PROBABILISTIC MODELLING OF GENOMIC TRAJECTORIES

KIERAN R. CAMPBELL

St John's College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2017

Kieran R. Campbell: *Probabilistic modelling of genomic trajectories*

ABSTRACT

The recent advancement of whole-transcriptome gene expression quantification technology - particularly at the single-cell level - has created a wealth of biological data. An increasingly popular unsupervised analysis is to find one dimensional manifolds or *trajectories* through such data that track the development of some biological process. Such methods may be necessary due to the lack of explicit time series measurements or due to asynchronicity of the biological process at a given time.

This thesis aims to recast trajectory inference from high-dimensional "omics" data as a statistical latent variable problem. We begin by examining sources of uncertainty in current approaches and examine the consequences of propagating such uncertainty to downstream analyses. We also introduce a model of switch-like differentiation along trajectories. Next, we consider inferring such trajectories through parametric nonlinear factor analysis models and demonstrate that incorporating information about gene behaviour as informative Bayesian priors improves inference. We then consider the case of bifurcations in data and demonstrate the extent to which they may be modelled using a hierarchical mixture of factor analysers. Finally, we propose a novel type of latent variable model that performs inference of such trajectories in the presence of heterogeneous genetic and environmental backgrounds. We apply this to both single-cell and population-level cancer datasets and propose a nonparametric extension similar to Gaussian Process Latent Variable Models.

PUBLICATIONS

Chapter 2 is published as

> *Campbell, K.R. & Yau, C., 2016. Order Under Uncertainty: Robust Differential Expression Analysis Using Probabilistic Models for Pseudotime Inference. PLoS computational biology, 12(11), p.e1005212.*

and

> *Campbell, K.R. & Yau, C., 2016. switchde: inference of switch-like differential expression along single-cell trajectories. Bioinformatics.*

Chapter 4 is published as

> *Campbell, K.R. & Yau, C., 2017. Probabilistic modeling of bifurcations in single-cell gene expression data using a Bayesian mixture of factor analyzers. Wellcome Open Research, 2, p.19.*

The software `Scater` used for much of the low-level analysis in this paper is published as

> *McCarthy, D.J. et al., 2017. Scater: pre-processing, quality control, normalization and visualization of single-cell RNA-seq data in R. Bioinformatics.*

ACKNOWLEDGMENTS

First and foremost I would like to thank Chris Yau for his wonderful supervision during my DPhil. He has taught me an incredible amount about statistics, genomics, and academic life in general, and in both our work and my plans in life he has encouraged me to go a level beyond that I would have before. On the rare occasion I'm convinced I've found a flaw in his thinking I'm swiftly proven wrong. I would also like to thank him for sending me to DisneyWorld.

I would like to thank Caleb Webber for his supervision on various projects and support and enthusiasm over the years. I am further indebted to Chris Ponting for his early supervision, but also his continued mentorship throughout my DPhil and advice in planning my future beyond these studies.

I'm also grateful to those with whom I've collaborated with scientifically both in Oxford and further afield. Davis McCarthy and Quin Wills on the development of Scater, Charmaine Lang on the Parkinson's project, Georg Holländer and Carlos Mayer on the Thymus project, and many others.

I am grateful to the Medical Research Council for funding my DPhil and also to St John's College Oxford for generous financial support for academic expenses and conference travel.

Many thanks to all my friends both in Oxford and further afield for the fun times we've had. I feel lucky to know such a great group of people.

I am deeply thankful to my parents for their love and support through 21 (!) continuous years of education and have enjoyed their visits to Oxford in the time I've been here.

Finally to Sally, for her support in listening to me complain about "the statistics of single-cell biology", but really for the adventures we've gone on during this time that have made it all such fun.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1

INTRODUCTION

1.1 SINGLE-CELL RNA-SEQUENCING

1.1.1 *Introduction*

It is hard to overstate the impact of single-cell whole-transcriptome expression quantification on recent scientific research. The first single-cell RNA-sequencing (scRNA-seq) method was introduced in 2009 [122], and gained widespread interest around three years later with the introduction of methods such as Smart-seq2 [101] that significantly reduced the cost of library preparation. It is quite incredible that in the space of a little over five years the technology has sufficiently advanced from profiling a few tens of cells to thousands and hundreds of thousands using technologies such as Drop-seq [82] and the 10x platform [147].

This mass-production of single-cell data has led to an explosion of methods development for downstream analysis by both computational biologists and statisticians. There are now a wide range of methods specific to single-cell data for common analyses such as normalisation [5, 75], differential expression [27, 59, 64], dimensionality reduction [102, 134], and "pseudotime" analyses (see section 1.2). Indeed, there are now over 100 dedicated tools for analysing single-cell data, the majority of which have been produced since 2015.

1.1.2 *Why quantify gene expression in single-cells?*

The basic logic behind quantifying gene expression in single cells is that we may expect some heterogeneity at the single cell level that is averaged over in bulk. For example, cells may undergo some biological process asychronously, where even time series analysis would average over progression at each point[1] [129]. Furthermore, multiple distinct cell types [63] or rare cell types [43] may exist within a population that a bulk assay would be unable to resolve. Some authors have even suggested that Simpson's paradox could mean that gene co-expression patterns observed in bulk are incorrect once groupings at the single-cell level are taken into account [128].

Arguably one of the most popular applications of single-cell RNA-sequencing has been to differentiating cells, tracking the changes in gene regulation as the cells progress. Early examples include tracking the differentiation of primary human myoblasts [129], which revealed switch-like changes in expression of key regulatory factors, and tracking the differentiation of alveolar cells in mice that allowed for the identification of bipotent progenitor cells [130]. Such ideas have been extended to many biological systems, such as differentiating hematopoietic stem cells [66, 148], sensory epithelial cells from the inner ear [15], and more exotic examples such as the differentiation of zebrafish thrombocytes [80] and neural stem cells in planarians (flatworms) [91].

A further popular application of single-cell RNA-seq has been identifying subtypes of neurons (see e.g. [105] for a full review). Possibly the first to do so was Usoskin et al. [133] who profiled 622 single mouse neurons from the dorsal root ganglion involved in the primary sensory system. Through a basic iterative PCA and clustering-by-eye procedure they identified 11 functionally distinct subtypes of primary sensory neurons. Shortly after Zeisel et al. [146] performed single-cell RNA-seq on 3005 cells from the mouse cerebral

1 Computational methods to correct for this (known as "pseudotime ordering") are the main focus of this thesis and are discussed in-depth in section 1.2)

cortex. Rather than rely on the ad-hoc procedure of clustering-by-eye, they developed an iterative biclustering algorithm called BackSPIN. This suggested 9 major distinct cell types, which was further partitioned into 47 by repeating the clustering on each major cell type. When attempting to identify novel cell types based on gene expression it is obviously advantageous to sequence as many cells as possible. Not content with the poor scalability of existing procedures, Macosko et al. [82] developed an entirely new library preparation procedure termed Drop-seq (see section 1.1.4) to cheaply profile thousands of cells at once. This allowed for the expression quantification of a staggering 44,808 cells from the mouse retina that were clustered into 39 transcriptionally distinct cell types using density-based clustering on a t-SNE projection. Although identifying cellular subtypes is a popular and intuitive idea, several issues remain. Surprisingly, there is no strict definition of a "cell type", a challenge to be tackled by the Human Cell Atlas [111]. Furthermore, practically every clustering algorithm - even those that choose the number of clusters "automatically" - have tunable parameters and modelling assumptions that leave the final number of cell types discovered open to subjectivity.

A recent development in single-cell technologies has been the ability to simultaneously profile "omics" other than gene expression alone (see [78] for a full review). For example, G&T-seq [79] allows for combined measurement of the genome and transcriptome in a single-cell. There are clear uses of such methods in cancer genomics, where the combined measurement of DNA mutations and RNA expression would allow for the quantification of clonality on gene expression and subsequently cellular regulation. Further methods have been used to jointly profile the methylation states of DNA at the single-cell level along with gene expression, known as scM&T-seq [4, 55]. A recent publication has extended this to simultaneously measure chromatin accessibility, DNA methylation, and gene expression in single-cells [26].

1.1.3 *Bulk expression quantification*

The first mass-produced technology able to quantify transcriptome-wide gene expression was DNA microarrays. To infer nucleic acid abundance, DNA fragments (after reverse transcription from mRNA transcripts for gene expression quantification) bind to "probes" of known nucleic acid sequence and fluoresce strongly if a large quantity of nucleic acid with the correct complementary sequence is present.

However, gene expression microarrays suffer several disadvantages. The requirement of cDNAs to bind to probes of known sequence precludes de novo transcriptome construction, quantification of differential splicing, and any somatic variant calling. Furthermore, DNA microarrays are known to suffer from technical artefacts, such as positional effects where the spatial position of the probe affects the expression estimate[2] (see e.g. [100, 145])

To overcome these issues researchers began direct sequencing of the cDNA fragments in a process known as RNA-sequencing (RNA-seq), which appeared in a trio of papers around the summer of 2008 [72, 92, 94]. RNA-seq normally begins with isolation of the RNA and depletion of any DNA using deoxyribonuclease to avoid genomic contamination. Optional enrichment of mRNAs may be performed using filtering for polyadenylation, before reverse transcription to cDNA and high-throughput ("next generation") short read sequencing.

1.1.4 *Single-cell library preparation*

Single-cell library preparation - the process of converting each cell's mRNA to barcoded cDNAs ready for sequencing - has come a long way since the manual isolation of Tang

2 This leads to the joke, "what's the difference between white noise and microarrays? White noise doesn't contain artefacts."

et al. in 2009 [122]. The most popular methods - such as STRT-seq [56], Smart-seq [109], Smart-seq2 [101], and CEL-Seq [47], all follow similar principles. Cells are (typically FACS) sorted into 96 (and more recently 384) well plates. In each well cells are lysed to release the RNA, which is then reverse transcribed to cDNA. There is then typically some amount of amplification to increase the overall quantity of cDNA for more reliable sequencing. Next, each cell is uniquely barcoded by appending a known sequence of nucleic acid to each cDNA molecule. This step greatly increased the throughput of single-cell RNA-seq by allowing multiplexed sequencing of all cells at once. Finally, the individual libraries are ready for sequencing.

A popular commercial solution in the early days of single-cell RNA-seq was the Fluidigm C1 system that attempted to automate as far as possible the library preparation process. By using microfluidics, the C1 could prepare libraries using nanolitre reaction volumes which reduced reagent costs and improved the accuracy of expression quantification [104]. However, the C1 system notably suffered from low capture rates and "doublets" whereby two cells were captured and sequenced rather than one.

While such plate-based and microfluidic assays were incredibly successful, the single-cell isolation and barcoding steps precluded the majority of studies exceeding 1000 cells. However, this changed in 2015 with the introduction of "droplet" based technologies, namely Drop-seq [82] and inDrop [61]. In such assays cells are encapsulated in nanolitre droplets along with distinctly barcoded beads that allow for multiplexed sequencing. This allows for an incredible increase in the number of cells - the original Drop-seq publication sequenced 44,808 - and is currently being commercialised through companies such as 10x genomics [147].

1.1.5 *Features of scRNA-seq data*

1.1.5.1 *Mean-variance relationship*

One feature of single-cell RNA-seq data (and indeed RNA-seq data in general) is a non-trivial relationship between the variance of the expression and its mean. Accurate characterisation of this relationship is particularly crucial for small sample size datasets in order to robustly estimate the variance for statistical tests of differential expression [115]. Statistical models of RNA-seq data typically fall in two camps when it comes to the distribution used for the likelihood. The first are the "count-based" methods that act on the raw counts (scaled by size factors) using negative binomial distributions such as DESeq(2) and EdgeR [2, 115]. The second are the "abundance-based" methods that log-transform[3] the (normalised) counts and model using a Gaussian likelihood, such as Limma Voom and Sleuth [68, 103].

In the case of the count-based methods, RNA-seq data are well-approximated by negative binomial distributions [2, 114, 115] which relates the variance σ^2 to the mean μ of the counts for a given gene by

$$\sigma^2 = \mu + \alpha\mu^2 \tag{1}$$

where α is a dispersion parameter and in the limit $\alpha \to 0$ a Poisson noise model is recovered. In other words the variance in expression increases quadratically with the mean number of counts for a particular gene. A common quantity to examine in such cases in the squared coefficient of variation $\mathrm{CV}^2 = \frac{\sigma^2}{\mu^2}$ which for the negative binomial model implies $\mathrm{CV}^2 = \alpha + \frac{1}{\mu}$. This relationship can be seen for a dataset of differentiating thymic epithelial cells in figure 1.

3 Typically with some offset c via $\log(x + c)$ to avoid divergence issues when $x \to 0$.

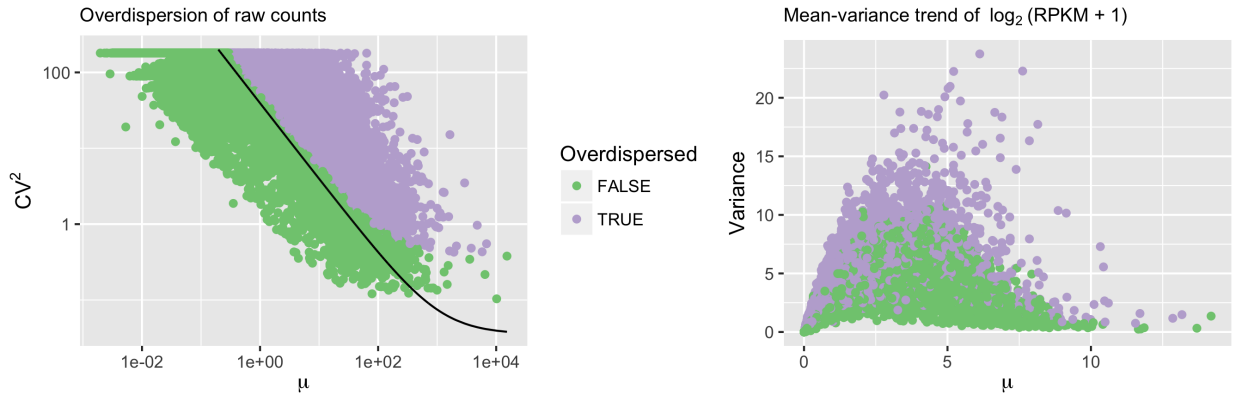


Figure 1: Mean variance relationships in an example single-cell dataset of thymic epithelial cells. Single-cell RNA-seq data shows a characteristic dependency on the $\mathrm{CV}^2 = \frac{\sigma^2}{\mu^2}$ on the mean counts for each gene (left). Methods such as [13] can then be used to identify genes that vary more than expected at random due to the technical noise in the dataset. An alternative view is to look at the mean variance trend in log space (right), where the overdispersed genes from [13] are often the most variable.

