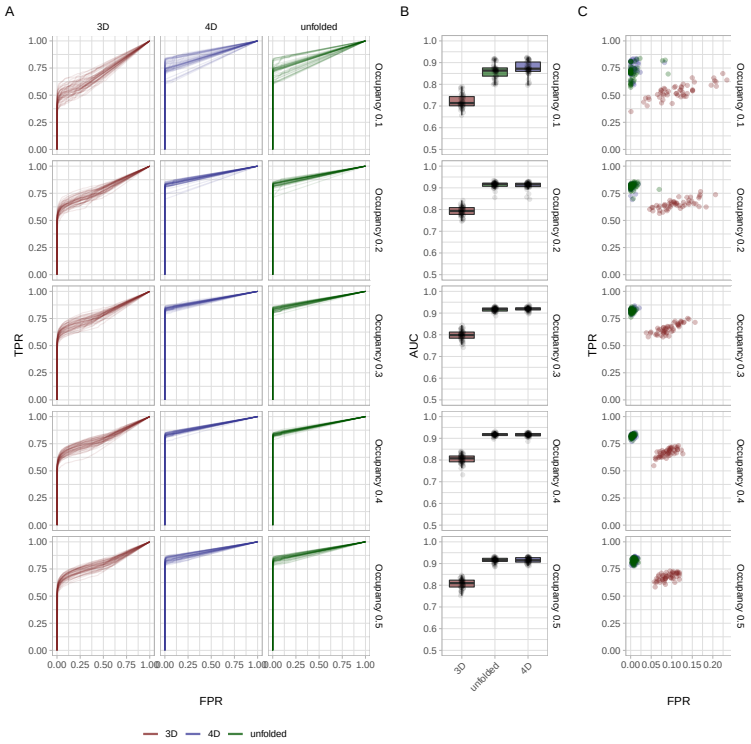

Figure 2.4: **Comparison of recovery of the gene loadings matrix by FPR and TPR.** On the 50 datasets at each level of sparsity we consider false positive rate and true positive rate for both $3D$ and $4D$ simulations. A) ROC curves for the three methods for the highest ELBO run on each dataset/method after component alignment using individual scores as described in the main text. B) Area under the curve for the ROC curves. C) FPR/TPR plotted when applying a threshold of 0.5 to the posterior inclusion probabilities.

2.4 Discussion

The application of sparsity inducing priors to a parallel factor analysis model for gene expression data was novel in [HVB+16] and has been shown to have application in large gene expression studies, in particular having power to detect trans eQTLs. We have extended this model to perform inference on four-dimensional data with gene expression data in mind, although we expect the approach to be more widely applicable.

The interpretation of the matrix B has been that of a tissue scores matrix and this has been recovered well. Similarly the matrix D, which we have used to define the variability of component expression over time has been recovered well - see for

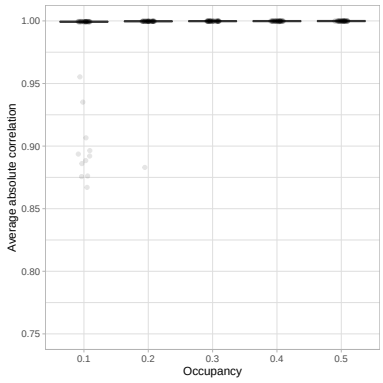

Figure 2.5: **Time Scores matrix is recovered well.**
For the highest negative free energy runs on each dataset we plot the average absolute correlation of the non-constant columns of the estimated time scores matrix with the true underlying time scores matrix, based on the component alignment carried out with the Individual scores matrix described in the main text. The plot above shows excellent recovery of the matrix. We note also that the sign and scale are indeterminate in the model, as shown in Figure 2.2.

an example Figure 2.2 and Figure 2.5. The combination of these two matrices and patterns being recovered is indicative of the power of both methods to detect 'discrete' patterns in the componentwise scores such as that seen in the tissue scores matrix, as well as the more continuous variation as captured in the time scores matrix.

We have shown that the four-dimensional method outperforms the three-dimensional method in recovering the gene loadings in components and the individual scores matrix, and observed excellent recovery of the tissue and time scores matrices. The results presented here suggest that the results from the 3D model are limited by the amount of data in a single time slice. When applied to a suitable unfolding of the four dimensional data, the results of the 3D model are comparable with that of the 4D model in gene loadings and individual scores for all but the most sparse simulations, when the 4D model outperforms the 3D model even on the unfolded data. However, the 4D PARAFAC method presented here reliably recovered the tissue and time scores across multiple components and simulations while using fewer parameters than the corresponding 3D model, and these "context" dimensions are very difficult to

disentangle or interpret in the results from an unrolled 3D dataset. We thus conclude that, when interpretability in the two additional context dimensions is desirable, or when components are expected to be sparse, the 4D model is a valuable extension.

While we present here a brief summary of results from a simulation study similar to that of [HVB$^+$16], we have observed qualitatively identical results in another scenario designed to be more symmetric in dimension, with a range of dimensions up to $50 \times 50 \times 50 \times 50$ datasets.

The current implementation of the model does not allow for missing data of any kind, and it is usual practice to remove any sample, tissue, time, or genes that are completely missing. However, in principle a four dimensional PARAFAC model as described in this chapter can accommodate missing data by simply masking the missing data points from the likelihood calculation. In particular, we note that care should be taken if extending and applying this model in such a way as missing time points could be a source of confounding between the time scores and individual scores.

We expect that with the increasing availability of gene expression data, time series data across multiple tissues will become available and this method will prove valuable in decomposing such datasets into active components with weightings (scores) for different features (dimensions). This approach has the potential to provide insight into components of genes involved in biological processes, as well as their differential expression over time. We note, however, that both the 3D and 4D model were designed with bulk RNA-seq data in mind. When large single cell RNA-seq data is used, the size of the datasets is large and a 2D (possibly group) factor analysis may be sufficient to capture much of the variation present while not constraining the results by requiring any structural modelling assumptions such as the PARAFAC decomposition.

CHAPTER 3

Annotation Informed Factor Analysis Model for scRNA-seq data

This chapter describes a new method for factor analysis leveraging annotations, named AnnotFA, for inferring structure in a single cell RNA-seq dataset. Such a dataset consists of counts for every cell in a sample, for a set of genes. The approach proposed here is to model this data in a factor analysis setting, thereby modelling the components of variation in expression that are shared across cells and genes. Currently, few methods in the literature exist to incorporate prior knowledge that may inform factor loadings. Examples of such prior information may include, for example, pathway annotations as available through the REACTOME database [JTGV$^+$05], or prior evidence of transcriptional regulation, such as transcription factor binding affinities for each gene in the dataset. We may also have information on cells, such as batch, or case-control status of cells from patients undergoing treatment. In the model described here, we make use of a spike-and-slab prior that informs sparsity on the component loadings, incorporating such prior information into the inclusion weights of genes in components. This method allows for automatic selection of relevant

information to the component gene inclusion probabilities, with immediate biological interpretability, while remaining scalable to existing large datasets. In this way, we make use of several of the features of existing factor analysis methods in the literature while incorporating prior information in our inference.

3.1 The Model

3.1.1 Heuristic description

In single cell RNA-seq datasets, biological processes are likely shared across many cells, and each cell is likely undergoing several processes at any one time. This leads to patterns of expression of genes that are shared across subsets of cells at differing levels. There will also be non-biological sources of variation present in the data such as technical artefacts including batch effects and noise. The method we describe in this chapter attempts to infer these sources of variation that are shared across cells, and their respective levels of expression in each cell.

3.1.2 Data model description

Throughout, we assume the data is a $J \times I$ matrix representing J cells and I genes, and we model the data mean values as AZ^T, where $A = (a_{jq})$ is a cell scores matrix and $Z = (z_{iq})$ a gene loadings matrix.

Additionally, we assume the presence of a set of annotation vectors $\{t_v\}_v$ representing prior information relevant to patterns of gene expression dynamics. For example, this may be inclusion or exclusion of genes in a given biological process, or vectors for each of a set of transcription factors, consisting of binding affinities (i.e likelihood of regulation) of that transcription factor to the promoter region of each gene.

We first describe the Gaussian AnnotFA model, before describing the closely related Poisson AnnotFA model. In later sections we describe our prior on model

parameters, and details of the inference procedure.

We model the data as a sum of Q latent components and an intercept component. Each component, or factor, is a pair of columns from the cell scores and gene loadings matrices respectively. In this way the model attempts to explain the variance present in the data across many cells and genes using a much smaller number of latent factors, each capturing covariance across cells and genes.

AnnotFA Gaussian model

The Gaussian model is described in Equation 3.1

$$y_{ji} = \sum_q a_{jq} z_{iq} + z_i^I + \epsilon_{ji} \tag{3.1}$$

where ϵ_{ji} is Gaussian noise, for $i = 1 \ldots, I$, $j = 1 \ldots J$, and $q = 1 \ldots Q$ and where z_i^I is an intercept loading for each gene. This can equivalently be written as $Y = AZ^T + \epsilon$ (by including an intercept component in the scores and loadings matrices). We assume heteroscedastic Gaussian residual noise, independent for each i and j with precision differing across cells,

$$\pi(\epsilon_{ji}) = \mathcal{N}(\epsilon_{ji} \mid 0, \lambda_j^{-1}).$$

AnnotFA Poisson model

As described in [SS20, SVTT18], there has been much discussion around the number of zero values present in scRNA-seq data, and some methods have modelled so-called dropout (for example [PY15]). However, when accounting for the technical noise created by the varying library size, it has been shown that the Poisson model is appropriate for single cell RNA-seq data [SS20, SVTT18].

Based on an identical prior structure to the Gaussian model, we have implemented a Poisson model as in Equation 3.2, where $\lambda : \mathbb{R} \mapsto \mathbb{R}_{>0}$ is the rate function $x \mapsto$

$\log(1+e^x)$ and $\tilde{n}_j = \sum_i y_{ji}$ is the library size for cell j, and we scale by some (usually large) pre-chosen constant ζ.

Specifically, for $i = 1, \ldots, I$ and $j = 1, \ldots, J$ we model the data y_{ji} as

$$y_{ji} \sim \text{Poi}\left(\tilde{n}_j \frac{\lambda(x_{ji})}{\zeta}\right), \text{ so} \tag{3.2}$$

$$\mathbb{P}(Y|\Theta) \propto \prod_{i,j} \left(\tilde{n}_j \frac{\lambda(x_{ji})}{\zeta}\right)^{y_{ji}} \exp\left(-\left(\tilde{n}_j \frac{\lambda(x_{ji})}{\zeta}\right)\right) \text{ where} \tag{3.3}$$

$$x_{ji} = \sum_q a_{jq} z_{iq} + z_i^I. \tag{3.4}$$

Incorporating library size in the model accounts for this source of variation in the data, similar to the suggestion in [SS20]. A similar approach is taken in [GI20], in the context of probabilistic cell type identification.

The rate function $x \mapsto \log(1+e^x)$ was chosen to facilitate inference based on [SB12] (see also Section 3.2.2) while maintaining the desirable features of the Gaussian model, modelling co-expressed genes and capturing both up-regulated (promoted) and down-regulated (repressed) factors. We note that it is also desirable that the components and their relationships are easily interpretable, and the rate function presents a challenge here, due to the non-linearity near zero. This is of particular concern as the inference method we use approximates the Poisson likelihood by a Gaussian. However, the prior structure specified in Section 3.1.3 results in a symmetric prior distribution with mean 0 on the x_{ji}, so a heuristic argument replacing $\lambda(x_{ji})$ with $\log(2)$ suggests an appropriate value of ζ is $I\log(2)$. For these reasons, in applications we have used values of ζ of the order of I, the number of genes in the model. We also tested this in simulations, some of which are presented in Section 4.1.3, demonstrating that the model achieves largest R^2 for such values of ζ. Since ζ is large, this concentrates the posterior in a region in which the link function is approximately linear [1].

[1] Notice that $\log(1+e^x) \simeq x$ for sufficiently large x.

3.1.3 Distributions on prior parameter values

Gene loadings

Most biological processes are likely to involve relatively few genes, and this corresponds to a belief that components that represent biological pathways or processes will be sparse. For this reason, we incorporate sparsity-inducing priors on the gene loadings. In doing so, we have aimed to maintain the features of other methods such as SDA [HVB+16] and MOFA [AVA+18], whilst also incorporating prior information into which genes are included in the loadings vector, in a data-driven way. To achieve this, we developed a prior structure that allows for sparsity in the gene inclusion in a factor loadings vector while being informed by prior information, and also allows for sparsity in the sense of shrinking components explaining little variance to zero. Building on the spike and slab prior structure of ([LCW+06, MB88]), we describe the new prior as follows:

We place a spike-and-slab prior on the columns of Z, consisting of a mixture of a "spike", being a point-mass at zero, and a "slab" being a more diffuse distribution, in this case a Gaussian centered at zero. The prior placed on an entry of the gene loadings matrix z_{iq} is

$$\pi(z_{iq}|\beta_q, p_{iq}) = (1 - p_{iq})\,\delta_0(z_{iq}) + p_{iq}\mathcal{N}(z_{iq}|0, \beta_q^{-1}) \tag{3.5}$$

where p_{iq} is an inclusion weight, β_q is the precision of the Gaussian slab, and δ_0 is a delta function at 0. One advantage of such priors is that the p_{iq} can be interpreted as probabilities of inclusion in the component. Therefore, these inclusion probabilities provide a means to model the gene-wise sparsity in a component. As discussed in Chapter 1, both MOFA [AVA+18] and SDA [HVB+16] make use of similar prior structures on gene loadings.

Since the atom at zero makes inference with such a distribution difficult, it is usual

to reparameterize this, following [TLG11] modelling the loading as a product of a Gaussian distributed w_{iq} and a Bernoulli variable s_{iq} with parameter p_{iq}. Thus, the prior structure can be written as follows:

$$\begin{aligned} z_{iq} &= w_{iq}s_{iq} && (3.6)\\ \pi(s_{iq}|p_{iq}) &= \text{Ber}(s_{iq}|p_{iq}) && (3.7)\\ \pi(w_{iq}|\beta_q) &= \mathcal{N}(w_{iq}|0, \beta_q^{-1}) && (3.8) \end{aligned}$$

The component-wide slab precision parameters β_q are given a conjugate Gamma prior distribution with parameters e, f.

$$\pi(\beta_q) = \mathcal{G}(\beta_q|e, f). \tag{3.9}$$

This corresponds to a so-called automatic relevance determination (ARD) prior. Effectively, the value of $1/\beta_q$ determines the importance of (i.e the amount of variance explained by) the q^{th} component. A large value of β_q corresponds to a component explaining little variance, and can force all loadings in the component to zero, while a small value of β_q corresponds to an important component. This ARD prior structure was also present in MOFA[AVA+18], SDA [HVB+16] and f-scLVM [BPM+17].

A main new feature of our model is to incorporate prior information into the gene loadings. We model the inclusion indicators s_{iq} probabilistically as informed by the prior information, which we call annotations, by modelling the log-odds of s_{iq} as linear combinations of the prior annotation vectors,

$$\log\left(\frac{p_{iq}}{1-p_{iq}}\right) = \eta_q^I + \sum_v \eta_{qv} t_{vi}, \tag{3.10}$$

where η_q^I is an intercept coefficient. This approach allows the annotations to be selected based on their importance in factors and also accommodates measurement

noise in the annotations or measurements.

The intercept coefficient serves two purposes, the first is that since patterns of co-expression may not be well known in advance (for example the data may contain information originating from previously unknown biological processes, or batch effects) and therefore may not be represented in the set of annotations used to inform the model, the intercept provides a means of returning non-zero probability across genes while not inferring any importance on any particular annotations. Secondly, we use the intercept to impose sparsity as we now describe. Previous approaches have imposed sparsity by having a prior with a large density near zero for p_{iq}. This suggests that the average value across a component should be small. To impose this, we first center the annotations by subtracting the mean of each annotation vector, and we include a pre-specified number of *padding genes* to each annotation with annotation value zero (the mean), and in our inference we assume the presence of a zero inclusion probability for each of the padding genes. This results in fitting an intercept which is negative, requiring evidence to incorporate an annotation with a positive coefficient including genes in a component. The number of zeros to use as padding is fixed throughout the inference and specified at the start of the algorithm. In practice we see that this creates a flexible way to impose sparsity within components.

In this way we incorporate the sparsity inducing prior structure from other spike and slab models. Moreover, we are able to infer components that are well represented by the annotations, and those that are not, without the need for a secondary type of "free" component, and we can capture components that are well represented by combinations of annotations without the need for multiple correlated components.

We note that if we include only annotations set to zero, then the model reduces to a single parameter $p_{iq} = p_q$, and the spike and slab prior with a component-wide inclusion weight is recovered.

Finally, we have a prior on the intercept loadings component, consisting of the

z_i^I, which is included in all cells but varies across genes, and is used to capture either issues with the centering of the data during normalization, as is usually applied before application of a Gaussian model, or a dense component which is common across cells. We give this a broad Gaussian prior,

$$\pi(z_i^I) = \mathcal{N}(z_i^I|0, \beta^{I^{-1}}).$$

We set $\beta^I = 10^{-4}$ as our default precision parameter to provide an uninformative prior.

Cell scores

In our default setting we give the cell scores standard Gaussian priors. This corresponds to a belief that cells and components are independent. Independently for each j and q, the prior is

$$\pi(a_{jq}) = \mathcal{N}(a_{jq}|0, 1).$$

We have also extended the model to include an annotation-informed spike-and-slab prior on the cell scores, with cell-annotation vectors, as an option in the implementation.

Gaussian noise precision

Recall that for the Gaussian model, the noise is modelled for each $j = 1, \dots J$ and each $i = 1, \dots, I$ as $\epsilon_{ji} \sim \mathcal{N}(0, \lambda_j^{-1})$. We additionally place a conjugate Gamma prior on the precision parameters λ_j, independently for each j as

$$\pi(\lambda_j) = \mathcal{G}(\lambda_j|u, v). \tag{3.11}$$

Hyperparameters

The full set of hyperparameters for the model is $e, f, \beta^I, \{\eta_{qv}\}_{q,v}$ and additionally u, v for the Gaussian model. The default settings are $e = 10^{-6}, f = 10^6$, $u = 10^{-6}, v = 10^6$, providing broad uninformative priors on the Gamma distributions, and $\beta^I = 10^{-4}$, again providing a broad uninformative prior on the intercept component.

The remaining hyperparameters $\{\eta_{qv}\}$ are not given priors but are instead inferred in an empirical Bayes framework, applying a variational EM algorithm to maximise a lower bound to the log marginal likelihood, or model evidence, with respect to both the hyperparameters $\{\eta_{qv}\}$, and the variational approximation to the posterior.

3.2 Inference under the Model

We fit the AnnotFA model by variational inference (VI) with a structured mean-field approach, which ensures the method scales well to the size of datasets we expect it to be used for. The reader is referred to Section A.1.2 for further references and discussion.

3.2.1 Updates with a Gaussian likelihood model

In deriving the AnnotFA inference algorithm we fully partition the model variables except for the pairs of spike and slab from the loadings (or, if this option is used, the cell scores) which are paired together, since they are strongly correlated [TLG11]. Denoting $\mathbb{E}_Q(.)$ by $\langle . \rangle$ for brevity, the updates for the variational distributions Q_i^* in the variational EM algorithm are as follows:

Gaussian Noise Precision

$$
\begin{aligned}
Q^*(\lambda_j) &= \mathcal{G}(\lambda_j | u_j^*, v_j^*) \\
u_j^* &= u + \frac{I}{2}; \\
v_j^* &= \left(\frac{1}{v} + \frac{1}{2} \sum_i \left\langle \left(y_{ji} - z_i^I - \sum_q a_{jq} z_{iq} \right)^2 \right\rangle \right)^{-1};
\end{aligned}
$$

Cell Scores

$$
\begin{aligned}
Q^*(a_{jq}) &= \mathcal{N}(a_{jq} | \mu_{jq}^*, (\omega_{jq}^*)^{-1}) \\
\omega_{jq}^* &= 1 + \langle \lambda_j \rangle \sum_i \langle z_{iq}^2 \rangle; \\
\mu_{jq}^* &= \frac{1}{\omega_{jq}^*} \langle \lambda_j \rangle \sum_i \langle z_{iq} \rangle \left(y_{ji} - \langle z_i^I \rangle - \sum_{k \neq q} \langle a_{jk} \rangle \langle z_{ik} \rangle \right);
\end{aligned}
$$

Gene Loadings

The intercept component:

$$
\begin{aligned}
Q^*(z_i^I) &= \mathcal{N}(z_i^I | \zeta_i^*, \pi_i^{*-1}) \\
\pi_i^* &= \beta_I^{-1} + \sum_j \langle \lambda_j \rangle; \\
\zeta i^* &= \pi_i^{*-1} \sum_j \langle \lambda_j \rangle (y_{ji} - \sum_q \langle a_{jq} \rangle \langle z_{iq} \rangle);
\end{aligned}
$$

Recall that $z_{iq} = w_{iq} s_{iq}$ and we pair w_{iq} and s_{iq} in the partition and infer $Q^*(w_{iq}, s_{iq})$, as first demonstrated in [TLG11]. This can be decomposed as $Q^*(w_{iq}, s_{iq}) = Q^*(w_{iq} | s_{iq}) Q^*(s_{iq})$, resulting in the following update equations:

The slab:

$$
\begin{aligned}
Q^*(w_{iq}|s_{iq}) &= \mathcal{N}(w_{iq}|s_{iq}m^*_{iq}, ((1-s_{iq})\langle\beta_q\rangle + s_{iq}\sigma^*_{iq})^{-1}) \\
\sigma^*_{iq} &= \langle\beta_q\rangle + \sum_j \langle\lambda_j\rangle\langle a^2_{jq}\rangle; \\
m^*_{iq} &= \frac{1}{\sigma^*_{iq}} \sum_j \langle\lambda_j\rangle\langle a_{jq}\rangle \left(y_{ji} - \langle z^I_i\rangle - \sum_{k\neq q} \langle a_{jk}\rangle\langle z_{ik}\rangle \right);
\end{aligned}
$$

The spike:

$$
\begin{aligned}
Q^*(s_{iq}) &= \mathcal{B}er(s_{iq}|\gamma^*_{iq}) \\
\gamma^*_{iq} &= \frac{1}{1+e^{-u^*_{iq}}}; \\
u^*_{iq} &= \frac{1}{2}\sigma^*_{iq}(m^*_{iq})^2 + \frac{1}{2}\log(\langle\beta_q\rangle) + \log(\frac{p_{iq}}{1-p_{iq}}) - \frac{1}{2}\log(\sigma^*_{iq}).;
\end{aligned}
$$

The precision:

$$
\begin{aligned}
Q^*(\beta_q) &= \mathcal{G}(\beta_q|e^*_q, f^*_q) \\
e^*_q &= e + \frac{I}{2}; \\
f^*_q &= \left(\frac{1}{f} + \frac{1}{2}\sum_i \langle w^2_{iq}\rangle \right)^{-1};
\end{aligned}
$$

It remains to describe the updates for η_{qv}: Approaching this as a variational EM algorithm we update the values for η_{qv} to maximise the ELBO. Since the only terms of the ELBO that rely on η_{qv} are the expected log likelihood of the (prior) spike

probabilities, this amounts to maximising the following function:

$$\begin{aligned}\tilde{F}(\eta) &= \sum_{i,q} \left(\langle s_{iq} \rangle \log(p_{iq}) + (1 - \langle s_{iq} \rangle) \log(1 - p_{iq}) \right) \\ &= \sum_{i,q} \left(\langle s_{iq} \rangle (\eta_q^I + \sum_v \eta_{qv} t_{vi}) - \log(1 + \exp(\eta_q^I + \sum_v \eta_{qv} t_{vi})) \right)\end{aligned}$$

where $\log(\frac{p_{iq}}{1-p_{iq}}) = \eta_q^I + \sum_v \eta_{qv} t_{vi}$, and η_q^I is the intercept coefficient. This is clearly separable across factors q, and within each factor is similar to a logistic regression, but with continuous valued response. We update each set of component coefficients $\{\eta_{qv}\}_v$ independently by a limited memory BFGS (L-BFGS) quasi-Newton algorithm.

3.2.2 Inference under the Poisson likelihood model

For inference under the Poisson model, we follow the general approach of [SB12], applying a Gaussian approximation to the Poisson likelihood, with some adjustments to accommodate some differences in our model. We now collectively think of Θ as denoting all parameters and variables in the AnnotFA model. We denote by $X(\Theta)$ the matrix obtained from the model, that is $X(\Theta)_{ji} = z_i^I + \sum_q a_{jq} z_{iq}$. and we write x_{ji} for $X(\Theta)_{ji}$. For brevity, we define $n_j = \frac{\tilde{n}_j}{\zeta}$ for the remainder of this chapter.

Recall that inference in variational Bayes proceeds by optimizing the ELBO, that is, the lower bound $F(Q)$ to the model evidence $\log(\mathbb{P}(Y))$. This can equivalently be written as minimizing the negative lower bound, that is, finding

$$\min_{Q(\Theta)}(-F(Q)) = \min_{Q(\Theta)}(\mathbb{E}_Q(-\log \mathbb{P}(Y|\Theta) + \mathrm{KL}(Q(\Theta)|\pi(\Theta))) \tag{3.12}$$

Now, returning to the Poisson model, we can write $-\log(\mathbb{P}(Y|\Theta)) = \sum_{i,j} f_{ji}(x_{ji})$, up to a constant term which depends only on the data Y. Here we define $f_{ji}(x_{ji}) = n_j \lambda(x_{ji}) - y_{ji} \log(n_j \lambda(x_{ji}))$ where λ is the rate function $x \mapsto \log(1 + e^x)$. Now, f_{ji} is twice differentiable, and in order to proceed with a Taylor expansion we first determine

a bound on f''_{ji}. Note that $\lambda''(x) = \frac{1}{1+e^{-x}}\frac{1}{1+e^{x}}$ and thus $\lambda''(x) \leq \frac{1}{4}$. Following [SB12], the second derivative of $-\log(n_j\lambda(x))$ is bounded above[2] by 0.17. It follows that

$$f''_{ji}(x_{ji}) \leq \frac{n_j}{4} + 0.17 \max_i(y_{ji}) =: \kappa_j$$

With this bound in place, by Taylor expansion, we can bound the coordinate functions of the log-likelihood as in Equation 3.13. Although throughout this derivation the bounds are tight, the presence of extremely large values in the data will degrade the bound κ_j, and the authors of [SB12] recommend clipping outlying large data values in practice.

$$f_{ji}(x_{ji}) \leq \frac{\kappa_j}{2}(x_{ji} - \xi_{ji})^2 + f'_{ji}(\xi_{ji})(x_{ji} - \xi_{ji}) + f_{ji}(\xi_{ji}) =: q_{ji}(x_{ji}, \xi_{ji}). \tag{3.13}$$

Now using these bounds and exchanging $\mathbb{E}_Q$ and $\min_{\xi_{ji}}$ in Equation 3.12, we can reframe inference as a new variational optimization problem:

$$\min_{Q(\Theta),\xi_{ji}} \left(\sum_{i,j} \mathbb{E}_Q(q_{ji}(x_{ji}, \xi_{ji})) + \mathrm{KL}(Q(\Theta)|\pi(\Theta)) \right) \tag{3.14}$$

A natural approach to this problem is to alternate between updates for ξ and the updates for $Q(\Theta)$. For the updates to ξ, notice that $\mathbb{E}_Q(q_{ji}(x_{ji}, \xi_{ji})) = q_{ji}(\mathbb{E}_Q(x_{ji}), \xi_{ji}) + C_{ji}$ where C_{ji} depends only on the first and second moments of Q, not on ξ_{ji}. Thus, conditional on $Q(\Theta)$, we may aim to minimize the term $q_{ji}(\mathbb{E}_Q(x_{ji}), \xi_{ji})$. Now, for any ξ_{ji},

$$\begin{aligned} q_{ji}(\mathbb{E}_Q(x_{ji}), \xi_{ji}) &\geq f_{ji}(\mathbb{E}_Q(x_{ji})) & (3.15) \\ &= q_{ji}(\mathbb{E}_Q(x_{ji}), \mathbb{E}_Q(x_{ji})). & (3.16) \end{aligned}$$

[2]This is claimed by inspection. Since for $x >> 0$, $\lambda(x) \simeq x$ we found in practice this identity holds to computational accuracy for $x > 10$, and $\lambda(x) \simeq 0$ for $x << 0$, again in practice for $x < -10$. Between these values the second derivative is clearly positive, and by inspection is bounded above by 0.17.

It follows that, conditional on $Q(\Theta)$, the optimal update for ξ_{ji} is to set $\xi_{ji} = \mathbb{E}_Q(x_{ji}) = \langle z_i^I \rangle + \sum_q \langle a_{jq} \rangle \langle z_{iq} \rangle$.

Now for the update of $Q(\Theta)$: Setting $\tilde{y}_{ji} = \xi_{ji} - f'_{ji}(\xi_{ji})/\kappa_j$, reveals that

$$q_{ji}(x_{ji}, \xi_{ji}) = \frac{\kappa_j}{2}(x_{ji} - \tilde{y}_{ji})^2 + \tau_{ji}, \tag{3.17}$$

where τ_{ji} is a constant with respect to Q. Thus, up to a term constant with respect to Q, the term $q_{ji}(x_{ji}, \xi_{ji})$ is the negative log-likelihood of $\mathcal{N}(\tilde{y}_{ji}|x_{ji}, \frac{1}{\kappa_j})$. Holding the ξ_{ji} fixed and comparing equations 3.14 and 3.12, this implies that the updates for Q are identical to those from a Gaussian model, with precision fixed at κ_j and the data replaced by pseudo-data points $\tilde{Y} = (\tilde{y}_{ji})$.

To summarise: to fit the Poisson model we alternate between updates for ξ_{ji} and updates for the variational distributions $Q(\Theta)$. The update for ξ_{ji} is $\xi_{ji} = \langle z_i^I \rangle + \sum_q \langle a_{jq} \rangle \langle z_{iq} \rangle$, and the pseudo-data $\tilde{y}_{ji}$ is then updated according to the following:

$$\tilde{y}_{ji} = \xi_{ji} - \frac{1}{\kappa_j}\left(n_j - \frac{y_{ji}}{\log(1 + e^{\xi_{ji}})}\right)\frac{1}{1 + e^{-\xi_{ji}}}.$$

The gene loadings and cell scores updates remain the same as in the Gaussian model, except the precision is now fixed at κ_j, and the data is replaced by the pseudo-data.

3.2.3 The ELBO

Recall that the ELBO is calculated as $\mathbb{E}_Q\left(\log \frac{\mathbb{P}(Y,\Theta)}{Q(\Theta)}\right)$. In practice this can be separated into the sum of the log-likelihood term $\mathbb{E}_Q(\log \mathbb{P}(Y|\Theta))$ plus the prior term $\mathbb{E}_Q(\log \mathbb{P}(\Theta))$ minus the variational term $\mathbb{E}_Q(\log \mathbb{Q}(\Theta))$. We describe first the general Gaussian form, and then comment on the adjustments required to formulate the objective function for inference under the Poisson model. We summarise the terms as follows:

Log-likelihood terms

$$-\frac{IJ}{2}\log(2\pi) + \frac{I}{2}\sum_j \langle \log(\lambda_j)\rangle - \frac{1}{2}\sum_{i,j}\langle\lambda_j\rangle \left\langle \left(y_{ji} - z_i^I - \sum_q a_{jq} z_{iq}\right)^2 \right\rangle .$$

For the Gaussian model, λ_j has a Gamma variational distribution, and thus

$$\langle \log(\lambda_j)\rangle = \psi(u_j^*) + \log(v_j^*),$$

where ψ is the digamma function.

When calculating this for the Poisson model, we can remove the constant terms, and replace λ_j with κ_j (which is constant) throughout these terms. The full Poisson objective is thus recovered by adding the sum of the constants τ_{ji} (see Section 3.2.2), and excluding the Gaussian noise precision terms.

Noise Precision Terms:

Prior λ_j terms:

$$\sum_j \left(-\log\Gamma(u) - u\log(v) + (u-1)\langle\log(\lambda_j)\rangle - \frac{1}{v}\langle\lambda_j\rangle \right).$$

Variational $Q(\lambda_j) = \mathcal{G}(\lambda_j | u_j^*, v_j^*)$ terms:

$$\sum_j \left(-\log\Gamma(u_j^*) - u_j\log(v_j^*) + (u_j^*-1)\langle\log(\lambda_j)\rangle - \frac{1}{v_{j^*}}\langle\lambda_j\rangle \right)$$

Unannotated Cell Scores:

Prior a_{jq} terms:

$$-\frac{JQ}{2}\log(2\pi) - \frac{1}{2}\sum_{j,q}\langle a_{jq}^2\rangle$$

Variational $Q(a_{jq}) = \mathcal{N}(a_{jq}|\mu_{jq}^*, \omega_{jq}^{*-1})$ terms:

$$-\frac{JQ}{2}\log(2\pi) + \frac{1}{2}\sum_{j,q}\log(\omega_{jq}^*) - \frac{JQ}{2}$$

Gene Loadings Terms:

Prior β_q terms:

$$\sum_q \left(-\log\Gamma(e) - e\log(f) + (e-1)\langle\log(\beta_q)\rangle - \frac{1}{f}\langle\beta_q\rangle\right).$$

Variational $Q(\beta_q) = \mathcal{G}(\beta_q|e_q^*, f_q^*)$ terms:

$$\sum_q \left(-\log\Gamma(e_q^*) - e_q^*\log(f_q^*) + (e_q^*-1)\langle\log(\beta_q)\rangle - \frac{1}{f_q^*}\langle\beta_q\rangle\right)$$

Prior spike s_{iq} terms:

$$\sum_{i,q}\left(\langle s_{iq}\rangle\log(p_{iq}) + (1-\langle s_{iq}\rangle)\log(p-t_{iq})\right)$$

Variational $Q(s_{iq}) = \mathcal{B}er(s_{iq}|\gamma^*_{iq})$ terms:

$$\sum_{iq} \left(\langle s_{iq} \rangle \log \langle s_{iq} \rangle + (1 - \langle s_{iq} \rangle) \log(1 - \langle s_{iq} \rangle) \right)$$

Prior slab w_{iq} terms:

$$-\frac{IQ}{2}\log(2\pi) + \sum_{i,q}\left(\frac{1}{2}\langle \log(\beta_q)\rangle - \frac{1}{2}\langle\beta_q\rangle\langle w_{iq}^2\rangle\right)$$

Variational $Q(w_{iq}|s_{iq}) = \mathcal{N}(w_{iq}|m^*_{iq}s_{iq}, (\sigma^*_{iq}s_{iq} + (1 - s_{iq})\langle\beta_q\rangle)^{-1})$ terms:

$$-\frac{IQ}{2}\log(2\pi) - \frac{IQ}{2} + \frac{1}{2}\sum_{i,q}\left(\langle s_{iq}\rangle \log(\sigma^*_{iq}) + (1 - \langle s_{iq}\rangle)\log(\langle\beta_q\rangle\right)$$

Prior intercept z_i^I terms:

$$-\frac{I}{2}\log(2\pi) + \frac{I}{2}\log(\beta_I) - \frac{1}{2}\beta_I \sum_i \langle (z_i^I)^2\rangle$$

Variational $Q(z_i^I) = \mathcal{N}(z_i^I|\zeta_i^*, \pi_i^{*-1})$ terms:

$$-\frac{I}{2}\log(2\pi) - \frac{I}{2} + \frac{1}{2}\sum_i \log(\pi_i^*)$$

3.3 Optional Annotated Cell Scores

In addition to prior information on expression of genes, one may have prior information on cells, such as, for example, batch labels, or case-control status, or some treatment status such as dosage. To accommodate this we have also implemented the inclusion of prior information in the cell scores via the same prior-informed spike and slab structure on the cell scores, as on the gene loadings. Since scale is unidentifiable in this model,

we maintain the precision of the slab at 1, so that the spike and slab prior can be implemented by writing the cell scores as a product of a Bernoulli random variable s^A_{jq} and a standard Gaussian random variable w^A_{jq}. In this case the updates are identical to the gene loadings updates, replacing the t_v information vectors (or annotations), with corresponding annotations for the cells, and replacing every occurrence of β_q with 1.

In this case we use the same idea of using padding with zeros in the logistic regression update to impose sparsity, and include this as an additional parameter to set in the model.

The cell scores ELBO terms described previously are for the unannotated cell scores. The ELBO terms for annotated cell scores follow the same structure as for the gene loadings terms, but setting $\beta_q = 1$ throughout, and excluding the corresponding terms depending on e, f, e^*_q, f^*_q.

3.4 Implementation

Hyper parameters have been chosen to provide uninformative priors whenever possible.

AnnotFA is implemented as an R [R C19] package, and will be released at a later date. We make extensive use of Rcpp [EF11] and RcppEigen [BE13] to enable the updates to scale well with sample size. The implementation of the updates scale linearly with the number of genes and number of cells, and quadratically in the number of components, $O(IJQ^2)$. The precise scaling of the algorithm is demonstrated for 50 components, plus the intercept component, for 5000 iterations in Figure 3.1. We note that the Poisson model is slower than the Gaussian model due to the pseudodata update. Although the algorithm is iterative and iterations cannot be parallelized, we have implemented multithreading through the use of openmp for the pseudodata update, the logistic regression updates, and all matrix operations.

In implementing the Logistic Regression update we make use of the gradient-based L-BFGS optimization procedure as provided by the header library included in the R(cpp) package RcppNumerical [QBB+19], and first described in [LN89]. The L-BFGS method is a quasi-Newton method that uses an analytical function and gradient computation but approximates the product of the (inverse) Hessian with the gradient, making use of a short history of updates for the function value and the gradient. The resulting algorithm, described briefly in Section A.2.2 is fast and memory-efficient. We run the optimization based on the default settings for the RcppNumerical function fastLR[3], for a maximum of 100 iterations, and we accept the update to the returned parameters only when the function $\tilde{F}$ increases with the updated parameters.