This relationship between the squared coefficient of variation and mean expression was exploited in a seminal single-cell RNA-seq paper [13] to identify genes that varied more than can be expected at random due to technical noise in the dataset. First a $\mathrm{CV}^2 -$ mean relationship is fitted for ERCC spike-ins[4] to infer a null model of the variance relationship we expect in the data due to technical variation alone. A p-value for the coefficient of variation of endogenous (biological) genes given their mean expression can then be computed under the null model and subsequently genes "overdispersed" at some significance threshold can be identified. An example of this for differentiating thymic epithelial cells can be seen in figure 1, with the overdispersed genes shown in purple.

In the second set of "abundance-based" methods there is no analytical expression for the variance of the log-counts in terms of the mean since in general $\mathrm{Var}[\log(X + c)] \neq \log(\mathrm{Var}[X] + c)$. An example of this for the same set of thymic epithelial cells can be seen in figure 1, coloured by the overdispersion test results from [13]. This forms

4 ERCC spike-ins are small panels of known RNA sequence added in known concentrations to experiments to both calibrate the molecular concentrations to RNA-seq counts and to infer the limit of detection.

a characteristic "hump" shape, with overdispersed genes typically exhibiting moderate mean expression but high variance in log space. Differential abundance methods such as Limma Voom [68] and Sleuth [103] fit smoothed curves (such as LOESS curves) to obtain a nonparametric numerical approximation to the relationship. We use this formulation in chapter 3, where we note that "interesting" genes lie close enough to the diagonal to posit a linear mean-variance relationship in log space, i.e. $\sigma^2 = \phi\mu$ for some constant $\phi > 0$.

1.1.5.2 *Dropout*

An often discussed feature of single-cell RNA-seq data is the idea of *dropout* where the probability a gene is detected in a cell is dependent on the mean expression of that gene. For each mRNA there is a stochastic probability of a failure to reverse transcribe into cDNA for sequencing. If only a few mRNAs exist for a given species (in other words, if a gene is lowly expressed) then this results as zero counts for that gene, rather than a small quantity detected. Obviously the more mRNAs present to begin with, the smaller the chance that reverse transcription misses all mRNAs of that species, inducing a dependence between a gene's (true) expression level and the dropout probability. This problem is further confounded by sequencing depth - if relatively few reads for a cell are sequenced then there is a high chance that a cDNA (which was lucky to be reverse transcribed in the first place) won't subsequently be sequenced. Thus typically the lower the read depth of a dataset the sparser the resulting count matrix is.

The characteristic pattern of proportion of cells in which a gene is expressed compared to the mean expression can be seen in figure 2A for an example dataset of thymic epithelial cells. Most single-cell datasets exhibit a set of genes that are not expressed (top-left corner), a set of housekeeping genes that are constitutively expressed at high

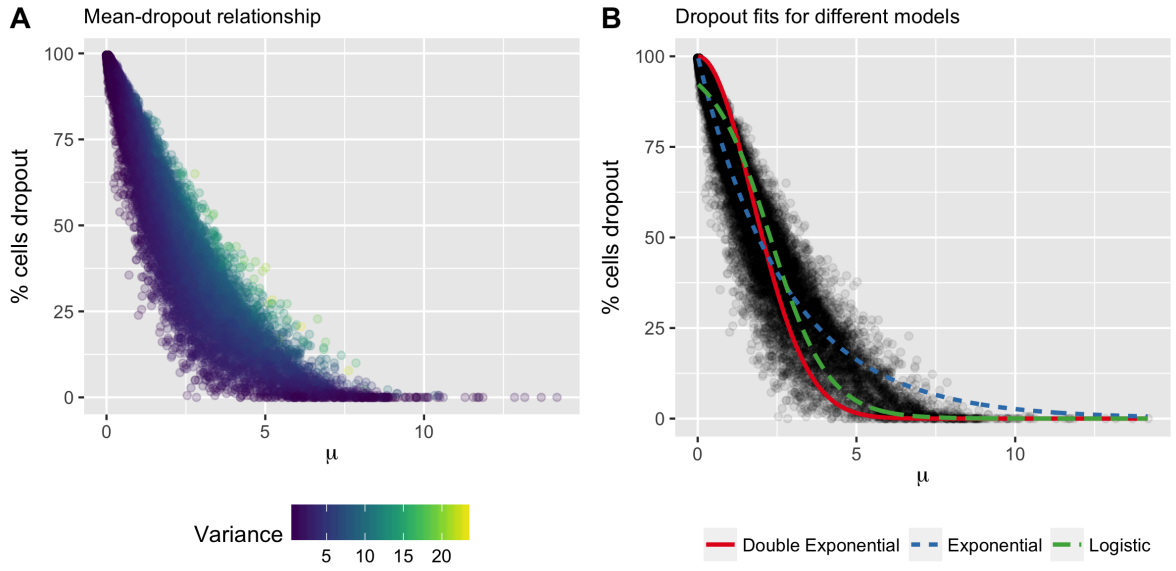


Figure 2: Relationship between dropout (proportion of cells not expressing a gene) and mean expression in an example single-cell dataset of thymic epithelial cells. Typically, the high variance genes lie along the right hand "edge" of the relationship (left). A number of models to account for dropout as a function of mean expression have been proposed, including exponential, double exponential and logistic (right).

levels (bottom right), and a set of high-variance genes that contribute to heterogeneity in the dataset inbetween.

Several methods have attempted to correct for dropout through statistical modelling. An early example was SCDE [59] that uses a mixture model of a point pass at zero (the dropout component) and an "amplified" component. Noting that whether a gene is dropout is not a constant probability but dependent on the latent expression, they used a mixture-of-experts model where the probability of being amplified was logistic on the latent expression, i.e. $p_{\text{dropout}} = \text{Logistic}(\beta_0 + \beta_1\mu)$ where μ is the latent expression and β_0 and β_1 are free parameters. An example of the logistic regression fit can be seen in figure 2B.

A different dropout model was implemented in the dimensionality reduction model ZIFA (Zero Inflated Factor Analysis, [102]). ZIFA models the probability of a dropout as $p_{\text{dropout}} = \exp(-\lambda x^2)$ for a "dropout rate" λ and latent (true) expression x. A

factor analysis model then operates on x (see section 1.3.3) to find a reduced dimension representation of the data accounting for dropout. A further model called M3Drop [3] models dropout using Michaelis-Menten enzyme kinetics to select highly-variable, though the authors admit that their model is a specific case of SCDE.

Some methods take a different approach to dropout and treat it as a missing data problem that requires imputation. MAGIC [29] uses a diffusion-based approach (similar to diffusion maps, see section 1.3.4.2) that computes a low-dimensional graph embedding before "imputing" each cell's expression by a weighted sum of its neighbours. A similar method SAVER [149] models expression using a Poisson likelihood with a latent true expression that is dependent on similar genes in the same cell (rather than the same gene in similar cells as per MAGIC).

1.2 PSEUDOTIME & TRAJECTORIES

1.2.1 *The pseudotime estimation problem*

In many single-cell assays cells undergo some transformation over time. Examples include the differentiation of stem cells into neurons or skin cells, cells progressing through the cell cycle, or cells undergoing apoptosis (cell death). Ideally we would like to track expression changes over time, revealing key gene expression changes that correlate with the biological progression of interest.

However, most gene expression quantification methods to date - particularly transcriptome-wide ones such as single-cell RNA sequencing - destroy the cell during the measurement process, thus prohibiting repeated time-series measurement on the same cell. A first solution would be to repeatedly measure sets of cells at different time points, analogous to bulk RNA sequencing. However, the transcriptional heterogeneity at the single-cell level

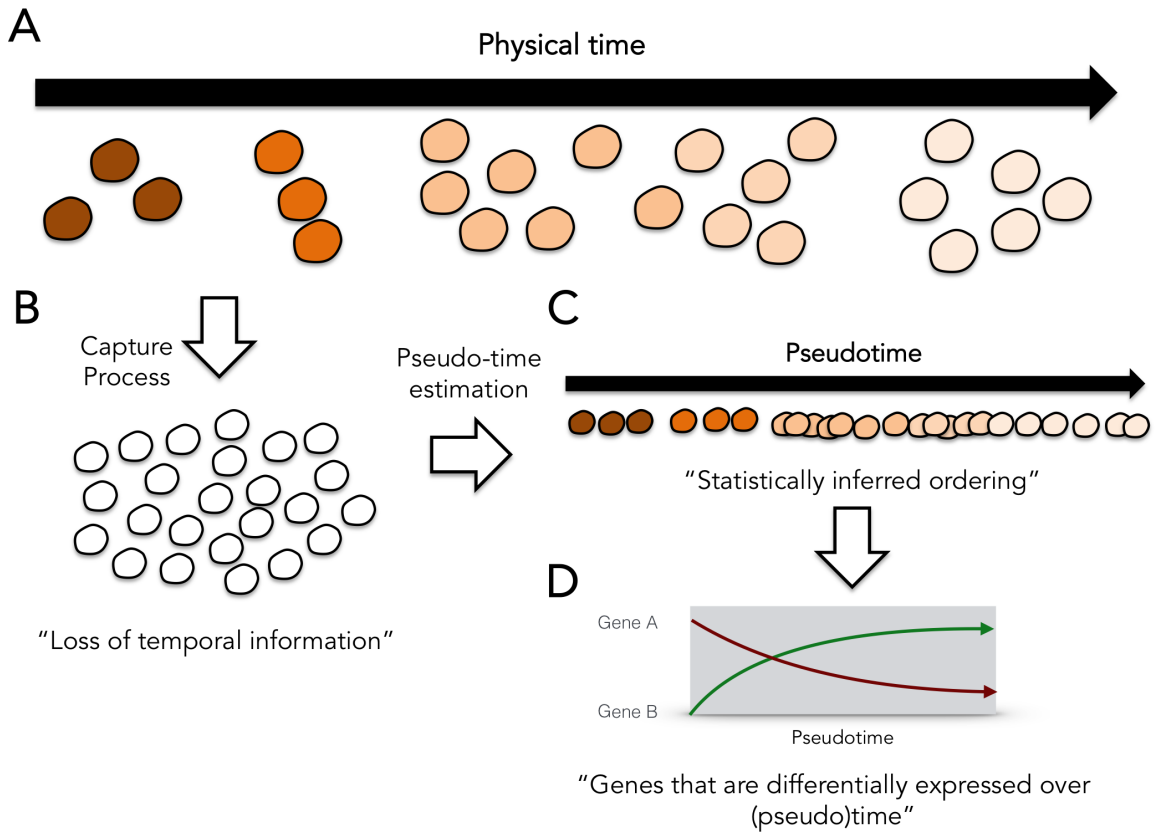


Figure 3: The pseudotime estimation problem. **A** Cells undergo some physical time process such as differentiation, cell-cycle, or apoptosis. **B** The cell capture and expression quantification method - such as single-cell RNA-seq or single-cell mass-cytometry - leads to a loss of temporal information (if it were possible to measure in the first place). **C** Pseudotime algorithms attempt to reconstruct the original time series using gene expression data alone. **D** Downstream analysis such as differential expression proceeds with the pseudotimes in place of the physical times.

leads to an asynchronicity of expression [129], meaning some cells at a given time point will be transcriptionally more similar to those at the next and some transcriptionally more similar to those at the previous.

As a solution to these issues the idea of assigning cells a *pseudotime* was proposed. In such a setting an algorithm takes the transcriptome-wide gene expression measurements for all cells and maps each on to a one-dimensional pseudotime[5]. The pseudotime of a

5 Mathematically, a pseudotime algorithm is just a function $f : \mathbb{R}^{N \times G} \to \mathbb{R}^N$ for N cells and G genes, mapping the gene expression matrix to a vector of pseudotimes (one for each cell). Strictly it is not a function $f : \mathbb{R}^G \to \mathbb{R}$ since the pseudotime for a given cell is (typically) dependent on all other

particular cell quantitatively represents that cell's progression through the biological process of interest and is not necessarily supposed to represent the physical (capture) time due to the transcriptional asynchronicity mentioned above. Consequently, each cell acts as a surrogate time point, the transcriptome of a given cell representing the supposed transcriptional signature of the biological process at that cell's assigned (pseudo-)time point. The set of cells ordered by pseudotime is often described as a *trajectory* representing the continuous progression of transcription along the process of interest (figure 3).

Several downstream analysis typically follow the assignment of pseudotimes to cells. Examples include identifying genes differentially expressed along the trajectory [18] or gene clusters differentially regulated [129]. However, care must be taken to realise that the pseudotimes are an uncertain estimate derived from exceptionally noisy input data and not a "perfectly" measured quantity such as physical capture time; this is the subject of chapter 2.

1.2.2 *Early applications to bulk expression data*

The ideas behind pseudotime orderings were first introduced in the context of ordering bulk microarray samples by Magwene et al. in 2002 [83]. They examine the case of tracking gene expression from tumour samples in mouse models from disease inception and argue that "the samples obtained from a mouse model of cancer do not represent a time series with a well-defined developmental order" because early in cancer multiple tumours within the same tissue may progress asychronously. Magwene et al. therefore suggest that obtaining an ordering of samples based on the gene expression measurement alone may more accurately depict cancer development.

cells. Such an explicit functional form reveals that pseudotime algorithms are in fact just a form of dimensionality reduction.

Their algorithm is motivated by the fact that the noise-free progression of microarray samples would trace out a smooth one-dimensional curve embedded in high dimensional expression space. They begin by fitting a minimum spanning tree[6] (MST) to the data using a modified distance function that is the "standard pairwise dissimilarity" if two points are "relatively similar" and the sum of the pairwise dissimilarities between two samples if "relatively dissimilar".

If the MST represents a path (i.e. it has no branches) then this is taken to be the ordering of cells. However, if the MST does have branches then the "diameter path" (the longest path through the MST) is computed and various heuristics are run to assess whether the diameter path essentially represents the dominant mode of variation through the data. If so then the diameter path is taken to be the ordering and if not a set of orderings representing the uncertainty in path variations is returned. Magwene et al. applied this algorithm to time series data of bacterial gene expression and found that it reconstructed the true time ordering of the samples accurately.

Gupta and Bar-Joseph (2008, [44]) expanded on the ideas of Magwene et al. in three important ways. Firstly, they formally proved that a method could reconstruct the temporal order of expression profiling measurements. Interestingly, this proof shows that expression profiles of samples near each other in time will be more similar than those further apart in time, provided the change in expression is correlated over time[7] with high probability. Secondly, they applied their method to static cancer data rather than simply recovering the capture time from time-series microarrays, though in practice the algorithm of Gupta et al. could be applied here also. Finally, their method recovered expression profiles for individual genes using spline fits. Interestingly, their model is close to a nonlinear factor analysis model like that considered in chapter 3, but performs a

6 A minimum spanning tree is a graph that connects all vertices together without any cycles using the minimum overall edge weight, a little like joining all the dots using the least ink possible.

7 i.e. if a gene is upregulated at a given time point, it is more likely to be upregulated at the next and vice versa.

line search at each iteration of an expectation-maximisation (EM) algorithm to find the pseudotimes of each sample that minimises the mean squared error, rather than maximum likelihood or Bayesian inference of the pseudotimes.

Following this was the introduction of *Sample Progression Discovery* (SPD) by Qui et al. in 2011 [106]. This built on similar ideas of using MSTs to connect together microarray samples in a time ordering, with several important differences. SPD performs consensus clustering on the expression matrix to identify correlated gene expression modules and calculates a MST for each of them. It then chooses a subset of gene modules that have concordant MSTs and calculates an overall progression MST using only these. The authors emphasise that such an approach provides a feature selection ability, highlighting modules of genes associated with changes in the MST.

1.2.3 *Single-cell pseudotime inference algorithms*

1.2.3.1 *Overview*

Ordering bulk microarray samples was arguably a niche area, with the three studies discussed above constituting the main work in the field. However, the advent of single-cell sequencing lead to an explosion of interest in the field, mostly due to the higher inter-sample heterogeneity present in single-cell data. The first single-cell pseudotime algorithm was published in March 2014 [129]; at time of writing (May 2017) there are now approximately 33 single-cell pseudotime inference algorithms that have at least been preprinted[8], a rate of roughly one per month.

This renaissance was largely spurred by the development of `Monocle` [129], discussed in detail in section 1.2.3.2. Monocle's two step procedure - an initial dimensionality reduction step using independent component analysis followed by a "cell ordering" step

8 A useful spreadsheet of single-cell software is maintained at https://tinyurl.com/mqabyur.

Algorithm	Reference	Dimensionality reduction	Cell ordering	Probabilistic	Branching
Monocle	[129]	ICA[a]	MST[b]	No	Yes
Wanderlust	[8]	N/A	KNN-graph[c]	No	No
Monocle 2	[108]	DDR-tree	Distance from root cell	No	Yes
Scuba	[85]	t-SNE[d]	Principal curves	No	Yes
Diffusion pseudotime	[46]	N/A	Diffusion distance from root cell	Interpretation[e]	Yes
SLICER	[138]	LLE[f]	Principal curves	No	Yes
DeLorean	[112]	N/A	GPLVM[g]	Yes	No
TSCAN	[57]	PCA	Cluster-based MST	No	Yes
Waterfall	[120]	PCA	Cluster-based distances	No	No
Ouija	[17]	N/A	Nonlinear factor analysis	Yes	No
MFA	[19]	N/A	Mixture of factor analysers	Yes	Yes

Table 1: An overview of some pseudotime algorithms. Most involve a dimensionality reduction step followed by pseudotime assignment ("cell ordering") in the reduced space, though arguably this constitutes a single dimensionality reduction step. Methods shaded in grey are introduced in this thesis.

a Independent component analysis
b Minimum spanning tree (cells are projected onto the longest path through the MST).
c k-nearest neighbour graph. Trajectory inferred by an ensembl of random walks.
d See section 1.3.4.5
e DPT has an interpretation in terms of the probability of one cell state transitioning into the other. However, it does not define a generative probabilistic model.
f See section 1.3.4.4
g See section 1.3.5

to order the cells using MSTs - greatly influenced the algorithms that followed. Examples include `TSCAN` that uses PCA for dimensionality reduction followed by clustering and MSTs for cell ordering; `embeddr` that uses laplacian eigenmaps for dimensionality reduction and principal curves for cell ordering; and `Waterfall` that uses PCA for dimensionality reduction followed by connecting clusters in the reduced space for cell ordering. An overview of some pseudotime algorithms is given in table 1 with several important contributions discussed in detail in the following sections.

1.2.3.2 *Monocle*

Monocle was designed to understand the gene expression dynamics that accompany the differentiation of stem cells into human skeletal muscle myoblasts (HSMM). The experimental design was to take stem cells and engineer a serum switch to induce differentiation, then perform single-cell RNA-sequencing at 0h, 24h, 48h and 72h afterwards. The authors noted that the expression patterns of key marker genes over time (such as switch-like inactivation of *ID1*) was absent, and suggested asynchronicity of cellular development could be the underlying cause and thus a pseudotemporal ordering of cells was required.

Monocle begins with a gene selection step to identify which should be retained for ordering the cells. In the original publication the authors used those genes differentially expressed[9] between the time points, though emphasise that other approaches (such as selecting highly variable genes) would work also.

The algorithm proceeds by using independent component analysis (ICA) to reduce the dimensionality of the dataset down to two dimensions. ICA attempts to find latent components in the data that are statistically independent and assumes that each com-

9 Using a Tobit likelihood model; see chapter 2 for further discussions on differential expression over pseudotime.

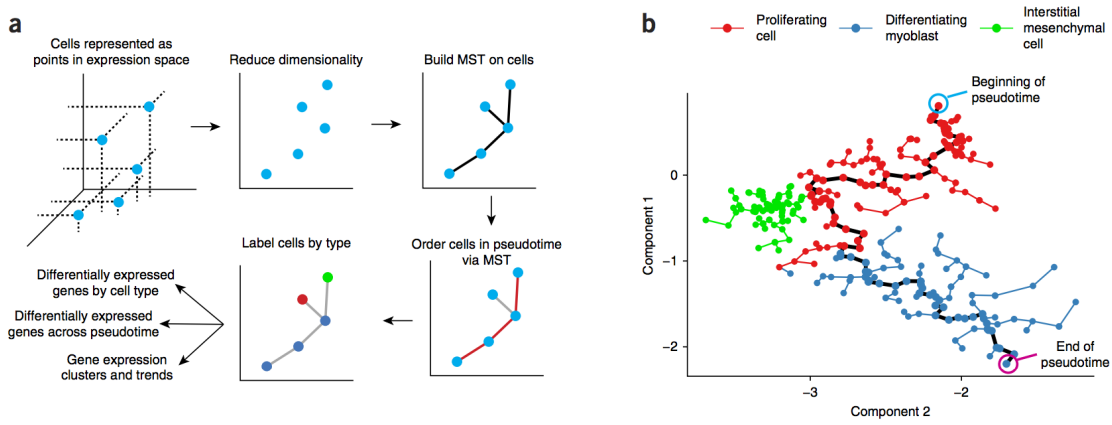


Figure 4: The Monocle pseudotime algorithm (reused with permission). **a** Dimensionality reduction is performed on the expression matrix using ICA. A MST is then fitted in the reduced space and cells are ordered along the diameter path, with branching also detected. Downstream analysis then includes differential expression and gene clustering. **b** Monocle applied to human skeletal muscle myoblasts. The authors identified three cell types (proliferating cells, differentiating myoblasts, and interstitial mesenchymal cells). The main pseudotime trajectory progresses from the proliferating cells to the differentiating myoblasts, with the contaminating mesenchymal cells appearing on a separate branch.

ponent is non-Gaussian[10]. Monocle then proceeds with a secondary cell ordering step in the reduced space that largely follows the procedure of Magwene et al. [83] (see section 1.2.2) in constructing a MST in the reduced space and finding the "diameter path" (i.e. the longest path through the MST) as the trajectory. In order to find biological branches (rather than technical ones), the algorithm finds branches from "indecisive" vertices (those with degree greater than two) and returns the k longest, where k is an integer set by the user in line with prior knowledge of the expected number of terminally differentiated cells in the sample.

1.2.3.3 *Wanderlust*

Published just a month after Monocle was Wanderlust [8], an algorithm developed to uncover human B cell lymphopoiesis using single-cell mass cytometry data. Compared to transcriptome-wide RNA sequencing, mass cytometry measures a smaller number of markers. In this study, a custom panel of 44 makers designed to characterise B-cells were measured, including "phenotypic proteins, transcription factors, regulatory enzymes, cell-state indicators, and activation of regulatory signalling molecules".

Wanderlust begins by constructing a k-nearest neighbour graph between all cells using euclidean distance and selects a small random subset of cells as "waypoints". It then creates a random ensemble of graphs by subsampling cells to mitigate the impact of "short circuits" on the trajectory construction[11]. Wanderlust then assigns each cell an initial time as the shortest path through the graph from a manually chosen "early" cell, and performs an iterative procedure where each cells pseudotime is a weighted average of the distances to the waypoint cells, which is repeated until convergence. The overall pseudotime for each cell is taken as the average over the ensemble of graphs.

10 ICA finds a projection that maximises the kurtosis in the latent space, compared to PCA that maximises the variance (see section 1.3.1).

11 Short circuits are defined as "spurious edges between distant cells" - two cells distant in pseudotime are connected due to random fluctuations in their gene expression. In theory such a problem will be worse in the 40 dimensional mass cytometry data than 1000+ dimensional single-cell RNA-seq.

1.2.3.4 *DeLorean*

DeLorean [112] was published approximately two years after Monocle and Wanderlust and is mentioned here as an example of an entirely different method. DeLorean departs from previous approaches in using a probabilistic model called a Gaussian Process Latent Variable Model (GPLVM, discussed in section 1.3.5 and the subject of most of chapter 2).

The GPLVM learns a probabilistic mapping from the one dimensional latent pseudotimes to the observed gene expression profiles. The basic idea is that if two cells have similar pseudotimes then their transcriptomes will be highly correlated and thus will have similar measured expression profiles[12]. Inference in the DeLorean model proceeds using the Stan probabilistic programming language, which performs automatic Bayesian inference using either Hamiltonian Monte Carlo (HMC) or Automatic-Differentiation Variational Inference (ADVI).

DeLorean allows for the specification of informative priors on the latent space centred around the capture times of the cells. If k_n is the capture time of cell n, then the prior on n's pseudotime t_n takes the form $t_n \sim \mathcal{N}(k_n, \sigma_n^2)$, so the confidence that the pseudotime lies close to the cell's capture time σ_n may be set by the user. The characteristic length scale of the GPLVM kernel (see section 1.3.5) is also set by the user, which corresponds to the expected number of fluctuations of the gene expression profile across pseudotime. Further, DeLorean specifies a generative model of the observed gene expression space, though this requiers parameters for each output (gene) which can scale poorly, so in practice pseudotime fitting is limited to a small number of genes chosen *a priori*.

A particular strength of DeLorean is the ability to critically evaluate it because of its formulation as a probabilistic model. Given the underlying continuity assumption

12 This is essentially a continuity assumption - that cells close in pseudotime will have expression profiles "close" in expression space. This assumption was discussed in [44] (see section 1.2.2) who showed the change in expression over (pseudo-)time must be correlated for it to be true. This logic underlies many if not all pseudotime inference methods, such as DPT [46].

common to all pseudotime algorithms we would expect they are all multimodal to some extent, or at least have multiple solutions that are dependent on algorithm parameters or initial values. However, their ad-hoc algorithmic nature precludes any assessment of this, while in the DeLorean model these can be recognised as a multimodal posterior distribution and steps taken to mitigate this such as the systematically incorporating prior information in the form of capture times to constrain the latent space.

1.3 STATISTICAL LATENT VARIABLE MODELS

1.3.1 *Principal component analysis*

Principal component analysis (PCA, [58]) is a ubiquitous linear dimensionality reduction technique. PCA has several interpretations[13], but the most common is that of a linear projection to a low-dimensional space such that the variance of the data points in the projected space is maximised. Intuitively, this can be thought of as finding a linear subspace of the high-dimensional data space that best "explains" the data.

Mathematically, we start with G-dimensional data points $\{\mathbf{y}_n\},\ n = 1, \ldots, N$ and wish to find the Q principal axes $\boldsymbol{\lambda}_q,\ q = 1, \ldots, Q$ that act as a mapping from the observed to latent space such that the variance of the input points projected to the latent space is maximal. To find $\boldsymbol{\lambda}_q$ the eigenvectors of the sample covariance matrix

$$\mathbf{S} = \frac{1}{N} \sum_{n=1}^{N} (\mathbf{y}_n - \bar{\mathbf{y}})(\mathbf{y}_n - \bar{\mathbf{y}})^T \tag{2}$$

are calculated where $\bar{\mathbf{y}}$ is the sample mean of $\mathbf{y}_n$. The Q principal axes are then the Q eigenvectors of $\mathbf{S}$ with the largest eigenvalues. The latent space projections $\mathbf{z}_n$ of each

13 Lior Pachter has an excellent blog post on this at https://liorpachter.wordpress.com/2014/05/26/what-is-principal-component-analysis/

data point may then be obtained via $\mathbf{z}_n = \mathbf{\Lambda}(\mathbf{y}_n - \bar{\mathbf{y}})$ where $\mathbf{\Lambda}$ is the $Q \times G$ matrix whose rows are $\boldsymbol{\lambda}_q$ ordered by decreasing eigenvalue.

PCA is frequently used in computational genomics. It is commonly used for visualisation of genomic variation data such as single nucleotide polymorphism (SNP) datasets, where it has an elegant interpretation in terms of the underlying genealogical history of samples [88]. It is often applied to microarray gene expression data for tasks such as data visualisation (see e.g. [113]) and for clustering samples [142].

Since the advent of single-cell RNA-sequencing, PCA has been the go-to dimensionality reduction algorithm for exploratory data analysis[14]. Examples include latent space projections to understand cell types and hierarchies in the developing lung [130] and the transcriptional states defining differentiation of embryonic stem cells under different serums [62]. PCA is also used for clustering cells, either as a preprocessing step prior to clustering algorithms like $k-$means or as part of likelihood-based clustering [141].

PCA also forms an initial step in many pseudotime algorithms such as in TSCAN [57] and Waterfall [120]. However, no studies have actually considered that a principal component of the data itself could *be* the pseudotemporal trajectory. Intuitively such an idea is appealing since we would expect the pseudotemporal process to be the dominant source of variation within the data.

1.3.2 *Probabilistic principal components analysis*

One weakness of standard PCA is the absence of any probabilistic framework or interpretation. In a landmark paper [124] Tipping & Bishop derived[15] probabilistic PCA (PPCA) that provides an explicit generative model for PCA and relates the maximum likelihood estimates of parameters to the algorithmic estimation we discussed previously.

14 Though faces stiff competition from t-Stochastic Neighbour Embedding (tSNE).
15 By first considering a factor analysis model that we discuss in section 1.3.3.