The only scaling we are not able to bound analytically is the L-BFGS algorithm, though we have found the L-BFGS algorithm scales very well and for up to 100 annotations we find little to no impact on the inference time in practice. We also found that updating the coefficients in this step was unnecessary on every iteration, and instead we set default values of arguments to update the coefficients every 10 iterations for the first 2000 iterations, and every 50 iterations thereafter.

There is a certain amount of flexibility in implementing an algorithm such as this, in particular in the order of the updates for variational posteriors. In practice for the cell scores and gene loadings the updates within a component are independent and we found we gained efficiency by simultaneously updating parameters within a component and looping over the components for each update type. By grouping the updates of all gene loadings (or all cell scores), the vast majority of the computations in the loop could be efficiently precomputed. In practice, in each iteration we update first the cell scores, then the gene loadings, the loadings precision, the noise precision and then the logistic regression updates. In the Poisson model, the pseudodata update precedes all other updates. By default we only calculate the ELBO every 50 iterations.

[3]fastLR is a function in the RcppNumerical package that implements a fast logistic regression with L-BFGS optimization, but does not include the padding terms we have implemented here.

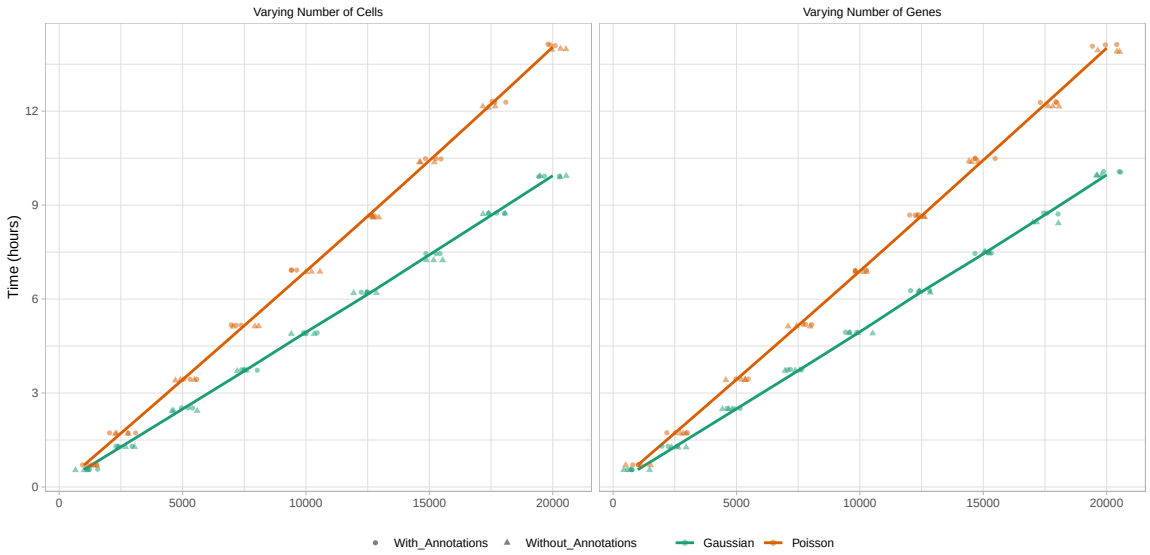

Figure 3.1: **Scaling of the AnnotFA algorithm.** We ran AnnotFA with default settings, including 50 components and the intercept, for 5000 iterations on datasets simulated to scale one of the dimensions while holding the other fixed. We ran 3 initialisations with 50 annotations, and 3 initialisations with a single zero vector as an annotation (labelled as no annotations). In both cases we see AnnotFA scales linearly with the dimension, completes in less than 24 hours, and the annotations are negligible in runtime. To vary the number of cells, we fix the number of genes at $I = 10000$, and run the algorithm on simulated datasets with $J = 1000, 2500, 5000, 7500, \ldots, 20000$ cells. To vary the number of genes, we fix the number of cells at $J = 10000$, and run the algorithm on simulated datasets with $I = 1000, 2500, 5000, 7500, \ldots, 20000$ genes.

CHAPTER 4

Simulations of scRNA-seq data and associated annotations

In Chapter 3 we described AnnotFA, a new variational Bayes method for sparse Bayesian factor analysis incorporating prior information in the form of vectors that inform the inclusion probability for factor loadings. AnnotFA includes implementations of two versions of the model, the first incorporates a Poisson likelihood model for count data, and the second a Gaussian likelihood model. In this chapter we present several simulations to compare the approaches, and to evaluate the impact of including such prior information. We note that although other approaches exist, there are two main modelling approaches to analysing scRNA-seq data in the literature. Namely, modelling the data as Poisson, as suggested by recent analyses [SS20, SVTT18, GI20] and a more common approach that attempts to normalize the data and apply models and methods based on a Gaussian assumption. Moreover, despite other methods providing both Gaussian and Poisson likelihood models, we are not aware of any direct comparison of the two approaches. In this chapter we first simulate count data and compare the two modelling assumptions, and secondly we validate the AnnotFA

Gaussian model on simulated data with a Gaussian noise model.

4.1 Simulations of Count Data

Gene expression is a well orchestrated process, with many genes being expressed due to the binding of certain transcription factors which activate expression of collections of genes and together provide a mechanism for initiation or continuation of biological processes. Beyond this, it is known that certain transcription factors can bind to DNA in such a way as to repress expression. To capture this variation in a simulation for factor analysis, we simulate both a cell scores matrix and a gene loadings matrix in a manner close to the AnnotFA model described in Chapter 3. Corresponding columns of the gene loadings and cell scores matrices constitute one factor, or component, of the underlying pattern of expression. Such a pattern is simulated to be positive and can be considered to be the underlying expression levels of RNA in cells, from which a single cell RNA-seq assay essentially samples.

In this chapter we present several simulation scenarios, with some common threads throughout. In all cases, the annotations in the annotation matrix V are simulated to be non-zero in several relatively small blocks of genes with partial overlap, as well as three more dense annotations (Figure 4.1(a)). From these blocks we assign a value between 0 and 1 drawn from a $\mathcal{B}(3, 2)$ distribution. Following this, we append an intercept vector of 1's, and define an annotation dependency matrix η, such that $V\eta$ is the log-odds of gene inclusion in the corresponding components of the gene loadings matrix. η is defined with an intercept of -10 for all components, reflecting a belief that true biological components depend strongly on annotations and are typically sparse. Each of these initial 8 components is simulated to depend on between 1 and 5 annotations, some are simulated to be very sparse, others to be dense. Several are simulated with overlapping annotations representing the complex interplay of biological

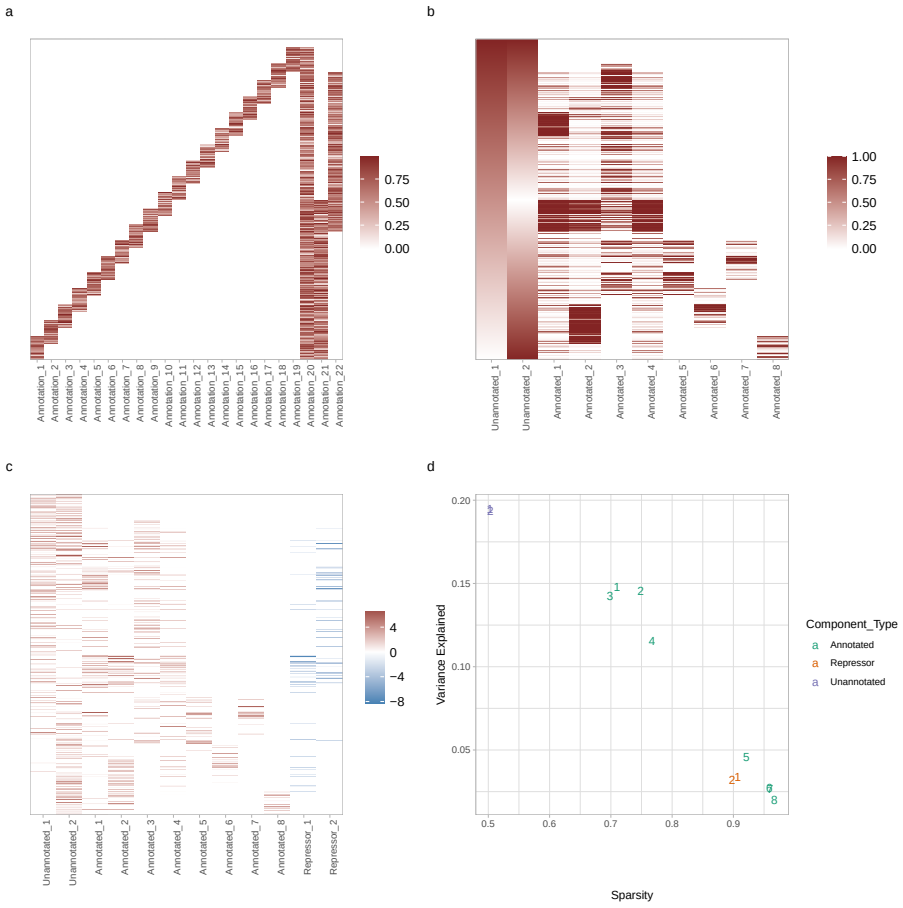

Figure 4.1: **Simulating Gene Expression Matrices.** (a) Example annotation matrix. (b) Example gene loadings probability matrix. (c) Example simulated gene loadings matrix. (d) Plot of variance explained against sparsity, of the simulated components.

drivers (and repressors) of expression. We also append two dense components which are not simulated to depend on the annotations. For an example annotation matrix and corresponding gene loadings probability matrix, see Figure 4.1(a,b).

For each component, the gene loadings are sampled from the product of a Bernoulli random variable using the corresponding column of the probabilities matrix, and the absolute value of a sample from a Gaussian distribution with mean 0 and variance 3. The cell scores are sampled, for each cell and component, from a Gaussian distribution with mean 2 and variance 1, and truncated at 0. This defines cell scores and gene loadings for 10 components, 8 of which depend on annotations.

To further include repressors, we assume that repressor components will act on specific processes or expression patterns and the effect will vary across cells. As such, we simulate cell scores to determine the amount of repression that is active in the cell as positive, and the gene loadings in repressor components to be negative. Specifically, we select two (disjoint) pairs of annotated components to repress, selected to ensure that the intersection of non-zero gene loadings in each pair of components is at least 10% of genes. For each pair, each gene loading of the corresponding repressor component is zero unless both components in the pair have non-zero loading, in which case it is non-zero with probability drawn from a beta distribution with shape parameters 6 and 2, whence the repressor loading is the sum of the respective loadings. For each cell, the repressor cell scores are sampled to be non-zero with probability 0.5, and in this case, the repressor cell score is a proportion, sampled from a $\mathcal{B}(3, 1)$ distribution, of the minimum of the two cell scores.

This results in a 12 component simulation, consisting of 8 annotated components, 2 unannotated components, and 2 repressor components. To simulate a single cell RNA-seq dataset in the form of a DGE matrix, the product of the cell scores matrix and gene loadings matrix AZ^T is used as the underlying pattern matrix, and the rows of this matrix correspond to underlying mean expression levels present in cells. As such, we sample from these under the assumption of independent Poisson samples with means given by the pattern entries, conditional on a library size, which is sampled from the library sizes of the post-QC DGE matrix from [JWR+19] scaled to the number of genes in the simulation[1]. Denoting the library size of cell j by n_j, the simulated mean for each gene and cell pair is hence $n_j \frac{\sum_q a_{jq} z_{iq}}{\sum_{i,q} a_{jq} z_{iq}}$.

To investigate the effect of both including annotations, and the normalisation procedure on signal capture, we simulated 5 datasets with 4000 cells, and 5000 genes following the procedure described above.

[1]This amounts to row-wise normalising the underlying pattern matrix, and sampling from a multinomial distribution

To apply AnnotFA with the Poisson likelihood model, we use the count data y_{ji} as simulated. To apply AnnotFA with the Gaussian likelihood model, we first normalize the library size, by dividing each UMI count by the total UMI count for the corresponding cell, and multiplying by the median cell total UMI count, and then scale reads from each gene to unit variance across cells. For both the Poisson and Gaussian models, we include an intercept component, and run the inference algorithm for 20000 iterations, both with and without annotations. When the model is provided with annotations, these are the simulated annotation matrix as described above, otherwise, the annotation matrix provided is just an intercept term. For each dataset and each type of run we initialise AnnotFA 10 times and select the run achieving the highest ELBO. Throughout we compare to SDA run on the data normalised following the steps described in [JWR+19], first normalizing cell library sizes, then applying the square root variance stabilizing transformation to account for the Poisson mean-variance relationship, and lastly we scale reads from each gene to unit variance across cells. In order to get meaningful results from SDA we found we needed to remove any columns of zeros from the dataset[2], which represent genes that are not sampled in any cell. This is a common step in most scRNA-seq data preparations, but such filtering may limit the ability of the AnnotFA model to leverage the information in these genes across annotations. For ease of comparison, we also remove these columns from the data for the Gaussian AnnotFA results presented here.

4.1.1 Simulation metrics

Since the model is invariant to permutation and sign of components, to evaluate how well components are captured we first align simulated and inferred cell scores. Since the Poisson model, and the Gaussian normalisation procedure both account for library size, the simulated cell scores must be divided by the simulated means row sums for

[2]This is due to the gene-wise noise precision term in the SDA model, which dominated the terms of the optimization.

comparison with the inferred cell scores. More precisely, the cell scores are aligned with a suitably normalized cell scores matrix $\tilde{A}$ such that $\tilde{a}_{jq} = \frac{a_{jq}}{\sum_{i,k} a_{jk} z_{ik}}$ (note that $\tilde{A}Z^T$ is the row normalised pattern matrix). To align the cell scores we thus calculate the absolute correlation of each pair of columns from the simulated cell scores $\tilde{A}$ and the inferred cell scores $\hat{A}$. We iteratively select the highest absolute correlation among the non-zero unassigned components. If $\hat{A}$ has more non-zero columns than the number of simulated components then this returns a subset of the non-zero columns of $\hat{A}$ and a permutation to align the two matrices.

We assess recovery of the cell scores matrix by root mean squared error (RMSE) as in Equation 4.1. Since the model is unidentifiable in terms of sign and scale of components, and RMSE is sensitive to scale, we scale the simulated and inferred cell scores to have unit variance in each component, and reverse sign where the correlation is negative.

$$RMSE(A, \tilde{A}) = \sqrt{\frac{1}{JQ} \sum (a_{jq} - \tilde{a}_{jq})^2} \tag{4.1}$$

To assess how well the gene loadings are recovered, we align the columns of the gene loadings according to the subset and permutation determined from the cell scores. From this, we reverse any scaling of the loadings during normalisation, and consider the absolute correlation of the inferred component posterior mean gene loadings (the product of the posterior inclusion probability, and the posterior mean of the slab), and the simulated gene loadings components.

We further assess the ability of the method to infer the inclusion of genes in the posterior inclusion probability matrix. For each component and each gene, this is a value between 0 and 1 indicating the posterior probability of inclusion $\langle s_{iq} \rangle$ of gene i in component q. We consider the false positive rate (FPR) and true positive rate (TPR) and by varying the threshold we construct a receiver operating characteristic curve for each component, and derive an area under the curve (AUC) statistic for comparison between components, runs, and models.

$$TPR(X, \hat{X}) = \frac{\sum \mathbb{1}(X_{cl} \neq 0 \ \& \ \hat{X}_{cl} \neq 0)}{\sum \mathbb{1}(X_{cl} \neq 0)} \tag{4.2}$$

$$FPR(X, \hat{X}) = \frac{\sum \mathbb{1}(X_{cl} = 0 \ \& \ \hat{X}_{cl} \neq 0)}{\sum \mathbb{1}(X_{cl} = 0)} \tag{4.3}$$

We note that AnnotFA can incorporate annotations, and when these are provided to the model, we compare the simulated coefficients to those inferred under the logistic regression update step.

We also consider the overall R^2 of the model, calculated for a centered simulated matrix Y and centered inferred matrix $\hat{Y}$, as in Equation 4.4. The simulated matrix is the centered matrix of simulated Poisson means, obtained by multiplying the simulated cell scores matrix and gene loadings matrix, and scaling the cellwise sums to be the library size, followed by centering.

$$R^2_{(Y,\hat{Y})} = 1 - \frac{\sum_{i,j}(y_{ij} - \hat{y}_{ij})^2}{\sum_{i,j}(y_{ij})^2}. \tag{4.4}$$

For the Gaussian model, we calculate $\hat{Y}$ by multiplying the inferred cell scores matrix $\hat{A}$ and the transpose of the inferred gene loadings matrix $\hat{Z}$, to obtain the inferred mean, and reverse the normalising transformation that was applied to the count data. For the Poisson model, we calculate the inferred means as $\hat{y}_{ij} = n_j \log(1 + \exp(\sum_q(\langle a_{jq} \rangle \langle z_{iq} \rangle)))$. In both cases, for calculating the R^2 we center the matrix.

To determine the relative importance of simulated and inferred components, we aim to compare the proportion of the variance of the model that is explained by each component. Note that if Y is a data vector, and $\hat{Y}$ is a model based inferred estimate of Y, and both are centered, then equation 4.4 is a measure of the variance explained by $\hat{Y}$. We make use of this as follows: Consider $Y = \hat{A}\hat{Z}^T$, the product of the cell scores and the transpose of the gene loadings matrix, and determine the variance

explained by component i as one minus the variance explained (of Y) by all other components. Denoting the cell scores and gene loadings excluding component q by $\hat{A}_{-q}$, and $\hat{Z}_{-q}$ respectively, we thus define $PVE(q)$ to be $1 - R^2_{Y,\tilde{Y}}$ where $Y = \hat{A}\hat{Z}^T$ and $\tilde{Y} = \hat{A}_{-q}\hat{Z}^T_{-q}$.

4.1.2 Count data simulation results

Since we have evaluated five different models for multiple initialisations on several datasets, throughout this chapter we refer to the AnnotFA models as simply the Gaussian or Poisson model run with annotations or without annotations.

Cell scores

To assess how well and specifically the different methods captured each component across the five datasets, we compared the number of inferred components exceeding absolute correlation of 0.4 with each simulated component. The results are summarised in Figure 4.2(a). We found that the Poisson model consistently missed the repressor components in the highest ELBO runs, though a few initialisations did find them. The Poisson model essentially captures the first 6 components consistently, that is, the 6 components explaining the most variance in the underlying expression pattern (see Figure 4.1(d)). The Poisson model inferred the seventh only once, and split the first unannotated component in one dataset. We note that the Poisson model also inferred all other components to be zero, so in the run where a component was split, this splitting of the component is likely conflating other variation in the data into two correlated components. All non-zero non-intercept components had correlation at least 0.4 with one simulated component. By this measure, both AnnotFA Gaussian models with and without annotations, and SDA captured all of the components. SDA occasionally split the first two unannotated components, and occasionally missed the most sparse annotated components, similarly to the AnnotFA Gaussian model when

run without annotations. The AnnotFA Gaussian model with annotations reliably inferred all except the single most sparse component, which was missed only once, and occasionally split the most variable unannotated component into two correlated components.

We further compare inferred and simulated components of cell scores by absolute correlation directly in Figure 4.2(b). The repressor components (which were not captured by the Poisson model at all) were less well captured by the Gaussian model and SDA in comparison to other components. While SDA and the unannotated Gaussian model captured repressors similarly well, we note that the annotated Gaussian model performed slightly better in this respect. The unannotated components were captured well in all cases (see Figure 4.2(b)), though both AnnotFA models with and without annotations inferred components which exceeded the correlation achieved by those inferred by SDA. The AnnotFA Gaussian initialisations incorporating annotations slightly improve on all other runs in this respect. We note also that Figure 4.2(b) should be interpreted with care as the correlation of the annotated components decreases as they become more sparse (see Figure B.1 for the corresponding plot by component), and the Poisson plots do not include the components that were not captured. We note in particular that the AnnotFA Gaussian model slightly outperforms SDA for all components.

Finally we compare the inferred and simulated cell scores by RMSE. Figure 4.2(c) shows both SDA and the AnnotFA Gaussian models outperforming the Poisson models in most instances, likely due to the Poisson model underestimating the number of components. SDA achieves both the maximum and minimum RMSE, while the AnnotFA Guassian models appear more consistent, and the model incorporating annotations mainly outperforming SDA.

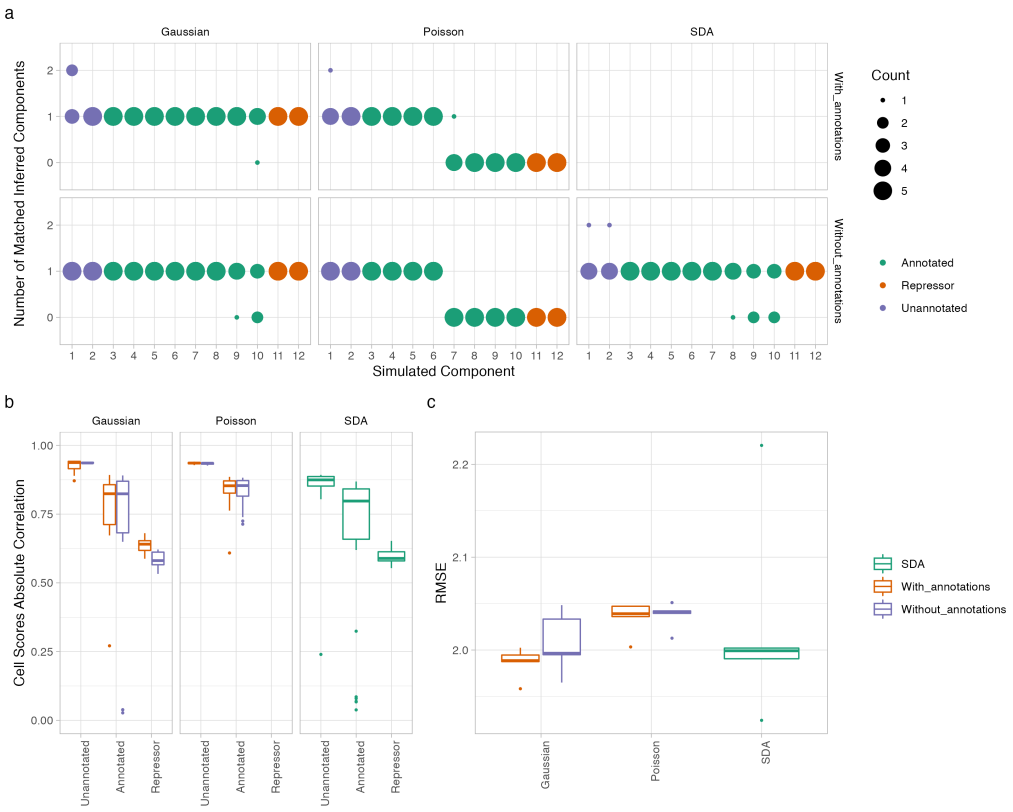

Figure 4.2: **Count Data Simulation Cell Scores.** (a) Counting an inferred component as matching with a simulated component if the correlation of cell scores is at least 0.4, this plot shows the frequency of match counts across datasets, run with and without annotations in both Poisson and Gaussian AnnotFA models, and SDA. (b) Absolute correlation of the inferred and simulated cell scores, with components matched as described in the main text. A further breakdown by component is Figure B.1(a). (c) RMSE of inferred and simulated cell scores, as described in the main text.

Gene loadings

Applying the component alignment derived from the cell scores, the inferred component gene loadings can be compared with simulated gene loadings. For both the AnnotFA Poisson and Gaussian models we observe in Figure 4.3(a) increases in the absolute correlation of inferred gene loadings with simulated gene loadings when incorporating annotations in the model. The AnnotFA Gaussian model clearly outperforms the Poisson model for repressor components and unannotated components when annotations are incorporated, and slightly outperforms SDA. In particular, the repressor components are clearly captured with absolute correlation exceeding 0.9 in the AnnotFA Gaussian model, while these components were not captured at all in the Poisson model. For the annotated components, the AnnotFA Gaussian model performs almost identically to SDA for the 4 most dense annotated components, slightly worse than the Poisson model. For both SDA and the AnnotFA Gaussian model, the correlations increase as the annotated components become more sparse, and the AnnotFA Gaussian model incorporating annotations outperforms both SDA and the Gaussian model without annotations (see Figure B.1(b)).

To compare inclusion of genes in components, we consider the true positive rate (TPR) and false positive rate (FPR) by a receiver operating characteristic curve by varying a threshold on the inferred posterior inclusion probability (PIP) for the gene loadings spike and slab. The resulting ROC curves are plotted in Figure 4.3(b), clearly demonstrating the significant improvement of incorporating the annotations in both Poisson and Gaussian AnnotFA models. This demonstrates again that the Gaussian model is generally capturing the gene inclusion probabilities significantly better than the Poisson model. For annotated and repressor components the Gaussian model without annotations performs similarly to SDA, which is extremely conservative achieving a low FPR, and where components were captured these are similar to the results for the Poisson model without annotations. For both annotated components

and repressor components the AnnotFA Gaussian model with annotations outperforms all models. Among unannotated components, and all models without annotations, the Poisson model achieves the best ROC curve, and the Poisson model with annotations shows slight improvement over the Poisson model without annotations. The AnnotFA Gaussian model appears to perform similarly to, but more consistently than, SDA. Including annotations in the AnnotFA Gaussian model has increased FPR and TPR showing that the model is making some use of the annotations, and is benefiting from the better resolved annotated and repressor components. We note also that the differences in pre-processing are evident here, the Poisson model has more information to resolve inclusion probabilities as in some genes all annotations were zero and the corresponding columns of zeros were included for the Poisson runs only. This may have the effect of inflating the true positive rate and decreasing the false positive rate for the Poisson results. Overall, this clearly demonstrates the utility of incorporating annotations in the model, and the AnnotFA Gaussian results with annotations are exceptionally good.

For all inferred components the model infers logistic regression coefficients. From these we predict the prior inclusion probabilities and compare with the simulated probabilities in Figure 4.3(c). The correlation is near perfect for the annotated components and falls for the Gaussian model on the unannotated components (simulated components which were simulated without annotations). This reflects the poorer inference of the PIPs as discussed and demonstrated in the ROC curves.

Furthermore, since the log-odds of inclusion of a gene in a component is simulated as depending linearly on the annotations, we compare the inferred coefficients from the logistic regression update steps. In Figure 4.3(d) we plot the non-intercept coefficients that are greater than -10. It is clear that the annotations the components depend on are identified as non-zero, and are much further separated from the approximately zero coefficients in the Gaussian model than the Poisson model. Moreover, among

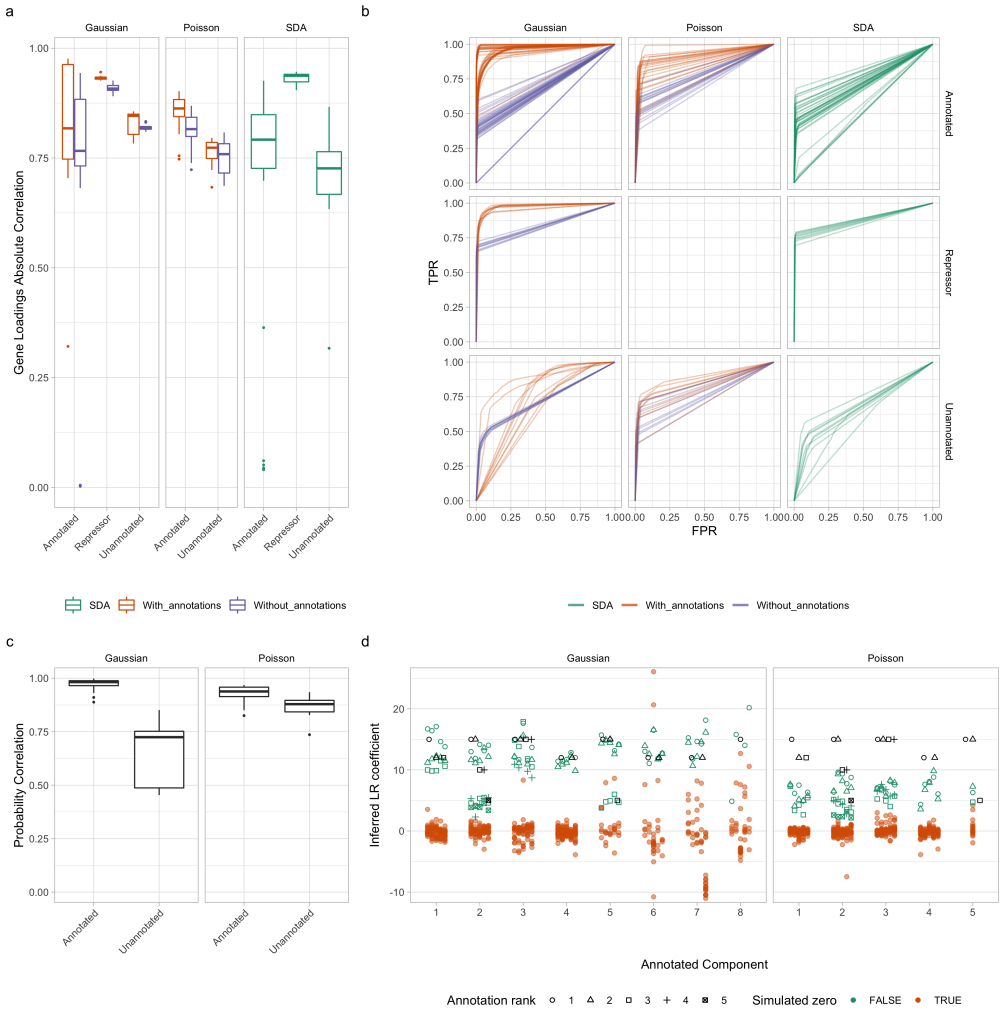

Figure 4.3: **Count Data Simulation Gene Loadings.** (a) Absolute correlation of components of gene loadings matched by cell scores. A further breakdown by component is Figure B.1(b).(b) Componentwise ROC curves for the Gaussian and Poisson model, and for SDA. See Figure B.2 for the corresponding AUC. (c) Correlation of the predicted prior inclusion probabilities derived from the inferred logistic regression coefficients. (d) Comparison of simulated and inferred logistic regression coefficients. The x-axis labels indicate the index of the corresponding annotated component, we plot the results for each of the results from an initialisation achieving the highest ELBO for each dataset. Each horizontal position corresponds to one dataset and one component. Green colour indicates coefficients simulated to be non-zero while orange indicates coefficients simulated to be zero. Shape corresponds to rank of the simulated coefficient, and the simulated values are plotted in black.

annotated components the Gaussian model does very well at identifying the non-zero coefficients. For component 5, which is very sparse but depends on three annotations, the model incorrectly incorporates several annotations as non-zero. The Poisson model only identified the first 5 annotated components and the truly non-zero coefficients were highest ranking in all results. However, the separation of the zero and non-zero coefficients is less clear, reflecting the poorer ROC curves.

Inference of expression means

To compare the variance explained by the Gaussian and Poisson models, we compare the R^2 of the simulated means and inferred means, after reversing the normalisation applied to the DGE matrix for each method. The results are presented in Figure 4.4. We note that the Gaussian model outperforms the Poisson model in R^2. We found the R^2 to be consistent across datasets and across runs with or without annotations (see Figure 4.4), and the AnnotFA models outperform SDA. However, while the Poisson imputed means appear to be unbiased, the Gaussian model does show signs of bias (Figure 4.4(b)), systematically underestimating the larger simulated means. We note that in all simulations incorporating Gaussian noise that we have investigated in the course of this work, including the simulations presented in Section 4.2, the AnnotFA Gaussian model infers unbiased estimates of the simulated means. We therefore believe the bias evident here is an artefact of the normalisation procedure, and may indicate that the Poisson mean-variance relationship has not been fully accounted for.

In conclusion, despite the data normalisation procedure, the Gaussian model captures more subtle components and outperforms the Poisson model in terms of inclusion of genes in components and capture of the gene loadings. Moreover, the Poisson model has underestimated the number of components, missing true components that explain variance in the data. We note that despite the poor performance in capturing the weaker components, the Poisson model nevertheless does capture

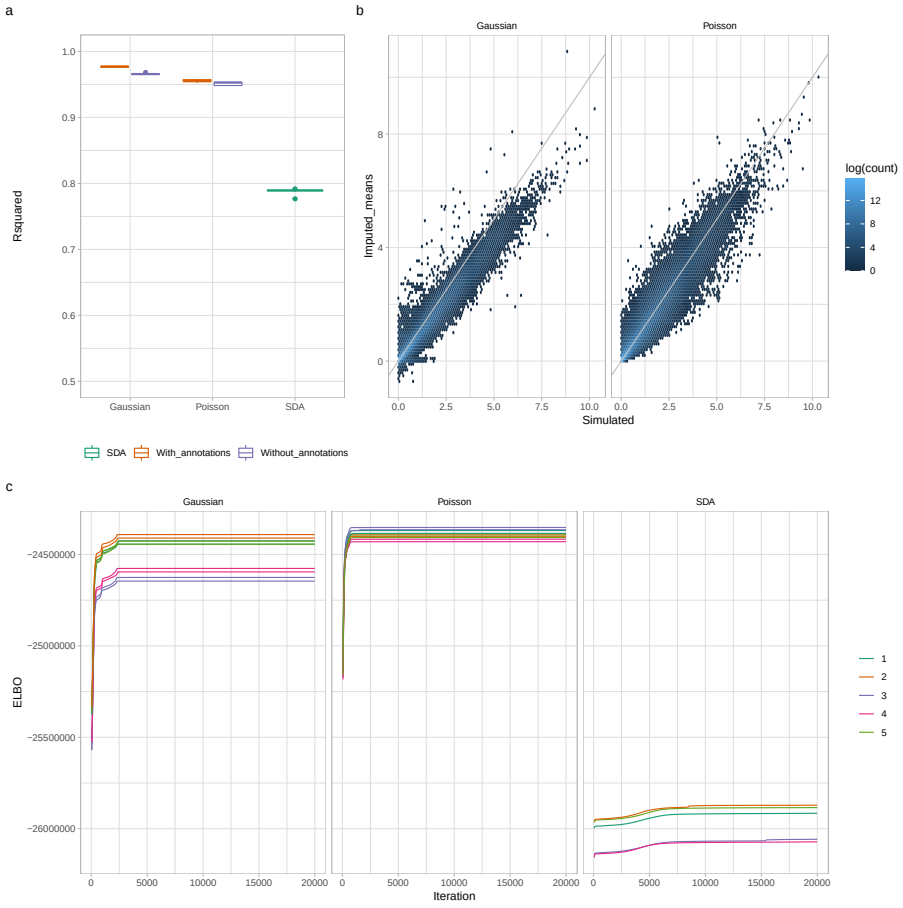

Figure 4.4: **Count Data Simulation R^2.** (a) R^2 for each model as described in the main text. (b) Example plots of imputed means vs simulated means for both models. The Gaussian imputed means have been transformed by inverting the normalisation transform applied to the DGE matrix. (c) ELBO traces for the initialisations achieving highest ELBO. Colour indicates simulated dataset.

meaningful components well.

The Poisson model is inferred by means of a Gaussian approximation to the likelihood, and the inference algorithm effectively fixes the cell-wise precision and iteratively changes the data being fitted to. This is likely obfuscating the more sparse components that explain less variance in the model. Furthermore, we note that the rate function involved in the Poisson likelihood model is non-linear near zero but linear for large values, which can impede the algorithms ability to fit to the data through an underlying linear factor analysis model. To illustrate this, we note that the interpretation of a loading in the Poisson model is conditional on the cell scores and loadings for all other components. For example, if all other components reconstruction from cell scores and gene loadings is negative or close to zero in a particular cell and gene (before application of the rate function λ), then any small or negative additional component will contribute to the inferred mean in a region in which the rate function is well approximated by the exponential $x \mapsto \exp(x)$, so multiplicatively. Whereas, if all other components reconstruction is sufficiently large, then the contribution will be in a region in which the rate function is approximately linear. Moreover, this could vary between cells for a particular gene. Besides the difficulties in interpreting results that this causes, we suggest that this may also account for the Poisson models inability to infer more than the coarsest componentwise structure in the data.

Considering the challenges in interpretability posed by the Poisson model, as well as the inherent approximations in the inference procedure and significantly slower inference times in comparison to the Gaussian model, and that normalisation is well-studied and routinely applied to scRNA-seq data, we recommend the use of the Gaussian model over the Poisson model in practice.

4.1.3 On the effect of the gene-wise padding and library size scaling

Having demonstrated in the previous section that incorporating the annotations in the model is effective, we briefly demonstrate that for the Poisson model including annotations, padding the logistic regression update has had a positive impact on the results observed. We also consider the effect of the combination of library size scaling and padding on the Poisson model.

We initialised the Poisson method with a range of scaling values, that is, scaling the library size by $\zeta = 1, 5000, 10000, 15000, 20000$, and 25000, and for each scaling choice, we initialised with padding of either 0 or 50000. The R^2, as plotted in Figure 4.5(a) clearly demonstrates that the scaling of 5000 with padding is the best choice of parameterisation. Moreover, when compared with padding of 100000, we see that the results are stable as the padding increases. This is similar to results we have seen in other scenarios, that it is necessary to use some padding, at least as large as the number of genes in the dataset, but the precise choice is not important. We also plot the sparsity of the inferred components in Figure 4.5(b), which shows the components are dense unless padding is used. For the count data simulations we observe the same behaviour for the Gaussian model, whereas for the Gaussian simulations in Section 4.2, we find padding is not needed.