The generative model for PPCA is given by

$$\begin{aligned} \mathbf{z}_n &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\ \mathbf{y}_n &\sim \mathcal{N}(\mathbf{\Lambda}\mathbf{z}_n + \boldsymbol{\mu}, \sigma^2\mathbf{I}) \end{aligned} \tag{3}$$

where $\boldsymbol{\mu}$ is the expectation of $\mathbf{y}$ and σ^2 is the isotropic measurement variance. In other words, if we centre $\mathbf{y}$ so that $\mathrm{E}[\mathbf{y}] = \mathbf{0}$ then PPCA corresponds to a Gaussian noise model with a mean given by standard PCA and isotropic covariance. Tipping and Bishop further derived closed form expressions for the maximum likelihood estimates (MLEs) of $\mathbf{\Lambda}$ and σ^2 along with the conditional distribution of the latent variables $p(\mathbf{z}_n|\mathbf{\Lambda}, \mathbf{y}_n)$. They demonstrated that in the limit $\sigma^2 \to 0$ the MLE estimate of $\mathbf{\Lambda}$ is identical (up to arbitrary rotation) to that of standard PCA. Furthermore, the MLE estimate of σ^2 is given by

$$\sigma^2_{\mathrm{MLE}} = \frac{1}{G-Q} \sum_{j=Q+1}^{G} \zeta_j \tag{4}$$

where ζ_j are the eigenvalues of the sample covariance matrix ordered by decreasing size. In other words σ^2 is the variance "lost" in the projection, averaged over the remaining dimensions.

1.3.3 *Factor analysis*

Factor analysis (FA) precedes both PCA and PPCA, dating back to Spearman's work on "general intelligence" [121] (see section 1.3.3.1). The overall model is very similar to

Student	Maths	Physics	Biology
A	8	9	7
B	3	1	5
C	6	6	8
D	4	2	7

(a) School test scores for children across subjects

Cell	*MYOD*	*MYF5*	*MYOG*
A	8	9	7
B	3	1	5
C	6	6	8
D	4	2	7

(b) Gene expression values of cells for transcription factors

Table 2: Children's test scores across subjects are correlated due to latent factors (a) in a similar way to the gene expression of key markers during differentiation (b).

that of PPCA with the only difference being the measurement (co-)variance is diagonal rather than isotropic. This gives a generative factor analysis model of the form

$$\begin{aligned} \mathbf{z}_n &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\ \mathbf{y}_n &\sim \mathcal{N}(\mathbf{\Lambda}\mathbf{z}_n + \boldsymbol{\mu}, \mathbf{\Sigma}) \end{aligned} \tag{5}$$

where $\mathbf{\Sigma} = \text{diag}(\sigma_1^2, \ldots, \sigma_G^2)$.

Maximum likelihood inference of factor analysis models may be performed using iterative procedures such as expectation-maximisation (EM) or Bayesian inference performed through MCMC methods or variational inference.

Factor analysis notably suffers from the *rotation problem*. Given a $Q \times Q$ orthogonal rotation matrix $\mathbf{R}$ the likelihood is invariant under simultaneous rotation of both the factor matrix $\mathbf{\Lambda}$ and latent values $\mathbf{z}$[16]. Note that if $Q = 1$ the rotation problem is essentially a scaling problem (we can divide $\mathbf{\Lambda}$ by k and multiply $\mathbf{z}$ by k to achieve the same likelihood) but this is solved by fixing the scales of $\mathbf{\Lambda}$ and $\mathbf{z}$ through priors.

1.3.3.1 *Origins of factor analysis and connection to pseudotime*

Factor analysis was first introduced by Spearman in 1904 [121] by considering children's test scores in subjects such as that in table 2(a). He noticed that the scores are correlated - for example, if a child has a high score in physics they are more likely to have a high score in maths, and vice versa. Spearman's insight was that rather than the scores being correlated with each other, they were correlated with an underlying (one-dimensional) hidden factor he termed *general intelligence*. Mathematically this can be expressed as

$$\mathbf{y}_n = \boldsymbol{\lambda} z_n + \epsilon_n \tag{6}$$

where $\mathbf{y}_n$ is the vector of child n's test scores, $\boldsymbol{\lambda}$ is a vector of subject specific constants (the "loadings"), z_n is the "general intelligence" and ϵ_n is any noise not explained by the model[17].

One can see in hindsight the parallels to the pseudotime estimation problem: we measure some high-dimensional quantity $\mathbf{y}_n$ over various samples and have a hazy and somewhat under-defined one-dimensional generating process z_n. In the context of psychology, this process is intelligence: something unmeasurable that we use to make sense of observations in life such as performance in tests. Similarly in biology we have the concept of differentiation trajectories: something equally unmeasurable but which we indirectly observe through molecular measurements such as gene expression quantification (table 2(b)).

Given the parallels between the original interpretation of factor analysis and differentiation trajectories we may suspect that it is particularly suited to the pseudotime

16 Given measurement noise $\boldsymbol{\epsilon}$ the likelihood is unchanged under $\mathbf{y} = \boldsymbol{\Lambda}\mathbf{z} + \boldsymbol{\epsilon} = \boldsymbol{\Lambda}\mathbf{R}\mathbf{R}^T\mathbf{z} + \boldsymbol{\epsilon} = \boldsymbol{\Lambda}'\mathbf{z}' + \boldsymbol{\epsilon}$ where $\boldsymbol{\Lambda}' = \boldsymbol{\Lambda}\mathbf{R}$ and $\mathbf{z}' = \mathbf{R}\mathbf{z}$ are the loadings and projections in the rotated space.

17 Of course the idea of a single "general intelligence" goes against common sense. If there truly is a single latent factor it is possible to form the *tetrad equations*. Later studies showed that deviations in the tetrad equations could not be explained by sampling noise alone. For an overview see https://www.stat.cmu.edu/~cshalizi/350/lectures/12/lecture-12.pdf.

inference. Indeed, chapters 3-5 are devoted to various modifications of factor analysis to infer such trajectories.

1.3.4 *Manifold learning*

While not strictly a class of statistical latent variable model, manifold learning has found much success in single-cell genomics. While methods such as PCA and FA attempt to infer low-dimensional linear subspaces embedded in high-dimensional space, the field of manifold learning extends this to the nonlinear setting. Several manifold learning algorithms applied to single-cell genomics are reviewed below and utilised in chapter 2.

1.3.4.1 *Laplacian eigenmaps*

Laplacian eigenmaps [6] are part of a larger class of *spectral methods* including diffusion maps (section 1.3.4.2). Starting with the $N \times G$ matrix $\mathbf{Y}$, laplacian eigenmaps seeks a $N \times Q$-dimensional embedding $\mathbf{Z}$ with row vectors $\mathbf{z}_n$ for each cell through minimisation of the quantity

$$\sum_{n,n'} W_{n,n'} ||\mathbf{z}_n - \mathbf{z}_{n'}||^2 \tag{7}$$

subject to the constrain that $\mathbf{z}_q^T \mathbf{z}_q = 1 \; \forall q$ where $\mathbf{z}_q$ are the column vectors of $\mathbf{z}$. $\mathbf{W}$ is an $N \times N$ similarity matrix between the samples (cells) with the intuition that if $W_{n,n'}$ is large then the distance between $\mathbf{z}_n$ and $\mathbf{z}_{n'}$ is heavily penalised placing them close together in the reduced space. Conversely, if $W_{n,n'}$ is small then a large distance between $\mathbf{z}$ and $\mathbf{z}_{n'}$ has little effect on the optimisation problem. Solutions for $\mathbf{Z}$ can readily be found by solving an eigenvalue equation. Laplacian eigenmaps were used for pseudotime

inference in the `embeddr` package [20] using a symmetrised k-nearest neighbour graph for $\mathbf{W}$.

1.3.4.2 *Diffusion maps*

Diffusion maps are closely related to laplacian eigenmaps and have been successfully applied to single-cell RNA-seq data both in the context of visualisation [45] and pseudotime inference [46]. The basic idea is to consider points on the manifold in terms of a diffusion process, with a sample more likely to diffuse to one closer to it than further away. It begins by constructing a transition matrix

$$P_{n',n} = \frac{1}{Z(\mathbf{y}_{n'})} \exp\left(-\frac{||\mathbf{y}_n - \mathbf{y}_{n'}||^2}{2\sigma^2}\right) \tag{8}$$

where $Z(\mathbf{y}_{n'}) = \sum_n \exp\left(-\frac{||\mathbf{y}_n - \mathbf{y}_{n'}||^2}{2\sigma^2}\right)$ and σ^2 is a characteristic length scale. $P_{n',n}$ can be thought of as the probability of transitioning or *diffusing* from cell n to n'. A renormalised transition matrix $\tilde{\mathbf{P}}$ can then be defined that takes into account the local density of samples in the space. One can then decompose the diffusion distances into a sum over the eigenvectors of $\tilde{\mathbf{P}}$ weighted by eigenvalues, implying that retaining eigenvectors for the first k ordered eigenvalues captures the major structure of the manifold and are therefore useful for visualisation. Haghverdi et al. [45] further derive a heuristic for selection of the kernel width σ in terms of the effective number of neighbours of each cell.

1.3.4.3 *Multidimensional scaling*

Multidimensional scaling (MDS) has previously been used for the visualisation of large genomic datasets [132] and more recently was used as the initial dimensionality reduction step for the pseudotime algorithm `SCORPIUS` [23]. It is motivated by the problem of trying

to place cities on points on a map if we are given the distances between them. Given an $N \times N$ distance matrix $\mathbf{D}$ MDS attempts to minimise the quantity

$$\text{Stress}(\mathbf{Z}) = \left(\sum_{n \neq n'=1}^{N} (d_{nn'} - ||\mathbf{z}_n - \mathbf{z}_{n'}||)^2 \right)^{\frac{1}{2}} \tag{9}$$

where $\mathbf{z}_n$ is the low dimensional embedding of sample n. The intuition is if n and n' are close we minimise the distance between $\mathbf{z}_n$ $\mathbf{z}_{n'}$ and if n and n' are far apart we need to maximise the distance between $\mathbf{z}_n$ and $\mathbf{z}_{n'}$ so that it is as close to $d_{nn'}$ as possible.

1.3.4.4 *Locally linear embedding*

Locally linear embedding (LLE) was used successfully by `SLICER` [138] as a nonlinear dimensionality step after highly-variable gene selection. LLE begins by defining an $N \times N$ weight matrix $\mathbf{W}$ where $W_{nn'}$ represents how useful sample n' is for reconstructing n. An optimal $\mathbf{W}$ is found by minimising

$$\sum_n ||\mathbf{y}_n - \sum_{n'} W_{nn'} \mathbf{y}_{n'}||^2 \tag{10}$$

where $\mathbf{W}$ is sparsely constrained so that each point is only reconstructed by its k nearest neighbours and so that the row sums of $\mathbf{W}$ are 1. The Q (reduced) dimensional reconstructions $\mathbf{z}_n,\ n = 1, \ldots, N$ are then found via minimising

$$C(\mathbf{Z}) = \sum_n ||\mathbf{z}_n - \sum_{n'} W_{nn'} \mathbf{z}_{n'}||^2 \tag{11}$$

where $\mathbf{W}$ is kept fixed in the second optimisation step. In other words, we seek a low dimensional embedding such that the points have approximately the same relationship to each other in the reduced space as in the full (G-dimensional) space. Minimisation of $C(\mathbf{Z})$ can subsequently be performed via a sparse eigenvalue problem.

1.3.4.5 *t-distributed stochastic neighbour embedding*

t-distributed stochastic neighbour embedding (t-SNE) [76] has become incredibly popular for the visualisation of single-cell RNA-seq data and as the initial dimensionality step for several pseudotime algorithms. Similarly to diffusion maps[18], it begins by defining a conditional transition matrix

$$P_{n'|n} = \frac{1}{Z(\mathbf{y}_{n'})} \exp\left(-\frac{||\mathbf{y}_n - \mathbf{y}_{n'}||^2}{2\sigma_n^2}\right) \tag{12}$$

which can be interpreted as the probability under a Gaussian likelihood of n choosing n' as its neighbour. This is then symmetrised to form $P_{n'n} = \frac{1}{2N}(P_{n'|n} + P_{n|n'})$.

It then defines similarities in the latent space as

$$Q_{nn'} = \frac{(1 + ||\mathbf{z}_n - \mathbf{z}_{n'}||^2)^{-1}}{\sum_{m \neq m'}(1 + ||\mathbf{z}_m - \mathbf{z}_{m'}||^2)^{-1}} \tag{13}$$

which is equivalent to measuring distances in the latent space with a Student-t distribution with one degree of freedom. Values of $\mathbf{z}_n$ are found by minimising the Kullback–Leibler (KL) divergence $\mathrm{KL}(\mathbf{P} \parallel \mathbf{Q}) = \sum_{n \neq n'} P_{nn'} \log \frac{P_{nn'}}{Q_{nn'}}$.

It is hard to overstate how popular t-SNE has been for visualising single-cell gene expression data. Examples include as the initial dimensionality reduction step in `SCUBA` [86] or for visualisation of branch structure of single-cell mass cytometry data [117]. Criticisms of t-SNE include the required specification of the kernel widths σ_n (which can be interpreted in terms of an effective number of nearest neighbours or *perplexity*) and the number of iterations of the (stochastic) gradient descent algorithm.

18 Note that t-SNE differs in defining a different kernel width for each data point.

1.3.5 *Gaussian process latent variable models*

Gaussian Process Latent Variable Models (GPLVM) are the subject of chapter 2 but are mentioned here for completeness. Technically GPLVM is a form of probabilistic manifold learning that learns an explicit map from the latent space to the observed space but may also be seen as a form of nonlinear factor analysis. In the factor analysis model of equation 34 typical estimation proceeds by marginalising over $\mathbf{z}_n$ to give a marginal likelihood $\mathbf{y}_n \sim (\boldsymbol{\mu}, \boldsymbol{\Lambda}\boldsymbol{\Lambda}^T + \boldsymbol{\Sigma})$ followed by direct optimisation. However, Lawrence [69] instead marginalised over the mapping $\boldsymbol{\Lambda}$ through a prior of the form $p(\boldsymbol{\Lambda}) = \prod_{q=1}^{Q} \mathcal{N}(\boldsymbol{\lambda}_q|\mathbf{0}, \alpha^{-1}\mathbf{I})$. This introduces a coupling between different samples $\mathbf{y}_n$. Let $\mathbf{Y}$ be the full $N \times G$ data matrix with row vectors $\mathbf{y}_n$ and column vectors $\mathbf{y}_g$ and $\mathbf{Z}$ the $N \times Q$ matrix of latent values with row vectors $\mathbf{z}_n$. The likelihood marginalised over the mapping is then

$$p(\mathbf{Y}) = \prod_{g=1}^{G} \mathcal{N}(\mathbf{y}_g|\mathbf{0}, \alpha^{-1}\mathbf{Z}\mathbf{Z}^T + \boldsymbol{\Sigma}) \tag{14}$$

where $\boldsymbol{\Sigma}$ is the diagonal noise covariance matrix as before.