4.2 Gaussian Data Simulation

In this section, we consider the performance of the AnnotFA Gaussian model on data simulated with a Gaussian noise model. This section serves to validate the Gaussian model. We simulate data in a manner similar to that in Section 4.1, with a range of sparse and dense components, use an annotation matrix simulated in an identical fashion to the count data simulation. For each simulation we include 2000 cells and

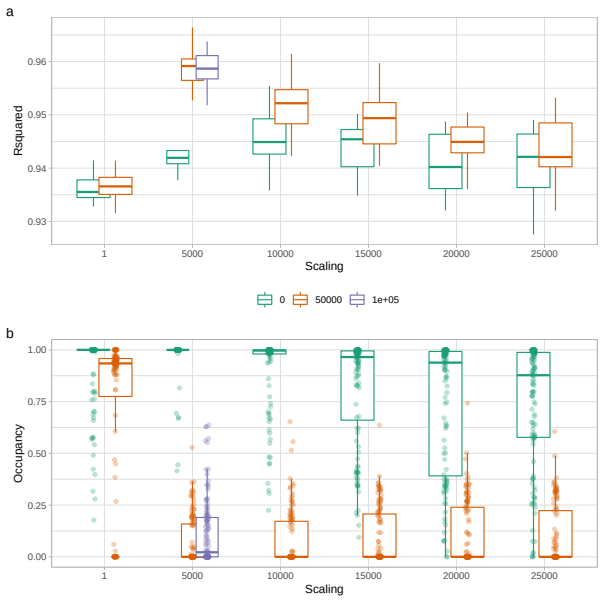

Figure 4.5: **Varying scaling and padding.** (a) R^2 for a range of scaling and padding values for the Poisson model on the dropseq simulated data. (b) Occupancy, defined as the proportion of PIPs greater than 0.5, of inferred components for a range of padding and scaling values

5000 genes, and 10 components. The components included here correspond to the unannotated and annotated components. The cell scores are simulated from $N(0, 1)$ and the gene loadings are simulated to be included in a component with probability given from the probability matrix, and if so, then the value is sampled from a standard Gaussian distribution $N(0, 1)$. Gaussian noise is added with a cell-wise variance sampled from a Poisson distribution with mean 20. We simulate 5 such datasets, and run the Gaussian model with no intercept but at a range of logistic regression padding values, namely $0, 20000, \ldots, 100000$. We run the Gaussian model with and without annotations for 10000 iterations, and SDA for 30000 iterations, as it took longer to converge, see Figure 4.6(e). All models were run with 20 components, double the simulated number.

4.2.1 Gaussian data results

We compare the results of the AnnotFA Gaussian model with and without annotations against SDA. The best results we see for the AnnotFA Gaussian model in this scenario are without padding the logistic regression updates, so we focus primarily on this model here. In Section 4.2.2 we briefly discuss the effect of the padding in this setting.

Applying a componentwise PVE threshold of 0.005 for inclusion, all methods returned the correct number 10 non-zero components. On decreasing the threshold to 0.001 SDA increased to 20 while the AnnotFA Gaussian model maintained 10 components when run with or without annotations, demonstrating a clearer determination of the underlying components.

All models captured all components to absolute correlation at least 0.9 for each component cell scores. Although the aligned cell scores agreed with the simulated cell scores almost identically across different models, the correlation of inferred components gene loadings with simulated loadings were improved for the AnnotFA Gaussian model with annotations, compared to SDA and to the AnnotFA Gaussian model without annotations. We note that the AnnotFA Gaussian model without annotations slightly exceeded SDA in this measure (Figure 4.6(c,d)).

Similarly the AnnotFA Gaussian models outperformed SDA in terms of true positive rate, achieving higher AUC than SDA both with and without annotations and performing excellently with annotations (Figure 4.6(a,b)). We note however, that SDA appears to be more conservative on false positive rate as discussed in Chapter 1.

4.2.2 Effect of padding on the Gaussian model

We also initialised the AnnotFA model with a range of padding levels between 0 and 120000. The trend is well represented by padding levels 0, 40000, 80000 so we restrict our attention to these values here and summarise some important aspects of the results in Figure 4.7.

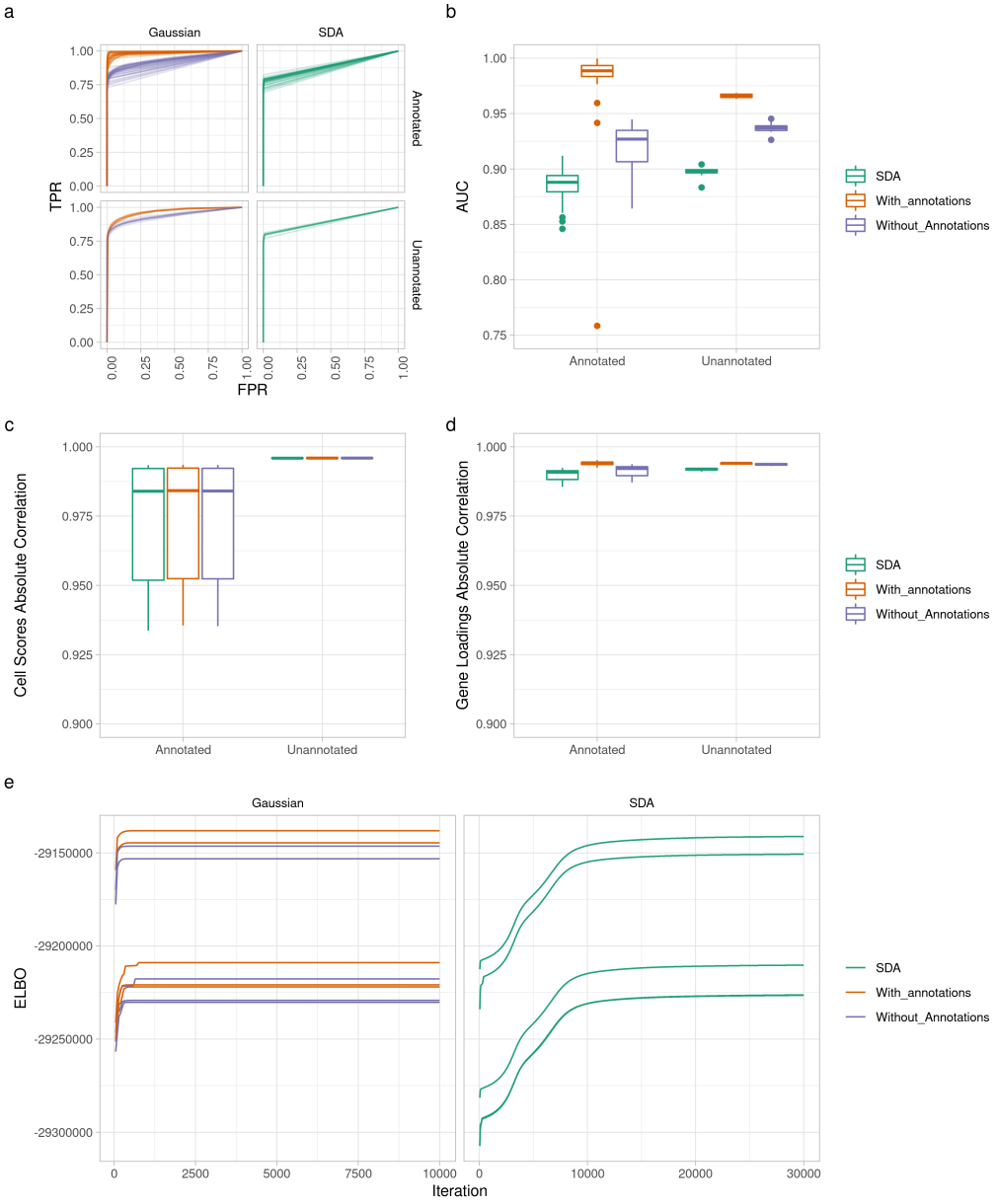

Figure 4.6: **Gaussian Simulation with zero padding.** All plots in this panel are for zero padding runs, and the highest ELBO initialisation for each dataset. (a) Componentwise ROC curves, for components aligned using the cell scores separated by model and component type. (b) Area under the curve (AUC) for the ROC curves in (a). (c) After alignment of components, the absolute correlation of of each cell scores component with the corresponding simulated component. (d) As for (c), Absolute correlation of inferred gene loadings with simulated gene loadings following alignment using cell scores as described in the text. (d) ELBO traces for the different datasets and models.

For a threshold of 0.002 on the componentwise proportion of variance explained, SDA and the AnnotFA Gaussian model with and without annotations, all captured the correct number of components when run without padding. As padding increased, the number of components passing this threshold increased to 11 for one out of 5 initialisations of the AnnotFA Gaussian model with annotations, and to 13 or more components when run without annotations (Figure 4.7(b). This is reflected in the number of components with absolute correlation greater than 0.4 between cell scores with the simulated components in Figure 4.7(c). This shows that for the runs with annotations, increasing the padding has led to splitting the unannotated components, while for the annotated components (those simulated to depend directly on the annotations) exactly one component is captured in each run. For the runs without annotations, both the annotated components, and the unannotated components can be split into more than one component. Figure 4.7(a) shows that for 0 padding the ROC curves for all components were very high, with AUC close to 1, whereas the SDA and runs without annotations are extremely conservative but achieve lower TPR. The unannotated components are captured slightly less well in the runs with annotations, than the annotated components. Moreover, as padding increases the runs with annotations maintain similar ROC curves while the runs without annotations have much lower ROC curves, with decreased TPR. Finally, the absolute correlation of aligned cell scores and gene loadings (Figure 4.7(d,e)) were all very high for cell scores, with the runs with annotations slightly outperforming the runs without annotations in all cases, and outperforming SDA when run without padding. For 0 padding the same is true for gene loadings, but the absolute correlation of gene loadings dropped significantly when the padding was increased and annotations were not included in the model.

We conclude that our model outperforms SDA on all the simulations we run here, the prior structure is making excellent use of the annotations which improve

the inferred parameters that are available in other models, and provides additional information as to which annotations are informative for each component.

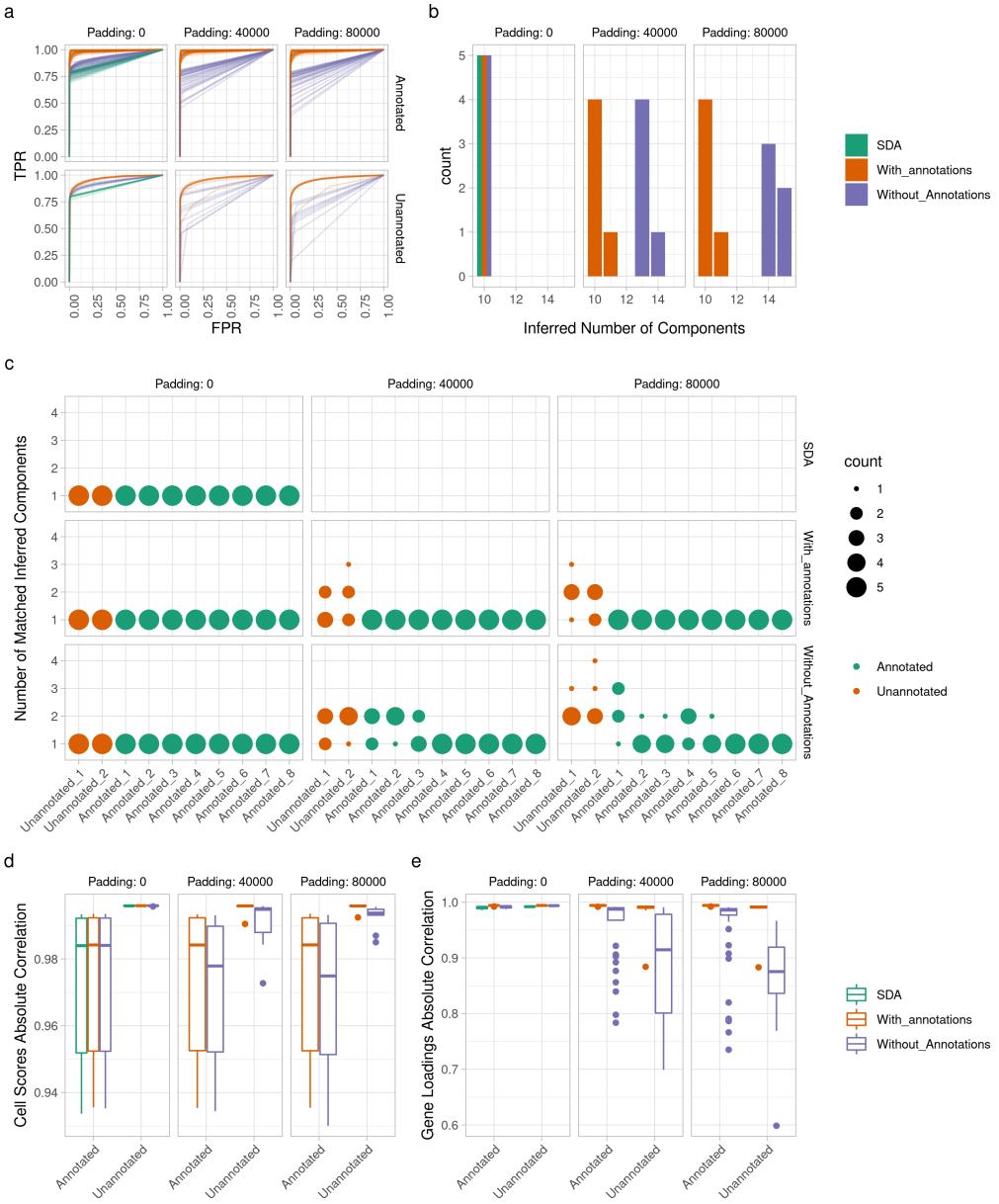

Figure 4.7: **Gaussian Simulation with range of padding.** (a) ROC curves separated by component type and the level of padding applied during inference. (b) Number of inferred components exceeding the threshold on the proportion of variance explained of 0.002. (c) Number of components with absolute correlation of cell scores exceeding 0.4 with simulated components, counted across the 5 datasets, for the best run at each padding level. AnnotFA has split the unannotated components when padding was used, and split the annotated components when annotations were not provided, as described in the main text. (d)(e) Absolute correlation of matched component cell scores and gene loadings

CHAPTER 5

Application to scRNA-seq data

In this chapter we apply AnnotFA to an existing scRNA-seq dataset previously studied in [JWR+19]. This is a dataset consisting of cells from mouse testis, a heterogeneous tissue consisting of somatic cells and germ cells undergoing spermatogenesis, a highly dynamic process that operates continually in mature male mammals and is essential for evolution and fertility, and maintaining diversity. In this chapter we present the key results from applying AnnotFA to this dataset with vectors of binding affinities of transcription factors to promoter regions of genes as the provided annotations informing gene inclusion in components. In particular we find that the components capture broad stages of spermatogenesis, highlighting in particular meiosis and mutant pathology. We demonstrate that new components are found, examine inferred components across initialisations to demonstrate that most are reproducibly inferred, and show that the algorithm makes use of the annotations provided to inform gene inclusion in the inferred components. We highlight the important annotations in a family of components, and the complexity of the dependence on annotations. Furthermore, we

discuss the observed patterns of inferred prior and posterior inclusion probabilities across the developmental trajectory present in these cells.

5.1 Biological Background

Spermatogenesis is the process by which male gametes, named spermatozoa, are produced. In mammals this takes place in the seminiferous tubules of the male testes. Broadly, spermatogenesis can be understood as a three stage process [dKLM$^+$98]. The first stage is mitotic division of spermatogonial stem cells into type A and B cells. The former of which serve to replenish the stem cells, and the latter of which, known as primary spermatocytes, have a complete diploid genome, and are committed to the second stage of spermatogenesis, undergoing meiosis. Through meiosis, each such cell results in four daughter cells, each of which has a full haploid genome. These haploid spermatids then undergo the third stage, spermiogenesis, during which the immature spermatids mature to functioning, haploid spermatozoa. During spermiogenesis, a tail is formed by developing microtubules to form an axoneme, DNA is repackaged around protamines [YHF$^+$18], becoming highly condensed and transcription is silenced. A number of reorganisations then take place including the formation of the acrosome, forming mature but immotile spermatozoa. Cells completing spermatogenesis have been in contact with somatic Sertoli cells throughout the process, before finally being released into the lumen of the seminiferous tubule, transported to the epididymis where the now mature spermatozoa gain motility. Each of these stages is strongly controlled with various biological checkpoints for which failure leads to apoptosis.

Meiosis, and in particular recombination, is the basis of Mendelian inheritance, providing a mechanism to maintain diversity in the population and is typically separated into meiosis I and meiosis II. In meiosis I during Prophase I, maternal and paternal homologous chromosomes pair and exchange information in homologous

recombination, before the first division in which two haploid cells are produced (each containing a single sister chromatid pair of each chromosome). Prophase I is further separated into Leptotene, Zygotene, Pachytene, Diplotene [dKLM$^+$98]. The first stage of prophase I is Leptotene, meaning "thin threads" [SSJ97], where the synaptonemal complex begins to form, and double strand breaks are created by SPO11. Leptotene is followed by Zygotene, meaning "paired threads" [SSJ97], also known as the "bouquet stage" [LJ19] during which the telomeres are clustered, the synaptonemal complex is formed and homologous chromosomes become synapsed. Following Zygotene is Pachytene, meaning "thick threads" [SSJ97]. This is the stage at which double strand breaks are repaired and chromosomal crossovers result in homologous recombination. Following Pachytene is Diplotene, meaning "two threads" during which the synaptonemal complex is disassembled and homologous chromosomes separate except at crossovers where they are connected by chiasmata. The remaining stages of meiosis I are Metaphase I, Anaphase I, Telophase I. During these stages, a complex interplay of processes involving microtubules reshape the cell, elongating it, and also pull the homologous chromosomes apart while cohesin is cleaved but remains on the centromeres [Pie16], allowing the separation of chromosomes (while sister chromatids remain connected). This allows for the pairs of sister chromatids to be separated to opposite poles and the cell to cleave creating two cells.

Meiosis II is the second meiotic division, separating sister chromatids, resulting from centrosomes moving to opposite poles and spindle fibers attaching to opposite centromeres (at the kinetochores), the cohesin connecting the sister chromatids is cleaved and the sister chromatids separate. At this point the chromosomes lengthen and decondense, nuclear envelopes form and cells separate producing four haploid daughter cells.

These processes are highly dynamic and require a specific transcriptional program and environment. In particular, spermatogenesis is the main function of the testes,

which have been found to have the largest number of tissue specific genes [JWR+19], and to share an unusual transcriptional similarity with the cerebral cortex [GHS+05]. Moreover, cells undergoing spermatogenesis are the only cells in the male body with sex chromosome inactivation [YM09].

5.2 Single Cell RNA-seq Data Preparation and Quality Control

The data we study here has been previously analysed using the SDA factor analysis approach in [JWR+19]. The dataset initially consisted of 57,600 cells from the testes of wild-type mice and mice with gonadal defects due to disruption of the genes Mlh3, Hormad1, Cul4a or Cnp. The cells studied here had sequencing libraries generated according to the drop-seq protocol described in [MBS+15] (the reader is referred to [JWR+19] for further details). Following sequencing processing, filtering and alignment as described in [MBS+15], DGE matrices were generated for each experimental batch and combined. Cells with fewer than 200 UMI counts or fewer than 50 genes expressed were removed, as were cells for which the total UMI count, or number of genes expressed were more than one standard deviation below the mean for that experiment. A further homogeneous group characterized by low library size, and high mitochondrial gene expression as well as coexpressed genes from early and late meiosis were also removed on the suspicion of poor quality or doublets. After these steps 20,322 cells and 28,893 genes remained. This post-QC DGE matrix is the start of our work. We normalize the library size to the median for the dataset, dividing each entry of the DGE by the library size for that cell and multiplying by the median library size. Secondly, we remove all genes in the lower third of expression means and normalize the genes to unit variance. The final matrix (the normalised DGE matrix) consists of 20,322 cells and 19,262 genes. These cells were jointly analysed in [JWR+19], and

we take a similar approach here relying on the inherent features of a factor analysis model to identify technical artefacts, spermatogenic gene regulatory components, and transcriptomic signatures of the mutant pathologies. Throughout this chapter we draw comparison with the previously published SDA analysis of [JWR+19], and adopt the convention that those components are identified with a "V" prefix throughout the text.

5.3 Annotations

A key feature of AnnotFA is the integration of external data in the form of annotations that may represent biological heterogeneity and inform inclusion of genes in a component. For this first application of the method we have used a set of transcription factor binding vectors from [JWR+19, Fig. 7B]. These were generated using motifs from the Hocomoco database [KVY+17], subject to some thinning, using the motifFinder software first described in [DHA+16], to assign to each gene and transcription factor a probability of binding to the promoter region. Due to availability of the transcription start site annotations for certain genes, we had information for only 18,513 genes of the 19,262 genes in the dataset, so we further filter the DGE matrix to these genes.

In order to remove collinearity in the annotation matrix we iteratively selected the transcription factor vector that had absolute correlation greater than 0.7 with the maximum number of other vectors, and removed all annotations that correlated beyond the threshold 0.7 with it. This procedure resulted in thinning from 1276 vectors to 469 transcription factor binding vectors that we include in the model as annotations and which we regard as representing groups of correlated transcription factors.

5.4 Application of AnnotFA

Considering the results of Chapter 4 we apply the Gaussian model to a normalised expression matrix.

Since variational inference can be sensitive to initialisation, we initialised each of five runs of AnnotFA with different seeds and 50 components and ran the algorithm for 20000 iterations. To assess convergence we considered the change in ELBO, the change in proportion of PIPs below 0.5, and also the correlation of each components' gene loadings and cell scores between successive checkpoints 2000 iterations apart. In all cases we saw that 20000 iterations was sufficient for convergence, see Figures 5.1 and B.4. We later initialised at the same seeds running to 50000 iterations as confirmation and saw minimal change, confirming that the algorithm has converged.

5.5 Inferred Components

We selected the initialisation achieving the highest ELBO, run 5, for further downstream analysis and interrogation of components. We note that most components in the results of each run have relatively low correlations. See Figure 5.3(a) for the components from run 5. The intercept component was usually the last component to converge to its final state (this component is qualitatively the largest difference between the cell scores correlation and gene loadings correlations in Figure B.4). We adopt the convention that non-intercept components from run 5 will be referred to with a "C" prefix. Some initial observations on the non-intercept components are presented in Figure 5.2. The posterior mean of β_q correlates well with the variance of the loadings for each component, and many of the most variable components are predominantly one-sided. That is, the large loadings are mostly positive or mostly negative. This also holds for the cell scores for which the within component variance is more stable across components, consistent with the model. Moreover, most components are sparse, with

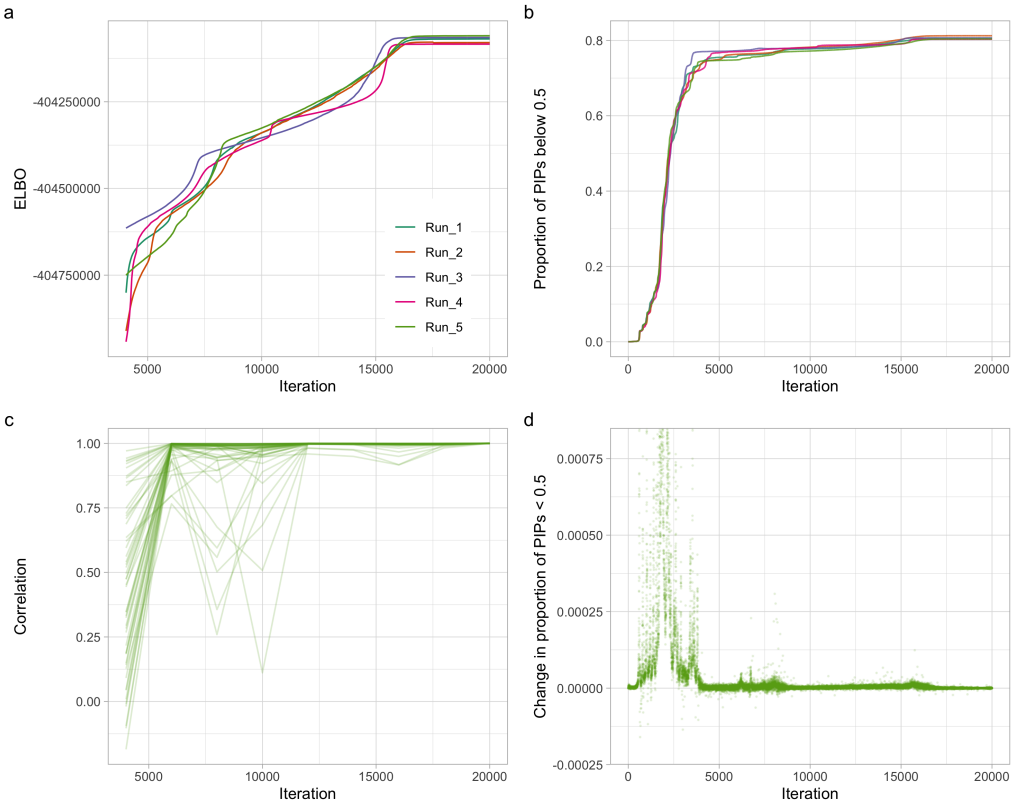

Figure 5.1: **Convergence of AnnotFA.** (a) ELBO trace for five initialisations over 20000 iterations, showing clearly that they have converged and Run 5 has achieved the highest ELBO. (b) Trace of the overall sparsity of the model captured as the proportion of PIPs below 0.5 across 5 initialisations.(c) Componentwise trace of the correlation with the corresponding component in the preceding 2000 iteration checkpoint as described in the main text. (d) Change in proportion of PIPs below 0.5 per iteration in Run 5.

the majority of components having less than 20% of PIPs over 0.5 (Figure 5.2(b)). The majority of PIPs are close to zero or one (this is clear from the slope of the component points in Figure 5.2(c)), and the spike and slab structure of the loadings has captured much of the gene inclusion structure in the components (Figure 5.2(d)).

Whilst a factor analysis approach can provide a useful dimension reduction, inferring components of co-expressed genes in the gene loadings and their respective expression profile in cell scores, care is still required when interpreting the components. To address this, we make use of the cell scores and gene loadings from the previous analysis from [JWR+19], which we label as "SDA" components. To distinguish in the

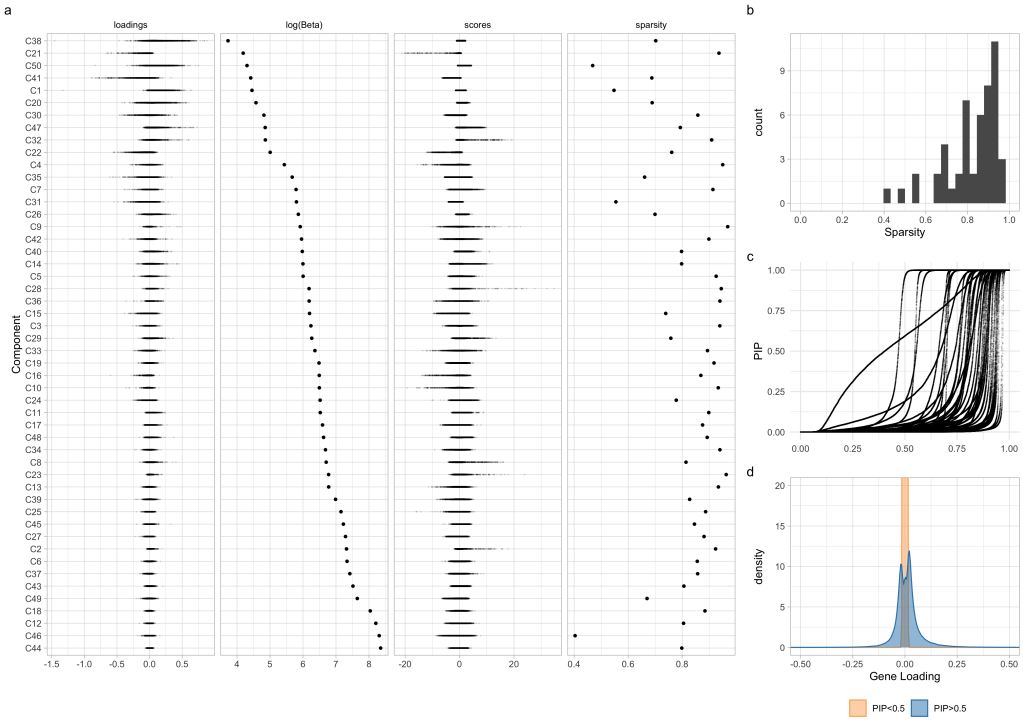

Figure 5.2: **Inferred Components** (a) Scatter plots by component of the posterior means of gene loadings, cell scores, the log-transformed posterior mean of β_q, and the component sparsity. Here, we define sparsity to be the proportion of PIPs below 0.5. (b) Histogram of component-wise sparsity for AnnotFA run 5. (b) PIPs ordered left to right by magnitude within each component, showing all 50 inferred components. (d) density plot of the gene loadings, coloured according to the gene being included in the component.

text, we will henceforth follow the notation from the figures of [JWR+19], referring to the SDA components with a "V" prefix. These SDA components are extremely useful for comparison and to get a sense of the developmental stage of the cells involved. For a direct comparison of run 5 components and SDA components we plot a heatmap and scatter plots in Figure B.5 showing clearly that many of the SDA components are well correlated with AnnotFA components, while approximately 20 components are not well matched by AnnotFA components and vice versa.

We find that most of the AnnotFA components are well conserved between initialisations and that component correlations between AnnotFA initialisations are often higher than those with SDA components (Figure 5.3(b)). In particular 12 of the inferred AnnotFA components had gene loadings with maximal correlation with an SDA component less than 0.5, suggesting these are novel components of co-expressed genes. The intercept component consistently captured the gene-wise mean normalised expression (Figure 5.3(c)), similar to observed behaviour on simulations. This component therefore has an effect equivalent to centering the columns of the data matrix and running without an intercept. We consider that most of the differences between the AnnotFA and SDA results are due to the inclusion of annotations in the inference, and have compared the prior and posterior inclusion probabilities of genes in components between SDA and AnnotFA components in Section 5.6. In particular, the AnnotFA components achieve similar levels of sparsity to SDA components, but show evidence of having inclusion probabilities strongly informed by the fitted prior derived from the annotations. Viewing the prior annotation informed inclusion probabilities as predictors for posterior gene inclusion, AnnotFA components achieve AUC ranging from 0.67 to 0.97 (mean 0.8), while SDA components achieve AUC ranging from 0.65 to 0.77 (mean 0.69) for SDA. We describe this in more detail in Section 5.6.

Since post-multiplication of the cell scores matrix by an orthogonal matrix and corresponding pre-multiplication of the loadings matrix results in an identical re-

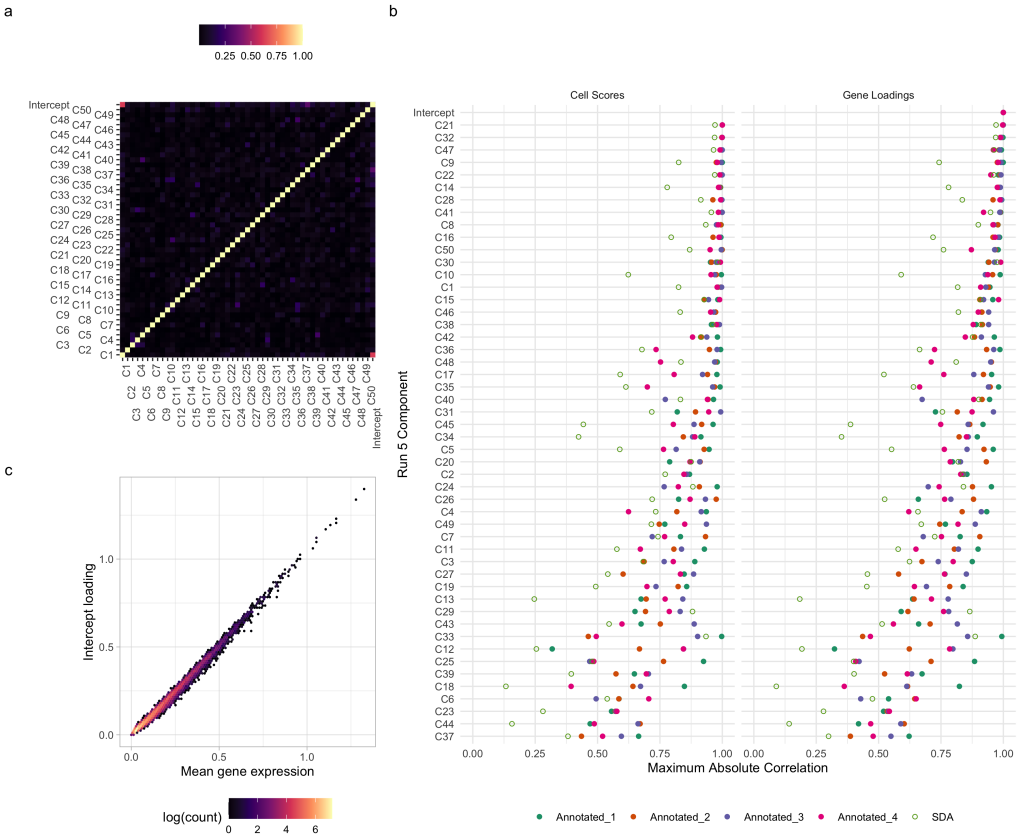

Figure 5.3: **Correlation between components** (a) Between component (absolute) correlation matrix of AnnotFA run 5 cell scores. (b) For each AnnotFA component we plot the maximum absolute correlation with a component from each of the other runs, and the SDA components. The left panel captured the cell scores correlation and the right panel the gene loadings. (c) Plotting the intercept component gene loading against the mean gene expression across cells for each gene.

construction, explaining the same variance in the data, we must consider whether this is the relationship between different factor analysis results here. To address this, we make use of the Procrustes transformation, an orthogonal transformation (i.e multiplication by an orthogonal matrix) of one matrix in such a way as to minimize the sum of squares differences with another matrix. We have calculated such transformations of cell scores matrices towards both the SDA scores matrix and the run 5 scores matrix and plotted the resulting correlations between components in Figure B.6. As expected, since informing sparsity in the loadings goes some way to addressing this orthogonal invariance, the conclusions are similar to those from

direct comparison of components. The vast majority of AnnotFA components were well conserved between runs of AnnotFA and better correlated between AnnotFA runs than with the SDA components. The post-Procrustes transformed AnnotFA runs correlation with the SDA components revealed a number of SDA components $V8, V46, V14, V50, V4, V1, V10, V16, V3, V7$ that were particularly poorly matched with AnnotFA components (correlation less than 0.25). It is noted in [JWR+19] that $V1, V4, V8, V14$, and $V46$ are expressed in only one or very few cells, and are hence described as single cell components. $V50$ differentiates stem cells, $V10$ is potentially a sertoli cell component, $V16$ is labelled sertoli (rare) $V3$ marks T lymphocytes, and $V7$ marks differentiating early stage spermatogonia. With the exception of $V7$, these components appear to be evident in relatively few cells in the population, and often with relatively low cell scores, suggesting that they explain little variance. These were also the lowest correlated with the untransformed cell scores matrices in Figure B.5. Comparing the sets of SDA components that were highly correlated with AnnotFA components and those with procrustes transformed AnnotFA scores matrices reveals that most of the highest correlated components from the latter are represented in the highest ranked from the former set. However, two components stand out, components V5 and V34 have Procrustes correlation between 0.931 and 0.943 and between 0.95 and 0.966 with components from AnnotFA initialisations respectively, while they have largest correlations -0.779 and 0.733 respectively with components C14 and C4 from Run 5. Inspection of the rotation matrix shows that Component V5 has been expressed as a linear combination with coefficients for C12, C14, C15, C35 all with modulus greater than 0.1 and all other coefficients below 0.1, while V34 has three most prominent coefficients corresponding to C3, C4, and C5. The orthogonal transformation of the SDA cell scores to the AnnotFA cell scores is the transpose of the one previously calculated. From this we note that Component C14 achieves a correlation of 0.93 with a component of the procrustes transformation of the SDA

cell scores, while the maximal component-wise correlation was 0.779 (with component V5). The largest coefficients in the procrustes transformation matrix were -0.83 (V5 (pre)Leptotene), 0.35 (V44 Zygotene), -0.23 (V23 pre-pachytene & Hormad1), and 0.21 (V2 B Spermatogonia), and -0.2 (V38 X activation (Hormad1)). These components are all early stage meiotic components and go some way to describing some of the differences between C14 and V5 that we will discuss in a later section. We note in particular that C14 is strongly informed by annotations, with inclusion probabilities achieving AUC of 0.87, while V5 has AUC of 0.72, and each of V44, V23, V2 having AUC less than 0.74 (See Section 5.6 and Figure 5.13 for further details). Similarly C12, C14, C15, C35 have AUCs of 0.87, 0.87, 0.81, and 0.86 respectively. This suggests that the AnnotFA model has explained the variance in this portion of the data in a manner consistent with the prior information provided. Since the annotations used here are biologically relevant for expression dynamics, we suggest that the annotation informed components are likely more biologically representative.

The results of [JWR+19] included separating somatic cells and cells undergoing spermatogenesis, with the latter being assigned a pseudotime ordering based on a principal curve fitted to a t-SNE dimension reduction of the inferred cell scores, on which cells are ordered from early stem cells through to mature spermatids in a clockwise direction. To get a sense of how the pseudotime ordering may be similar if inferred from our components, we calculated UMAP and t-SNE dimension reductions from components C1 - C50 cell scores, after removing components that appeared to be batch effects, or whose large loadings were dominated by pseudo genes. Such components were C2, C7, C10, C15, C17, C19, C23, C24, C27, C36, C45, and C48. UMAP dimension reductions were calculated using the R package umap [Kon20], and t-SNE reductions using the R package Rtsne [Kri15, vdMH08, van14], in both cases without a PCA initialisation. Being stochastic methods, results are dependent on the random seed. However, we found the results were generally qualitatively similar

and we provide some representative dimension reductions in Figure B.7, coloured by the inferred pseudo time from [JWR$^+$19]. The UMAP figures (plots (c) and (d)) are characteristically similar in shape to each other and qualitatively similar to the t-SNE plot (a) from [JWR$^+$19], showing only a few cells that we might likely place at a later pseudo time. The t-SNE shown (b) similarly shows cells on a more or less continuous trajectory again showing a few cells which we might infer as later stage. As such, and considering the previous analysis based around the published t-SNE figure, we decided to make use of the t-SNE coordinates and pseudotime from [JWR$^+$19] to inform further downstream analyses. Many inferred components show clear patterns of expression over pseudotime. This is representative of, as expected, transcriptomic signatures of a highly dynamic well orchestrated biological process. For a subset of these components, see Figure 5.4(a-e).

To collectively visualise and order all inferred components over pseudotime, we employed a two stage process. We first restricted to cells which have pseudotime, and assigned to each component a weighted average pseudotime, where the weights are the absolute value of the corresponding cell scores, including only those cell scores with absolute value greater than 1. However, a number of components are more naturally considered to be somatic components, being primarily active in somatic cells, which do not have an inferred pseudotime. To identify somatic components, we included somatic cells by ordering them by library size, and extended pseudotime to be negative for these cells (to distinguish them from the non-somatic cells), and recalculated the weighted average. This placed ten components below all other components, with negative weighted average pseudotime. We have thus labelled these components as somatic and we confirmed the labelling by inspecting the cell scores plotted on the t-SNE coordinates (Figure B.8), noting the clusters determined in [JWR$^+$19]. The ten identified somatic components are C9, C10, C21, C22, C28, C32, C33, C41, C42, C47. Comparing with the cluster-labelled t-SNE diagram from [JWR$^+$19] confirms

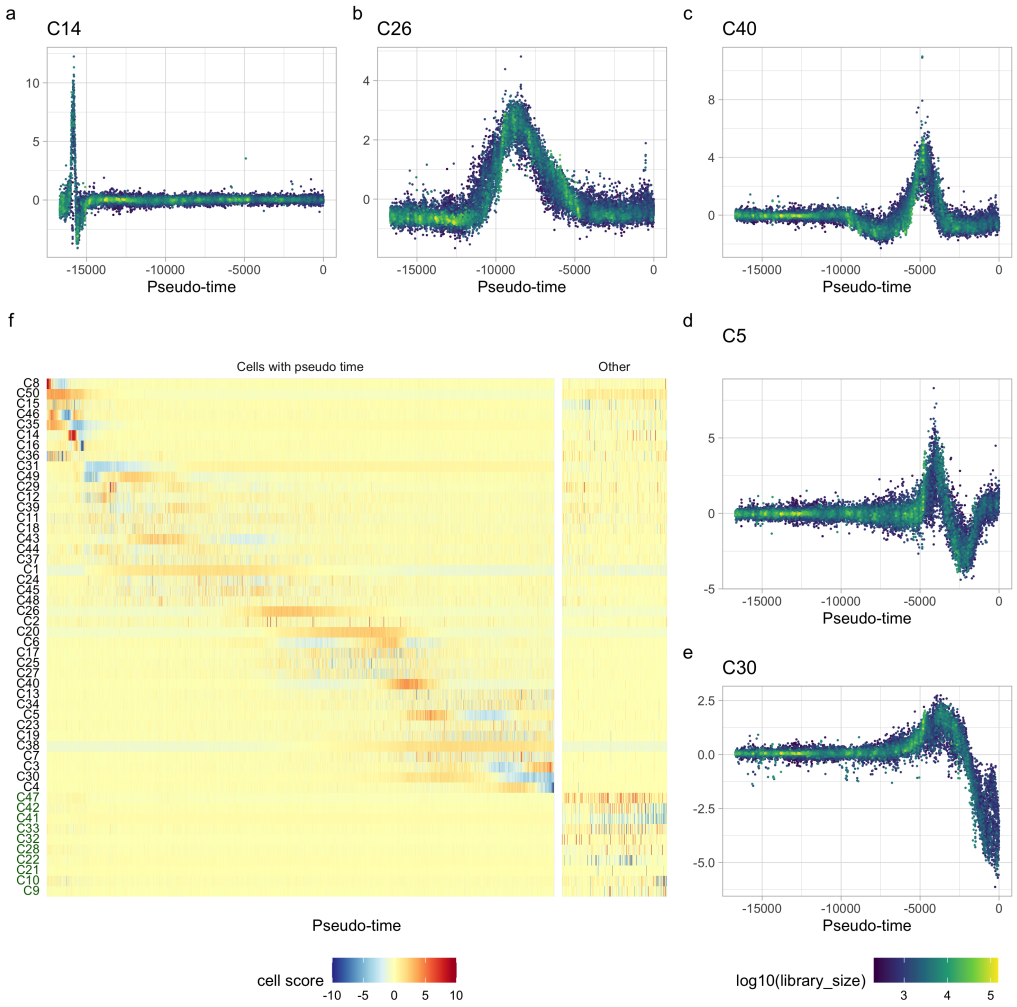

Figure 5.4: **Cell Scores over pseudo-time** (a-e) cell scores for selected components over pseudotime, plotted chronologically left to right. (f) Cell scores, capped at 10 across all components, by pseudotime. Somatic components are labelled in green text, and ordered alphabetically, non-somatic components are ordered by a weighted average of pseudo time as described in the main text. Cells without pseudotime, considered to be somatic cells, are order by library size.

that C9, C10, and C41 are all active primarily in Leydig cells, while C42 differentiates Leydig cells, similarly to V26, C21 is a Macrophage component, C22, C28, and C47 are active in Sertoli cells, and C32 is active in Sertoli cells and Telocytes, while C33 is qualitatively similar to V49, marking somatic cells generally.

In Figure 5.4(f) we visualise component cell scores over pseudotime in a heatmap and plot the non-somatic component cell scores on t-SNE coordinates, ordered according to the same weighted pseudotime in Figure B.9. These two figures show clearly the temporal dynamics present in many components as cell scores are silent over some period, before becoming strongly positive or negative. This demonstrates well the benefit of a factor analysis approach to modelling gene expression data since a hard clustering would fail to capture these temporal signals which overlap considerably. By comparing Figure B.9 with the cell type clustering and identification from [JWR+19], we can identify that the AnnotFA components capturing variation in known stages of spermatogenesis. For example, C38 is negative through meiosis and positive through spermiogenesis. C50 is a broad early meiotic component, with C8 to C36 capturing early meiosis, C31 through C1 are primarily active in pachytene through to the meiotic divisions, C26 is expressed around the final meiotic division and early acrosomal stage. C20 is broadly acrosomal and C40 is round spermatid stage.

We now highlight some components with particularly striking features.

Meiotic sex chromosome inactivation components

A key stage in meiosis is the synapsis of chromosomes preceding double strand break repair and recombination. At this stage, unsynapsed chromatin is silenced in a process known as meiotic silencing of unsynapsed chromatin (MSUC), in particular since the X and Y sex chromosomes do not synapse, meiotic sex chromosome inactivation (MSCI) is observed [Tur07, Tur15]. In previous studies MSCI has been shown to occur from the start of pachytene [LBEW18] and this has been confirmed for this

dataset [JWR+19]. Importantly, we observe this silencing in the component scores and loadings for components C31 and C49. In Figure 5.5(a,c), the loadings for both C31 and C49 are positive on the X chromosome, and Figure 5.5(b,d) show the corresponding cell scores on the t-SNE dimension reduction, demonstrating that these components have strong negative scores through early pachytene, corresponding to a repression of X-linked genes.

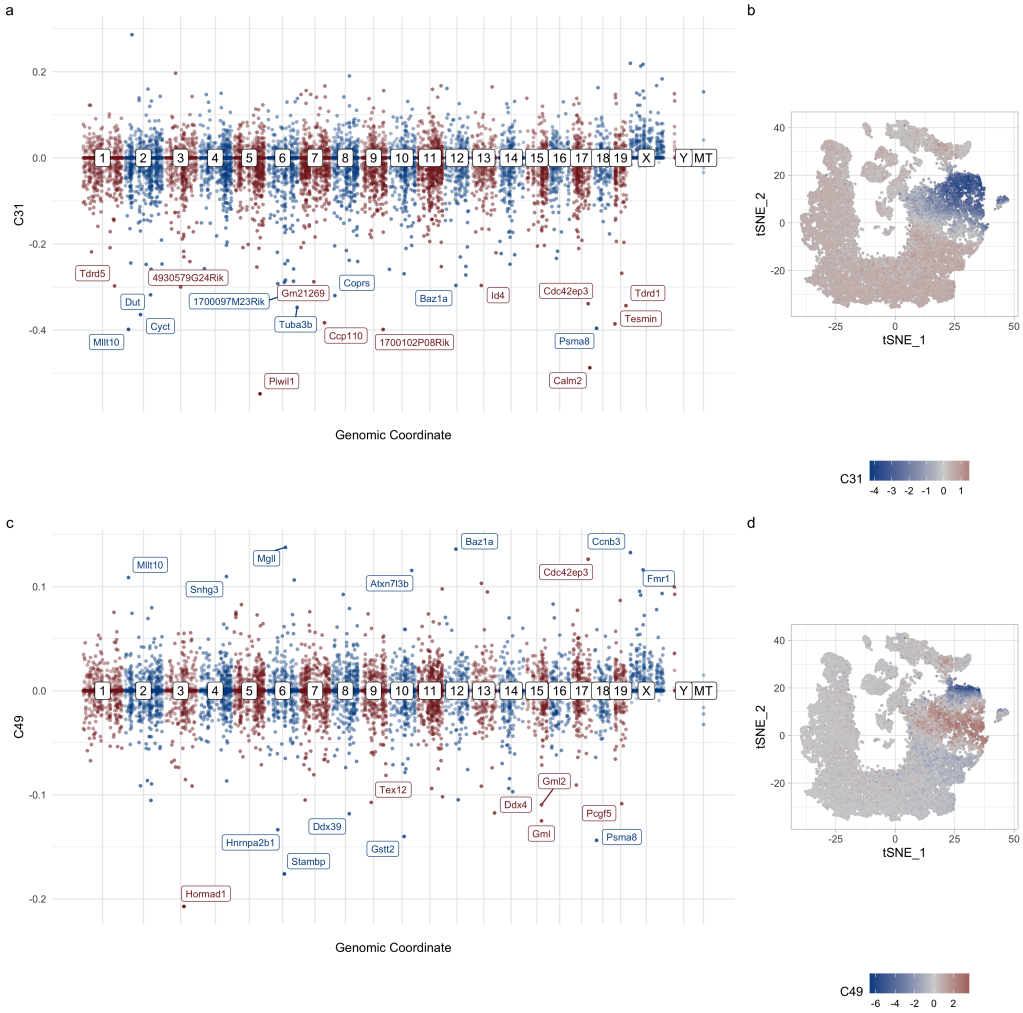

Figure 5.5: **MSCI Components C31 and C49** (a) Gene loadings for C31, labelling the 20 genes with largest loadings in absolute value. (b) Cell scores for C31 on t-SNE coordinates. (c) Gene loadings for C49, labelling the 20 genes with largest loadings in absolute value. (b) Cell scores for C49 on t-SNE coordinates.

Meiotic component C14

We now highlight particularly Component C14, both due to its key function in meiosis and to introduce several downstream inferences we make use of throughout the rest of the chapter. Component C14 is the 6th most highly conserved between initialisations but also shows some distinction from the corresponding SDA component V5 with which it has correlation -0.78 for cell scores (Figure 5.3(b)). See Figure 5.6(a,b) for the corresponding cell scores plotted over pseudotime and on the t-SNE coordinates. The cells in which the components are most active, having largest absolute cell scores, are highly conserved between the two components. In [JWR+19] a hard clustering was performed on the cells in this dataset and based on known major cell type markers these cells and component V5 were labelled (pre)Leptotene. We therefore consider our component C14 to also mark pre Leptotene cells. The most noticeable difference between V5 and C14 across cells being scores for the opposite sign (that is, negative cell scores for C14, and positive for V5) which occur before this main activity in V5 and after this main activity in C14. We also plot the gene loadings in Figure 5.6(c), labelling the 50 genes with the largest loadings. This component is of particular interest in meiosis, due to the number of high loading genes of known vital function. For example, stimulated by retinoic acid 8 (Stra8), the fourth ranked gene, is known to be first expressed in pre-leptotene cells and be expressed throughout leptotene. This important transcription factor binds to its own promoter and those of thousands of other meiotic genes triggering the transcriptional program that initiates meiosis in mice [KdRP19]. Zcwpw1 has the highest loading in this component and this is known to be recruited by Prdm9 (the gene with the 6th highest loading in this component), and is necessary for double strand break repair [WBM+20]. Prdm9, a speciation gene, is known to deposit histone marks that position double strand breaks [BBG+10, MBT+10, PPP10]. Hells is also known to be essential for meiotic progression [SDA+20] and forms a complex with Prdm9 to open chromatin to provide access for

double strand break formation. Beyond the genes highlighted here, a large number of noteworthy genes are in the highest 500 rankings for C14: genes coding for components of the cohesin complex Rad21l, Smc1b, Smc3, Stag3 and Esco2 [Ran15, JWR$^+$19]), genes involved in double strand breaks: Mei1, Ccdc36, proteins involved in formation and processing of single strand DNA intermediates and regulators Dmc1, Rad51, Brca2, Tex15, components of the synaptonemal complex Syce1, Syce2, Tex12, and meiotic telomere complex proteins Terb1, Terb2. Additionally, Meioc and Ythdc2 are known to act as post-transcriptional regulators, effectively extending meiotic prophase I despite the expression of many general cell cycle genes [KdRP19, ATR$^+$16]. Many of the genes listed above are known to be upregulated by Stra8 [KdRP19], and although Stra8 was not present in the annotations used in the inference performed here, it is strongly related to annotations that were used in the inference for this component (see Section 5.6).

In practice we see most components have both positive and negative loadings, and positive and negative cell scores. The natural interpretation is that these could be two components of expression that are opposing each other in expression dynamics across cells. That is, genes are coexpressed in such a way that positive cell scores represent the positive signed loadings being expressed, and the negative signed loadings being repressed, and vice versa for negative cell scores. This is the natural interpretation from the model and the way we interpret the inferred components here. To confirm this from the data, for each cell we first sum the total (normalised) expression of the 50 genes with highest ranked loadings on each sign of C14, labelling these signed components C14P and C14N (where the P and N suffix denote the positive loadings and negative loadings respectively) and view this across pseudotime in Figure 5.6(d). It is clear that the aggregated expression of the 50 largest negative loadings are most expressed in line with the large negative cell scores, and the aggregated expression of the 50 largest loadings is expressed most highly in line with the largest positive cell

scores, demonstrating that the components are capturing coexpressed genes across cells and that both sides of the component loadings should be considered separately. Secondly, we split the gene loadings into positive and negative parts (setting the loadings on the opposite side to zero in each case) and consider each of these to be "signed gene loadings components". For each such "signed" component, we obtained the residuals in the data by subtracting the reconstruction using only the remaining 49 components (and the intercept), and for each cell we regress the residual expression against the signed loadings to obtain signed cell scores. These inferred signed cell scores are plotted against pseudo time in Figure 5.6(e). Both components of signed cell scores showed qualitatively similar behaviour to the unsigned cell scores (those from the AnnotFA output), demonstrating that C14P does indeed capture a component of expression in the positive corresponding C14 cell scores, and repression in the negative, and similarly C14N captures the opposite. However, regressing the unsigned cell scores (the posterior means from the model) against the z-scores obtained from both C14P and C14N cell scores produced an R^2 of 0.998, and both significant coefficients of 0.751 and 0.236 for C14P, C14N respectively, demonstrating that, as expected from the magnitude of the loadings, the positive component C14P is the dominant component in explaining the variance across cells.

For all AnnotFA and SDA components, treating positive and negative loadings separately, we used the enrichGO function from the R package clusterProfiler [YWHH12] to perform gene ontology (GO) analysis, performing a hypergeometric test on whether the highest ranking 250 genes are enriched for GO terms compared to the background genes in the dataset and correcting for multiple testing using the Benjamini–Hochberg procedure.

Component C14 demonstrates the complexity and number of biological processes occurring during meiosis 1, since both C14P and C14N loadings clearly capture expression at closely related (pseudo)time points, which both occur within meiosis

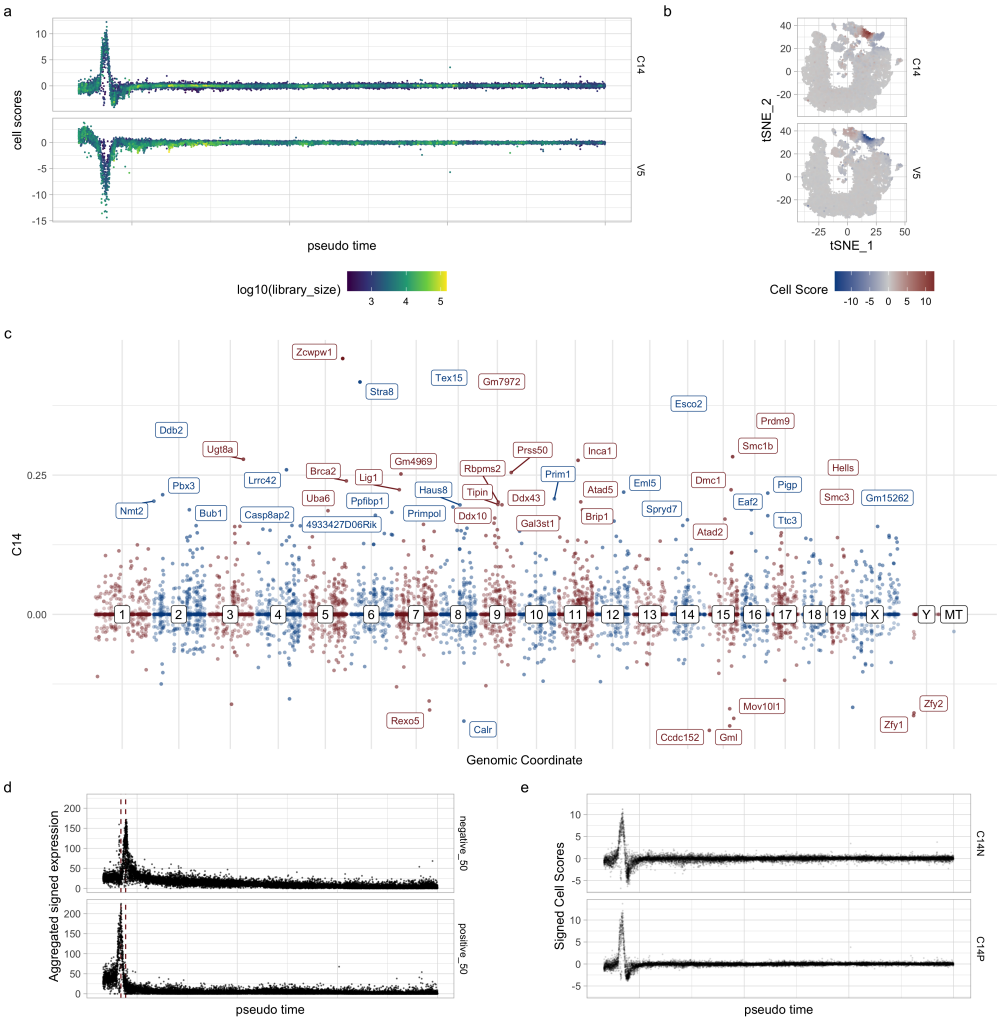

Figure 5.6: **Components C14 and V5.** (a) Cell scores over pseudotime for non-somatic cells for components C14 and V5. (b) Cell scores for all cells plotted on t-SNE coordinates. (c) Gene loadings for Component C14, labelling the 50 largest loadings in absolute value. (d) Aggregated positive and negative component expression for largest 50 loadings plotted across pseudotime, with red lines marking the maximum positive cell score and largest negative cell score. (e) Inferred signed cell scores as described in the text.

1. GO annotations for C14P and V5N both highlight this stage of meiosis, with shared terms meiosis 1, meiotic chromosome segregation, chromosome organization involved in meiotic cell cycle, homologous chromosome pairing at meiosis, DNA recombination, DNA repair, double strand break repair, homologous recombination, double-strand break repair via homologous recombination, recombinational repair and histone methylation, all with adjusted p-value less than 0.01.

As described earlier, the two components C14 and V5 are remarkably similar in the main burst of expression that is captured (for C14P and V5N), and the Procrustes transformation explained some of the differences in the way in which the models explain variance in the data. This difference was also evident in the GO analysis since the opposite sides (V5P and C14N) were enriched for strikingly different GO terms. We considered the complements of GO annotations for which V5P and C14N were enriched with adjusted p-value less than 0.01. The SDA component V5P is enriched for terms involved in cellular respiration and chromatid segregation that are not enriched in C14N, while C14N was enriched for terms involved in RNA splicing and gene silencing for which V5P was not enriched.

Components identifying mutant pathology

A careful investigation of all inferred components across cells and experimental batches has revealed a number of components identifying different mutant mouse strains. In particular, Hormad1 is known to be a meiosis-specific protein regulating chromosome recombination and synapsis. Hormad1 knockout mice cells fail to silence the X chromosome and become apoptotic due to pachytene checkpoint failure [DLH$^+$11]. This pathology was captured in Component C16 which is highly conserved between initialisations and has large negative cell scores for a subset of the Hormad1 KO cells (Figure 5.7(b)). These cells are clearly separated from the majority of other cells in the t-SNE plot (Figure 5.7(c)), consistent with the fact that they are not following

the usual developmental trajectory. Moreover, the majority of the largest loadings of C16 are on the X chromosome (Figure 5.7(a)), including Rhox2h, an X-linked gene which is reported in [JWR+19] to have not been expressed in cells at earlier stages but is highly expressed in the Hormad1 KO cells identified in the cell cluster that this component is most strongly expressed in (cf. Cluster 30 from [JWR+19]). A very similar component (V38) was identified and discussed in [JWR+19], which has loadings and scores with Pearson correlation 0.72 and 0.78 respectively with C16, and subject to reversing the sign has several very similar properties, including also having highest loadings on chromosome X, while still having non-zero loadings on autosomes. However, we note that V38 has AUC 0.69 based on our downstream analyses inferring the PIP prediction from annotations (see Figure 5.13), while C16 has AUC 0.79, demonstrating that C16 is informed by annotations. We see further evidence (see Figure 5.18(c,d)) that the prior inclusion for the negative loadings are complex, depending on many annotations, including MybA, E2f5, Zfx. MybA is also known as Mybl1 which is a master regulator of meiosis [BFBB+11]. Zfx is an X-linked transcription factor that binds many of the same genes as Zfy1 and Zfy2, which have large loadings in C16 and V38, and were suggested in [JWR+19] to potentially explain the large autosomal loadings in V38.

Another mutant mouse line associated with infertility is CNP knock-in mice. In previous work no component was identified to clearly explain the mutant phenotype. However we do identify one component that has an expression profile that identifies Cnp knock-in cells (henceforth CNP cells), namely Component C48. This component has a relatively flat cell score profile but scores for CNP cells are noticeably lower (negative and larger in absolute value) than other cells. To investigate this, we first consider signed cell scores derived from the positive and negative loadings of C48 to determine whether one, or both are driving the separation of the CNP cells from the remaining cells. It is clear from Figure 5.8(c,d) that while the negative loadings

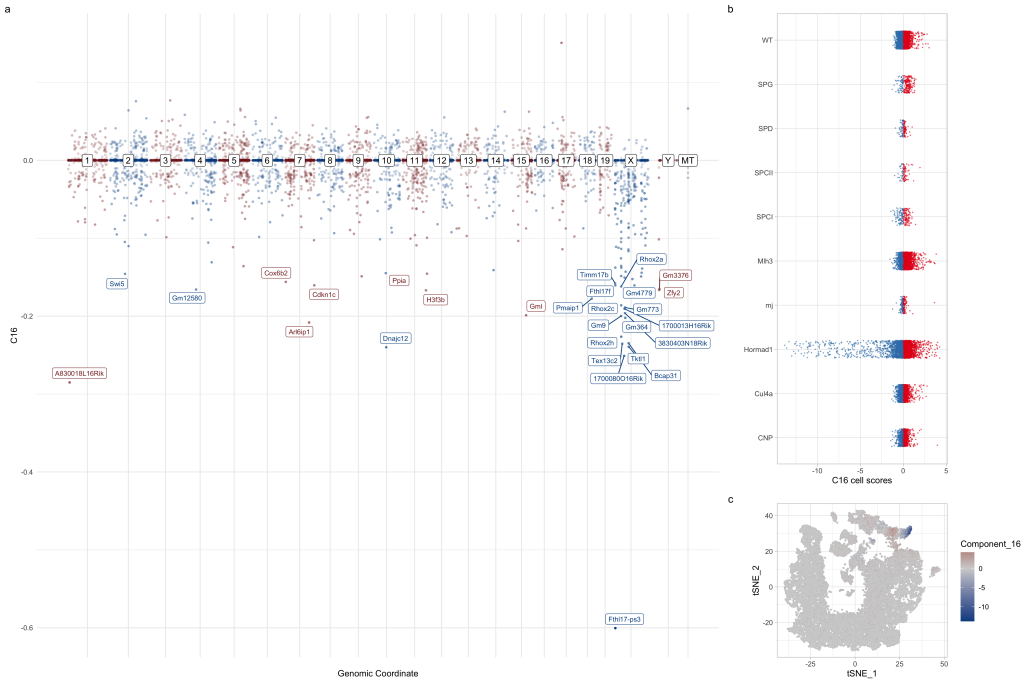

Figure 5.7: **Hormad1 KO pathology component C16** (a) Gene loadings for C16, labelling the 30 genes with largest loadings in absolute value. (b) Cell scores by experimental group. (c) cell scores on the t-SNE coordinates clearly identifying a separating group of Hormad1 KO cells that are not proceeding in the usual direction.

are consistent with lower cell scores, the positive component C48P is driving the separation. This is consistent with the component loadings for which the largest are overwhelmingly positive (92 of the largest 100 loadings were from positively weighted genes, while 49 of the largest 50 were positive[1]).

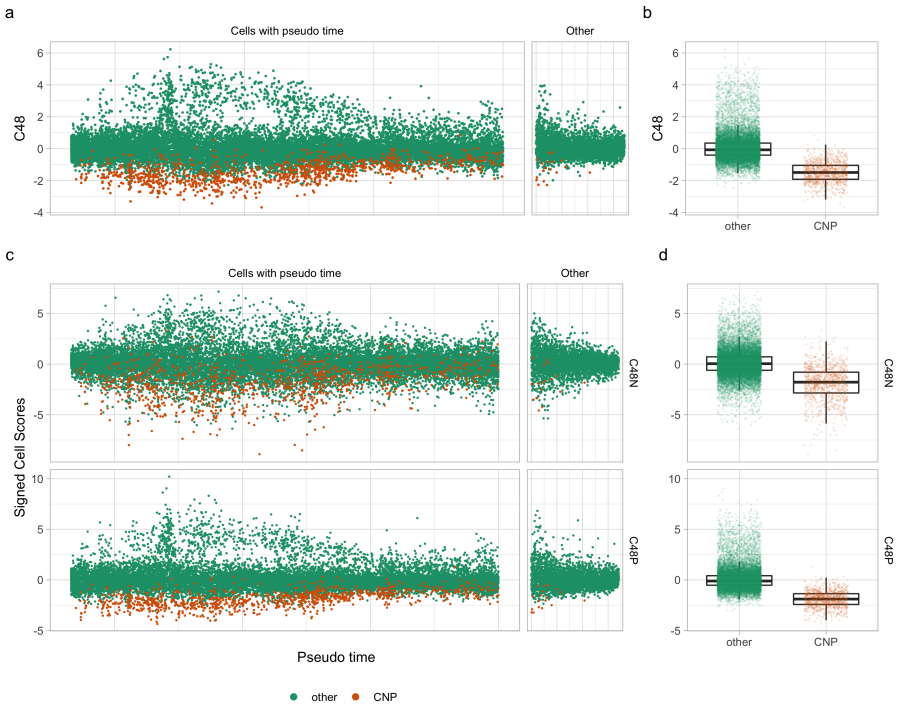

Figure 5.8: **CNP knock-in component C48.** (a) Component C48 cell scores ordered by pseudotime where it is available and by library size otherwise. Coloured by mouse cell line to indicate Cnp knock-in cells. (b) boxplot and scatter plots of the cell scores for C48, coloured as for (a). (c - d) Signed component cell scores similar to (a-b).

The CNP mice are transgenic, expressing ribosomal protein Rpl10a under the control of the Cnp promoter (described at https://www.jax.org/strain/009159, and [DDH+08]). Consistent with this, the most significant GO terms associated with the positive part of the component include ribosomal small subunit assembly (OR 34.7, p-value 8×10^{-14}, and q-value 2.59×10^{-11}), ribosome biogenesis (8.29, $3.73 \times 10^{-18}, 2.38 \times 10^{-15}$), ribosomal small subunit biogenesis (16.28, $5 \times 10^{-15}, 2.41 \times 10^{-12}$), ribosome assembly ($16.9, 1.9 \times 10^{-14}, 7.3 \times 10^{-12}$). The most significant term was

[1]the exception being the 11th ranked gene Csnk2b which had the largest negative loading.

cytoplasmic translation with an odds ratio of 20.7 and p-value 2.91×10^{-24}, q-value 5.58×10^{-21}. Both before and after correction for multiple testing these terms were the most significant for C48P. We note that the pattern of cell scores, loadings, and these terms are consistent with genes associated with these terms being under expressed in the CNP cells.

We also identified components clearly capturing Cul4a KO mutant pathology. Cul4a is a component of the E3 ubiquitin ligase complex CRL4 which regulates cell cycle, DNA replication, DNA repair and chromatin remodelling [DDWN18]. Previous studies of Cul4a knockouts observe some cells arresting at the pachytene checkpoint, and cells that complete meiosis result in malformed spermatozoa [YLK$^+$11]. This suggests that we should see at least one component capturing Cul4a KO pathology and we observe three such components, namely C19, C27, and C45 (see Figure 5.9). All three of these components correlate most highly with Component V25 from [JWR$^+$19] which was identified as a Cul4a knockout component. However, C19, C27, and C45 appear to be active in three different stages, C45 being the earliest, diverging from wild type cells in Pachytene/Division II. Component C27 appears to diverge during the round spermatid phase, and Component C19 diverges at the latest stage of elongating spermatids.

We inferred the signed cell scores and both signed components show the same divergence from wild type cells in all three of the components, and the aggregate expression shows expected patterns for the highest ranking genes in each signed component, suggesting that both signed components are diverging for Cul4a cells at the stage indicated by the cell scores. Component C45 cell scores diverge to be positive for Cul4a cells, while the wild type cells are noisily around zero. The highest ranking 250 loadings for C45P revealed no significant association with GO terms after correction for multiple testing. However, C45N has several GO terms associated such as cilium organization (OR 6, p-value 1.5×10^{-11}, q-value 2.76×10^{-8})

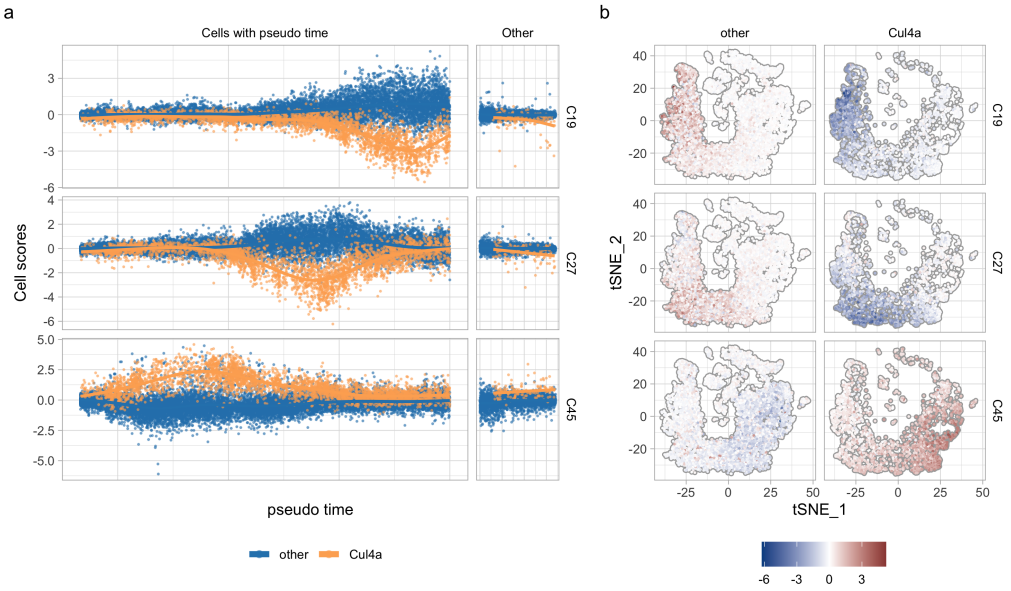

Figure 5.9: **Cul4a components C19, C27, C45.**

and cilium assembly (OR 5.48, p-value 2.86×10^{-9}, q-value 2.59×10^{-6}). This may well signify repressive regulation in the Cul4a mutant disrupting the later stages of spermatogenesis.

Components C27 and C19 both diverge to negative cell scores while the wild type cells stay closer to zero. The signed loadings associated to C27 and C19 had no significant associations (adjusted p-value < 0.01) with GO annotations after correction for multiple testing.

5.5.1 Component inclusion probabilities ordered by pseudotime

The model also infers posterior inclusion probabilities (PIPs) for genes in components, providing a valuable insight into the association between sets of genes involved in different stages of spermatogenesis. For completeness we examine these here, but we note that many of the key features are also reflected in the inferred prior inclusion probabilities which are derived directly from the annotations, and which we further

examine in Section 5.6.1.

Correlations of component PIPs reveal a dramatic switch between two sets of developmental components occurring at a particular stage of pseudotime, visualised in Figure 5.10(a). This heatmap shows a clear separation between the earliest 22 developmental components and the later 16 developmental components. Between the two blocks, components have typically uncorrelated or negatively correlated PIPs, pointing to a switch in genes involved in an early set of stages and the later stages of spermatogenesis. The two components on the cusp of this switch are components C2 and C26. Component C2 maintains low cell scores in the majority of cells except for a particular batch of wild type cells distinguished by the fact they are enzymatically dissociated, while all other wild type cells were mechanically dissociated. We therefore consider this component to have captured a batch effect most likely caused by the dissociation method used. On the other hand, C26, appears to be of biological interest: It has a clear temporally dynamic expression profile (Figure 5.4(b)) that is small but negative for early and late stage cells but switches sign and becomes strongly positive for a portion of pseudotime in a continuous pseudotime trajectory roughly crossing the threshold between the second meiotic division stage and the round spermatid stage (cf. https://www.testisatlas.ml for stage determination). The larger size of the early block of components is consistent with the complexity of the orchestra of biological processes occurring in meiosis, and this first block also exhibiting signs of further structure.

Component C26 PIPs show low but positive correlation with both the early and late clusters of components. We thus split the C26 PIPs according to the sign of the gene loadings, resulting in PIPs for signed components C26P and C26N and calculated the correlation with each of the (unsigned) components C1 - C50. Signed components C26P and C26N have correlation 0.69 and 0.56 with Component C26 respectively, all other components had lower correlation and are plotted in

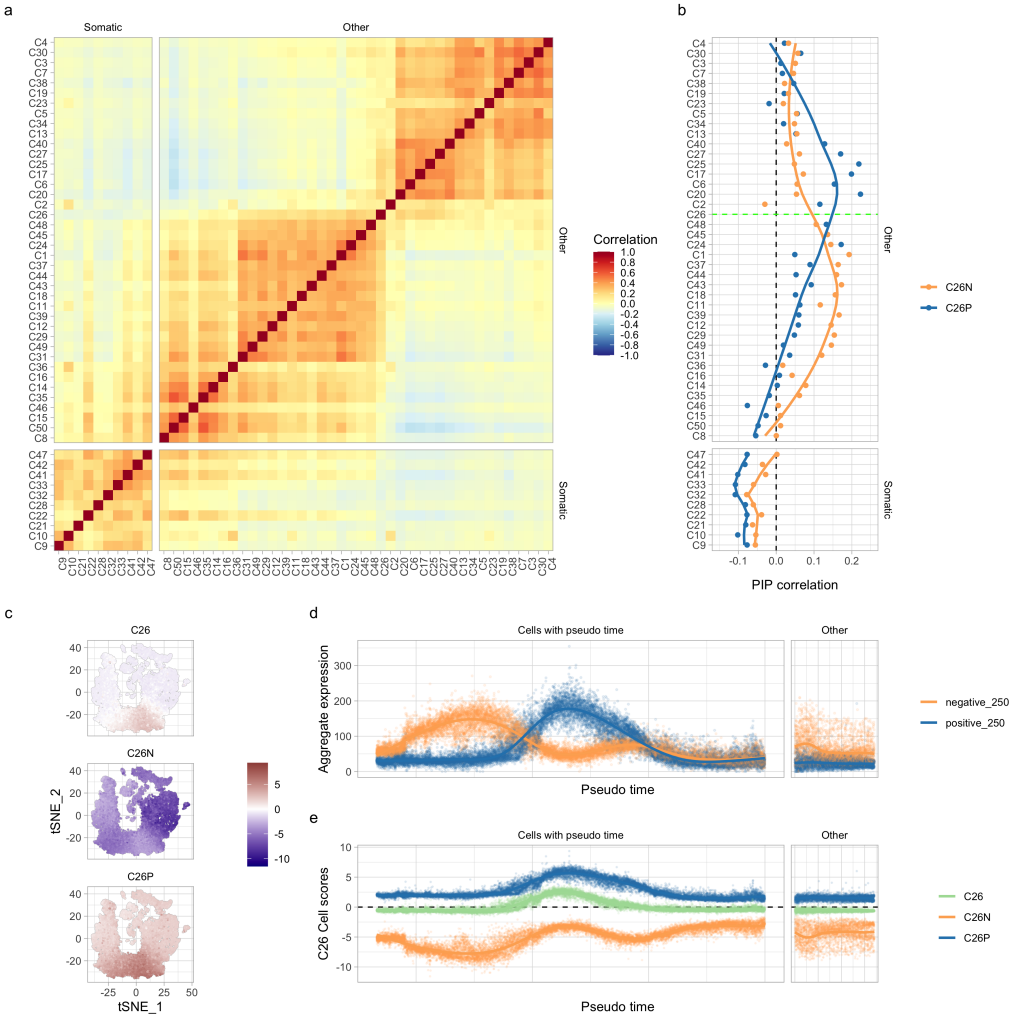

Figure 5.10: **PIP correlation and Component C26.(a)** Correlation between inferred posterior inclusion probabilities (PIPs), separated into somatic components and other components, with other components ordered by a weighted average of pseudo time as described in the main text, chronologically left to right. **(b)** Correlation of the signed PIPs C26P and C26N with components ordered as in (a). The dashed black line marks 0, and the dashed green line marks the $C26$ component, points for which are not plotted. **(c)** Component C26 and inferred signed cell scores C26N, C26P on t-SNE coordinates. **(d)** Aggregated normalised expression of the highest ranking 250 genes on either positive or negative side of C26 plotted over pseudo time. **(e)** Inferred cell scores C26P and C26N plotted with C26 over pseudo time.