Lawrence's key insight was that the term $\alpha^{-1}\mathbf{Z}\mathbf{Z}^T$ in the covariance matrix represents similarity between difference samples, since the covariance between samples n and n' is $\alpha^{-1}\mathbf{z}_n \cdot \mathbf{z}_{n'}$. Therefore, it can be replaced with any positive definite *kernel* $k(\mathbf{z}_n, \mathbf{z}_{n'})$ representing similarity between $\mathbf{z}_n$ and $\mathbf{z}_{n'}$. Popular examples include the squared exponential kernel

$$k_{\text{SQE}}(\mathbf{z}_n, \mathbf{z}_{n'}) = \sigma_f^2 \exp\left(-\frac{1}{2l^2}||\mathbf{z}_n - \mathbf{z}_{n'}||^2\right) \tag{15}$$

as used in [22] or the Matern family such as the $\text{Matern}_{3/2}$ kernel used in [112]:

$$k_{\text{Matern}_{3/2}}(\mathbf{z}_n, \mathbf{z}_{n'}) = \left(1 + \sqrt{3}|\mathbf{z}_n - \mathbf{z}_{n'}|\right) \exp\left(-\sqrt{3}|\mathbf{z}_n - \mathbf{z}_{n'}|\right). \tag{16}$$

GPLVM has been widely applied to single-cell expression data, including to single-cell qPCR data [14] and single-cell RNA-seq [22, 81]. Latent embeddings inferred using GPLVM are typically under-constrained leading to a number of studies that introduce "data-driven priors" to further constrain the model, such as incorporating the t-SNE cost function to preserve local structure in GPLVM [77].

1.4 PROBABILISTIC MODELLING OF GENOMIC TRAJECTORIES

The overall aim of this thesis is a comprehensive probabilistic treatment of the pseudotime problem. Chapter two exchanges the post-dimensionality-reduction cell ordering algorithms of many pseudotime methods for probabilistic curves using Gaussian Process Latent Variable Models and asks what the effects are of incorporating uncertainties into downstream modelling. It also introduces a differential-expression-over-pseudotime model. Chapter three introduces a generative model of pseudotime based on nonlinear factor analysis that replaces the dimensionality reduction and cell ordering steps with a single inference procedure. Chapter four considers Bayesian inference of bifurcations in single-cell data using a hierarchical mixture of factor analysers. Chapter five introduces a novel type of latent variable model that lets a secondary dataset to perturb the factor loadings in the first. This is applied to a diverse set of single-cell and bulk cancer studies and identifies novel interactions between phenotypic covariates and biological pathways. Finally, chapter six incorporates a discussion of some of the weaknesses of probabilistic models of pseudotimes and suggests future directions.

2

UNCERTAINTY IN SINGLE-CELL PSEUDOTIME

2.1 INTRODUCTION

Practically, current methods for pseudotime inference proceed via a multi-step process which we describe throughout using gene expression data as the focus of our discussion. First, gene selection and dimensionality reduction techniques are applied to compress the information held in the high-dimensional gene expression profiles to a small number of dimensions (typically two or three for simplicity of visualisation). The identification of an appropriate dimensionality reduction technique is a *subjective* choice and a number of methods have been adopted such as Principal and Independent Components Analysis (P/ICA) and highly non-linear techniques such as diffusion maps [45, 46] or stochastic neighbourhood embedding (SNE) [1, 52, 76]. This choice is guided by whether the dimensionality reduction procedure is able to identify a suitable low-dimensional embedding of the data that contains a relatively smooth trajectory that might plausibly correspond to the temporal process under investigation. Next, the pseudotime trajectory of the cells in this low-dimensional embedding is characterised. In Monocle [129] this is achieved by the construction of a minimum spanning tree (MST) joining all cells. The diameter of the MST provides the main trajectory along which pseudotime is measured. Related graph-based techniques (Wanderlust) have also been used to characterise temporal processes from single cell mass cytometry data [8]. In SCUBA [85] the trajectory itself is directly modelled using principal curves [48] and pseudotime is assigned to each cell by projecting its location in the low-dimensional embedding on to the principal curve. The estimated pseudotimes can then be used to order the cells and to assess differential expression of genes across pseudotime. Note that in the diffusion pseudotime framework

[46], all the diffusion components are used in the random-walk pseudotime model and there is no strict dimensionality reduction step. However, the derivation of the diffusion maps does lead to the compression of information into the first few diffusion components which is what enables successful visualisation [45].

A limitation of these approaches is that they provide only a single *point estimate* of pseudotimes concealing the full impact of variability and technical noise. As a consequence, the statistical uncertainty in the pseudotimes is not propagated to downstream analyses - such as differential expression of genes along pseudotime - precluding a thorough treatment of stability. Thus, the impact of this pseudotime uncertainty has not been explored and its implications are unknown as the methods applied typically do not possess a probabilistic interpretation. However, we can examine the stability of the pseudotime estimates by taking multiple random subsets of a dataset and re-estimating the pseudotimes for each subset. For example, we have found that the pseudotime assigned to the same cell can vary considerably across random subsets in Monocle (section 2.4.5).

In order to address pseudotime uncertainty in a formal and coherent framework, probabilistic approaches using Gaussian Process Latent Variable Models (GPLVM) have been used recently as non-parametric models of pseudotime trajectories [21, 81, 112]. These provide an explicit model of pseudotimes as latent embedded one-dimensional variables. These models can be fitted within a Bayesian statistical framework using priors on the pseudotimes [112], deterministic optimisation methods for approximate inference [81] or Markov Chain Monte Carlo (MCMC) simulations allowing full posterior uncertainty in the pseudotimes to be determined [21].

In this chapter we adopt this framework to assess the impact of pseudotime uncertainty on downstream differential analyses. We first introduce a novel model for differential expression along pseudotime that may be more suited to single-cell data. We then develop

a model for uncertainty in pseudotime based on GPLVM. We go on to show that pseudotime uncertainty can be non-negligible and when propagated to downstream analysis may considerably inflate false discovery rates. We demonstrate that there exists a limit to the degree of recoverable temporal resolution, due to intrinsic variability in the data, with which we can make statements such as "this cell precedes another". Overall, we outline a modelling and analytical strategy to produce more stable pseudotime based differential expression analysis.

2.2 A SWITCH-LIKE MODEL FOR PSEUDOTIME DIFFERENTIAL EXPRESSION

2.2.1 *Existing pseudotime differential expression*

Once a pseudotime has been assigned to each cell it is possible to identify genes that exhibit a strong pseudotemporal dependence through differential expression testing. An approach first introduced in Trapnell et al. [129] was to regress gene expression on pseudotime using cubic B-spline basis functions to model smooth expression profiles.

Single-cell RNA-seq data is also known to exhibit a large number of *dropouts* which manifest as zero-inflated the data (see e.g. [59]). This is due to a failure to reverse-transcript low abundance mRNAs, resulting in zero counts. To account for this, Trapnell et al. introduces a *Tobit* likelihood that models the observed expression y in terms of the true expression y^* and a limit of detection λ as

$$y = \begin{cases} y^*, & \text{if } y^* > \lambda, \\ \lambda, & \text{otherwise.} \end{cases} \tag{17}$$

which essentially deals with zero-inflation by enforcing any expression less than the detection limit to be latent.

However, the flexible nonparametric nature of the B-spline basis functions may lead to overfitting, biologically unrealistic shapes, and is also difficult to interpret without further post-hoc analysis. To our knowledge no other differential-expression-along-pseudotime models have been proposed.

2.2.2 *Statistical model*

As a solution to these issues we present `switchde`, a statistical model and accompanying `R` package for identifying switch-like differential expression analysis along single-cell trajectories. We model sigmoidal expression changes along pseudotime that provides interpretable parameter estimates corresponding to gene regulation strength and timing along with hypothesis testing for differential expression.

Let y_{ng} denote the $\log_2$ gene expression of gene g in cell n at pseudotime t_n then

$$y_{ng}(t_n) \sim \mathcal{N}(\mu_g(t_n),\ \sigma_g^2) \tag{18}$$

where

$$\mu_g(t_n) = \begin{cases} \mu_g^{(0)}, & \text{if gene } g \text{ not differentially expressed,} \\ \dfrac{2\mu_g^{(0)}}{1+\exp\left(-k_g(t_n - t_g^{(0)})\right)}, & \text{if gene } g \text{ differentially expressed.} \end{cases} \tag{19}$$

Under this model the parameter k_g can be thought of as an activation 'strength' relating to how quickly a gene switches on or off along pseudotime, while $t_g^{(0)}$ represents the pseudotime at which the gene switches on or off (figure 5A).

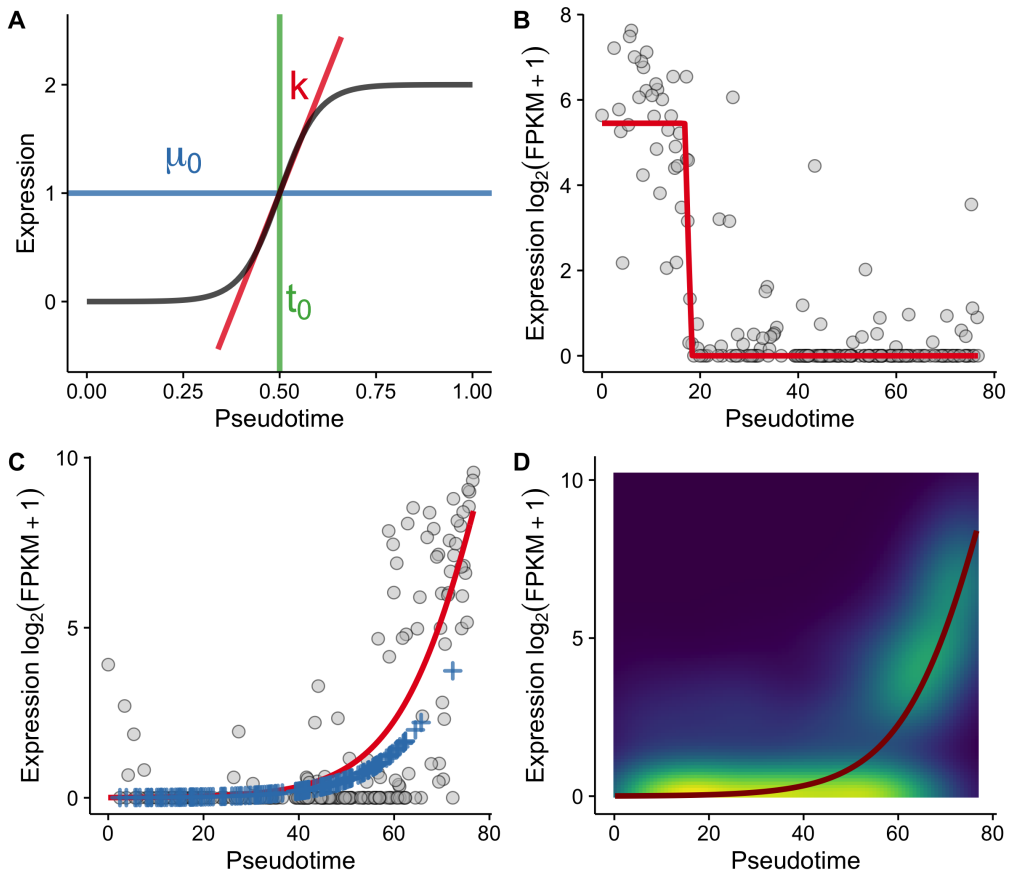


Figure 5: Sigmoidal expression across pseudotime. **A** The sigmoid curve as a model of gene expression along single-cell trajectories, parametrised by the average peak expression μ_0, the activation strength k and the activation time t_0. **B** An example using the *NDC80* gene from the Trapnell dataset [129], which had the lowest p-value of all genes tested. Gene expression measurements are shown as the grey points with the maximum likelihood sigmoid fit denoted by the dark line. The maximum likelihood parameter estimates were $\mu_g^{(0)} = 2.73$, $k_g = -8.71$ and $t_g^{(0)} = 17.61$. **C** Zero-inflated differential expression for the transcription factor *MYOG*. Solid line shows the MLE sigmoidal mean while crosses show imputed gene expression measured as zeroes. **D** Posterior predictive density for the zero-inflated model with the solid line denoting MLE sigmoidal mean.

We fit the model using gradient-based L-BFGS-B optimisation to find maximum likelihood estimates (MLEs) of the parameters (appendix A.1). By setting $k_g = 0$ we identify a nested constant-expression model where $y_{ng} \sim \mathcal{N}(\mu_g^{(0)}, \sigma_g^2)$ and so can perform a likelihood ratio test for differential expression, where twice the difference in the log-

likelihood MLE between the constant and sigmoidal models asymptotically follows a χ^2 distribution with two degrees of freedom.

2.2.2.1 *Modelling zero-inflation*

To account for the aforementioned zero-inflation in the data we further propose a model that incorporates dropouts in a similar style to [102]:

$$\begin{aligned} \mu_g(t_n, \theta_g) &= \frac{2\mu_g^{(0)}}{1 + \exp\left(-k_g(t_n - t_g^{(0)})\right)} \\ x_{ng} &\sim \mathcal{N}(\mu_g(t_n, \theta_g), \sigma_g^2) \\ h_{ng}|x_{ng} &\sim \text{Bernoulli}(\exp(-\lambda_g x_{ng}^2)) \\ y_{ng} &= \begin{cases} x_{ng}, & \text{if } h_{ng} = 0 \\ 0, & \text{if } h_{ng} = 1 \end{cases} \end{aligned} \tag{20}$$

which can be fitted with an expectation-maximisation algorithm (appendix A.2). One advantage of such a model is it allows us to effectively "impute" zero or dropout counts as can be seen in figure 5C.

2.2.3 *Properties*

2.2.3.1 *Expression profiles of marker genes*

In [129] the genes *CDK1* and *ID1* are identified as markers for the myoblast differentiation trajectory. Zero-inflated `switchde` fits for these two genes are shown in figures 6A&B respectively along with the imputed dropout expression (blue crosses). Using the zero-inflated likelihood ratio test these two genes have p-values of 2.371947×10^{-73} and 1.550464×10^{-8}.