Figure 5.10(b) ordered in pseudotime. This shows that the correlation for the later stage components is partially explained by the positive signed component C26P and the early stage components are correlated most highly with C26N. Patterns of aggregated normalised expression over pseudotime reveal also that the highest ranked 250 genes of the positive side of C26 follow a continuous expression trajectory over pseudotime similar to the cell scores of C26, while those of the negative side of the component are primarily expressed at an earlier stage of the developmental timeline, with a second weaker wave of expression at the end of the main C26P wave (Figure 5.10(d)), consistent with this component repressing the C26N gene loadings during the positive wave of C26P expression. The signed cell scores, inferred as described for C14, reveal a similar pattern of expression (Figure 5.10(c,e)). The loadings for component C26N are enriched for GO terms mitotic sister chromatid segregation (OR 4.95, p-value 2×10^{-5}, q-value 7.7×10^{-3}), mitotic nuclear division (OR 4.08, p-value 8×10^{-6}, q-value 6.3×10^{-3}) and other terms consistent with the second division occurring in meiosis II such as mitotic spindle organization, mitotic spindle assembly, positive regulation of G2/M transition of mitotic cell cycle (OR = 7, p-value 8×10^{-3}), nuclear division and nuclear chromosome segregation. The positive signed component C26P has 250 highest ranked genes enriched for GO terms consistent with development of the axoneme as a first step toward the spermatid developing a tail. In particular, we note the association of GO terms axonemal dynein complex assembly (OR 12.26, p-value 5×10^{-5}, q-value 9.9×10^{-3}), axoneme assembly ($8.42,\ 5.8 \times 10^{-6}$,$1.2 \times 10^{-3}$) cilium assembly ($5.97, 0, 1 \times 10^{-7}$), cilium organization ($5.6, 0, 1 \times 10^{-7}$), and cilium movement ($5.35, 4.2 \times 10^{-6}, 2 \times 10^{-3}$), as well as cilium-dependent cell motility ($5.31, 6.7 \times 10^{-5}, 1 \times 10^{-2}$).

Some remaining structure in Figure 5.10(a) highlights batch effects and issues with the ordering of components, but there is evidence that the more subtle structure is biologically relevant. In particular, C36 and C46 appear to disrupt the earlier block

(pre-C26), and on closer inspection, they are likely placed too late in pseudo-time (cf. Figure 5.4(f)). The remaining structure in the early block is subtle but there appears to be a switch roughly separating the pairs C14, C16 and C31, C49 (in pseudotime order, having identified C36 as an earlier component). This is consistent with the start of pachytene. Recall that C31 and C49 mark X chromosome inactivation, as noted in Section 5.5. However, we note that the change in PIP correlation observed here is not signified by exclusion of the X chromosome from the PIPs since both components included X-linked genes on one side of the loadings.

The spermiogenesis block (the late stage block, occurring after C26) also has some structurally interesting features. Component C23 appears out of place, and on closer inspection appears to be a batch effect identifying the same batch of cells as C2. Some more subtle structure in this later block is a visual change between C40 and C13. Component C40 has a beautifully continuous trajectory when plotted against pseudotime (Figure 5.4(c)), and on comparison with the stages from [JWR+19], marks the end of the round spermatid stage, as the spermatid matures into the elongating spermatid stage of development.

5.6 Annotation Informed Inclusion Probabilities

5.6.1 Annotation informed prior inclusion probabilities

A novel feature of AnnotFA is the incorporation of annotations into the prior for the inclusion probabilities. As a part of the inference algorithm, AnnotFA infers the important annotations for each component and constructs the prior through a logistic regression update (see Section 3.2). In this section we consider the prior inclusion probabilities (pIPs) and posterior inclusion probabilities (PIPs), and the evidence that the model has made effective use of the annotations.

The density plot in Figure 5.2(d) shows some gene loadings density at 0 for PIPs

greater than 0.5. We consider this to occur for two reasons. Firstly, if the component-wide prior slab precision β_q is large then the spike and slab model for each gene in this component places a large density close to zero from the slab as well as the spike and this poses an identifiability issue. This results in little power to remove a gene with a small loading from a component by the PIP, and in the variational posterior for the gene loading is characterized by inferring a large PIP and centering the conditional mean of the slab at zero. Secondly, we consider this to also be consistent with the new prior structure in AnnotFA since the prior dependence on annotations can cause genes to have a strong prior for inclusion while the genes are not included in the component, and we may not have enough information in the data to remove the gene in the posterior. Figure B.10 confirms that the excess in PIPs being larger than 0.5 while having loading close to 0 is correlated with the posterior mean of β_q consistent with the comments above.

Since as part of the algorithm the prior is constructed through the inference of annotation coefficients, we have used these annotation coefficients to reconstruct the prior inclusion probabilities, as visualised in Figure 5.11(b). These show considerably more uncertainty than the posterior inclusion probabilities in Figure 5.2(c), but each component is positively correlated with the PIPs (Figure 5.11(a)), with correlation ranging from 0.31 to 0.89. The annotation coefficients were typically sparse, with many coefficients close to zero. We note also that while the correlation between prior and posterior inclusion probabilities is positive, it is clear from Figure 5.11(d) that the posterior inclusion probabilities can be high when the prior was low. However, when the prior is high enough, it appears there is little evidence to remove a gene (that is, to set the PIP to zero). Furthermore, the stratification in Figure 5.11(d) showing curves of high density where the prior is high but the posterior is relatively low are bounds on the PIPs within components achieved when the inferred loading is small. The bound depends on $\langle \beta_q \rangle$, as shown in Figures 5.12 and B.11. This is

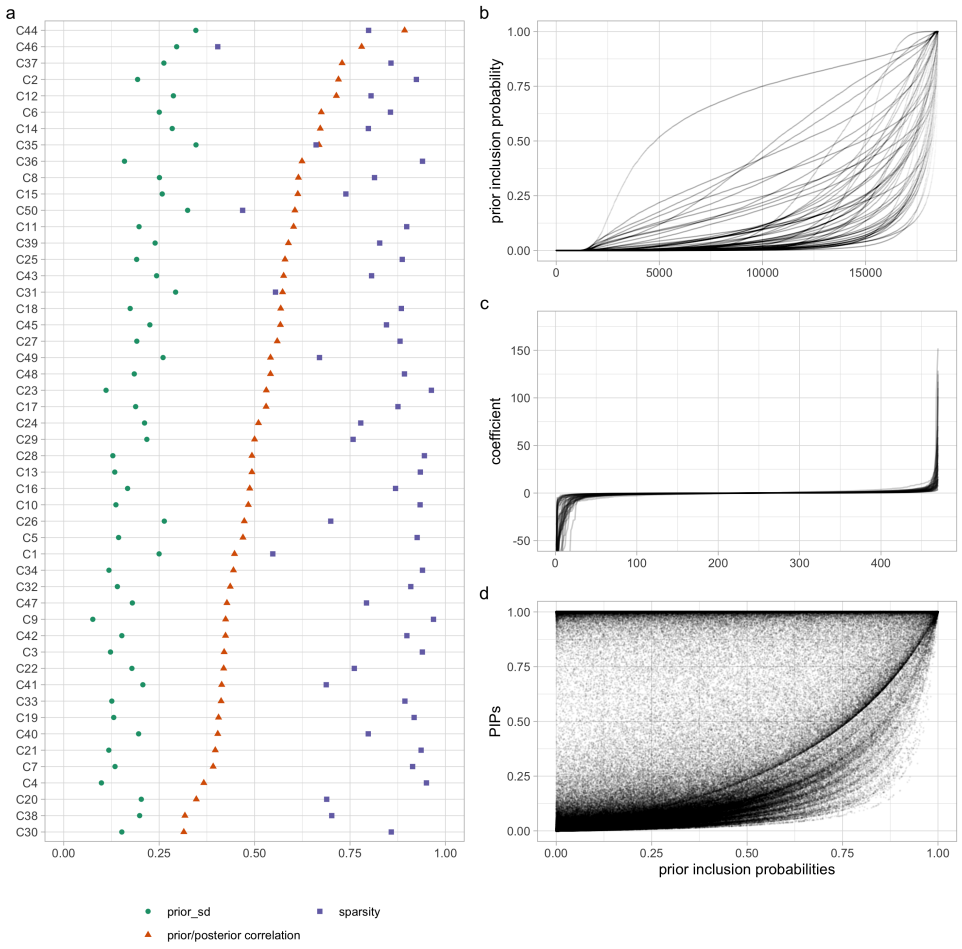

Figure 5.11: **Unsigned Prior and Posterior Inclusion Probabilities.** (a) Component correlation of prior and posterior inclusion probabilities, sparsity by PIP, and standard deviation of the prior inclusion probabilities. (b) prior inclusion probabilities ordered within component. (c) Ordered prior annotation coefficients for each component. (d) Scatter plot of prior and posterior inclusion probabilities.

representative of the identifiability issue as discussed earlier that we observe in the small loadings for large β_q values. Furthermore, the prior and posterior inclusion probabilities correlation is positively correlated with $\langle \beta_q \rangle$ (correlation coefficient 0.71, p-value 5×10^{-9}), plotted in Figure B.12.

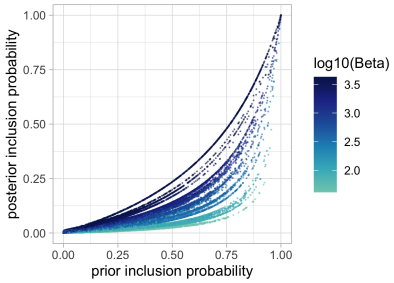

Figure 5.12: **Unsigned prior and posterior inclusion probabilities by** $\langle \beta_q \rangle$ Scatter plot of the posterior inclusion probabilities against the prior inclusion probabilities including only those genes and components for which loadings are less than 10^{-4}, and coloured by $\log_{10}(\langle \beta_q \rangle)$.

To quantify the impact of incorporating annotations into the model, in comparison to an analysis method which does not make use of annotations, we compare with the SDA components from [JWR+19]. To determine whether the SDA components are consistent with being informed by the annotations, we fit a logistic regression model to the SDA PIPs with the annotations as predictors and evaluate the effectiveness of the predicted inclusion probabilities as a predictor for SDA PIPs by considering the AUC from the associated ROC curve. It is clear from Figure 5.13, that although both SDA components and AnnotFA components achieve a similar range of sparsity, the AnnotFA PIPs are better predicted by annotations than the SDA PIPs, suggesting that the model is making effective use of the annotations. Further evidence is provided by the Spearman and Pearson correlation of the annotation predicted prior inclusion probabilities and the PIPs.

We have shown evidence that the prior inclusion probabilities are informative and that the posterior inclusion probabilities capture patterns of expression that align with stages of spermatogenesis, suggesting that transcription factor binding to promoter

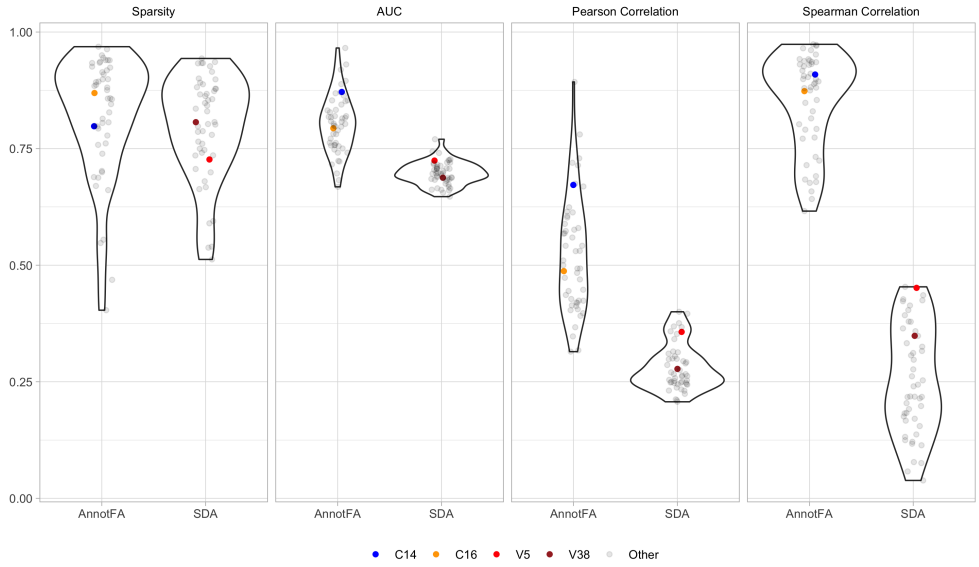

Figure 5.13: **Annotations inform inclusion probabilities.** Violin plots comparing SDA and AnnotFA components in terms of sparsity, AUC, and correlation of prior inclusion probabilities with posterior inclusion probabilities as described in the main text.

is associated to the spermatogenic stage of the cell and that the prior inclusion probabilities will capture this information. When ordered over pseudotime, the inter-component pIP correlations have a pattern similar to that of the posterior inclusion probabilities, clearly separating meiosis and spermiogenesis (Figures 5.10 and 5.14(a)). To further examine the structure present, we performed a hierarchical clustering on the correlation matrix which identified further rich structure (Figure B.14(b,c) and Table 5.1). The clusters detected capture broad stages of spermatogenesis, as indicated in Table 5.1. In particular, the major split is separation of components into two clusters of 33 and 17 components respectively. The larger cluster of 33 components consists of the two somatic, three meiotic, and the "Other" clusters as labelled in Table 5.1. The smaller cluster of 17 components consisted of two batch effect components (C2, C23, as described in Section 5.5.1), and three clusters which broadly capture early, mid and late spermiogenesis. Almost all components in clusters are consistent with the labels attached to the cluster, with the exception of C46, which is most likely associated to Spermatogonia and commitment to meiosis (based on pseudotime and

Cluster	Component
Somatic cluster 1	C9, C10, C21, C28, C32, C33, C42, C46
Somatic cluster 2	C22, C41, C47
Early meiosis cluster	C14, C15, C16, C35, C50
Pachytene cluster	C1, C18, C24, C29, C31
Late meiosis cluster	C11, C26, C37, C39, C43, C44, C45, C48
Early spermiogenesis	C6, C17, C20, C25, C27, C40
Mid spermiogenesis	C13, C19, C38, C34
Late spermiogenesis	C3, C4, C5, C7, C30
Batch effect	C2, C23
Other	C8, C12, C36, C49

Table 5.1: Clusters of component prior inclusion probabilities as indicated in the dendrogram in Figure B.14(c)

comparison of cell scores with those from a highly correlated SDA component labelled B spermatogonia in [JWR+19]). The cluster labelled "Other" consists of Component C8, a very early meiotic component, Component C36 is a batch effect for a batch of spermatogonial stem cells, and components C49 and C12. The expression of C49 is highly specific to the early pachytene cells in which X chromosome inactivation initially occurs, with a large loading on Hormad1, a gene which is responsible for MSCI, so we consider this to be a component which likely captures this unusual and highly specific process. The identification as an outlier here may well signify a transcriptional regulation related signature. C12 has largest cell scores expressed at a similar pseudotime in opposite directions for two different mutant strains so we consider this to likely be a component capturing mutant pathology.

Having identified the clusters as described above, there is a clear pattern of development in the correlations, as shown in Figure 5.14(b) where we have ordered (left to right and bottom to top) these clades roughly according to pseudotime as somatic clades, meiotic clades and spermiogenic clades, followed by the batch clade of C2 and C23, the mixed clade consisting of C8, C12, C36, C49, and finally the outlying component C46. This ordering reveals a great deal of heterogeneity between blocks of somatic components, meiotic components, spermiogenic components, and a number of

outliers that correlate most closely with the meiotic components. In particular, we note the clear separation of meiotic and spermiogenesis components, which are mostly negatively correlated, suggests a clear separation of transcriptional expression activity, and likely signalling a significant switch in transcriptional regulation. Furthermore, the clade of somatic components C22, C41, C47, most closely clustered with the meiotic components forms a clear bridge between the somatic and meiotic components, with correlation structure blending with the early meiotic components, in particular with C50 and C15, which occur roughly before C14 in pseudotime, and therefore relate to spermatogonial stem cell differentiation and commitment to meiosis. This correlation of regulation may well be indicative of early signalling which may be shared between spermatogonial stem cells and the neighbouring "nurse" sertoli cells [MLH$^+$00]. The pachytene cluster is clearly present, and bridges the preceding early meiotic and later meiotic clades. Similarly, early spermiogenesis clades show clear structure with the structure being weaker as components move through spermiogenesis. This clustering therefore provides another perspective into the post-meiotic switch as evidenced in Figure 5.10(a), revealing structure in the patterns of transcriptional regulation involved in spermatogenesis which are predicted by sequence features.

To elucidate the information regarding shared regulatory features captured in the prior inclusion probabilities, we further considered rank correlation between pIPs and certain annotations. In addition to the annotations that were included in the model we also include a binary annotation indicating whether there is a CpG island within 2kb of the transcription start site, derived from [KdRP19, Supplementary Information]. In Figure 5.15 we show a subset of the results. It is clear that the somatic and meiotic components share regulatory features distinct from the spermiogenesis components. This switch in regulation at the meiotic division is consistent with findings in [JWR$^+$19], which asserted that there is a switch in groups of transcription factors regulating expression between pre-division cells and post division cells. It is

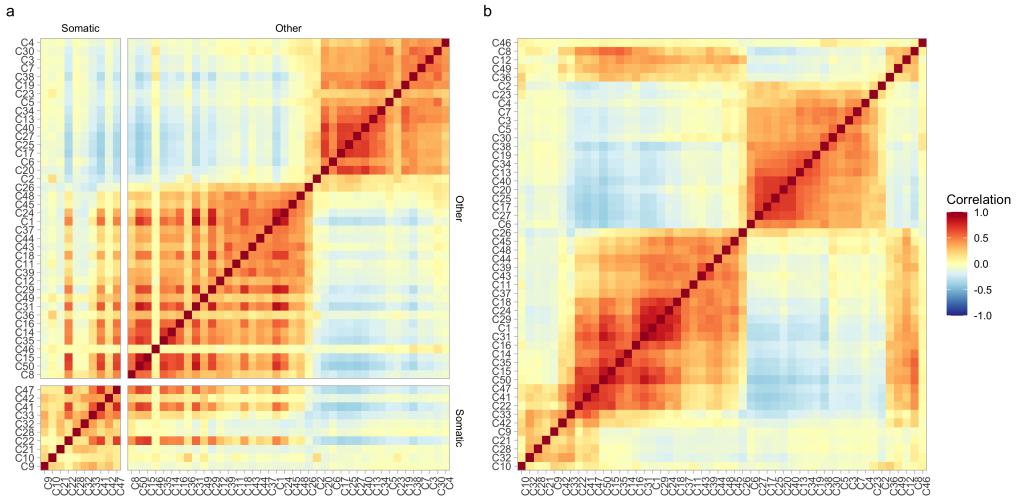

Figure 5.14: **Prior inclusion probability correlation** (a) Correlation between prior inclusion probabilities, with components identified as somatic and ordered according to pseudotime in an identical ordering to Figure 5.4(f). (b) Correlations between prior inclusion probabilities with ordering based on the hierarchical clustering presented in Figure B.14(b,c)

noted in [JWR+19] that many of the motifs associated with pre-division gene loadings contain CpG dinucleotides, and this is consistent with the pattern observed here for the CpG annotation, that gene inclusions in the pre-division components are positively correlated with inclusion of a CpG island at the promoter, and negatively correlated for post-division components. Here we see only the Rfx3 motif spanning the divide, and we note that this is also consistent with the observation that Rfx2, which is a master regulator of spermiogenesis [KBL+15], spans the divide in [JWR+19], since Rfx3 is the representative of the transcription factors Rfx1-6 (see Section 5.3).

We see strongest correlation for AP2D, concentrated in Component C35 and C14, which we consider to be a signature of the Stra8 initiated transcriptional program, noting particularly that the AP-2 family of transcription factors were noted for being most similar to the Stra8 motif described in [KdRP19] (which was not included in our analysis), while AP-2 genes were found to not be expressed in preleptotene cells, the stage at which C14 is primarily expressed. Similarly both E2F5 and E2F6 are present in the set of annotations correlating with meiotic components, and E2f family

transcription factors were noted to have motifs enriched near Stra8-activated genes transcription start sites. A further striking feature is the strong negative correlation that the Kaiso annotation presents. This annotation shows positive association with the pre-division components followed by strong negative correlation with the post-division components. Furthermore, Kaiso is known to act as a methylation-dependent transcriptional repressor, binding either to a sequence-specific site or to methylated CpG dinucleotides [SZPP15], so the strong negative correlation here may suggest a role for Kaiso in the sudden transcriptional silencing of genes that were active in meiotic cells pre-division. However, the only detected expression of Kaiso in this dataset is in early meiotic cells.

5.6.2 Signed component prior inclusion probabilities

We have shown in Section 5.5 that there is reason to consider the components as signed components, taking the positive and negative loadings separately. To study the PIPs further we split the PIPs for each component into two "signed" components labelled P and N corresponding to the positive and negative loadings respectively, and setting PIPs for which the loading takes the opposite sign to zero in each case.

We now further investigate the evidence that transcription factor binding to promoter sequence may be predictive of sets of genes that are coexpressed through spermatogenesis by considering the informativeness of the prior inclusion probabilities. To identify the signed prior inclusion probabilities, we infer coefficients in a logistic regression on the signed PIPs in R using the glm function with family "quasi-binomial", with the same predictors as used in the model. From this we predict the inclusion probabilities and we refer to these as signed prior inclusion probabilities (signed pIPs) and distinguish these from the unsigned prior inclusion probabilities that were inferred in the model. When inferring the signed pIPs we set genes for which the loading takes a certain sign to zero, so the signed PIPs are necessarily more sparse than the

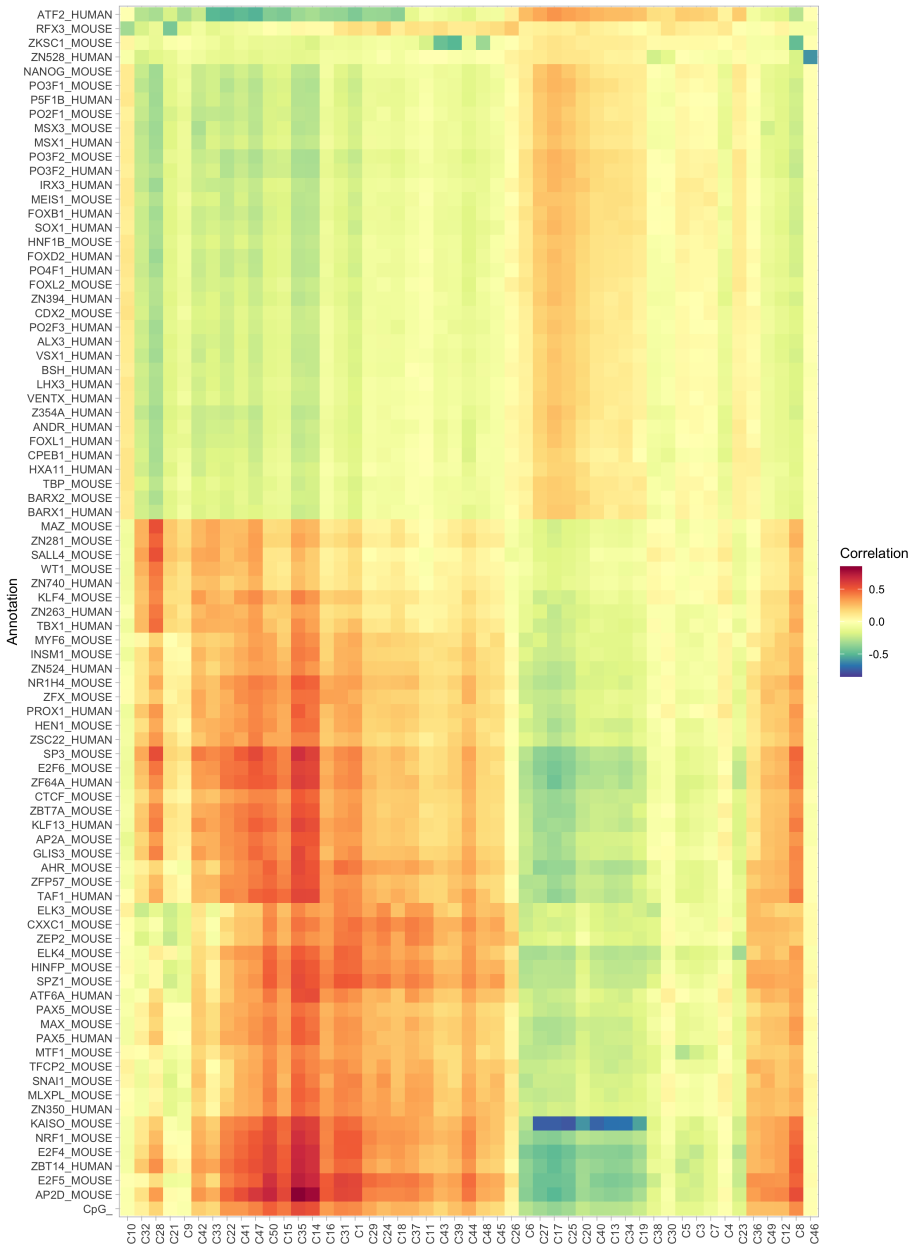

Figure 5.15: **Spearman Correlation of pIPs with annotations.** The annotation matrix used in the model was filtered to include only annotations for which the absolute rank correlation with component prior inclusion probabilities exceeds 0.4 for at least one component. The annotations are ordered by hierarchical clustering of the correlation vectors, and the first row is a CpG annotation as described in the main text.

unsigned PIPs, and this is reflected in the corresponding pIPs (see Figures 5.11(b) and 5.16(b)). We also find that the regression coefficients for unsigned pIPs are relatively sparse. Each component has one signed component for which the pIPs are highly correlated with the unsigned prior (Figure B.13(a)). Furthermore, for each component the correlation between the two corresponding vectors of signed pIPs are positive, with coefficients ranging from 0.11 to nearly 0.78 (Figure 5.16(c,d)). The components for which the correlation of signed priors is low, each have one signed pIP which correlates well with the unsigned pIPs and one that does not (Figure B.13). Moreover, in this case the signed pIPs are often more correlated with other components of signed pIPs (Figure B.13). The correlation between signed pIPs for a component has correlation 0.66 with $\langle \beta_q \rangle$, and as $\langle \beta_q \rangle$ decreases, the components tend to be predominantly one sign or the other (Figure 5.2(a)). On inspection, the components which are predominantly one sign or the other in loadings have lowest correlation between signed pIPs. This suggests that the unsigned pIPs are correlated with the dominant side of the loadings, and the corresponding signed pIPs reflect this.

We have taken two approaches to determine the informativeness of the prior. First, we define the informativeness score of a component's signed pIPs as the standard deviation divided by the maximal possible standard deviation from a component with identical mean. That is, we scale the standard deviation by $\sqrt{p(1-p)}$ where p is the mean pIP. An informativeness score of 0 corresponds to a uniform prior distribution on the inclusion probabilities, while a 1 corresponds to a distribution of probabilities with maximal variance among those with mean p. Figure 5.16 shows the pIPs have a range of informativeness, which does not appear to depend on the sparsity of the component, and is weakly correlated with the standard deviation of the prior inclusion probabilities. The vast majority have an informativeness score between 0.25 and 0.5 with only 2 components falling below 0.25, and five above 0.5, demonstrating that all components infer informative prior distributions of probabilities for inclusion.

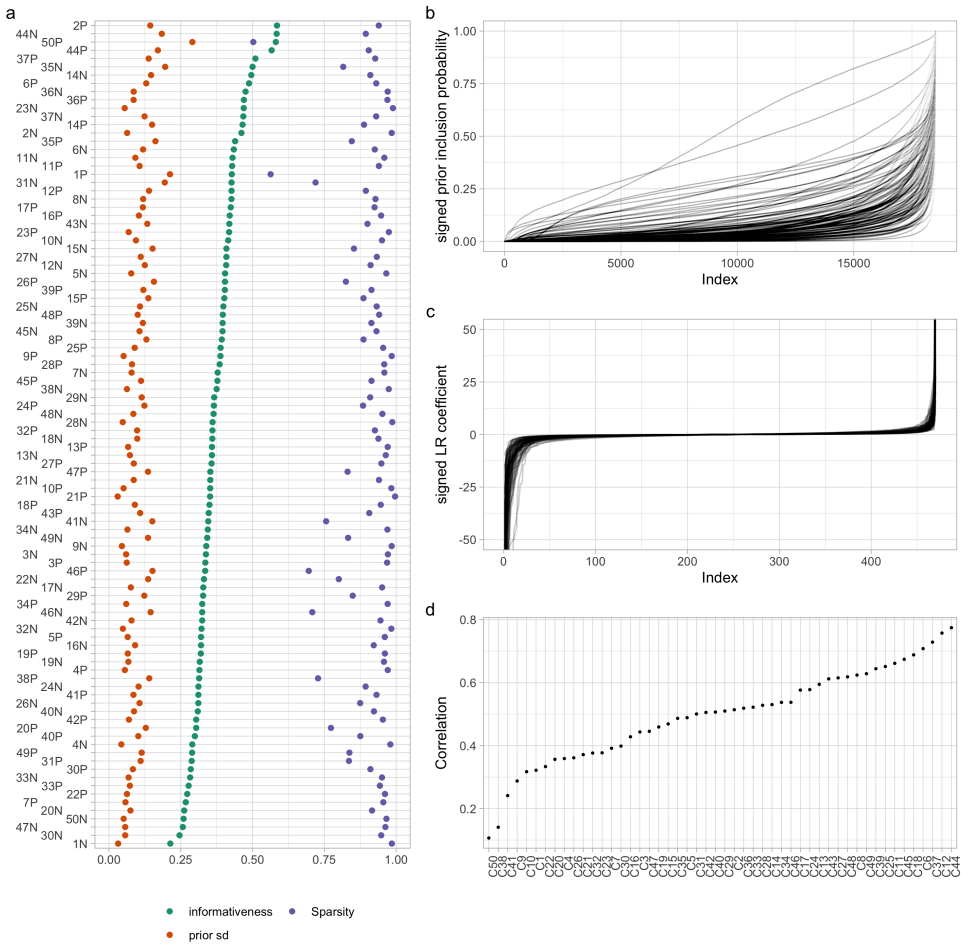

Figure 5.16: **Signed prior inclusion probabilities information content.** (a) Informativeness, sparsity and prior sd for signed component prior inclusion probabilities as described in the main text. Ordered by informativeness score. (b) Prior inclusion probabilities ordered within signed component. (c) Ordered prior annotation coefficients for each signed component. (d) Correlation of positive and negative prior inclusion probabilities for each unsigned component.

Secondly, we note that the PIPs are relatively certain taking values close to 0 or to 1, (see Figure 5.2(c)) so we also consider whether the prior inclusion probabilities are good predictors of the PIPs by considering them as a classifier. That is, for unsigned components, and signed components, we round the PIPs to be either 0 or 1 and calculate the ROC curve and AUC for each of the component classifiers (priors) by varying a threshold. The resulting curves in Figure 5.17 show that in all cases the priors are predictive of the posterior inclusion probabilities, with AUC ranging from approximately 0.7 to 0.95. In most cases, one signed prior is a better predictor than either the unsigned prior or the alternate side, and often the worst predictor within a component (the P, N or unsigned component) is the unsigned component. However, Component C46 stands out as being very well predicted by the pIPs, with AUC over 0.9 while the signed components C46N and C46P have AUC between 0.75 and 0.8. This component is unusual in that it has the most uncertainty in the PIPs, being the component with smallest slope in Figure 5.2(c), and has cell scores that vary from very strongly positive to strongly negative in a small early window of pseudotime. It is also the most dense component (Figure 5.2(a)) and has the second largest posterior mean β_q-value, suggesting that the unusually high prediction of the unsigned PIPs may be due to the identifiability issue noted earlier. That is, the high PIPs will strongly resemble those predicted from the annotations. However, the component is reliably inferred between different initialisations (Figure 5.3(b)), and the largest loadings for both C46P and C46N have strong enrichments for gene ontology terms. Component C38 also stands out, with the lowest signed AUC for C38P and the lowest unsigned AUC for C38, both just below 0.7, while still achieving AUC approximately 0.85 for the signed component C38N. On inspection, C38 appears to have positively skewed loadings, has the smallest $\langle\beta_q\rangle$ (Figure 5.2(a)) and the cell scores range from being around -1 through much of meiosis, before switching to approximately 2 through spermiogenesis in a continuous trajectory through pseudotime. Large loadings for both

C38N and C38P are significantly enriched for gene ontology terms consistent with their expression stage. We note also that C38 is consistently inferred between initialisations, and, having the smallest posterior β_q suggests it has PIPs strongly informative for inclusion and exclusion of genes in the component. We therefore suggest that the findings here are representative of a pattern of expression that is not as well predicted by the annotations present. This could be due to lack of annotations, poor quality of relevant annotations, or potentially due to the fact that C38 captures a broad wave of expression across spermiogenesis for which C38N captures early expression and later repression of a component of expression that is well predicted by the annotations, while C38P captures early repression and later expression of a component that is poorly predicted by the annotations.

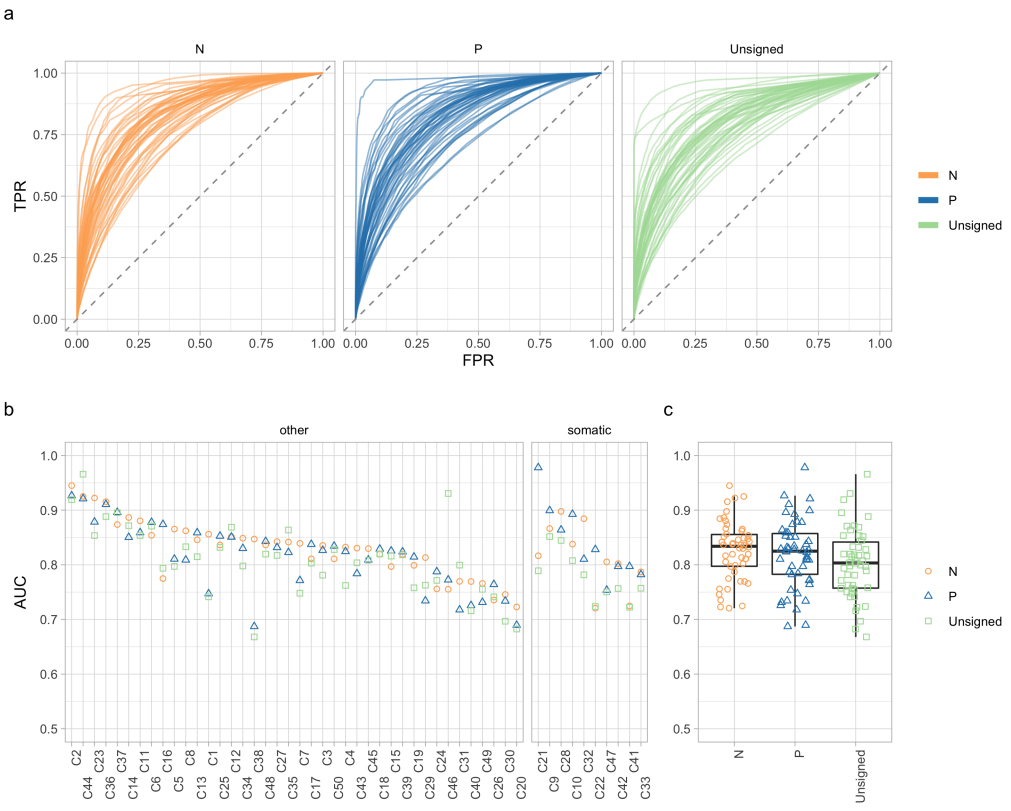

Figure 5.17: **ROC and AUC for prior and posterior inclusion probabilities.** (a) ROC curves for the prior inclusion probabilities as classifiers of the posterior inclusion probabilities. (b) AUC for the ROC curves from (a), ordered by maximal AUC for a signed component. (c) Boxplots of AUC from (b), points have horizontal jitter added.

To investigate which annotations are important in explaining the variance in the PIPs, and the complexity of the dependence on annotations, we performed forward stepwise logistic regressions on the signed PIPs with annotations as predictors. The first 30 steps in the stepwise regressions for signed components C14P and C16N are visualised in Figure 5.18(a,c). These two components show clearly that even after 30 steps, the AIC and deviance are both still decreasing, showing that the regulation of these components by annotations is complex. C14P is clearly most well explained by the AP2D transcription factor annotation, but the deviance and AIC continue to decrease as more annotations are included. Moreover, this provides evidence that a feature unique to our model, the dependence of components on multiple annotations, is appropriate to model the complexity of such biological systems. It has been noted that AP-2 family transcription factors are not highly expressed in preleptotene cells, but that the Stra8 motif is most similar to AP-2 family motifs [KdRP19], so we consider that the selection of AP2D in this component is indicative of the Stra8 regulation expected at this stage. Furthermore, [KdRP19] found that many Stra8 activated genes are also bound by E2f family transcription factors, and the transcription start sites of Stra8 bound genes had among the most enriched known motifs those of the E2f transcription factors. We note that the C14P annotations presented here include E2f and Nrf genes, both of which feature among the closest known matches to denovo motifs inferred from ChIP-seq peaks near Stra8-activated genes [KdRP19].

5.6.3 Signed pIPs over pseudo time

We further consider the correlations between signed prior inclusion probabilities and the progression over pseudotime. To order the signed pIPs over pseudotime we assigned a signed cell score to each cell and signed component as the absolute value of the cell scores that agree with the sign of the gene loading selected for this component, and 0 otherwise. By filtering by sign in this way, this results in cell scores that capture the

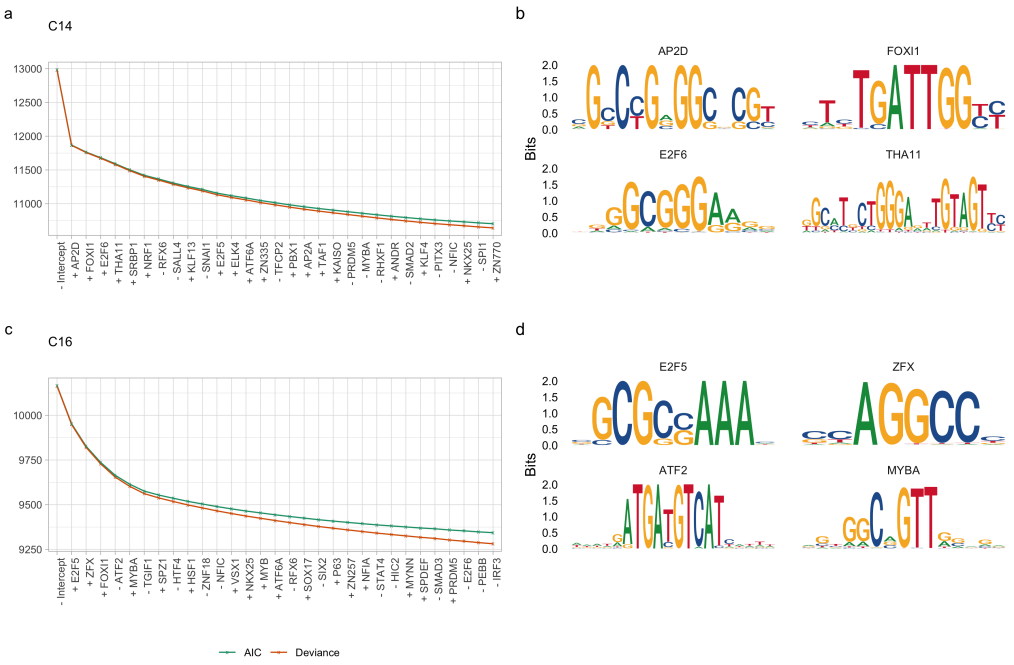

Figure 5.18: **C14 and C16 annotations.** In (a) and (c) we have carried out 30 steps of a forward stepwise logistic regression on the signed PIPs of C14P and C16N, the predominant signs of each component as described in the main text. Here we show the AIC and deviance trace over the first 30 steps, indicating the sign of the coefficient for each transcription factor below the name on the axis. (b) and (d) show example motifs explaining variance in the PIPs from the traces in (a) and (c).

expression of genes from each signed component. However, the thresholding on the scores can result in (signed) components that are expressed in early pseudotime and later pseudotime but with zero expression in the middle, for example C26N (cf. Figure 5.4(b)). For this reason, a weighted average of pseudotime as used in a previous section resulted in an ordering that was uninformative for maximal expression times, and we opted to use a rolling average of cell scores. We ordered the signed cell scores by pseudotime and for each cell assigned an average absolute signed cell score in a window of length 1000 cells centered on the cell in question. We assigned to each component the pseudotime of the maximal rolling average cell score. To identify somatic signed components we first extended the original pseudotime to include somatic cells (somatic cells were ordered by library size) and ordered components as described above, which identified 18 components for which the assigned pseudotime was outside of the range of pseudotime for non-somatic cells. Only C38N was wrongly assigned to be somatic, so we recalculated the ordering for the remaining 82 signed components and C38N, using only non-somatic cells. This results in a signed cell score ordering as illustrated in Figure B.15. We illustrate the correlation of the signed prior inclusion probabilities with components ordered by pseudotime in Figure 5.19.

The major structure present is a split between early and late pseudotime occurring in line with components C2P, C26P, and C47N. The vast majority of earlier components are positively correlated with the early components and negatively correlated with the components occurring later than these three. Similarly, the late components are positively correlated with late components and negatively correlated with the early components. At the transition between the two sets, these three components are unusual, C47N is more characteristic of the early components, C26P is correlated most positively with late components, and C2P is correlated with the later components. It is possible that issues with cell ordering in pseudotime, or uncertainty in the inferred cell scores may have caused C47N and C2P to be reversed in pseudotime. However,

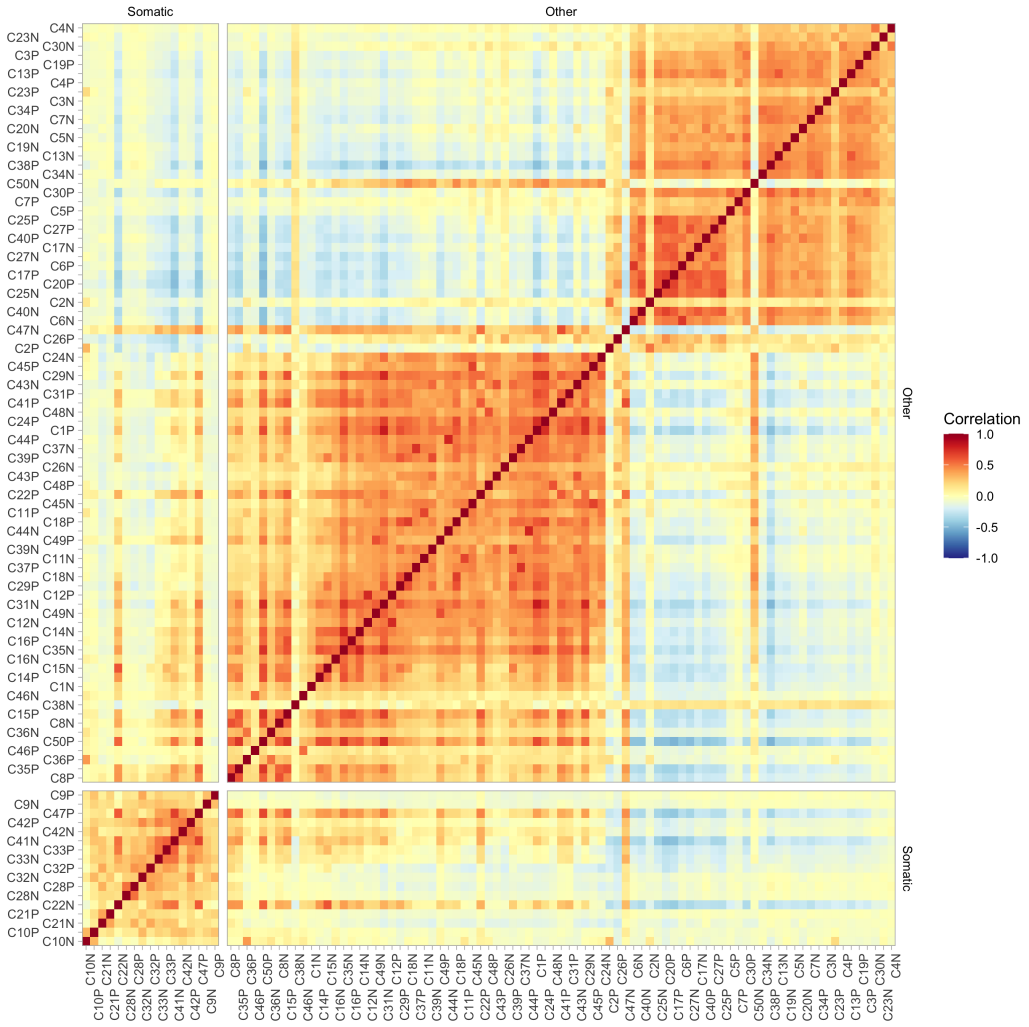

Figure 5.19: **Correlation of signed prior inclusion probabilities over pseudotime.** The somatic components are ordered alphabetically, identified as described in the main text, and the non-somatic components are ordered chronologically, left to right, with pseudotime according to a weighted average signed cell score.

C47 has loadings which are skewed to be positive and the positive cell scores C47P highlight a particular subset of somatic cells which are enriched for Hormad1 KO mouse cells, and among cells with pseudotime the largest scores occur in Hormad1 cells. The negative cell scores are fewer but largest in early pseudotime Hormad1 cells, so this rolling average has, in this case, been biased to be later by the prevalence of positive cell scores in early pseudotime (resulting in an abundance of zeros in the early pseudotime signed cell scores for C47N). Component C26P and C2P mark the early to late spermatogenesis transition consistent with our findings in Section 5.5.1.

A further component which stands out from Figure 5.19 is the late component C50N which appears to have a correlation pattern more consistent with the earlier meiotic component block. Component C50 is another component for which the large loadings are strongly positive and the scores primarily show a burst of expression in early pseudotime, roughly throughout the early stages of meiosis I until pachytene. The remaining cell scores are relatively small and negative throughout pseudotime. We note also that C50P and C50N pIPs have very low correlation so we consider that it is most likely that C50N is acting to repress genes that are consistent with the stage of spermatogenesis that C50P is expressed in. This is likely either caused by genes that are included in C50 through the dependence on the prior annotations or other components expressed at a similar time.

We further investigated the signed prior inclusion probabilities by examining the rank correlation with annotations. The correlations, including only annotations achieving at least correlation 0.4 with some signed component pIPs, are presented in Figure B.16 with components ordered identically to Figure 5.19. This shows broadly similar results to the corresponding figure for AnnotFA inferred component pIPs (see Figure 5.15), suggesting a clear switch in regulation of expression at meiotic division.

5.7 Discussion

In this chapter we have applied the AnnotFA software, as described in Chapter 3, to an existing scRNA-seq dataset from [JWR+19]. Through an extensive analysis of the inferred components from several initialisations of AnnotFA, we have shown that most components are well conserved between runs. Moreover, certain early meiotic components from [JWR+19] are explained by combinations of AnnotFA components, as described and determined using a procrustes transformation. By comparison with the previous analysis, we have shown that AnnotFA components achieve a wider range of sparsity than those inferred in an SDA analysis, and the inferred prior inclusion probabilities are informative for the posterior inclusion probabilities in the model. That is, the model has made use of the transcription factor binding annotation vectors to inform inclusion in components. We have identified novel components capturing mutant pathology, and identified components capturing broad changes in meiosis and spermatogenesis. Furthermore, inter-component correlations between both the prior and posterior inclusion probabilities capture broad stages of spermatogenesis, in particular a signature of a switch in expression at the end of meiosis. We have further investigated the annotations in the model, showing that some components are dependent on multiple annotations, and finding that the motifs and annotations indicated in the early meiotic stages are consistent with previous studies. In future work it would be useful to include an annotation based on the Stra8 motif, and carry out a similar study. Unfortunately we did not have such an annotation vector when carrying out this study. This additional vector, if distinct enough from the existing AP2D vector, would provide a useful case study for sensitivity analysis in the early meiotic cells. We would also like to include annotations derived from GO terms.

We note that the signature of dependence on annotations inferred in this chapter showed usually several components, closely aligned in pseudotime, were associated with annotations suggesting that the transcription factor activation may well capture

a broad part of the transcriptional dynamics, but other regulation may be driving the fine scale differences between components. A similar observation was made in [JWR+19], where it was suggested that the fine scale differences might be described by transcription factor binding to more distant enhancer regions, or other factors.

We would also like to carry out a similar analysis of other scRNA-seq datasets (for example [LBEW18]) from mouse testes, with a view to compare and contrast both the performance of AnnotFA on datasets generated from different scRNA-seq technologies, and the components inferred between different datasets.

There are several directions suggested by this study for future methodological development. Firstly, we note that the model implemented here incorporates components that have loadings (and scores) of both positive and negative sign. In this case we showed that often the negative sign loadings signify repression when the scores are positive and expression when the scores are negative (and vice versa for the positive loadings). Moreover, in certain components with a dominant sign, the opposite signed component was found to align more closely with other components expressed at similar times. We further analysed "signed components" where we split the loadings into one sign or the other for all components and found that these reflected similar structure to the "two-sided" components. However, this model has successfully incorporated annotations into the inference. We therefore suggest that a model with "single-sided" components, still incorporating annotations, and a suitable likelihood model for scRNA-seq data (that is, a Poisson measurement model [SS20]), which would remove the need for normalisation, would be a useful direction for future research.

One challenge we encountered is the collinearity in the matrix of annotations, and here we have been cautious and thinned the annotations. An alternative approach would be to use a penalized logistic regression update, such as a ridge or lasso regression penalty term. For example, the coefficient shrinkage and variable selection of the lasso

would yield more immediately interpretable results.

CHAPTER 6

Discussion

The widespread use of RNA-seq technology has catalysed advancements in the understanding of biology, highlighting previously unknown or poorly understood aspects of gene expression. For example revealing the extent of RNA splicing [WSL+08], the regulation of gene expression by non-coding RNAs [MM14], and enhancer RNAs [SGH19]. It is expected that bulk RNA-seq will remain a widely used and useful approach [SGH19], providing insights into differentially expressed genes between tissues. However, transcription is a highly dynamic process, the RNA profile of cells is constantly changing and the use of single cell RNA-seq data provides increased cellular resolution beyond that of bulk methods providing insights into developmental processes, cell types and cell-type transcriptional markers.

In this thesis we have focussed on latent variable models for the analysis of RNA-seq data, predominantly related to factor analysis, noting in particular that such an approach avoids the pitfalls of a hard clustering of cells, instead allowing for a "soft clustering" approach that infers both components of co-expressed genes and their

expression levels across cells. We have presented methods for incorporating additional contextual or annotation information (functional or regulatory, for example) into a (parallel) factor analysis model for single cell or bulk RNA-seq data.

The first method we developed here is presented in Chapter 2, building on an existing sparse Bayesian parallel factor analysis method for bulk RNA-seq data. Extending the SDA [Hor15, HVB+16] model from three dimensions to four dimensional gene expression data, this allows for an additional contextual or time series dimension in the data tensor. We have made the software for this method, SDA4D, available as an R package. In Chapter 2 we demonstrate on simulated data that additional contextual data provides power to more accurately infer the patterns of sample scores and gene loadings. Moreover, the four dimensional model accurately infers a parsimonious representation of the structure of the data. The same four dimensional data, with the tissue and time dimensions unfolded into a single context dimension led to almost identical recovery of the gene loadings and sample scores when using the 3D model. We note however, that the three dimensional model does not capture the four dimensional structure, and it is not clear how one might disentangle the inferred context dimension into separate time and tissue dimensions. We note also that we might expect that for many applications of bulk RNA-seq data much of the variability in the data will be explained in the context and gene dimensions, and the additional time dimension will provide valuable additional insights to the transcriptomic dynamics present in the data. Though we have not explored it here, the SDA4D software would benefit from extensions to accommodate missing data, as is present in many genetic datasets [Hor15], though care is needed in such applications, due to the possibility of confounding as discussed in Chapter 2.

The prevalence and size of single cell RNA-seq data presents an opportunity to integrate other external datasets into analysis pipelines. It is possible to apply methods such as SDA4D to scRNA-seq data. However, in this context we suggest that many

sources of context (tissue, time for example) may be less clearly defined. For example, time series context labels may not be appropriate as the stage of development of cells on a developmental trajectory might be more informative than the time a sample is assayed, and cells included from a tissue sample may include cells which are not tissue specific. For example, many tissue samples will include blood cells. Considering this, it may be better to model the variability across cells and genes directly. In this work we take such an approach, with a factor analysis model, introducing a sparse Bayesian prior structure that integrates annotations into the inclusion probabilities of genes in components and adaptively selects the informative annotations in a data driven way. The predominant trend in the literature has been to use a Gaussian noise model, but several recent findings have highlighted that after accounting for library size, single cell RNA-seq data is consistent with a Poisson distribution [Sve20, SS20]. For this reason we developed a model and inference algorithm for both Poisson and Gaussian likelihood models, with the Poisson model incorporating the library size. The model and implementation are described in Chapter 3.

In Chapter 4 we investigate the relative merits of each of the models and compare against the factor analysis model implemented as part of SDA, which has previously been applied to scRNA-seq data [JWR$^+$19]. Early simulations close to the model but not representing expression dynamics well were not included in this work, but served to highlight the interpretability issues we mentioned in Chapter 4. Specifically that the rate function $\lambda : x \mapsto \log(1+\exp(x))$, as suggested in [SB12] and used in other methods such as MOFA [AVA$^+$18], f-scLVM [BPM$^+$17], causes some difficulties in interpretation. We found that the model would explain variance in the data with a suitable number of components but the inferred components were not always linearly related to the simulated components. Instead, each inferred component is best understood through considering the link function which necessitates considering the reconstruction based on all other components. This inference method is similar to that used in MOFA

and f-scLVM (all based on that suggested in [SB12]) which approximates the Poisson likelihood by a Gaussian likelihood and iteratively updates the pseudo-data that the underlying Gaussian model fits to. In Chapter 4 we demonstrate on several datasets simulated from one protocol that the Poisson model reliably captures the most variable components but the finer grained components that are more sparse and explain less variance are not captured. On the other hand, after normalising the library size of the data, we found the Gaussian model reliably captured all of the components, explained more variance in the data and outperformed the Poisson model and SDA. Moreover, AnnotFA made effective use of the annotations when available, the inclusion probabilities were reliably inferred and the correct annotations were included in components. We note also that a feature of the model is the cell-wise Gaussian noise precision in the model. During development we tested the Gaussian model on the Poisson simulated data with normalisation carried out as described in [JWR+19]. This follows a library size normalisation, followed by square-root variance stabilisation and scaling genes to unit variance. However, we found that the inferred components were well correlated with the true components only when conditioning on the library size, in contrast to the SDA inferred components which were well correlated showing limited effect of library size. This is due to the variance stabilisation occurring after the library size normalization. After removing the intermediate step of square-root variance stabilisation for the Gaussian model we recovered excellent correlation with the simulated components, outperforming SDA. Furthermore, in applications in Chapter 5 we compared the dimension reductions resulting from UMAP and t-SNE on AnnotFA cell scores with those from the SDA cell scores from [JWR+19]. As described in [JWR+19] the SDA dimension reduction shows a strong radial effect of library size in which distance from the center is strongly correlated with library size, while there is no evident effect of library size from the AnnotFA dimension reductions. We conclude that after accounting for library size by row-sum normalising the DGE matrix, the

Gaussian model with cell-wise noise variance captures the sources of variance in the data well, and the prior structure described in this work makes effective use of the annotations in the model.

In Chapter 5 we applied AnnotFA to an existing single cell RNA-seq dataset, from a drop-seq experimental protocol applied to over 20000 cells originating from testes. The cells were from a variety of mouse lines including several mutant mice for which spermatogenic cells are known to arrest at certain stages, resulting in a dataset enriched for early meiotic cells. Following the original analysis [JWR+19] we analysed all cell lines together and with 50 components, plus an intercept. To incorporate annotations that may be informative for biological mechanisms driving transcription, we used a number of annotations from [JWR+19] consisting of probabilities of transcription factor binding to a promoter region of each gene. The inferred components showed consistency across initialisations and several components clearly explain similar variation to that explained by the components from [JWR+19]. By careful consideration of all inferred components we have identified a number of batch effects, a number of components that capture biological mutant pathology, and identified the signatures of testes-specific biological phenomena such as meiotic sex chromosome inactivation. We confirmed that the pseudotime from [JWR+19] was consistent with dimension reductions based on AnnotFA components scores, and used this pseudotime extensively. In particular, we observed that many components scores show a continuous trajectory as pseudotime progresses, demonstrating the power of factor analysis to explain variance that is continuous over time. Furthermore, the posterior inclusion probabilities show clear patterns of correlation that concur with known major biological stages of spermatogenesis, most strikingly a significant split between meiotic components and spermiogenic components which are often negatively correlated. We further demonstrated that this same signal is present in the prior inclusion probabilities which are constructed from the annotations, and confirmed that the prior inclusion

probabilities are informative for the posterior inclusion probabilities. On inspection of the correlation of the transcription factor annotations with the prior inclusion probabilities, we observe, similarly to [JWR+19] that the annotations driving meiotic components of expression have motifs typically enriched for CpG dinucleotides. The AnnotFA components are strongly informed by annotations, moreso than those of SDA which only aims to induce sparsity in components. Due to the biological relevance of the annotations used, we suggest that this means the AnnotFA components are more representative of the underlying biological processes. Furthermore, AnnotFA components achieve a wider range of sparsity than those of SDA, capturing components both more sparse and less sparse than those of SDA, and we find that the dependence on annotations is complex but meaningful. For example, a signature of motifs consistent with the Stra8 initiated meiotic transcriptional program is clearly present in early meiotic components.

Overall, this application confirms that the method is working effectively, can be applied to large datasets, and has recovered both known results from a previously studied dataset, and also uncovered previously unknown components describing mutant pathology. Moreover, by using transcription factor binding probabilities derived from motifs, we have shown that the variation explained by this model is informed by transcriptional regulation that is predicted from DNA sequence.

6.1 Future Directions

There are a number of extensions of the model that we would like to explore. First and foremost, we would like to improve upon the integration of the annotations. Currently the dependency on annotations is incorporated by an Empirical Bayes methodology making use of a logistic regression update. This has proven useful but we found with the application to large sets of annotations in Chapter 5 that several annotations were

collinear requiring thinning of the set. While we have not quantified the effects of this, it is possible that informative annotations for this dataset were removed. For this reason we suggest that a penalised logistic regression update might be used (such as a ridge or lasso regression), facilitating the use of a larger more complete set of annotations.

Further to this, many biologically informative annotations may be better incorporated through interaction terms in the regression. For example, while we have made use of transcription factor binding affinities as annotations, binding is only possible on open chromatin, so chromatin accessibility profiles for sorted cell lines will be informative for transcriptional regulation and may be best incorporated as interaction terms with transcription factor binding affinities.

To make further use of the current model and similar annotations corresponding to transcription factor binding affinities predicted from motifs, and noting that transcription factors can only bind to open chromatin, we would like to apply AnnotFA to single cell ATAC-seq data, which measures chromatin accessibility across the genome at a single cell resolution.

We also expect that the model would be powerful when used with other annotations, such as chromatin accessibility profiles, DNase activity profiles, GO annotations, histone mark profiles, or ChIP-seq assays. While several of these would only be available from separate studies, presenting challenges for integration, recent efforts in genomics have aimed to curate collections of such epigenetic maps, such as EpiMap [BJP$^+$21], providing a valuable resource for future work.

Beyond this, we have incorporated a similar prior structure that leverages annotations for the cell scores. We expect this to have wide applicability with use cases including tissue labels, time points, patient ID, case-control status, or treatment status, cellular phenotype for example. We have not tested such applications in this work but expect it to work similarly well.

A modelling extension would be to accommodate additional data types in a group factor analysis model similarly to MOFA [AVA+18], with a common cell scores matrix and the different data modalities each having a loadings matrix. This is a natural model to integrate different data types, and could accommodate different sets of annotations for each datatype. Although many applications are possible, it would be interesting to apply such a model to the abundances of unspliced (nascent) RNA and spliced RNA such as are used in the RNA velocity model [LMSZ+18]. It may be possible to identify components of variation that complement RNA velocity this way.

We also anticipate application of this model to existing bulk RNA-seq datasets and suggest that it may be useful to implement the prior structure in a parallel factor analysis model.

Factor analysis is a widely used statistical model. For example, in genomics it has been applied in a number of settings, from population genetics [ES10, FJ20], to determining latent factors for integration into polygenic risk scores [JHG+20], and gene expression studies. For this reason we expect that the model presented here can be applied in many analysis settings as an effective method of integrating external information. All of the above extensions will facilitate the use of the model in a wide variety of settings. We also note that the current implementation scales to the size of datasets we expected it to be used for. However, considering the exponential scaling of single cell RNA-seq data, and diverse number of applications that may be possible, we expect to implement a stochastic variational inference [HBWP13] version of the algorithm. Stochastic variational inference has been used in a (group) factor analysis context in MOFA+ [AAB+20] and applied to single cell data, but does not feature the prior structure facilitating incorporating prior information that we have developed in this work. We note in particular that a careful design of the software would facilitate fast stochastic variational inference while allowing for multiple use cases such as factor analysis, group factor analysis, and parallel factor analysis.

APPENDIX A

Supplementary Information

A.1 Empirical Bayes, Variational Bayes and Variational EM

A.1.1 The EM algorithm

Here, we briefly outline the Expectation-Maximisation (EM) algorithm [DLR77]. In maximum likelihood inference under a probabilistic model, it is common to have observed data X, and a set of parameters ξ, and often unobserved latent variables Θ. In models without latent variables where, for instance, the model is summarized by an analytical likelihood function $L(\xi) = \mathbb{P}(X|\xi)$, based entirely on the parameters ξ, inference of the parameters ξ is often approached by finding the maximum likelihood estimator $\hat{\xi}$, that is, the value of ξ that maximises the likelihood function.

In the presence of latent variables Θ, the likelihood $\mathbb{P}(X, \Theta|\xi)$ is a function of X, Θ, and the unknown parameters ξ. In this case, the maximum likelihood estimate is the

value of ξ maximising the marginal likelihood,

$$\mathbb{P}(X|\xi) = \int \mathbb{P}(X, \Theta|\xi) d\Theta. \tag{A.1}$$

In certain cases this is tractable to compute analytically, but in many cases, this is not possible. For example, even in a discrete model involving Θ such as a hidden markov model (HMM) with a discrete hidden state space, the latent variables are states in a sequence, so the number of possible states of the complete sequence grows exponentially with the length of the sequence of states. In such cases with intractable marginal likelihoods, the EM algorithm is often used. The EM algorithm typically proceeds by initialising a guess at the parameters $\xi^{(0)}$, and proceeds by iteratively updating the parameter estimate by alternating between the E-step, which calculates

$$Q(\xi|\xi^{(t)}) = \mathbb{E}_{\Theta|X,\xi^{(t)}} \log \mathbb{P}(X, \Theta|\xi); \tag{A.2}$$

and the M-step, which updates $\xi^{(t)}$ to $\xi^{(t+1)}$, a value which maximises $Q(\xi|\xi^{(t)})$.

It has been shown analytically that each step of the algorithm leads to an increase in the marginal likelihood (Equation A.1), and thus will eventually find a local optimum. An alternative formulation of the EM algorithm was given in [NH98] (see also [Bis06]), and we present this here as it provides justification that each step is increasing the marginal likelihood, and clearly emphasises the connection with variational Bayes and the inference method we apply in this work.

Choosing a distribution $q(\Theta)$ on the latent variables Θ, the log marginal likelihood separates into two terms as

$$\log \mathbb{P}(X|\xi) = \mathcal{F}(q, \xi) + \mathrm{KL}(q(\Theta)|\mathbb{P}(\Theta|X, \xi)), \tag{A.3}$$

where $\mathrm{KL}(q(\Theta)|\mathbb{P}(\Theta|X,\xi))$ is the Kullback–Leibler divergence, defined as

$$\mathrm{KL}(q(\Theta)|\mathbb{P}(\Theta|X,\xi)) = -\int q(\Theta)\log\left(\frac{\mathbb{P}(\Theta|X,\xi)}{q(\Theta)}\right)d\Theta, \tag{A.4}$$

and the term $\mathcal{F}(q,\xi)$ is defined as

$$\mathcal{F}(q,\xi) = \int q(\Theta)\log\left(\frac{\mathbb{P}(X,\Theta|\xi)}{q(\Theta)}\right)d\Theta. \tag{A.5}$$

The Kullback–Leibler divergence between two distributions is not symmetric, but is non-negative, and is equal to zero precisely when the two distributions are equal. It follows that $\mathcal{F}(q,\xi) \leq \mathbb{P}(X|\xi)$. That is, $\mathcal{F}(q,\xi)$ is a lower bound for the log marginal likelihood.

Viewed from this perspective, the EM algorithm proceeds by iterating between the E-step, which consists of holding $\xi = \xi^{(t)}$ and maximizing $\mathcal{F}(q,\xi^{(t)})$ with respect to the distributions q, and the M-step in which the resulting lower bound is maximized with respect to ξ.

The marginal likelihood is constant with respect to q, so in the E step, the lower bound $\mathcal{F}$ is maximised with respect to q when the Kullback–Leibler divergence is minimized. This is precisely when $q^{(t)}(\Theta) = \mathbb{P}(\Theta|X,\xi^{(t)})$. Now, in the M step, maximizing $\mathcal{F}(q^{(t)},\xi)$ with respect to ξ results in a new value $\xi^{(t+1)}$ such that $\mathcal{F}(q^{(t)},\xi^{(t+1)}) \geq \mathcal{F}(q^{(t)},\xi^{(t)})$. In general, $q^{(t)}(\Theta) \neq \mathbb{P}(\Theta|X,\xi^{(t+1)})$, so the Kullback–Leibler divergence increases to being non-zero and this demonstrates that the log marginal likelihood has been increased by the M step, and by more than the increase in the lower bound. Moreover, it can be shown that after the E-step, the lower bound $\mathcal{F}$ is equal (up to a term constant with respect to ξ) to $Q(\xi|\xi^{(t)})$, so the typical description of the EM algorithm, as previously given, is recovered in this setup.

A key point in this alternative view of the EM algorithm is that in both the E-step and the M-step, the objective is maximization. From this viewpoint, the algorithm

can be generalized in a number of ways when the original formulation may remain intractable. This is further explored and justified in [NH98], showing that if (q, ξ) is a local maxima of $\mathcal{F}(q, \xi)$ then ξ is a local maxima of the log marginal likelihood, while if (q, ξ) is a global maxima of $\mathcal{F}(q, \xi)$ then ξ is a global maxima of the log marginal likelihood.

A.1.2 Variational Bayes

A central aim of Bayesian inference is to compute the posterior distribution of parameters in a statistical model in which a generative model is specified for data, based on a likelihood model, and a prior structure specified for each of the latent variables and parameters. In many cases a precise description of the posterior distribution is intractable, and approximate methods can be useful. One such approximate inference, or Variational Inference (VI) approach is Variational Bayes (VB). Variational methods are well covered in the literature, see for example [BKM17, Bis06]. We note here that variational Bayes is known to be fast and produce good posterior mean estimates, but it is prone to underestimating posterior variance. We now briefly describe a commonly used approach to mean-field variational Bayes, sometimes referred to as coordinate ascent variational inference (CAVI).

For the purposes of this section we write Θ for all latent variables and parameters, and assume prior distributions have been specified for Θ.

In this setting the log marginal likelihood, or model evidence, is denoted $\mathbb{P}(X)$, and similarly to the discussion of the EM algorithm, we can express the log marginal likelihood as

$$\log \mathbb{P}(X) = \mathcal{F}(q) + \mathrm{KL}(q(\Theta)|\mathbb{P}(\Theta|X)), \tag{A.6}$$

where

$$\mathcal{F}(q) = \int q(\Theta) \log \left(\frac{\mathbb{P}(X, \Theta)}{q(\Theta)} \right) d\Theta. \tag{A.7}$$

Following an identical discussion to that given in Section A.1.1, minimising the Kullback–Leibler divergence is equivalent to maximizing the quantity $\mathcal{F}(q)$, which is a lower bound to the log marginal likelihood. This lower bound is often called the evidence lower bound, or ELBO, and is the objective function to be maximised in variational Bayesian inference.

Variational inference proceeds by placing some assumptions on the family of the distributions q over which to optimize the ELBO. Without such an assumption, the global optimum occurs at the posterior, and a balance must be struck between simplifying the optimization, and maintaining enough structure to adequately approximate the posterior. For a spike and slab prior structure, this is well demonstrated in [TLG11]. In (structured) mean field variational Bayes a partition is specified on Θ, partitioning the parameters into subsets $\{\Theta_i\}_i$. The inherent assumption in the method is that in the variational posterior, the different subsets are treated as independent. The posterior distribution is thus approximated by a distribution $q(\Theta) = \prod_i q_i(\theta_i)$, by minimizing the Kullback–Leibler divergence (see Equation A.4) between the posterior $\mathbb{P}(\Theta|X)$ and $q(\Theta)$. We note that since the posterior will likely have dependencies between variables, the family of distributions determined by these independence assumptions will rarely contain the true posterior, but this simplifying assumption facilitates a deterministic inference procedure.

Making use of the factorisation $q(\Theta) = \prod_i q_i(\Theta_i)$, we can rearrange the defining equation for the ELBO as follows:

$$\begin{aligned}
\mathcal{F}(q) &= \int q(\Theta) \log\left(\frac{\mathbb{P}(X,\Theta)}{q(\Theta)}\right) d\theta \\
&= \mathbb{E}_{q_k(\Theta_k)}\left(\int q_{-k}(\Theta_{-k})(\log(\mathbb{P}(X,\Theta) - \log(q(\Theta)))d\Theta_{-k}\right) \\
&= \mathbb{E}_{q_k(\Theta_k)}\left(\int q_{-k}(\Theta_{-k})(\log(\mathbb{P}(X,\Theta) - \log(q_{-k}(\Theta_{-k})))d\theta_{-k} - \log(q_k(\Theta_k))\right) \\
&= \mathbb{E}_{q_k(\Theta_k)}\left(\mathbb{E}_{q_{-k}(\Theta_{-k})}(\log(\mathbb{P}(X,\Theta))) - \mathbb{E}_{q_{-k}(\Theta_{-k})}(\log(q_{-k}(\Theta_{-k}))) - \log(q_k(\Theta_k))\right)
\end{aligned}$$

Since the second term on the right hand side is constant with respect to q_k, if we hold all q_j constant for $j \neq k$, the second term is constant and the variable terms can be rewritten as $-\mathrm{KL}(q_k(\Theta_k)|\tilde{\mathbb{P}}(X,\Theta)$ where $\tilde{\mathbb{P}}(X,\Theta)$ is the distribution defined by

$$\log \tilde{\mathbb{P}}(X,\Theta) = \mathbb{E}_{q_{-k}(\Theta_{-k})}(\log \mathbb{P}(X,\Theta)) + \text{constant}. \tag{A.8}$$

Thus, $\mathcal{F}(q)$ is maximized with respect to $q_k(\Theta_k)$ by

$$q_k^*(\Theta_k) = \frac{1}{z} \exp(\mathbb{E}_{q_{-k}(\Theta_{-k})}(\log \mathbb{P}(X,\Theta))), \tag{A.9}$$

where z is a normalising constant.

Equations A.8 and A.9 form the basis of coordinate ascent variational inference, and variational Bayes is thus implemented by initialising all $q_i(\Theta_i)$ and iteratively updating each q_i while holding each q_{-i} fixed. In practice, when conjugate priors are used, the form for each $q_k^*(\Theta_k)$ is analytical, and the distribution of $q_k^*(\Theta_k)$ is dependent on the moments of $q_i(\Theta_i)$ for $i \neq k$. For parameters where conjugate priors are not used, and no analytical form of the distribution is available, point estimates for the parameters are used, and the ELBO is optimized using standard optimisation methods.

In practice convergence of such inference algorithms is difficult to assess, and stopping conditions vary from implementation to implementation. For example, in SDA [HVB$^+$16], the stopping condition is based on monitoring rates of change of PIPs, while in practice in [JWR$^+$19] SDA was used without the stopping condition and the ELBO trajectory visualized to confirm convergence. Moreover, this form of inference is prone to finding local optima, and is often sensitive to initialisation so it is typically advisable to run several initialisations and keep the one achieving the highest ELBO.

A.1.3 Empirical Bayes

In empirical Bayes, a Bayesian model is defined with some unknown hyperparameters ξ which do not have prior distributions placed on them. In this case, the parameters ξ are determined by maximising the marginal likelihood.

In order to proceed with this approach in a Variational Bayesian methodology, we note that the algorithms described in Sections A.1.1 and A.1.2 are compatible. That is, we can consider the unknown hyperparameters ξ as additional parameters in the ELBO (Equation A.7) and the log-marginal likelihood, and alternate between maximising the ELBO with respect to the variational distributions q_i and the unknown hyperparameters ξ. This is equivalent to applying the EM algorithm with the E step replaced by an approximate E step, maximizing the lower bound with respect to q, subject to the mean-field factorisation in Section A.1.2. Such an approach results in updates identical to those in the variational Bayes algorithm for the distributions q_k and an M-step optimization of the ELBO for the unknown hyperparameters ξ. A similar approach is taken as a part of the inference algorithm for a Latent Dirichlet Allocation model in [BNJ03].