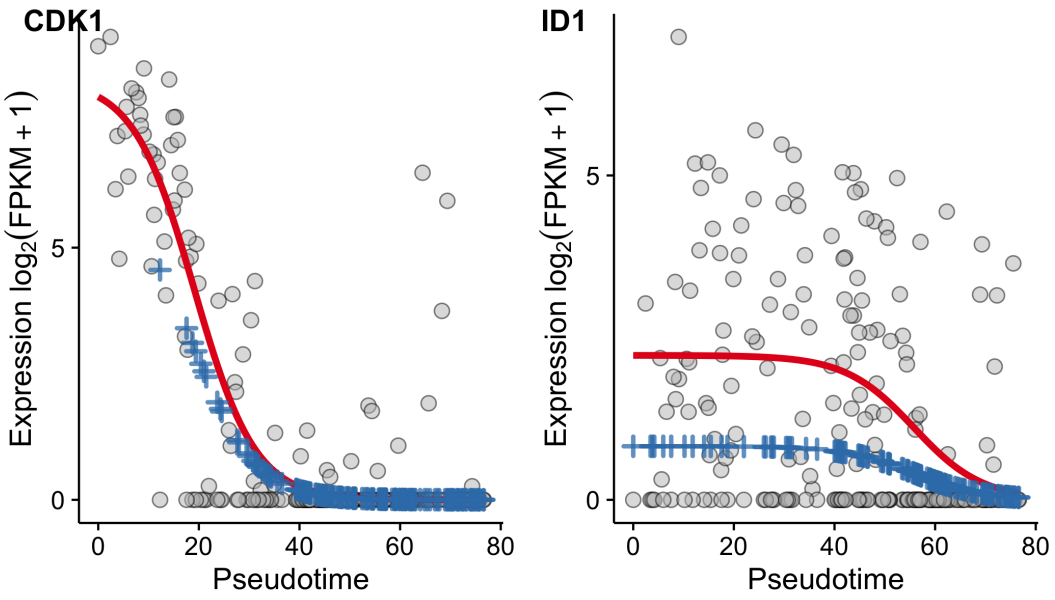


Figure 6: Expression profiles of *CDK1* (left) and *ID1* (right) along with MLE fits for the zero-inflated sigmoidal model (solid red line) with imputed dropout expression (blue crosses).

2.2.3.2 *Examples of large and small p-values*

In figure 7 we provide example (non-zero-inflated) fits for two genes with drastically different p-values: *NUSAP1* is shown in figure 7A with a p-value of 5.335782×10^{-69} and *GCLC* with a p-value of 0.9651487. It is clear from the expression plots and MLE sigmoidal fits that the gene with the very low p-value clearly follows the sigmoidal switch-like trend while the gene with the high p-value is well explained by a null (constant expression) model.

2.2.3.3 *Tracing activation times along pseudotime*

One advantage of our model is that one can identify genes up or down regulated at a particular part of the trajectory. To demonstrate this we found the genes with the closest t_0 values to 20%, 32%, 44%, 56%, 68% and 80% of the way through the trajectory, based on all genes significant at 1% FDR.

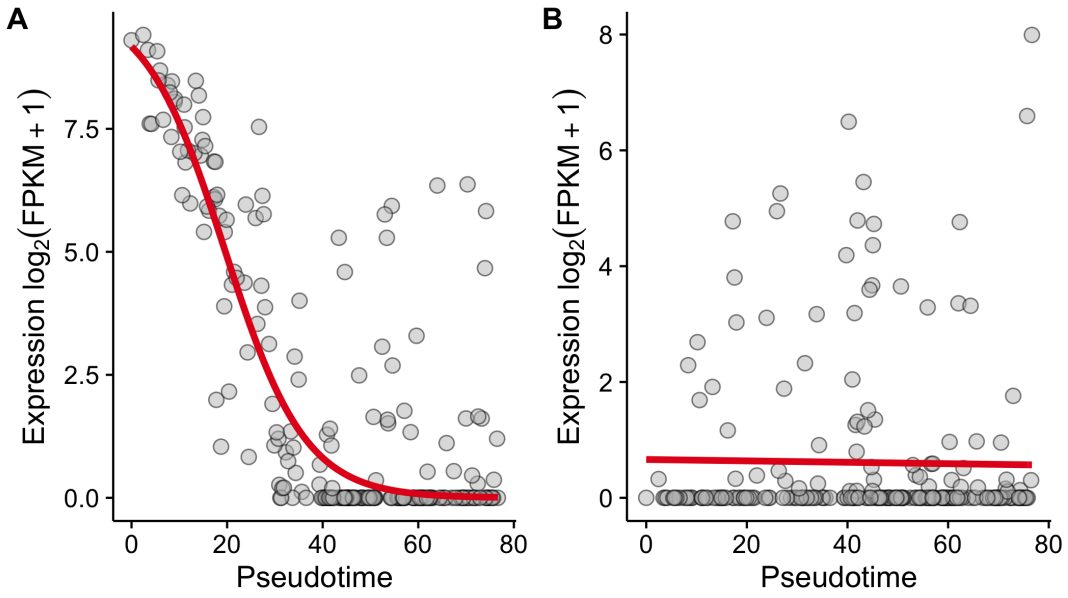


Figure 7: **A** Expression of *NUSAP1* with a p-value of 5.335782×10^{-69} and **B** expression of *GCLC* with a p-value of 0.9651487.

The results can be seen in figure 8, showing a clear 'cascade' of successive gene expression (in)-activations along the trajectory. This also sets the groundwork for identifying temporal gene networks along pseudotime as one can tell whether a given gene is regulated before another.

2.2.3.4 *Comparison of zero-inflated and standard models*

We sought to demonstrate the differences in parameter estimation when considering zero-inflation or otherwise by subsampling 200 genes and fitting both models for each. Figure 9A shows the comparisons of μ_0, demonstrating the zero-inflated estimate is typically higher which agrees with intuition. Figure 9B demonstrates that estimates of k are well calibrated between models. Figure 9C shows the comparison of t_0 estimates which is most dissimilar, with a spike in the non-zero-inflated model corresponding to when t_0 values barely deviate from their initial estimates. Finally, figure 9D compares the p-value estimates, which shows general concordance with no significant biases for either model.

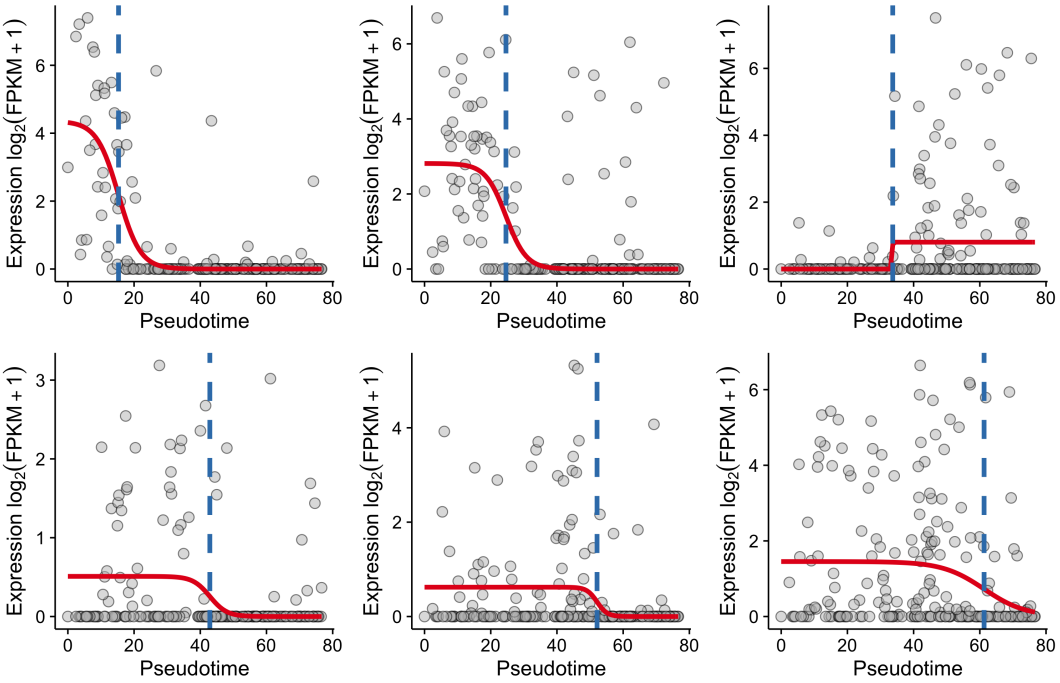


Figure 8: A 'cascade' of gene regulation along pseudotime. Each red curve corresponds to the MLE sigmoidal fit while the vertical blue dashed line corresponds to the MLE of t_0.

2.3 STATISTICAL MODEL FOR PROBABILISTIC PSEUDOTIME

We now turn to a probabilistic model of the pseudotimes themselves. As mentioned in the introduction, most pseudotime methods to date proceed by a two step procedure involving dimensionality reduction followed by inference of a one dimensional trajectory using techniques such as minimum spanning trees [57, 129] or principal curves [20, 23, 85]. This secondary step is analogous to curve fitting in the reduced-dimension space. While it is possible to create fully generative probabilistic models of pseudotime (ideas we explore in chapters 3 & 4), such models introduce additional assumptions and constraints. Therefore, in order to assess the uncertainty inherent in existing pseudotime inference algorithms we replace the second "curve-fitting" step with a model of probabilistic curves, in particular GPLVM.

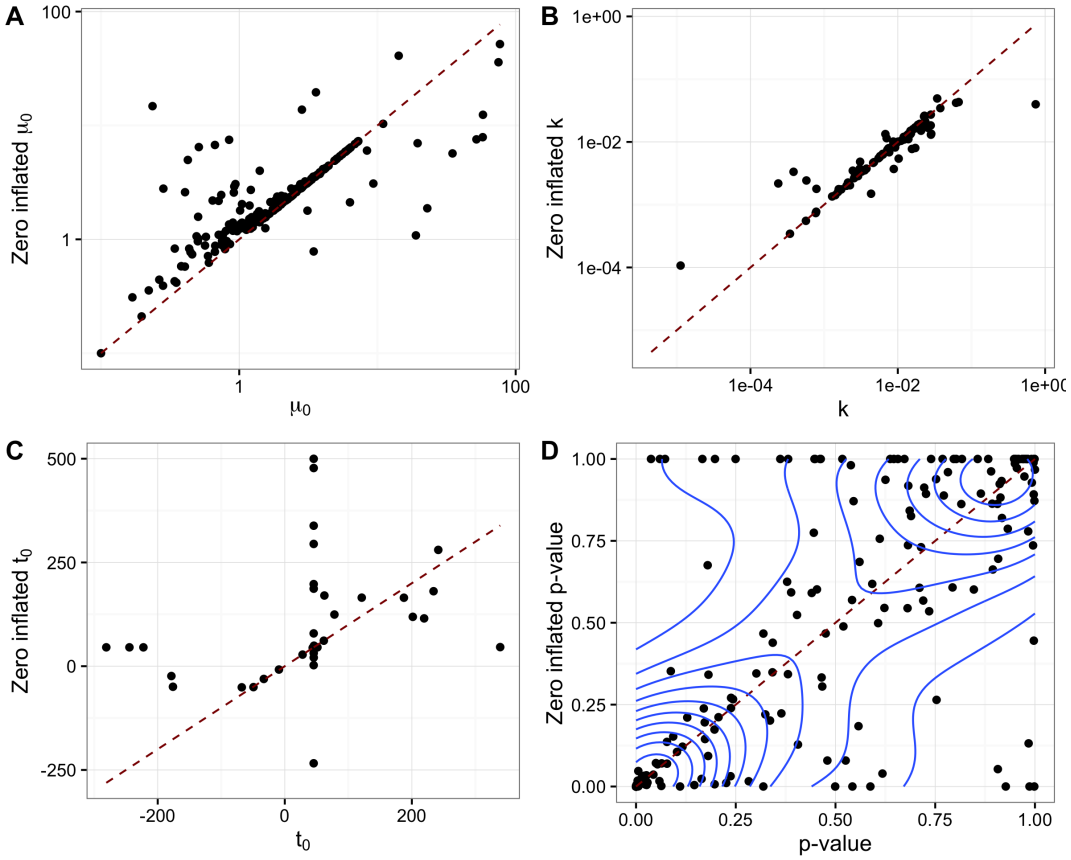


Figure 9: Comparison of MLE parameter estimates for zero-inflated and standard models, for **A** comparison of μ_0, **B** comparison of k, **C** comparison of t_0 and **D** comparison of p-values.

2.3.1 *Gaussian Processes and Gaussian Process Latent Variable Models*

A Gaussian Process (GP) is formally defined as *a collection of random variables, any finite number of which have a joint Gaussian distribution* [110]. They are completely characterised by their mean and covariance functions $m(\mathbf{x})$ and $k(\mathbf{x}, \mathbf{x}')$, typically denoted as

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \tag{21}$$

where the mean function $m(\mathbf{x})$ is typically taken to be zero.

In "real-world" applications we rarely observe samples from f itself but from some noisy realisation

$$y = f(\mathbf{x}) + \epsilon \tag{22}$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2)$. In this case the covariance function of the observed samples becomes

$$\text{cov}(y_i, y_j) = \text{cov}(f(\mathbf{x}_i) + \epsilon, f(\mathbf{x}_j) + \epsilon) = k(\mathbf{x}_i, \mathbf{x}_j) + \delta_{ij}\sigma^2. \tag{23}$$

Many covariance or *kernel* functions have been studied in the context of GPs, though by far the most commonly used is the *squared exponential kernel* given by

$$k_{\text{SE}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp(-\frac{1}{2l^2}|\mathbf{x} - \mathbf{x}'|^2) \tag{24}$$

where σ_f^2 is the *signal variance* and l the *length scale* [110]. An interpretation of the squared exponential kernel is that we maximise the covariance between observations when $\mathbf{x}$ is close to $\mathbf{x}'$ and minimise it when they are far apart. Variants of this kernel exist, such as that incorporating an automatic relevance determination (ARD) structure by allowing different length scales for each input dimension.

We can think of the GPs described so far as analogous to non-parametric regression where we observe some (possibly noisy) one-dimensional output y at a collection of Q-dimensional input points $\mathbf{x}_1, \ldots, \mathbf{x}_N$. However, provided the output dimensionality $D > 1$ and $Q < D$, GPs can also be used to infer the input points in a latent variable formulation termed Gaussian Process Latent Variable Models (GPLVM) [70].

To understand this, we begin with a factor analysis model. Let $\mathbf{Y}$ be an $N \times D$ matrix of observations, $\boldsymbol{W}$ be a $D \times Q$ weight matrix and $\mathbf{X}$ be a $N \times Q$ latent matrix. Let $\mathbf{y}_n$

by the n^{th} row vector of $\mathbf{Y}$, $\mathbf{y}_d$ by the d^{th} column vector, and $\mathbf{x}_n$ be the n^{th} row vector of $\mathbf{X}$. The factor analysis model generates a single observation $\mathbf{y}_n$ via

$$\mathbf{y}_n = \boldsymbol{W}\mathbf{x}_n + \boldsymbol{\epsilon} \tag{25}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}\boldsymbol{\sigma}^2)$ and $\boldsymbol{\sigma}^2$ is a D-dimensional vector of output variances. Introducing a prior over the weights of the form $p(\boldsymbol{W}) = \prod_{d=1}^{D} \mathcal{N}(\mathbf{w}_d|\mathbf{0}, \boldsymbol{I})$ induces a marginal distribution of the form

$$p(\mathbf{y}_d|\mathbf{X}, \boldsymbol{\sigma}^2) = \mathcal{N}(\mathbf{0}, \mathbf{X}\mathbf{X}^T + \boldsymbol{\sigma}^2\boldsymbol{I}). \tag{26}$$

The key insight from [70] is that the marginal distribution for a single output dimension is equal to N draws from a GP with kernel $\mathbf{K} = \mathbf{X}\mathbf{X}^T + \boldsymbol{\sigma}^2\boldsymbol{I}$, the sum of an inner product kernel and an independent noise process. We can therefore replace the inner product kernel with any positive-definite kernel that represents similarity such as the squared exponential kernel from before, and infer the latent variables through techniques such as maximum-likelihood [70] or Variational Bayes [125].

In the case of $D = 2$ and $Q = 1$, we have two output dimensions controlled by a single degree of freedom, thus representing a probabilistic curve where the latent variable is a probabilistic representation of pseudotime and we have defined a stochastic mapping between the pseudotime and the reduced dimensionality representation. If we apply Bayesian inference we can then approximate the marginal posterior distribution $p(\mathbf{X}|\mathbf{Y})$ allowing us to fully quantify uncertainty in the trajectory inference part of most pseudotime methods.

2.3.2 *Probabilistic pseudotime inference using Gaussian Process Latent Variable Models*

In the case of single-cell pseudotime the latent space is one dimensional with values representing some notion of cellular progression. The observed data is the $N \times G$ expression matrix $\mathbf{Y}$ for N cells and G genes. However, G is typically high-dimensional with 10^4 genes expressed per cell not unusual. We therefore perform an initial dimensionality reduction step to an $N \times P$ matrix $\mathbf{X}$, in keeping with most pseudotime algorithms. In practice we alternate between using PCA and laplacian eigenmaps, but any of the manifold learning algorithms referenced in 1.3.4 may equally be used.

The goal of Bayesian pseudotime inference is then to infer the posterior distribution $p(\mathbf{t}|\mathbf{X})$ where $\mathbf{t}$ is the N-length pseudotime vector $\mathbf{t} = [t_1, \ldots, t_n]$. If $\mathbf{\Theta}$ is the complete set of model parameters (other than $\mathbf{t}$) then Bayes rule gives

$$p(\mathbf{t}|\mathbf{X}) = \frac{\int_{\Omega_{\mathbf{\Theta}}} d\mathbf{\Theta} p(\mathbf{X}|\mathbf{t}, \mathbf{\Theta}) p(\mathbf{t}|\mathbf{\Theta}) p(\mathbf{\Theta})}{\int_{\Omega_{\mathbf{t}}} \int_{\Sigma_{\mathbf{\Theta}}} d\mathbf{t} d\mathbf{\Theta} p(\mathbf{X}|\mathbf{t}, \mathbf{\Theta}) p(\mathbf{t}|\mathbf{\Theta}) p(\mathbf{\Theta})} \tag{27}$$

where $\mathbf{\Omega_\Theta}$ and $\mathbf{\Omega_t}$ are the sample spaces of $\mathbf{\Theta}$ and $\mathbf{t}$ respectively. The the integrals in equation 2.3.2 are intractable so we turn to approximation methods, in particular Markov Chain Monte Carlo (MCMC) to obtain a numerical approximation from the posterior by drawing samples from the posterior distribution (see section 2.3.3 for further details). In the case of single-cell pseudotime, each sample corresponds to one possible trajectory *and* ordering for the cells with the set of samples providing an approximate distribution of pseudotimes. For interpretability the latent pseudotime values are constrained between measured 0 and 1 where a value of 0 corresponds to one end state of the temporal process and a value 1 to the other.

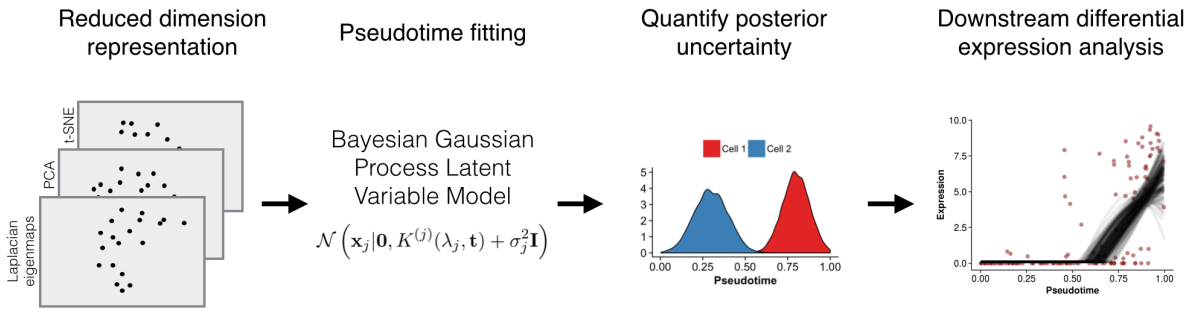


Figure 10: Workflow for fitting Bayesian Gaussian Process Latent Variable Model pseudotime models. Reduced-dimension representations of the gene expression data (from Laplacian eigenmaps, PCA and/or t-SNE) are created. The pseudotime can be fitted using one or more low dimensional representations of the data. Posterior samples of pseudotimes are drawn from a Bayesian GPLVM and these are used to obtain alternative pseudotime estimates. Downstream differential analyses can be performed on the posterior samples to characterise the robustness with respect to variation in pseudotime estimates.

The hierarchical model specification for probabilistic pseudotime using a Gaussian Process Latent Variable model is then

$$
\begin{aligned}
\gamma &\sim \text{Gamma}(\gamma_\alpha, \gamma_\beta), \\
\lambda_p &\sim \text{Exp}(\gamma), \;\; p = 1, \ldots, P, \\
\sigma_p^2 &\sim \text{InvGamma}(\alpha, \beta), \;\; p = 1, \ldots, P, \\
t_n &\sim \text{TruncNormal}_{[0,1)}(\mu_t, \sigma_t^2), \;\; n = 1, \ldots, N, \\
\mathbf{\Sigma} &= \text{diag}(\sigma_1^2, \ldots, \sigma_P^2) \\
K^{(p)}(t, t') &= \exp(-\lambda_p(t - t')^2), \;\; p = 1, \ldots, P, \\
\mu_p &\sim \text{GP}(0, K^{(p)}), \;\; p = 1, \ldots, P, \\
\mathbf{x}_n &\sim \mathcal{N}(\boldsymbol{\mu}(t_n), \mathbf{\Sigma}), \;\; i = 1, \ldots, N.
\end{aligned}
\tag{28}
$$

where $\mathbf{x}_n$ is the P-dimensional input of cell n (of N) found by performing dimensionality reduction on the entire gene set (for our experiments $P = 2$ following previous

studies). The observed data is distributed according to a multivariate normal distribution with mean function $\boldsymbol{\mu}$ and a diagonal noise covariance matrix $\boldsymbol{\Sigma}$. The prior over the mean function $\boldsymbol{\mu}$ in each dimension is given by a Gaussian Process with zero mean and covariance function K given by a standard double exponential kernel. The latent pseudotimes $t_1, \ldots, t_N$ are drawn from a truncated Normal distribution on the range $[0, 1)$. Under this model $|\boldsymbol{\lambda}|$ can be thought of as the arc-length of the pseudotime trajectories, so applying larger levels of shrinkage to it will result in smoother trajectories passing through the point space. This shrinkage is ultimately controlled by the gamma hyperprior on γ, whose mean and variance are given by $\frac{\gamma_\alpha}{\gamma_\beta}$ and $\frac{\gamma_\alpha}{\gamma_\beta^2}$ respectively. Therefore, adjusting these parameters allows curves to match prior smoothness expectations provided by plotting marker genes. The hyperparameters γ_α, γ_β, α, β, μ_t and σ_t^2 are fixed and values for specific experiments for given in appendix B.1.

The overall workflow can be seen in figure 10. We begin by reducing the dimensionality of the data using methods such as principal component analysis or laplacian eigenmaps. Next, the statistical model is fit to infer numerical approximations to the posterior distributions of the pseudotimes. This allows us to quantify the uncertainty in the pseudotime of each cell. Finally, we can propagate this uncertainty to downstream analysis such as differential expression, where each draw from the MCMC simulation is passed through the analysis to give us a distribution over the results.

2.3.3 *Inference*

The model defined in equation 2.3.2 is not conditionally conjugate, meaning we cannot use Gibbs sampling for inference. An alternative would be to use a Metropolis-Hastings algorithm[1] that can perform Bayesian posterior inference on any model for which the

1 Our initial implementation used exactly this.

likelihood can be evaluated. However, Metropolis-Hastings requires manual tuning of proposal distributions in order to explore the posterior space efficiently and can exhibit poor mixing in high-dimensional spaces.

In order to sidestep these issues we turn to the probabilistic programming language (PPL) Stan [39] that performs efficient inference on arbitrary (non-conjugate) Bayesian models. PPLs have become increasingly popular in the past few years with examples such as Stan, PyMC3 [116], Infer.net [136] and Edward [127]. They describe probabilistic models and perform inference in a (semi-) automated manner, taking much of the work out of statistical modelling and to some extent freeing models from assumptions that lead to easier inference.

We use Stan for the majority of the statistical inference in both chapters 2 & 3 so describe briefly how it works. Stan uses Hamiltonian Monte Carlo (HMC, see e.g. [95]), a form of monte carlo sampling that leverages gradient information to quickly traverse through "flat" regions of posterior probability to more effectively explore the space.

HAMILTONIAN DYNAMICS Hamiltonian Dynamics describe the evolution of a system defined by a position vector $\boldsymbol{\theta}$ and momentum vector $\mathbf{p}$. The dynamics of the system are governed by the Hamiltonian $H(\boldsymbol{\theta}, \mathbf{p}) = U(\boldsymbol{\theta}) + K(\mathbf{p})$ - the sum of a potential energy term $U(\boldsymbol{\theta})$ and kinetic energy term $K(\mathbf{p})$. The system then evolves according to Hamilton's equations

$$\begin{aligned} \frac{d\theta_i}{dt} &= \frac{\partial H}{\partial p_i} \\ \frac{dp_i}{dt} &= -\frac{\partial H}{\partial \theta_i} \end{aligned} \tag{29}$$

The kinetic energy is often defined as $K(\mathbf{p}) = \mathbf{p}^T \mathbf{M}^{-1} \mathbf{p}/2$ where M is a symmetric, positive definite mass matrix, which is typically diagonal. Note that this corresponds to the log probability function of a multivariate normal with mean $\mathbf{0}$ and covariance matrix

$\mathbf{M}$. Crucially, Hamiltonian Dynamics may be simulated on a computer by discretising the equations of motion, most typically using the leapfrog method (see [95]).

HAMILTONIAN MONTE CARLO The trick behind Hamiltonian Monte Carlo (HMC) is to choose $H(\boldsymbol{\theta}, \mathbf{p})$ such that the hamiltonian dynamics allow us to effectively explore the target distribution. To do this HMC borrows the idea of the canonical distribution from statistical physics, which for some energy function $E(\theta)$ is defined as[2] $p(\theta) = \frac{1}{Z}\exp(-E(\theta))$ for some normalising constant Z. Therefore, if the energy function is the Hamiltonian $H(\boldsymbol{\theta}, \mathbf{p}) = U(\boldsymbol{\theta}) + K(\mathbf{p})$ then the probability density factorises as

$$p(\boldsymbol{\theta}, \mathbf{p}) \propto \exp(-U(\boldsymbol{\theta}))\exp(-K(\mathbf{p})) \tag{30}$$

so $\boldsymbol{\theta}$ and $\mathbf{p}$ are independent. To target the Bayesian posterior density $p(\boldsymbol{\theta}|\mathbf{X})$ the potential energy U is set so that

$$U(\boldsymbol{\theta}) = -\log\left[p(\mathbf{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})\right]. \tag{31}$$

The HMC algorithm begins by initialising the parameters $\boldsymbol{\theta}$ and the momenta $\mathbf{p}$ (by drawing from the prior $\mathcal{N}(\mathbf{0}, \mathbf{M})$). A single HMC iteration involves simulating Hamiltonian dynamics on $(\boldsymbol{\theta}, \mathbf{p})$ according to equation 29 for L steps with a step-size of ϵ, ending at some position $\boldsymbol{\theta}^*$, at which point the momentum is negated giving a proposed state $(\boldsymbol{\theta}^*, \mathbf{p}^*)$. The proposed state is then accepted with probability

$$\min\left[1, \exp(-H(\boldsymbol{\theta}*, \mathbf{p}*) + H(\boldsymbol{\theta}, \mathbf{p}))\right]. \tag{32}$$

For each subsequent iteration the momentum variables are re-sampled from the prior.

2 We set the temperature that typically appears in the exponent to $T = 1$.

NO-U-TURN SAMPLER One issue with HMC is the need to specify the number of steps L and the step-size ϵ to which efficient inference is often highly sensitive. Stan is able to automatically determine both L and ϵ using what's known as the No-U-Turn Sampler (NUTS, [53]). The basic idea is to run HMC for enough iterations until the proposed location starts to move back towards itself - in other words, until the dot product of the displacement with the current momentum turns negative. The step size ϵ is then set to ensure the average acceptance probability is close to the optimal value.

2.4 RESULTS

2.4.1 *Sources of uncertainty in pseudotime inference*

We applied our probabilistic pseudotime inference to three published single-cell RNA-seq datasets of differentiating cells: myoblasts in Trapnell et al. (2014) [129], hippocampal quiescent neural stem cells in Shin et al. (2015) [120] and sensory epithelia from the inner ear in Burns et al. (2015) [15]. For the Trapnell and Shin datasets we used Laplacian Eigenmaps [6] for dimensionality reduction prior to pseudotime inference, while for the Burns dataset we used the PCA representation of the cells from the original publication. These particular choices of reduced dimensionality representations gave visually plausible trajectory paths in two dimensions.

An implicit assumption in pseudotime estimation is that proximity in pseudotime should reflect proximity in the observation or data space. That is, two cells with similar pseudotime assignments should have similar gene expression profiles but, in practice, cell-to-cell variability and technical noise means that the location of the cells in the observation space will be variable even if they truly do have the same pseudotime. We plotted posterior mean pseudotime trajectories for the three datasets learned using the

GPLVM in figure 11A-C and the posterior predictive data distribution $p(\mathbf{X}^*|\mathbf{X})$. The posterior predictive data distribution gives an indication of where *future* data points might occur given the existing data. Notice that for all three data sets, this distribution can be quite diffuse due to the cell-to-cell expression variability manifesting as a spread of data points around the mean trajectory.