A.2 L-BFGS algorithm

In this section we briefly describe the low memory BFGS algorithm for numerical optimisation.

A.2.1 The Newton-Raphson Method

We consider a differentiable function $f : \mathbb{R}^n \mapsto \mathbb{R}$. We briefly recall the Newton–Raphson method optimizing multivariate functions of this form before describing the BFGS and L-BFGS algorithms. The standard Newton–Raphson method for functions $g : \mathbb{R} \mapsto \mathbb{R}$ can be generalised for multivariate functions such as f by replacing the

derivative $\frac{dg}{dx}$ by the gradient $\nabla f(\mathbf{x}) = \left(\frac{\partial f}{\partial x_i}\right)$ and the inverse of the second derivative $\frac{d^2g}{dx^2}$ by the inverse $H_f(\mathbf{x})$ of the Hessian matrix, that is,

$$H_f(\mathbf{x}) = \left(\frac{\partial^2 f}{\partial x_i \partial x_j}\right)^{-1}.$$

A standard derivation of the Newton–Raphson method involves approximating f by the second order Taylor expansion of f, and iteratively updating point estimates $\mathbf{x}_i$ to $\mathbf{x}_{i+1}$ in the locally optimal direction. The direction of the Newton update $t(\mathbf{x})$ is obtained by setting the first derivative of the second order Taylor expansion of f to zero, resulting in

$$t(\mathbf{x}) = -H_f(\mathbf{x})\nabla f(\mathbf{x}).$$

An update for $\mathbf{x}_i$ is then typically defined as $\mathbf{x}_{i+1} = \mathbf{x}_i + \gamma_i t(\mathbf{x}_i)$ for some $\gamma_i > 0$. A basic Newton–Raphson update takes the value $\gamma_i = 1$, but often other methods are used to choose appropriate values of γ_i, for example backtracking where γ_i is decreased systematically until the objective function $f(\mathbf{x})$ decreases.

A.2.2 The BFGS and L-BFGS algorithm

In a family of algorithms collectively known as quasi-Newton methods, the inverse Hessian is not computed but is approximated using updates based on the history of the computation of the gradient [Sha70], generalising the secant method in which gradients are replaced by quotients of differences. The Broyden–Fletcher–Goldfarb–Shanno (henceforth BFGS) algorithm is a popular quasi-Newton method.

To be more precise, quasi-Newton methods use a quadratic model of the objective function f:

$$m_i(\mathbf{t}) = f(\mathbf{x}_i) + \nabla f(\mathbf{x}_i)^T\mathbf{t} + \frac{1}{2}\mathbf{t}^T\mathbf{B}_i\mathbf{t} \tag{A.10}$$

where B_i is a symmetric positive definite matrix and is updated at every iteration.

The minimiser $\mathbf{t}_i = -\mathbf{B}_i^{-1}\nabla f_i$ of Equation A.10 is the step direction for iteration $i+1$. The next iterate is defined to be $\mathbf{x}_{i+1} = \mathbf{x}_i + \gamma_i \mathbf{t}_i$ where γ_i is usually determined by a line search to satisfy certain conditions. This procedure is typically repeated until convergence, which is measured by the norm of the gradient falling below a tolerance. For brevity, we omit the precise mathematical formulation and motivation but refer the reader to the details in the original papers [Sha70, Bro70, Fle70, Gol70].

After each iteration, two quantities are updated, $\mathbf{s}_i = \mathbf{x}_{i+1} - \mathbf{x}_i$ and $\mathbf{y}_i = \nabla f(\mathbf{x}_{i+1}) - \nabla f(\mathbf{x}_i)$, and the BFGS update for the approximate inverse Hessian matrix is as follows:

$$\mathbf{H}_{i+1} \;=\; \left(I - \rho_i \mathbf{s}_i \mathbf{y}_i^T\right) \mathbf{H}_i \left(I - \rho_i \mathbf{y}_i \mathbf{s}_i^T\right) + \rho_i \mathbf{s}_i \mathbf{s}_i^T \text{ with } \rho_i = \frac{1}{\mathbf{y}_i^T \mathbf{s}_i}.$$

The limited memory BFGS algorithm (L-BFGS) maintains a history of length, say, m, of vectors $\mathbf{s}_i, \mathbf{y}_i$ and the above update is iteratively expanded as a sequence of $O(m)$ vector-vector products and a matrix-vector product with the initial H_0. For further information and references, see [Mur12].

APPENDIX B

Supplementary Figures

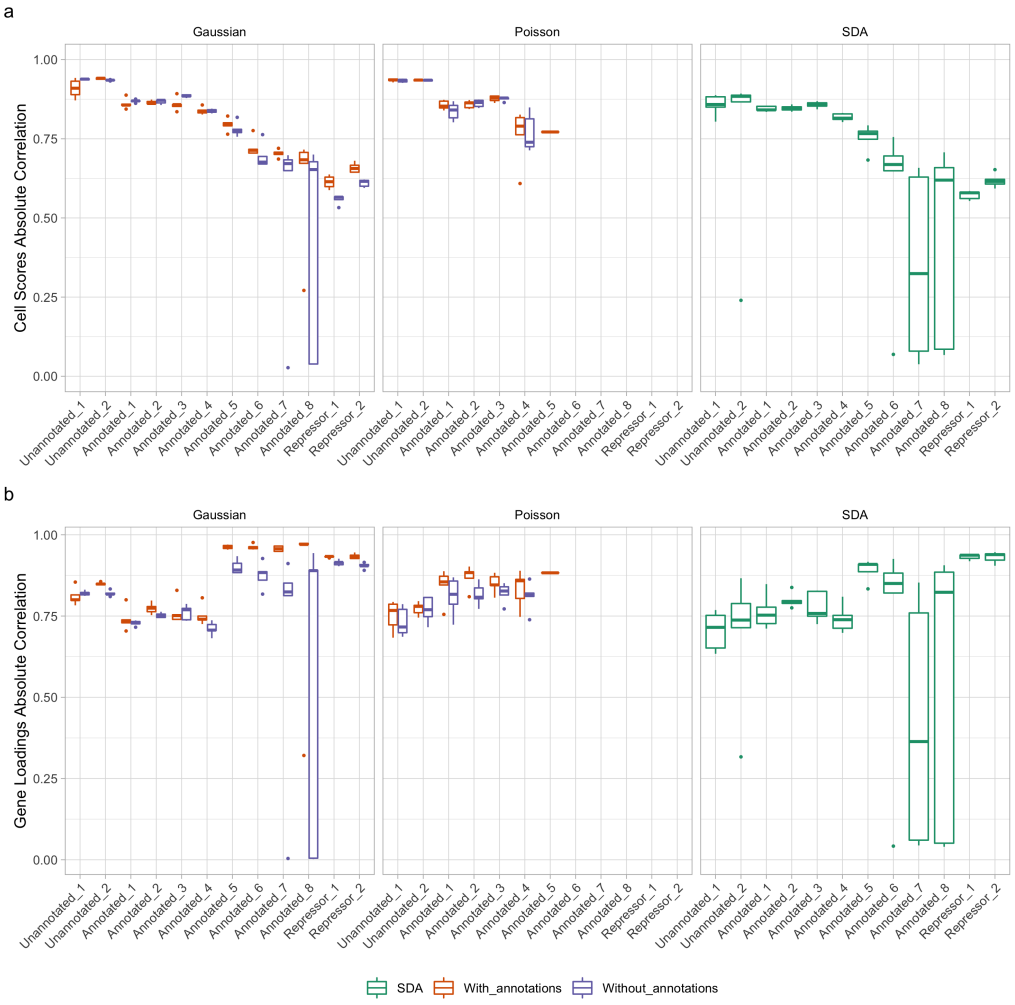

Figure B.1: **Absolute Correlation of aligned components.** Absolute correlation of aligned components from the count data simulation in Section 4.1.2.

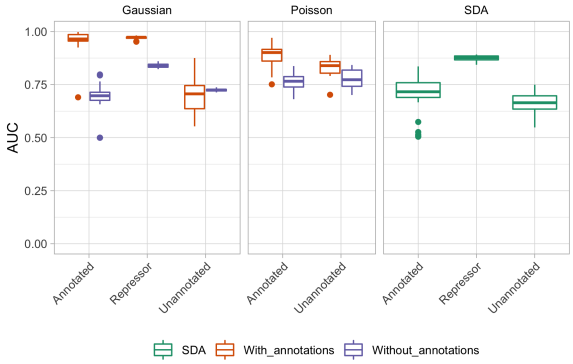

Figure B.2: **Componentwise PIP AUC.** AUC of posterior inclusion probabilities from AnnotFA and SDA models as predictors for the simulated component gene inclusions in the count data simulation from Section 4.1.2.

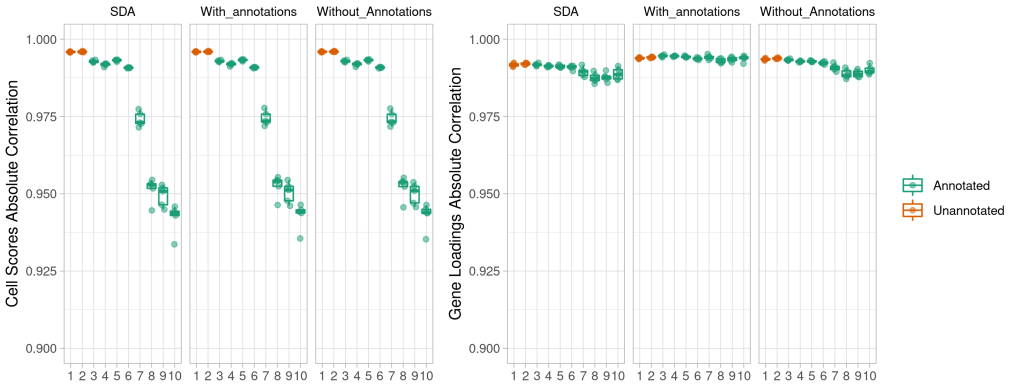

Figure B.3: **Absolute Correlation of Aligned Components.** Absolute correlation of aligned components from the Gaussian simulation in Section 4.2.

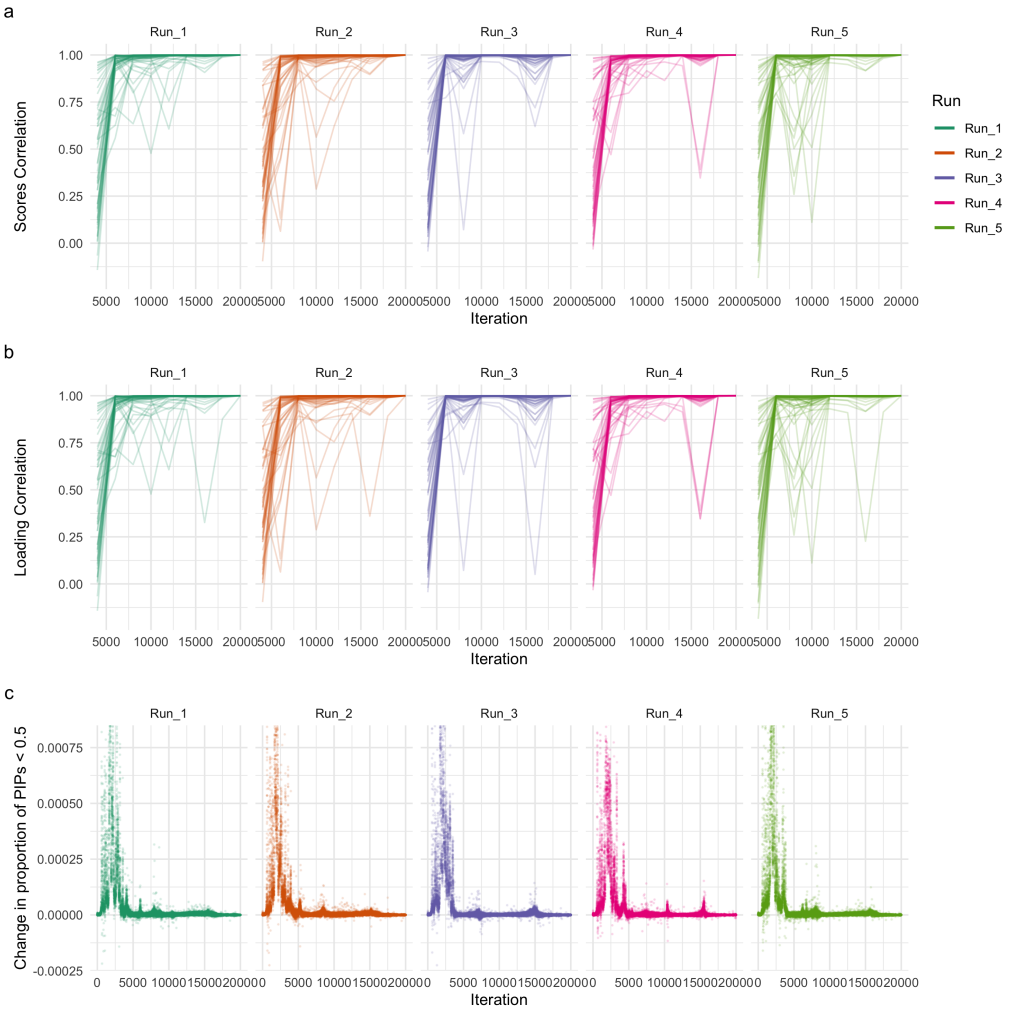

Figure B.4: **Convergence of AnnotFA across runs.** (a) Componentwise correlation to previous checkpoint 2000 iterations before for (a) cell scores, and (b) gene loadings . (c) Change in proportion of PIPs below 0.5 per iteration.

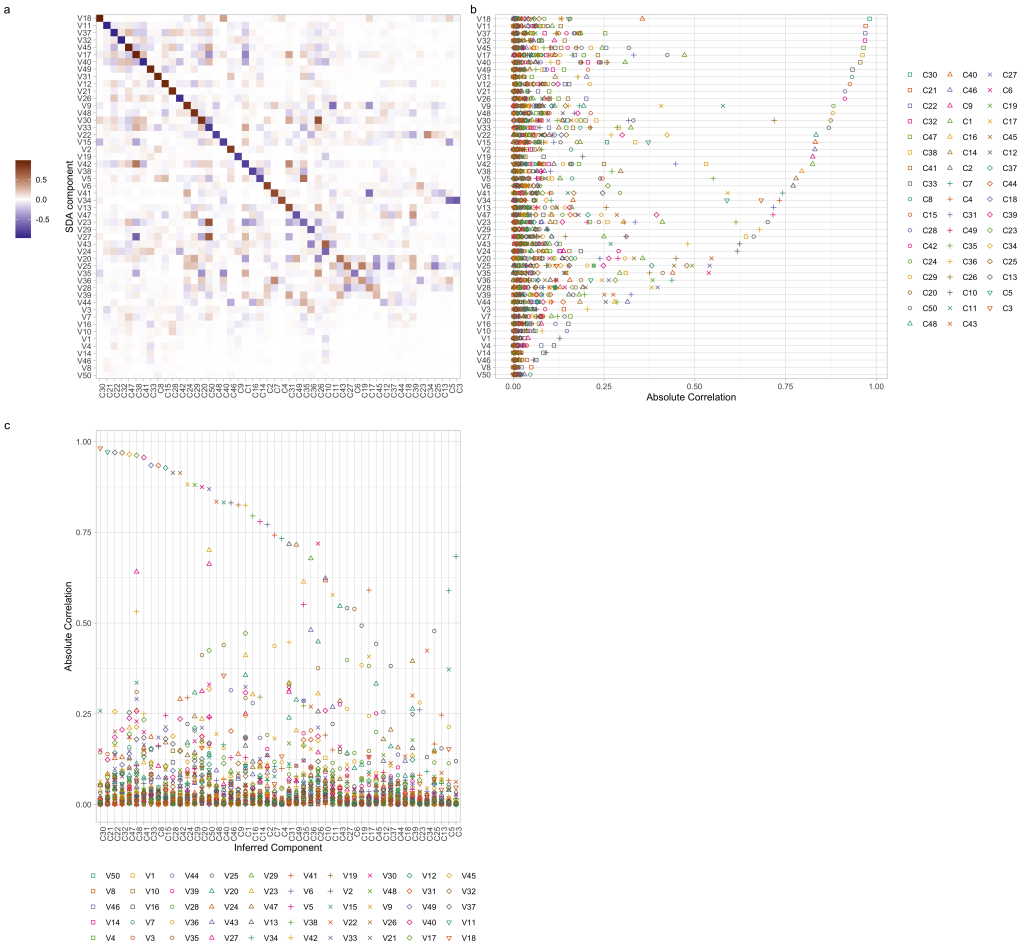

Figure B.5: **SDA and AnnotFA cell score correlation.** (a) Heatmap showing the correlation of cell scores between Run 5 of AnnotFA and the SDA results from [JWR+19]. (b-c) Absolute correlation between the cell scores

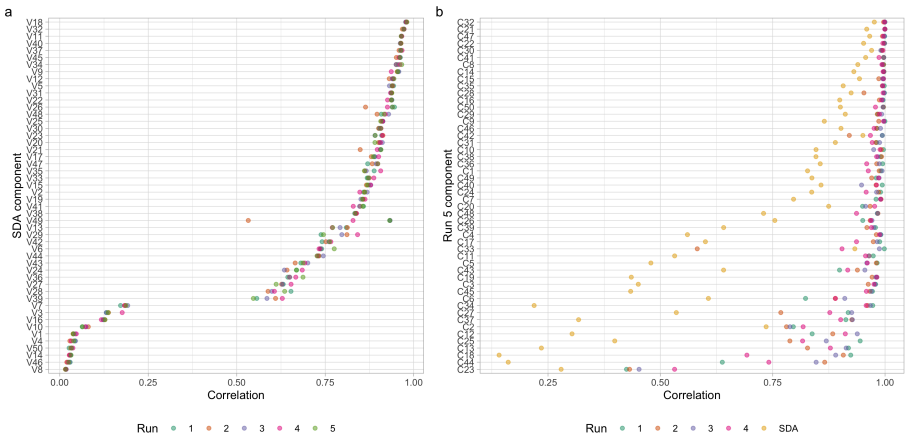

Figure B.6: **Procrustes rotated correlations.** (a) We plot here the maximal correlations of the components cell scores from each run of AnnotFA with SDA components, after Procrustes rotation to match the SDA results. (b) We plot here the maximal correlations of the component cell scores from each run of AnnotFA and the SDA results, after Procrustes rotation to match the run 5 results.

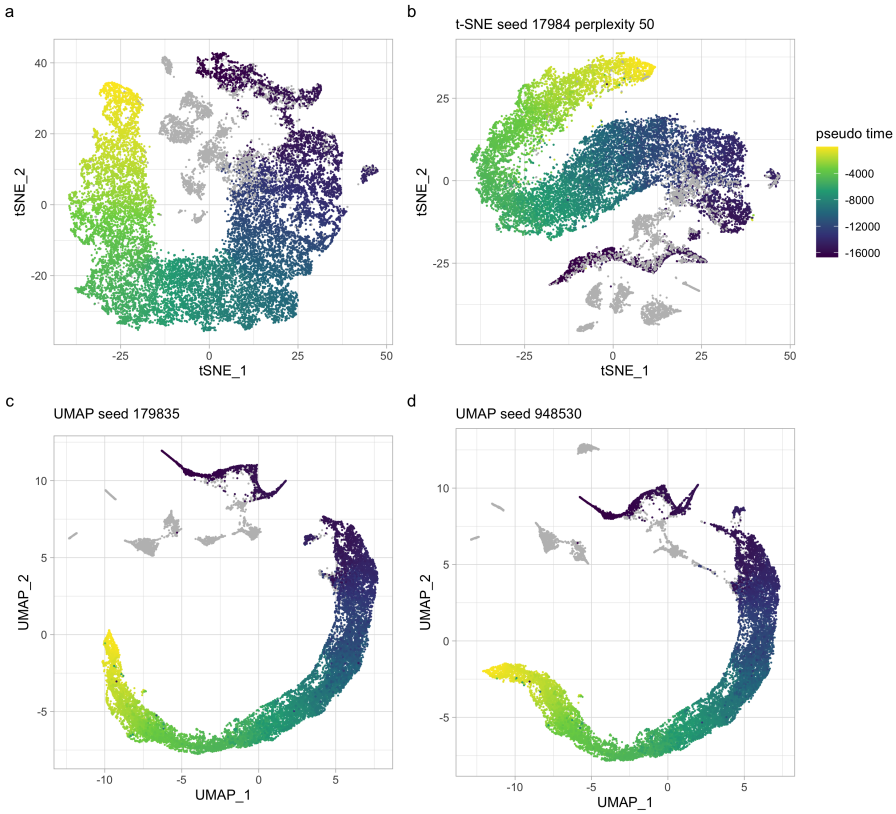

Figure B.7: **t-SNE, UMAP, and pseudotime examples.** (a) The t-SNE coordinates coloured by pseudotime ordering as inferred in [JWR+19]. All panels in this figure are coloured with the same pseudotime ordering. (b) t-SNE dimension reduction of the AnnotFA components as described in the main text. (c,d) Two UMAP dimension reductions of the AnnotFA components as described in the main text.

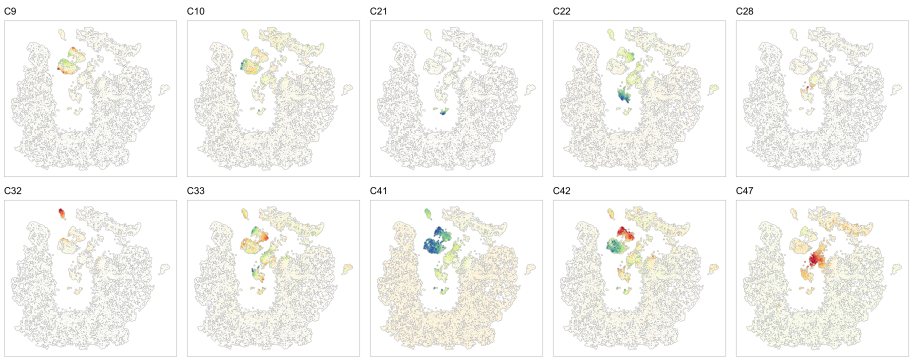

Figure B.8: **Somatic components on t-SNE coordinates.** Cell scores for somatic components, as identified in Figure 5.4(f), and described in the main text, plotted on the t-SNE coordinates from [JWR+19].

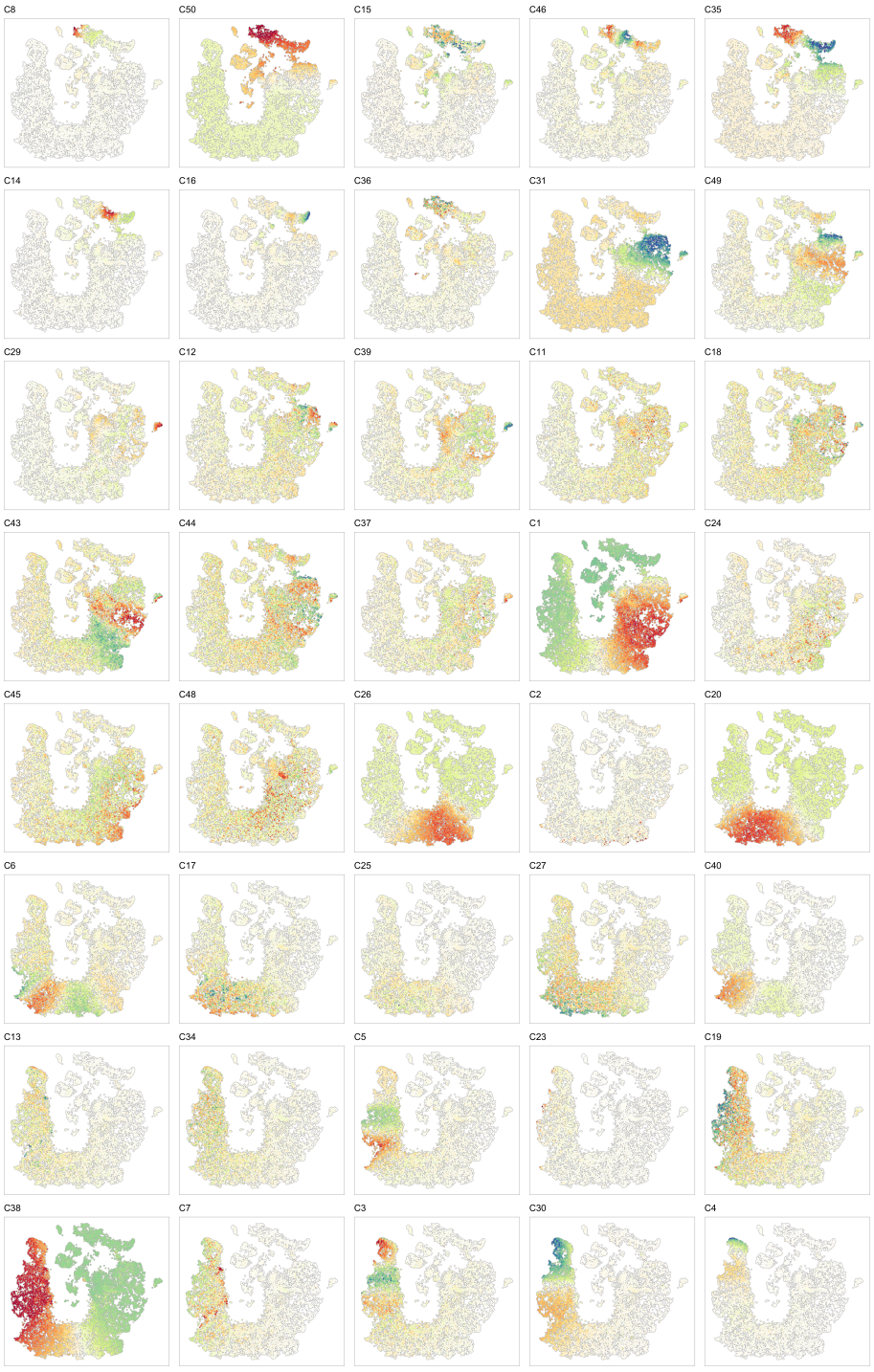

Figure B.9: **Spermatogenesis components ordered by pseudo time.** The cell scores of all non-somatic components, in order of pseudotime as in Figure 5.4(f), plotted on the t-SNE coordinates from [JWR+19].

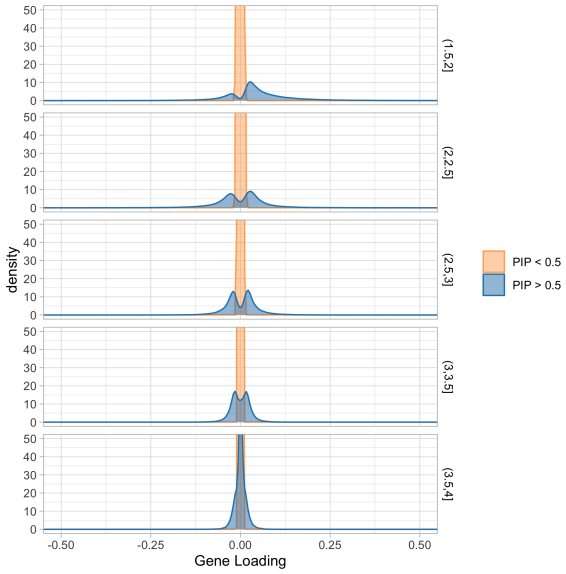

Figure B.10: **Gene loading distribution by PIP $\langle s_{iq} \rangle$ and $\langle \beta_q \rangle$.** Each row includes the loadings for components with $\log_{10}(\langle \beta_q \rangle)$ in the interval indicated on the right, where β_q is the component-wide loadings precision parameter from the AnnotFA model. This includes the 50 components from the 5th run of AnnotFA, as described in Chapter 5.

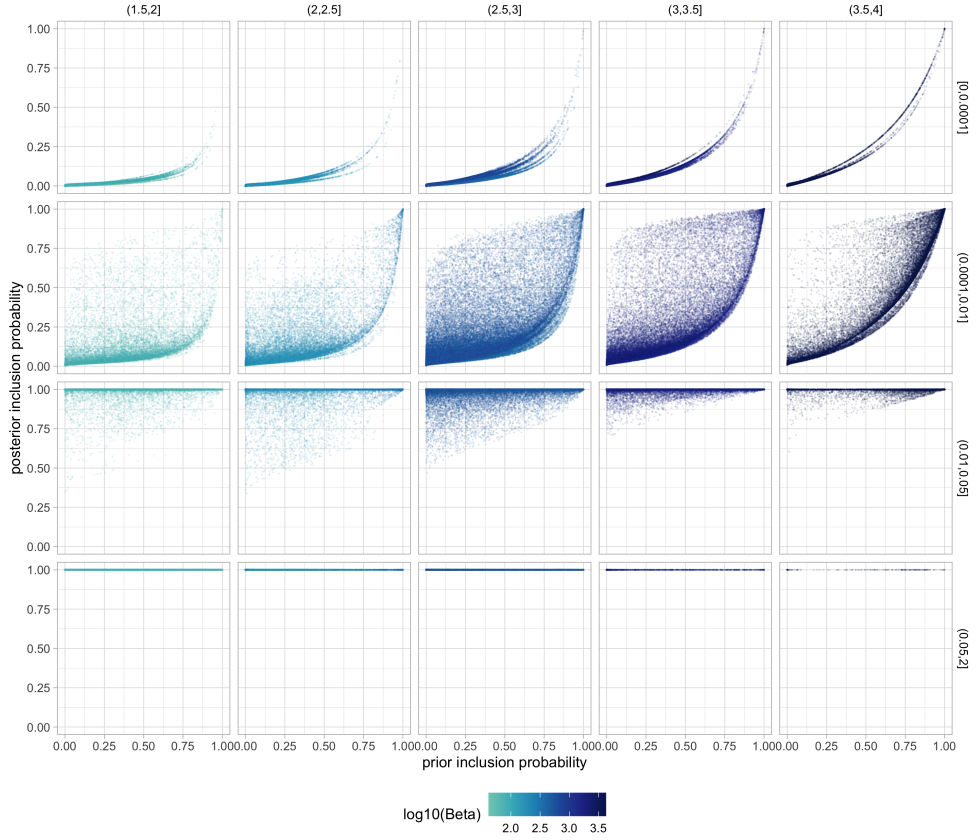

Figure B.11: **PIP vs. pIP split by gene loading and β_q.** Panels show the posterior inclusion probabilities vs prior inclusion probabilities. Each row selects genes for which the absolute value of the gene loadings falls in the indicated interval, and each column the components for which $\log_{10}(\langle\beta_q\rangle)$ falls in the indicated interval, where β_q is the component-wide loadings precision parameter from the AnnotFA model.

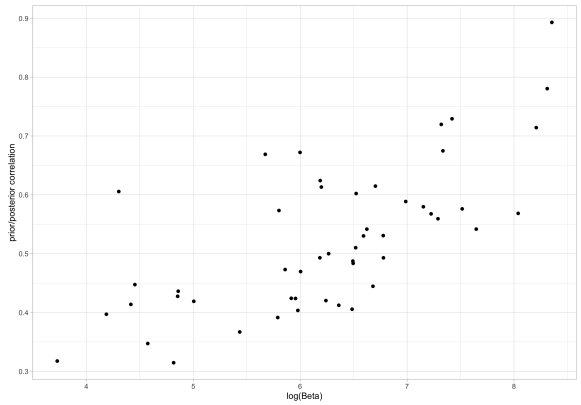

Figure B.12: **pIP and PIP correlation by** $\log\langle\beta_q\rangle$**.** The Pearson correlation between prior inclusion probabilities and posterior inclusion probabilities, plotted against the log of the posterior mean of the component slab precision $\langle\beta_q\rangle$.

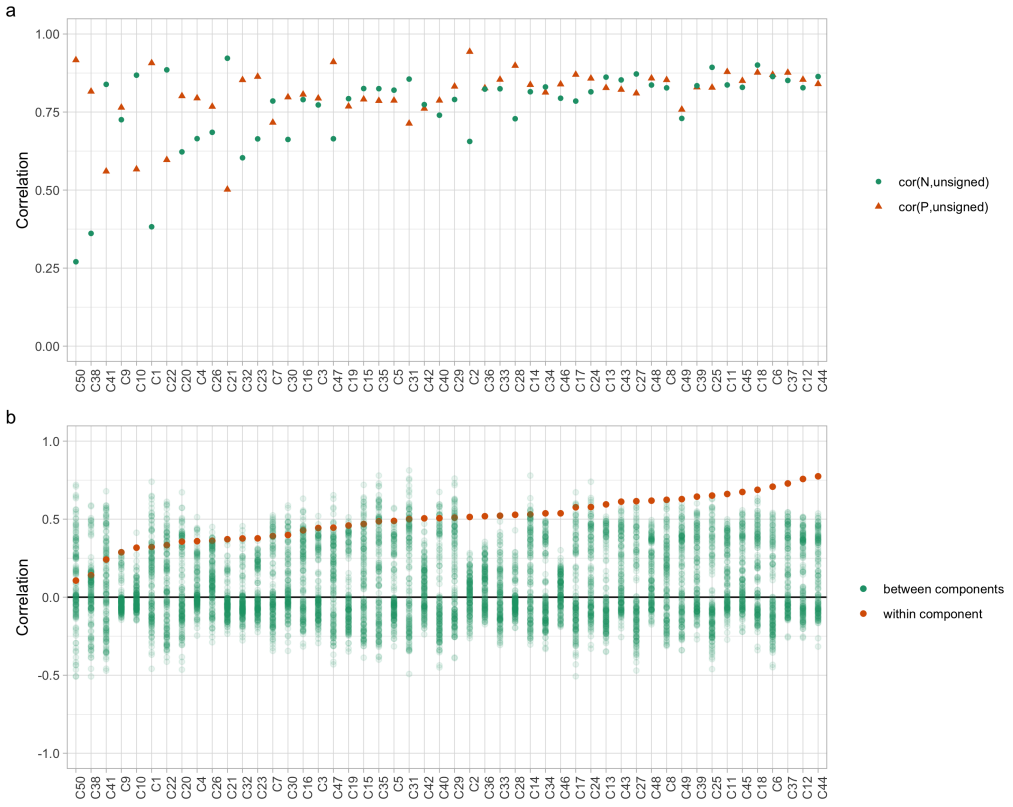

Figure B.13: **Signed pIP correlation between and within components.** (a) For each component we indicate the correlation of the signed prior inclusion probabilities with the unsigned prior inclusion probabilities inferred in the algorithm. Colour and shape indicate the sign of the signed prior inclusion probabilities. (b) Within component correlation shows the correlation of the positive and negative signed prior inclusion probabilities from a given component, indicated in orange. The green points indicate the correlations of both signed prior inclusion probabilities for this component with the signed prior inclusion probabilities from all other components.

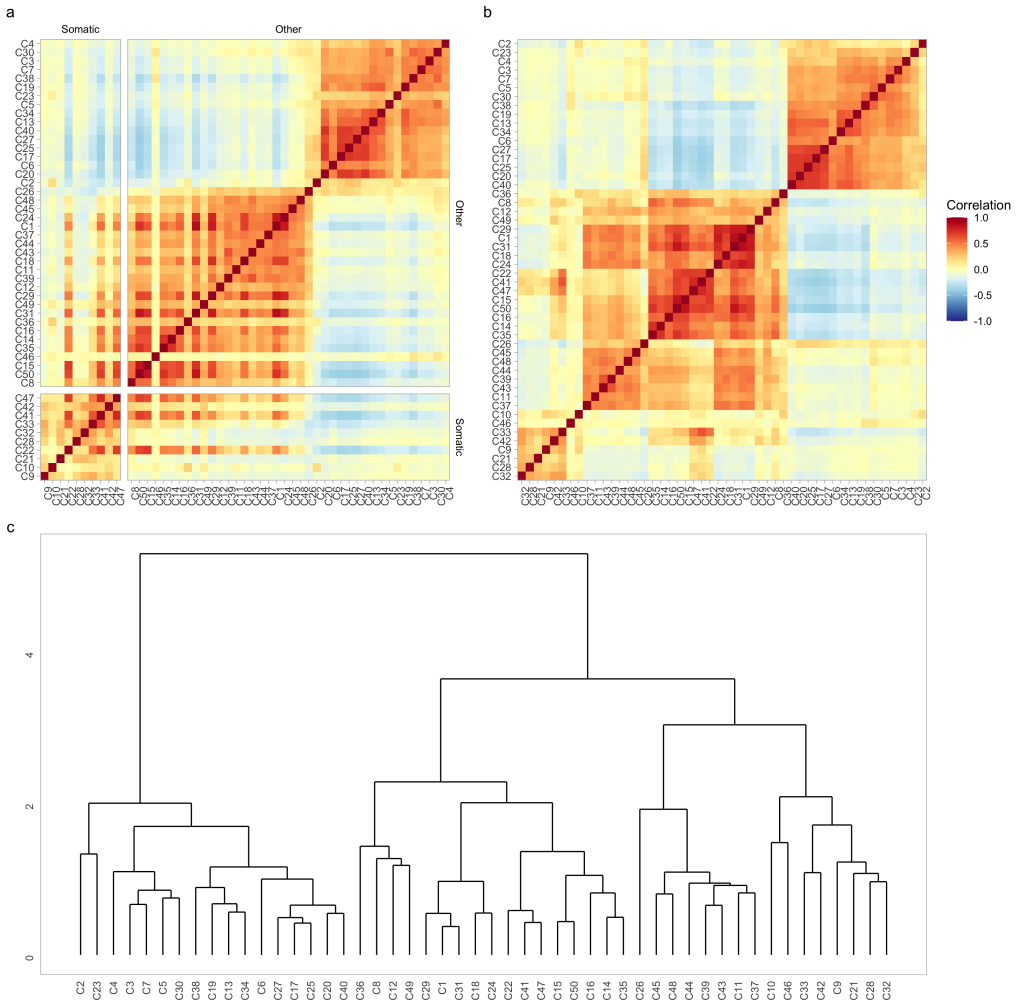

Figure B.14: **Unsigned pIP correlation plot ordered by pseudotime and clustering.** (a) Heatmap of the correlations of prior inclusion probabilities, with somatic components identified, and components ordered with respect to pseudotime as described in the main text for Figure 5.4(f) (b) The prior inclusion probability correlations from (a), with components ordered from a hierarchical clustering, corresponding to the dendrogram in (c).

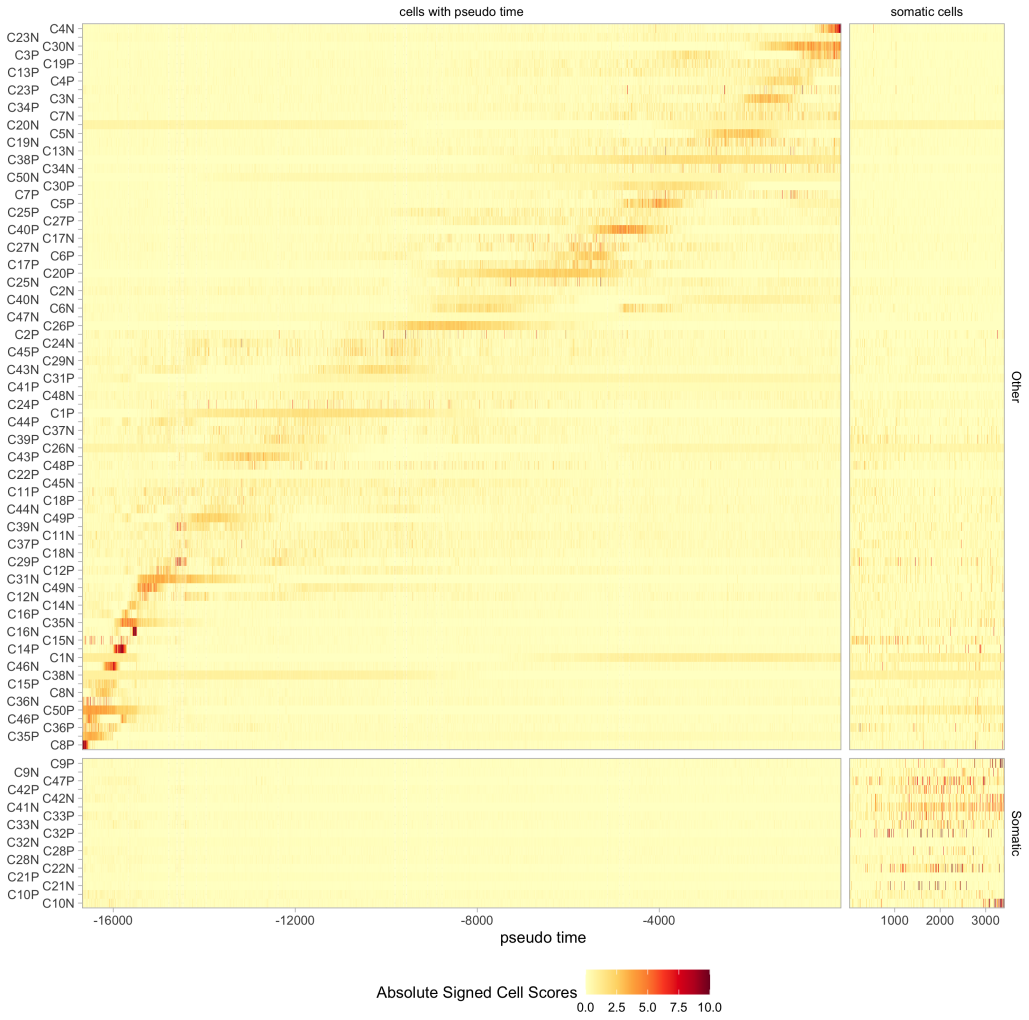

Figure B.15: **Signed cell scores ordered by pseudotime.** As described in Section 5.6.3.

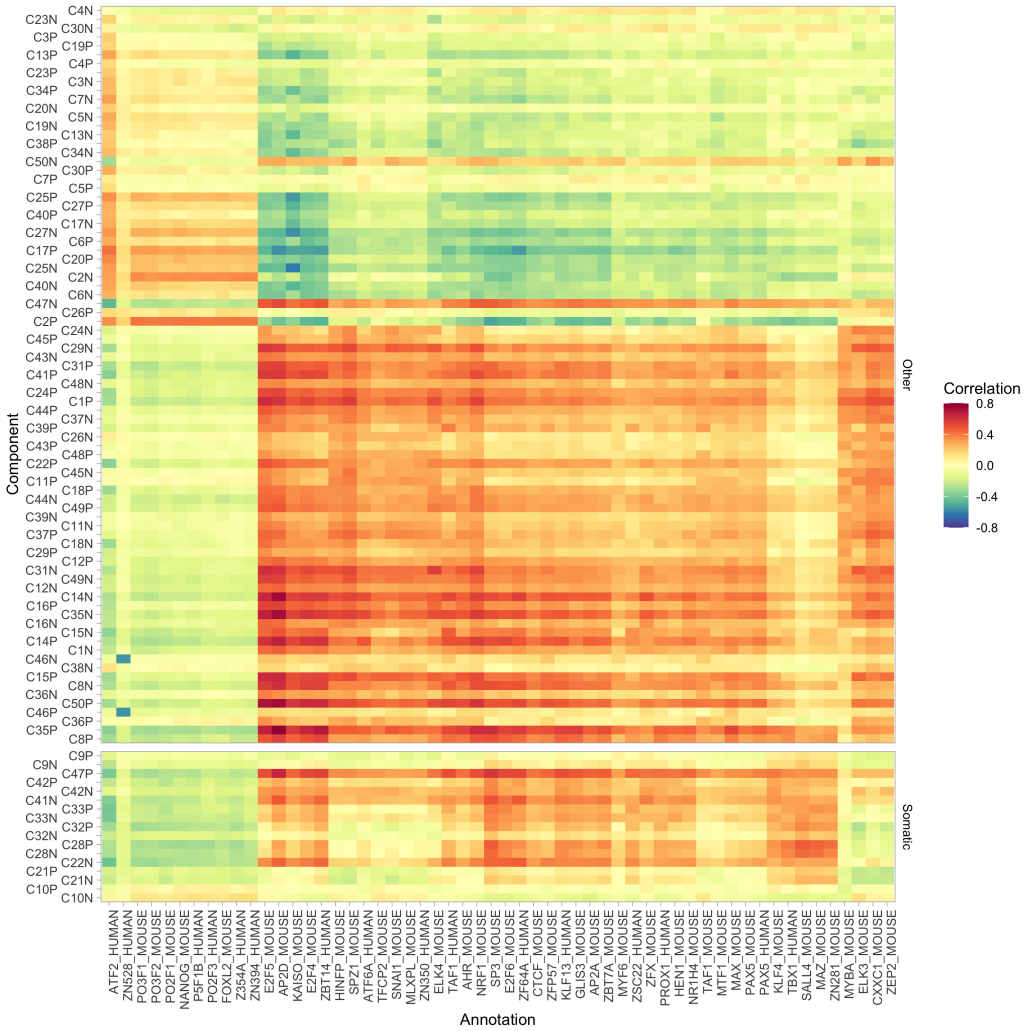

Figure B.16: **Signed pIP regulating annotations.** Rank correlation of annotations with signed prior inclusion probabilities, including only annotations which achieve at least a correlation of 0.4 with some component. As described in Section 5.6.3.

Bibliography

[AAB+20] Ricard Argelaguet, Damien Arnol, Danila Bredikhin, Yonatan Deloro, Britta Velten, John C Marioni, and Oliver Stegle. MOFA+: a statistical framework for comprehensive integration of multi-modal single-cell data. *Genome Biology*, 21(1):111, 2020.

[AH18] Tallulah S. Andrews and Martin Hemberg. Identifying cell populations with scRNASeq. *Molecular Aspects of Medicine*, 59:114–122, 2018.

[AT20] Sarah Aldridge and Sarah A Teichmann. Single cell transcriptomics comes of age. *Nat Commun*, 11(1):4307, Aug 2020.

[ATR+16] Emilie Abby, Sophie Tourpin, Jonathan Ribeiro, Katrin Daniel, Sébastien Messiaen, Delphine Moison, Justine Guerquin, Jean-Charles Gaillard, Jean Armengaud, Francina Langa, Attila Toth, Emmanuelle Martini, and Gabriel Livera. Implementation of meiosis prophase I programme requires a conserved retinoid-independent stabilizer of meiotic transcripts. *Nat Commun*, 7:10324, Jan 2016.

[AVA+18] Ricard Argelaguet, Britta Velten, Damien Arnol, Sascha Dietrich, Thorsten Zenz, John C Marioni, Florian Buettner, Wolfgang Huber, and Oliver Stegle. Multi-Omics Factor Analysis—a framework for unsupervised integration of multi-omics data sets. *Molecular Systems Biology*, 14(6), Jun 2018.

[BBG+10] F Baudat, J Buard, C Grey, A Fledel-Alon, C Ober, M Przeworski, G Coop, and B de Massy. Prdm9 is a major determinant of meiotic recombination hotspots in humans and mice. *Science*, 327(5967):836–840, Feb 2010.

[BE13] Douglas Bates and Dirk Eddelbuettel. Fast and elegant numerical linear algebra using the RcppEigen package. *Journal of Statistical Software*, 52(5):1–24, 2013.

[BFBB+11] Ewelina Bolcun-Filas, Laura A Bannister, Alex Barash, Kerry J Schimenti, Suzanne A Hartford, John J Eppig, Mary Ann Handel, Lishuang Shen, and John C Schimenti. A-myb (mybl1) transcription factor is a master regulator of male meiosis. *Development*, 138(15):3319–3330, Aug 2011.

[BGLL08] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, Oct 2008.

[BHS+18] Andrew Butler, Paul Hoffman, Peter Smibert, Efthymia Papalexi, and Rahul Satija. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature Biotechnology*, 36(5):411–420, Jun 2018.

[Bis06] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[BJP+21] Carles A Boix, Benjamin T James, Yongjin P Park, Wouter Meuleman, and Manolis Kellis. Regulatory genomic circuitry of human disease loci by integrative epigenomics. *Nature*, 590(7845):300–307, Feb 2021.

[BKM17] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[BNJ03] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(4-5), 2003.

[BPM+17] Florian Buettner, Naruemon Pratanwanich, Davis J. McCarthy, John C. Marioni, and Oliver Stegle. f-scLVM: Scalable and versatile factor analysis for single-cell RNA-seq. *Genome Biology*, 18(1):212, Nov 2017.

[Bro70] C. G. Broyden. The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations. *IMA Journal of Applied Mathematics*, 6(1):76–90, Mar 1970.

[BVW+16] Maayan Baron, Adrian Veres, Samuel L Wolock, Aubrey L Faust, Renaud Gaujoux, Amedeo Vetere, Jennifer Hyoje Ryu, Bridget K Wagner, Shai S Shen-Orr, Allon M Klein, Douglas A Melton, and Itai Yanai. A single-cell transcriptomic map of the human and mouse pancreas reveals inter- and intra-cell population structure. *Cell Syst*, 3(4):346–360, Oct 2016.

[CC70] J. Douglas Carroll and Jih-Jie Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35(3):283–319, Sep 1970.

[CLT+20] Robert Lorenz Chua, Soeren Lukassen, Saskia Trump, Bianca P. Hennig, Daniel Wendisch, Fabian Pott, Olivia Debnath, Loreen Thürmann, Florian Kurth, Maria Theresa Völker, Julia Kazmierski, Bernd Timmermann, Sven Twardziok, Stefan Schneider, Felix Machleidt, Holger Müller-Redetzky, Melanie Maier, Alexander Krannich, Sein Schmidt, Felix Balzer, Johannes Liebig, Jennifer Loske, Norbert Suttorp, Jürgen Eils, Naveed Ishaque, Uwe Gerd Liebert, Christof von Kalle, Andreas Hocke, Martin Witzenrath, Christine Goffinet, Christian Drosten, Sven Laudi, Irina Lehmann, Christian Conrad, Leif Erik Sander, and Roland Eils. COVID-19 severity correlates with airway epithelium–immune cell interactions identified by single-cell analysis. *Nature Biotechnology*, 38(8):970–979, 2020.

[CSQ+19] Junyue Cao, Malte Spielmann, Xiaojie Qiu, Xingfan Huang, Daniel M. Ibrahim, Andrew J. Hill, Fan Zhang, Stefan Mundlos, Lena Christiansen, Frank J. Steemers, Cole Trapnell, and Jay Shendure. The single-cell transcriptional landscape of mammalian organogenesis. *Nature*, 566(7745):496–502, 2019.

[DDH+08] Joseph P Doyle, Joseph D Dougherty, Myriam Heiman, Eric F Schmidt, Tanya R Stevens, Guojun Ma, Sujata Bupp, Prerana Shrestha, Rajiv D Shah, Martin L Doughty, Shiaoching Gong, Paul Greengard, and Nathaniel Heintz. Application of a translational profiling approach for the comparative analysis of cns cell types. *Cell*, 135(4):749–762, Nov 2008.

[DDWN18] Wolfgang Dubiel, Dawadschargal Dubiel, Dieter A Wolf, and Michael Naumann. Cullin 3-based ubiquitin ligases as master regulators of mammalian cell differentiation. *Trends Biochem Sci*, 43(2):95–107, Feb 2018.

[DHA+16] Benjamin Davies, Edouard Hatton, Nicolas Altemose, Julie G. Hussin, Florencia Pratto, Gang Zhang, Anjali Gupta Hinch, Daniela Moralli, Daniel Biggs, Rebeca Diaz, Chris Preece, Ran Li, Emmanuelle Bitoun, Kevin Brick, Catherine M. Green, R. Daniel Camerini-Otero, Simon R. Myers, and Peter Donnelly. Re-engineering the zinc fingers of PRDM9 reverses hybrid sterility in mice. *Nature*, 530(7589):171–176, 2016.

[dKLM+98] D.M. de Kretser, K.L. Loveland, A. Meinhardt, D. Simorangkir, and N. Wreford. Spermatogenesis. *Human Reproduction*, 13(1):1–8, 04 1998.

[DLH+11] Katrin Daniel, Julian Lange, Khaled Hached, Jun Fu, Konstantinos Anastassiadis, Ignasi Roig, Howard J Cooke, A Francis Stewart, Katja Wassmann, Maria Jasin, Scott Keeney, and Attila Tóth. Meiotic homologue alignment and its quality surveillance are controlled by mouse HORMAD1. *Nat Cell Biol*, 13(5):599–610, May 2011.

[DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data Via the EM Algorithm . *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1), 1977.

[EF11] Dirk Eddelbuettel and Romain Francois. Rcpp: Seamless R and C++ Integration. *Journal of Statistical Software, Articles*, 40(8):1–18, 2011.

[ES10] Barbara E. Engelhardt and Matthew Stephens. Analysis of population structure: A unifying framework and novel methods based on sparse factor analysis. *PLOS Genetics*, 6(9):1–12, 09 2010.

[FJ20] Olivier François and Flora Jay. Factor analysis of ancient population genomic samples. *Nature Communications*, 11(1):4661, 2020.

[Fle70] R. Fletcher. A new approach to variable metric algorithms. *The Computer Journal*, 13(3):317–322, Jan 1970.

[GBE13] Chuan Gao, Christopher D Brown, and Barbara E Engelhardt. A latent factor model with a mixture of sparse and dense factors to model gene expression data with confounding effects. arXiv(1310.4792v1), 2013.

[GHS+05] J H Guo, Q Huang, D J Studholme, C Q Wu, and Z Zhao. Transcriptomic analyses support the similarity of gene expression between brain and testis in human as well as mouse. *Cytogenet Genome Res*, 111(2):107–109, 2005.

[GI20] Isabella N. Grabski and Rafael A. Irizarry. Probabilistic gene expression signatures identify cell-types from single cell rna-seq data. *bioRxiv*, 2020.

[Gol70] Donald Goldfarb. A family of variable-metric methods derived by variational means. *Mathematics of Computation*, 24(109), 1970.

[GWP+15] Minzhe Guo, Hui Wang, S. Steven Potter, Jeffrey A. Whitsett, and Yan Xu. SINCERA: A Pipeline for Single-Cell RNA-Seq Profiling Analysis. *PLOS Computational Biology*, 11(11):1–28, Nov 2015.

[HBWP13] Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley. Stochastic variational inference. *J. Mach. Learn. Res.*, 14(1):1303–1347, May 2013.

[HL94] Richard A. Harshman and Margaret E. Lundy. PARAFAC: Parallel factor analysis. *Computational Statistics & Data Analysis*, 18(1):39–72, 1994.

[Hor15] Victoria Hore. *Latent Variable Models for Analysing Multidimensional Gene Expression Data*. PhD thesis, University of Oxford, 2015.

[HVB+16] Victoria Hore, Ana Viñuela, Alfonso Buil, Julian Knight, Mark I McCarthy, Kerrin Small, and Jonathan Marchini. Tensor decomposition for

multiple-tissue gene expression experiments. *Nature Genetics*, 48(9):1094–1100, Aug 2016.

[JHG+20] Bradley S Jermy, Saskia P Hagenaars, Kylie P Glanville, Jonathan RI Coleman, David M Howard, Gerome Breen, Evangelos Vassos, and Cathryn M Lewis. Using major depression polygenic risk scores to explore the depressive symptom continuum. *bioRxiv*, 2020.

[JTGV+05] G. Joshi-Tope, M. Gillespie, I. Vastrik, P. D'Eustachio, E. Schmidt, B. de Bono, B. Jassal, G. R. Gopinath, G. R. Wu, L. Matthews, S. Lewis, E. Birney, and L. Stein. Reactome: A knowledgebase of biological pathways. *Nucleic Acids Research*, 33(DATABASE ISS.), 2005.

[JWR+19] Min Jung, Daniel Wells, Jannette Rusch, Suhaira Ahmad, Jonathan Marchini, Simon R. Myers, and Donald F. Conrad. Unified single-cell analysis of testis gene regulation and pathology in five mouse strains. *eLife*, 8, Jun 2019.

[KB09] Tamara G. Kolda and Brett W. Bader. Tensor Decompositions and Applications. *SIAM Review*, 51(3):455–500, Aug 2009.

[KB19] Dmitry Kobak and Philipp Berens. The art of using t-SNE for single-cell transcriptomics. *Nature Communications*, 10(1):5416, 2019.

[KBL+15] W. Stephen Kistler, Dominique Baas, Sylvain Lemeille, Marie Paschaki, Queralt Seguin-Estevez, Emmanuèle Barras, Wenli Ma, Jean-Luc Duteyrat, Laurette Morlé, Bénédicte Durand, and Walter Reith. Rfx2 is a major transcriptional regulator of spermiogenesis. *PLOS Genetics*, 11(7):1–28, 07 2015.

[KdRP19] Mina L Kojima, Dirk G de Rooij, and David C Page. Amplification of a broad transcriptional program by a common factor triggers the meiotic cell cycle in mice. *Elife*, 8, Feb 2019.

[Kon20] Tomasz Konopka. *umap: Uniform Manifold Approximation and Projection*, 2020. R package version 0.2.7.0.

[Kri15] Jesse H. Krijthe. *Rtsne: T-Distributed Stochastic Neighbor Embedding using Barnes-Hut Implementation*, 2015. R package version 0.15.

[KVY+17] Ivan V Kulakovskiy, Ilya E Vorontsov, Ivan S Yevshin, Ruslan N Sharipov, Alla D Fedorova, Eugene I Rumynskiy, Yulia A Medvedeva, Arturo Magana-Mora, Vladimir B Bajic, Dmitry A Papatsenko, Fedor A Kolpakov, and Vsevolod J Makeev. HOCOMOCO: towards a complete collection of transcription factor binding models for human and mouse via large-scale ChIP-Seq analysis. *Nucleic Acids Research*, 46(1):252–259, Nov 2017.

[LBEW18] S Lukassen, E Bosch, A B Ekici, and A Winterpacht. Characterization of germ cell differentiation in the male mouse through single-cell rna sequencing. *Sci Rep*, 8(1):6521, Apr 2018.

[LCW+06] Joseph Lucas, Carlos Carvalho, Q Wang, Andrea Bild, J Nevins, and Mike West. *Sparse statistical modelling in gene expression genomics*, volume Bayesian Inference for Gene Expression and Proteomics 1, pages 155–176. Jan 2006.

[LJ19] Jana Link and Verena Jantsch. Meiotic chromosomes in motion: a perspective from Mus musculus and Caenorhabditis elegans. *Chromosoma*, 128(3):317–330, Sep 2019.

[LMSZ+18] Gioele La Manno, Ruslan Soldatov, Amit Zeisel, Emelie Braun, Hannah Hochgerner, Viktor Petukhov, Katja Lidschreiber, Maria E. Kastriti, Peter Lönnerberg, Alessandro Furlan, Jean Fan, Lars E. Borm, Zehua Liu, David van Bruggen, Jimin Guo, Xiaoling He, Roger Barker, Erik Sundström, Gonçalo Castelo-Branco, Patrick Cramer, Igor Adameyko, Sten Linnarsson, and Peter V. Kharchenko. RNA velocity of single cells. *Nature*, 560(7719):494–498, 2018.

[LN89] Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1-3):503–528, Aug 1989.

[LT19] Malte D Luecken and Fabian J Theis. Current best practices in single-cell RNA-seq analysis: a tutorial. *Molecular Systems Biology*, 15(6), Jun 2019.

[LTH17] Peijie Lin, Michael Troup, and Joshua W. K. Ho. CIDR: Ultrafast and accurate clustering through imputation for single-cell RNA-seq data. *Genome Biology*, 18(1):59, 2017.

[MB88] T. J. Mitchell and J. J. Beauchamp. Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83(404):1023–1032, 1988.

[MBS+15] Evan Z Macosko, Anindita Basu, Rahul Satija, James Nemesh, Karthik Shekhar, Melissa Goldman, Itay Tirosh, Allison R Bialas, Nolan Kamitaki, Emily M Martersteck, John J Trombetta, David A Weitz, Joshua R Sanes, Alex K Shalek, Aviv Regev, and Steven A McCarroll. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell*, 161(5):1202–1214, May 2015.

[MBT+10] Simon Myers, Rory Bowden, Afidalina Tumian, Ronald E Bontrop, Colin Freeman, Tammie S MacFie, Gil McVean, and Peter Donnelly. Drive against hotspot motifs in primates implicates the prdm9 gene in meiotic recombination. *Science*, 327(5967):876–879, Feb 2010.

[MHSG18] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. UMAP: Uniform Manifold Approximation and Projection. *Journal of Open Source Software*, 3(29):861, Sep 2018.

[MLH+00] X Meng, M Lindahl, M E Hyvönen, M Parvinen, D G de Rooij, M W Hess, A Raatikainen-Ahokas, K Sainio, H Rauvala, M Lakso, J G Pichel, H Westphal, M Saarma, and H Sariola. Regulation of cell fate decision of undifferentiated spermatogonia by gdnf. *Science*, 287(5457):1489–1493, Feb 2000.

[MM14] Kevin V Morris and John S Mattick. The rise of regulatory rna. *Nat Rev Genet*, 15(6):423–437, Jun 2014.

[Mur12] Kevin Patrick Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[NH98] Radford M Neal and Geoffrey E Hinton. A View of the Em Algorithm that Justifies Incremental, Sparse, and other Variants. In *Learning in Graphical Models*, chapter 12. 1998.

[Pie16] B.A. Pierce. *Genetics: A Conceptual Approach*. W. H. Freeman, 2016.

[PPP10] Emil D. Parvanov, Petko M. Petkov, and Kenneth Paigen. Prdm9 controls activation of mammalian recombination hotspots. *Science*, 327(5967):835–835, 2010.

[PY15] Emma Pierson and Christopher Yau. ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biology*, 16(1), Nov 2015.

[QBB+19] Yixuan Qiu, Sreekumar Balan, Matt Beall, Mark Sauder, Naoaki Okazaki, and Thomas Hahn. *RcppNumerical: 'Rcpp' Integration for Numerical Computing Libraries*, 2019. R package version 0.4-0.

[R C19] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2019.

[Ran15] Susannah Rankin. Complex elaboration: making sense of meiotic cohesin dynamics. *FEBS J*, 282(13):2426–2443, Jul 2015.

[RRSRT17] Orit Rozenblatt-Rosen, Michael J T Stubbington, Aviv Regev, and Sarah A Teichmann. The human cell atlas: from vision to reality. *Nature*, 550(7677):451–453, Oct 2017.

[RTL+17] Aviv Regev, Sarah A Teichmann, Eric S Lander, Ido Amit, Christophe Benoist, Ewan Birney, Bernd Bodenmiller, Peter Campbell, Piero Carninci, Menna Clatworthy, Hans Clevers, Bart Deplancke, Ian Dunham, James Eberwine, Roland Eils, Wolfgang Enard, Andrew Farmer, Lars Fugger, Berthold Göttgens, Nir Hacohen, Muzlifah Haniffa, Martin Hemberg, Seung Kim, Paul Klenerman, Arnold Kriegstein, Ed Lein, Sten Linnarsson, Emma Lundberg, Joakim Lundeberg, Partha Majumder, John C Marioni, Miriam Merad, Musa Mhlanga, Martijn Nawijn, Mihai Netea, Garry Nolan, Dana Pe'er, Anthony Phillipakis, Chris P Ponting, Stephen Quake, Wolf Reik, Orit Rozenblatt-Rosen, Joshua Sanes, Rahul Satija, Ton N Schumacher, Alex Shalek, Ehud Shapiro, Padmanee Sharma, Jay W Shin, Oliver Stegle, Michael Stratton, Michael J T Stubbington,

Fabian J Theis, Matthias Uhlen, Alexander van Oudenaarden, Allon Wagner, Fiona Watt, Jonathan Weissman, Barbara Wold, Ramnik Xavier, and Nir Yosef. The human cell atlas. *Elife*, 6, Dec 2017.

[SB12] Matthias Seeger and Guillaume Bouchard. Fast variational bayesian inference for non-conjugate matrix factorization models. In *Proc. of AISTATS, LA*, 2012.

[SDA+20] Catrina Spruce, Sibongakonke Dlamini, Guruprasad Ananda, Naomi Bronkema, Hui Tian, Kenneth Paigen, Gregory W Carter, and Christopher L Baker. HELLS and PRDM9 form a pioneer complex to open chromatin at meiotic recombination hot spots. *Genes Dev*, 34(5-6):398–412, Mar 2020.

[SdVBP20] Valentine Svensson, Eduardo da Veiga Beltrame, and Lior Pachter. A curated database reveals trends in single-cell transcriptomics. *Database*, 2020, Nov 2020.

[SGH19] Rory Stark, Marta Grzelak, and James Hadfield. Rna sequencing: the teenage years. *Nat Rev Genet*, 20(11):631–656, Nov 2019.

[Sha70] D. F. Shanno. Conditioning of quasi-Newton methods for function minimization. *Mathematics of Computation*, 24(111), 1970.

[SKN+18] Nicholas Schaum, Jim Karkanias, Norma F. Neff, Andrew P. May, Stephen R. Quake, Tony Wyss-Coray, Spyros Darmanis, Joshua Batson, Olga Botvinnik, Michelle B. Chen, Steven Chen, Foad Green, Robert C. Jones, Ashley Maynard, Lolita Penland, Angela Oliveira Pisco, Rene V. Sit, Geoffrey M. Stanley, James T. Webber, Fabio Zanini, Ankit S. Baghel, Isaac Bakerman, Ishita Bansal, Daniela Berdnik, Biter Bilen, Douglas Brownfield, Corey Cain, Michelle B. Chen, Min Cho, Giana

Cirolia, Stephanie D. Conley, Aaron Demers, Kubilay Demir, Antoine de Morree, Tessa Divita, Haley du Bois, Laughing Bear Torrez Dulgeroff, Hamid Ebadi, F. Hernán Espinoza, Matt Fish, Qiang Gan, Benson M. George, Astrid Gillich, Geraldine Genetiano, Xueying Gu, Gunsagar S. Gulati, Yan Hang, Shayan Hosseinzadeh, Albin Huang, Tal Iram, Taichi Isobe, Feather Ives, Robert C. Jones, Kevin S. Kao, Guruswamy Karnam, Aaron M. Kershner, Bernhard M. Kiss, William Kong, Maya E. Kumar, Jonathan Y. Lam, Davis P. Lee, Song E. Lee, Guang Li, Qingyun Li, Ling Liu, Annie Lo, Wan-Jin Lu, Anoop Manjunath, Andrew P. May, Kaia L. May, Oliver L. May, Marina McKay, Ross J. Metzger, Marco Mignardi, Dullei Min, Ahmad N. Nabhan, Norma F. Neff, Katharine M. Ng, Joseph Noh, Rasika Patkar, Weng Chuan Peng, Robert Puccinelli, Eric J. Rulifson, Shaheen S. Sikandar, Rahul Sinha, Rene V. Sit, Krzysztof Szade, Weilun Tan, Cristina Tato, Krissie Tellez, Kyle J. Travaglini, Carolina Tropini, Lucas Waldburger, Linda J. van Weele, Michael N. Wosczyna, Jinyi Xiang, Soso Xue, Justin Youngyunpipatkul, Macy E. Zardeneta, Fan Zhang, Lu Zhou, Andrew P. May, Norma F. Neff, Rene V. Sit, Paola Castro, Derek Croote, Joseph L. DeRisi, Geoffrey M. Stanley, James T. Webber, Ankit S. Baghel, Michelle B. Chen, F. Hernán Espinoza, Benson M. George, Gunsagar S. Gulati, Aaron M. Kershner, Bernhard M. Kiss, Christin S. Kuo, Jonathan Y. Lam, Benoit Lehallier, Ahmad N. Nabhan, Katharine M. Ng, Patricia K. Nguyen, Eric J. Rulifson, Shaheen S. Sikandar, Serena Y. Tan, Kyle J. Travaglini, Linda J. van Weele, Bruce M. Wang, Michael N. Wosczyna, Hanadie Yousef, Andrew P. May, Stephen R. Quake, Geoffrey M. Stanley, James T. Webber, Philip A. Beachy, Charles K. F. Chan, Benson M. George, Gunsagar S. Gulati, Kerwyn Casey Huang, Aaron M. Kershner, Bernhard M. Kiss, Ahmad N.

Nabhan, Katharine M. Ng, Patricia K. Nguyen, Eric J. Rulifson, Shaheen S. Sikandar, Kyle J. Travaglini, Bruce M. Wang, Kenneth Weinberg, Michael N. Wosczyna, Sean M. Wu, Ben A. Barres, Philip A. Beachy, Charles K. F. Chan, Michael F. Clarke, Seung K. Kim, Mark A. Krasnow, Maya E. Kumar, Christin S. Kuo, Andrew P. May, Ross J. Metzger, Norma F. Neff, Roel Nusse, Patricia K. Nguyen, Thomas A. Rando, Justin Sonnenburg, Bruce M. Wang, Irving L. Weissman, Sean M. Wu, Stephen R. Quake, The Tabula Muris Consortium, Overall coordination, Logistical coordination, Organ collection, processing, Library preparation, sequencing, Computational data analysis, Cell type annotation, Writing group, Supplemental text writing group, and Principal investigators. Single-cell transcriptomics of 20 mouse organs creates a tabula muris. *Nature*, 562(7727):367–372, 2018.

[SNL+17] Valentine Svensson, Kedar Nath Natarajan, Lam-Ha Ly, Ricardo J Miragaia, Charlotte Labalette, Iain C Macaulay, Ana Cvejic, and Sarah A Teichmann. Power analysis of single-cell RNA-sequencing experiments. *Nature Methods*, 14(4):381–387, 2017.

[SOAC+18] Genevieve L. Stein-O'Brien, Raman Arora, Aedin C Culhane, Alexander V Favorov, Lana X Garmire, Casey S Greene, Loyal A Goff, Yifeng Li, Aloune Ngom, Michael F Ochs, Yanxun Xu, and Elana J Fertig. Enter the Matrix: Factorization Uncovers Knowledge from Omics. *Trends in Genetics*, 34(10):790–805, 2018.

[SS20] Abhishek Sarkar and Matthew Stephens. Separating measurement and expression models clarifies confusion in single cell RNA-seq analysis. *bioRxiv*, page 2020.04.07.030007, Apr 2020.

[SSJ97] D.P. Snustad, M.J. Simmons, and J.B. Jenkins. *Principles of Genetics*. Wiley, 1997.

[Sve20] Valentine Svensson. Droplet scrna-seq is not zero-inflated. *Nat Biotechnol*, 38(2):147–150, Feb 2020.

[SVTT18] Valentine Svensson, Roser Vento-Tormo, and Sarah A Teichmann. Exponential scaling of single-cell rna-seq in the past decade. *Nature Protocols*, 13(4):599–604, 2018.

[SZPP15] Valeriya Sergeevna Shumskaya, Nadezhda Alekseevna Zhigalova, Anna Valerievna Prokhorchouk, and Egor Borisovich Prokhorchouk. Distribution of kaiso protein in mouse tissues. *Histochem Cell Biol*, 143(1):29–43, Jan 2015.

[TBW+09] Fuchou Tang, Catalin Barbacioru, Yangzhou Wang, Ellen Nordman, Clarence Lee, Nanlan Xu, Xiaohui Wang, John Bodeau, Brian B. Tuch, Asim Siddiqui, Kaiqin Lao, and M. Azim Surani. mRNA-Seq whole-transcriptome analysis of a single cell. *Nature Methods*, 6(5):377–382, 2009.

[TCG+14] Cole Trapnell, Davide Cacchiarelli, Jonna Grimsby, Prapti Pokharel, Shuqiang Li, Michael Morse, Niall J. Lennon, Kenneth J. Livak, Tarjei S. Mikkelsen, and John L. Rinn. The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature Biotechnology*, 32(4):381–386, 2014.

[TLG11] Michalis Titsias and Miguel Lázaro-Gredilla. Spike and slab variational inference for multi-task and multiple kernel learning. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances*

in Neural Information Processing Systems, volume 24. Curran Associates, Inc., 2011.

[TLS11] Fuchou Tang, Kaiqin Lao, and M Azim Surani. Development and applications of single-cell transcriptome analysis. *Nature Methods*, 8(4):S6–S11, 2011.

[Tur07] James M A Turner. Meiotic sex chromosome inactivation. *Development*, 134(10):1823–1831, May 2007.

[Tur15] James M A Turner. Meiotic silencing in mammals. *Annu Rev Genet*, 49:395–412, 2015.

[TWvE19] V. A. Traag, L. Waltman, and N. J. van Eck. From Louvain to Leiden: guaranteeing well-connected communities. *Scientific Reports*, 9(1):5233, 2019.

[van14] L.J.P. van der Maaten. Accelerating t-sne using tree-based algorithms. *Journal of Machine Learning Research*, 15:3221–3245, 2014.

[vdMH08] Laurens van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[VKKK12] Seppo Virtanen, Arto Klami, Suleiman A. Khan, and Samuel Kaski. Bayesian group factor analysis. In Neil D. Lawrence and Mark A. Girolami, editors, *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics (AISTATS-12)*, volume 22, pages 1269–1277, 2012.

[WBM+20] Daniel Wells, Emmanuelle Bitoun, Daniela Moralli, Gang Zhang, Anjali Hinch, Julia Jankowska, Peter Donnelly, Catherine Green, and Simon R

Myers. Zcwpw1 is recruited to recombination hotspots by prdm9 and is essential for meiotic double strand break repair. *Elife*, 9, Aug 2020.

[WSL+08] Eric T Wang, Rickard Sandberg, Shujun Luo, Irina Khrebtukova, Lu Zhang, Christine Mayr, Stephen F Kingsmore, Gary P Schroth, and Christopher B Burge. Alternative isoform regulation in human tissue transcriptomes. *Nature*, 456(7221):470–476, Nov 2008.

[YHF+18] Kosuke Yamaguchi, Masashi Hada, Yuko Fukuda, Erina Inoue, Yoshinori Makino, Yuki Katou, Katsuhiko Shirahige, and Yuki Okada. Re-evaluating the localization of sperm-retained histones revealed the modification-dependent accumulation in specific genome regions. *Cell Rep*, 23(13):3920–3932, Jun 2018.

[YLK+11] Yan Yin, Congxing Lin, Sung Tae Kim, Ignasi Roig, Hong Chen, Liren Liu, George Michael Veith, Ramon U Jin, Scott Keeney, Maria Jasin, Kelle Moley, Pengbo Zhou, and Liang Ma. The E3 ubiquitin ligase Cullin 4A regulates meiotic progression in mouse spermatogenesis. *Dev Biol*, 356(1):51–62, Aug 2011.

[YM09] Wei Yan and John R McCarrey. Sex chromosome inactivation in the male. *Epigenetics*, 4(7):452–456, Oct 2009.

[YWHH12] Guangchuang Yu, Li-Gen Wang, Yanyan Han, and Qing-Yu He. clusterProfiler: an R package for comparing biological themes among gene clusters. *OMICS: A Journal of Integrative Biology*, 16(5):284–287, 2012.

[ŽY16] Justina Žurauskienė and Christopher Yau. pcaReduce: hierarchical clustering of single cell transcriptional profiles. *BMC Bioinformatics*, 17:140, Mar 2016.