The GPLVM applied assumes a homoscedastic noise distribution which is uniform along the pseudotime trajectory. However, it is clear that the variability of the data points can change along the trajectory and a heteroscedastic (non-uniform) noise model may be more appropriate in certain scenarios. Unfortunately, whilst models of heteroscedastic noise processes can be applied [71], these typically severely complicate the statistical inference and require a model of how the variability changes over pseudotime which is likely to be unknown. The important point here is that the posterior probabilities are always calculated with respect to a given model. The misspecification inherent to this model may lead to reduced posterior variance estimates and poorly performing posterior predictive distributions.

Returning to the intrinsic cell-to-cell variability, we next considered the conditional posterior predictive data distributions $p(\mathbf{X}^*|t^*, \mathbf{X})$ which are shown in figure 11D-F. These distributions show the possible distribution of future data points given the existing data *and* a theoretical pseudotime t^* and, in this example, we condition on pseudotimes $t^* = 0.5$ and $t^* = 0.7$. Although the two pseudotimes differ by a magnitude of 0.2, the conditional predictive distributions are very close or overlapping. This means that cells with pseudotimes of 0.5 or 0.7 could have given rise to data point occupying these overlapping regions. This variability is what ultimately limits the temporal resolution that can be obtained.

It is important to note that the posterior mean trajectories correspond to certain *a priori* or subjective smoothness assumptions (specified as hyperparameters in the

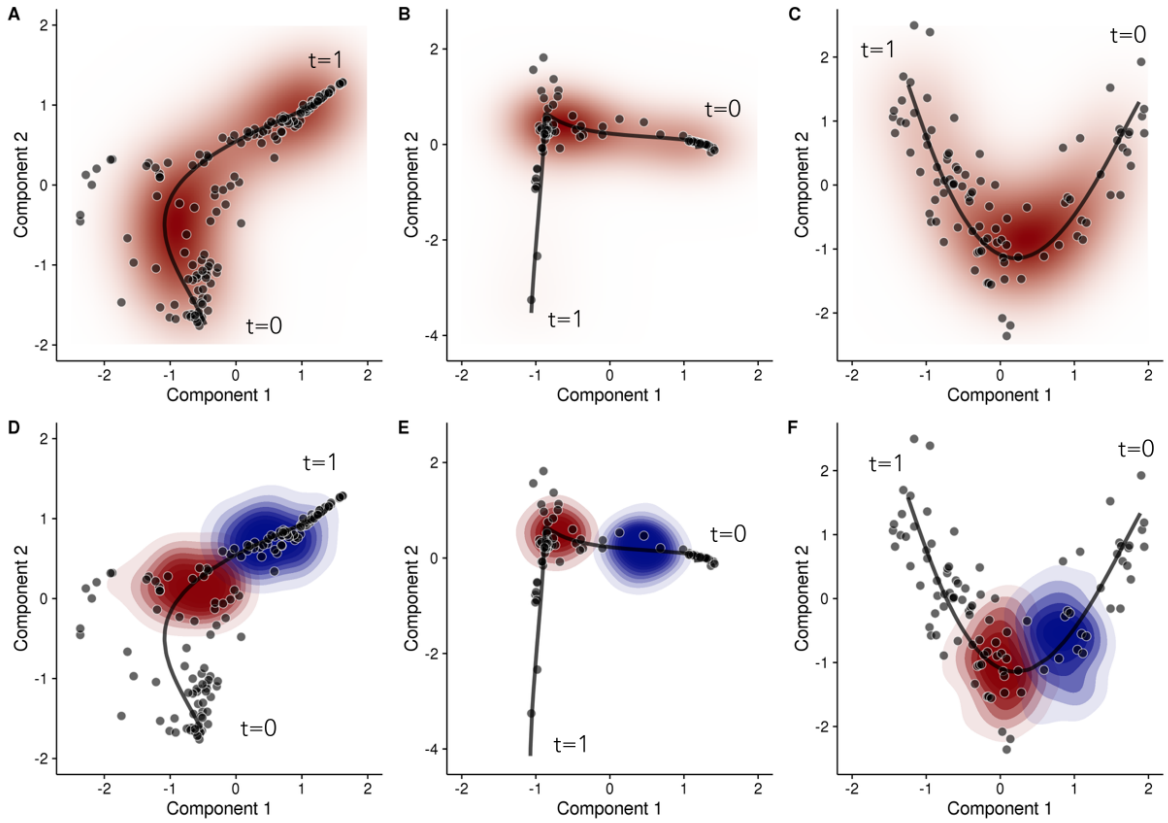


Figure 11: Posterior pseudotime trajectories for three single-cell RNA-seq datasets. Posterior pseudotime trajectories shown in a two-dimensional reduced representation space for (left) a Laplacian eigenmaps representation of Trapnell et al. (2014) [129], (centre) Laplacian eigenmaps representation of Burns et al. (2015) [15] and (right) PCA representation of Shin et al. (2015) [120]. Each point represents a cell and the black line represents the mean pseudotime trajectory. Plots **A-C** shows the overall posterior predictive data density (red) whilst **D-F** shows the conditional posterior predictive data density for $t = 0.5$ (red) and $t = 0.7$ (blue).

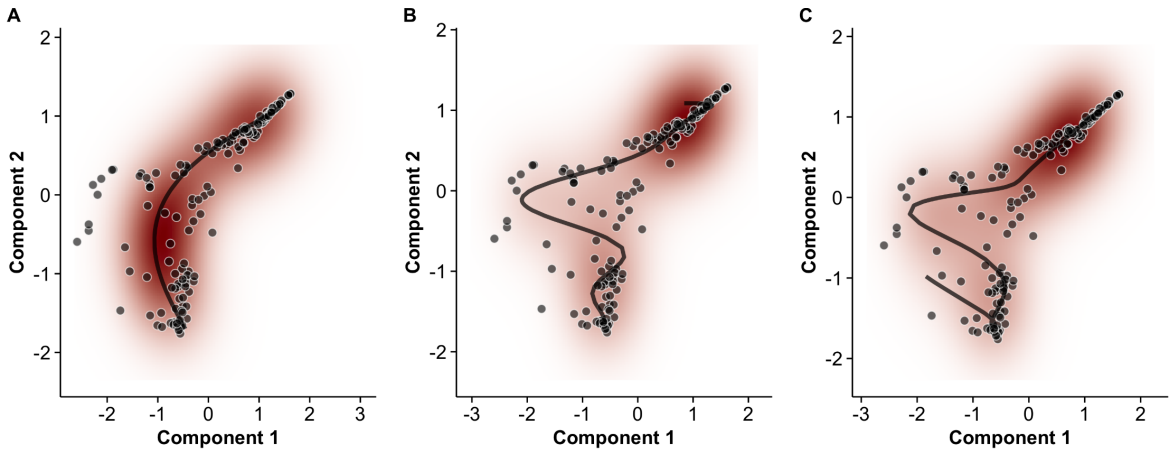


Figure 12: Effect of prior expectations on pseudotime trajectories. The prior probability distribution (defined in terms of hyperparameters $(\gamma_\alpha, \gamma_\beta)$ in our model) on the expected smoothness of pseudotime trajectories can fundamentally change the inferred progression path. Examples shown using the data of Trapnell et al. (2014) [129]. Red - shows the density of the posterior predictive data distribution. Black - shows the mean pseudotime trajectory. Shrinkage hyperparameters $(\gamma_\alpha, \gamma_\beta)$ of (30, 5), (5,1) and (3,1) were used for **A**, **B** and **C** respectively.

model specification) which dictate the curvature properties of the trajectory. Figure 12 shows three alternative posterior mean pseudotime trajectories for the Trapnell data based on different hyperparameters settings for the GPLVM. In a truly unsupervised scenario all three paths could be plausible as we would have little information to inform us about the true shape of the trajectory. This would become an additional source of uncertainty in the pseudotime estimates. However, we favoured hyperparameter settings that gave rise to well-defined (unimodal) posterior distributions that resulted in multiple independent Markov Chain Monte Carlo runs converging to the same mean trajectory rather than settings that give rise to a "lumpy" posterior distribution with many local modes corresponding to different interpretations of the data.

We next examined the posterior distributions in pseudotime assignment for four cells from the Trapnell dataset in figure 13A. Uncertainty in the estimate of pseudotime is assessed using the highest probability density (HPD) credible interval (CI), the Bayesian equivalent of the confidence interval. The 95% pseudotime CI typically covers around

one quarter of the trajectory, suggesting that pseudotemporal orderings of single-cells can potentially only resolve a cell's place within a trajectory to a coarse estimate (e.g. 'beginning', 'middle' or 'end') and do not necessarily dramatically increase the temporal resolution of the data. One immediate consequence of this is that it is unlikely that we can make definite statements such as whether one cell comes exactly before or after another. This is illustrated in figures 13B-D which displays the estimated pseudotime uncertainty for all three datasets. In all the datasets, the general progression is apparent, but the precise ordering of the cells has a non-trivial degree of ambiguity.

It is worth noting that such an analysis examines the overlap in the posterior marginal densities and may overstate the uncertainty in the orderings between two cells. If the posterior densities of the pseudotimes of two cells t_A and t_B are correlated then visually there may be significant overlap in the marginal densities but under the joint density the overlap could be quite small. This can be readily assessed by iterating over the MCMC traces and counting the number of times $t_A < t_B$ (making sure each pair (t_A, t_B) comes from the same iteration to maintain the joint distribution) and forming empirical probability $p(t_A < t_B)$. A heuristic criterion can then be formed to assess whether the ordering between two cells is uncertain such as $p(t_A < t_B) \in [0.05, 0.95)$. We adopt this approach to understanding the uncertainty in orderings in section 3.7.

2.4.2 *Failure to account for pseudotime uncertainty leads to increased false discovery rates*

The previous section addressed the sources of statistical uncertainty in the pseudotimes. We next explored the impact of pseudotime uncertainty on downstream analysis. Specifically, we focused on the identification of genes that are differentially expressed across pseudotime. Typically, these analyses involve regression models that assume the input

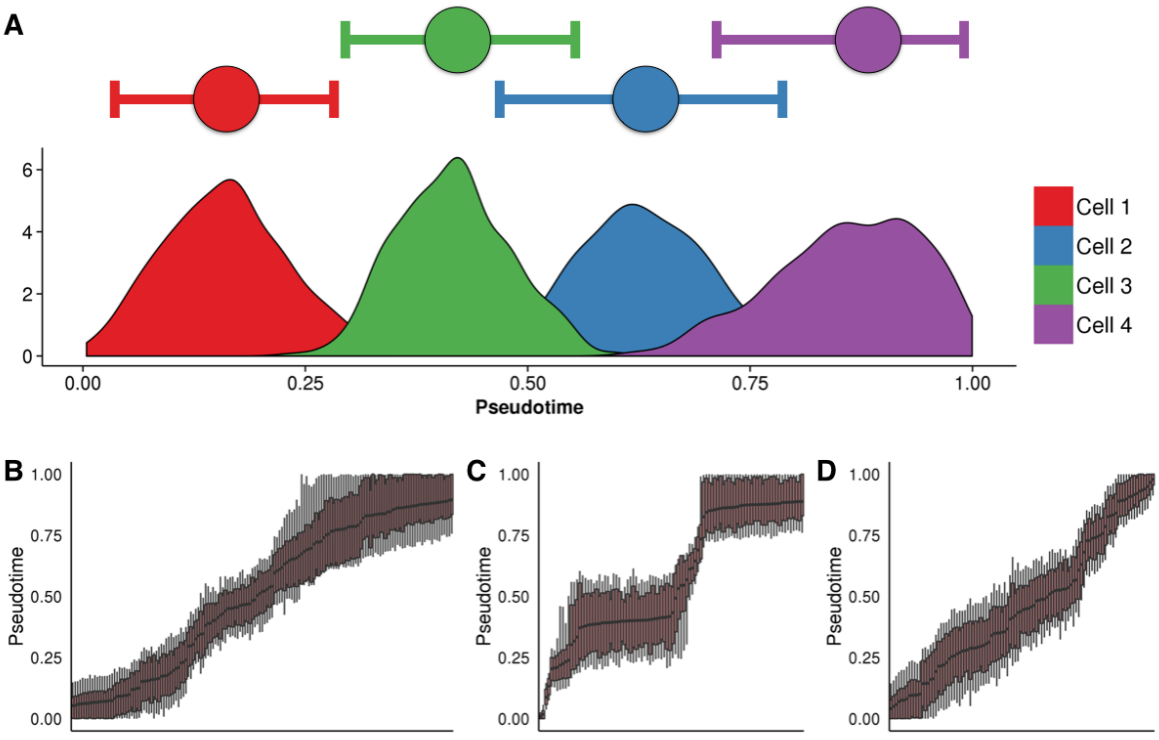


Figure 13: Posterior uncertainty in pseudotime trajectories. **A** Posterior uncertainty in pseudotimes for four randomly selected cells from the Trapnell et al. (2014) dataset. Horizontal bars represent the 95% highest probability density (HPD) credible interval (CI), which typically covers around a quarter of the pseudotime trajectory. **B-D** Boxplots showing the posterior uncertainty for each cell from the Trapnell et al. (2014) datasets. The edges of the boxes and tails correspond to the 75% and 95% HPD-CIs respectively.

variables (the pseudotimes) are both fixed and certain but, with our probabilistic model, we can use the posterior samples from our Bayesian model to refit the regression model to each pseudotime estimate. In doing so we can examine which genes are called as significant in each of the posterior samples and assess the stability of the differential expression analysis to pseudotime uncertainty by recording how frequently genes are designated as significant across the posterior samples. This allowed us to re-estimate the false discovery rate (FDR) fully accounting for the variability in pseudotime. As there are a multitude of sources of uncertainty on top of this (such as biological and technical variability) this allows us to put a lower bound on the FDR of such analyses in general.

Precisely, we fitted the Tobit regression model from [129] for each gene for each sample from the posterior pseudotime distribution, giving us a per-gene set of false-discovery-rate-corrected Q-values. We then compared the proportion of times a gene is called as differentially expressed (5% FDR) across all pseudotime samples to the Q-value using a point pseudotime estimate based on the maximum *a posteriori* (or MAP) estimate. We reasoned that if a gene is truly differentially expressed then such expression will be robust to the underlying uncertainty in the ordering. Note for comparison, our MAP estimates with the GPLVM correlate strongly with Monocle derived pseudotime point estimates (see figure 20).

Figures 14A&B show two analyses for two illustrative genes (*ITGAE* and *ID1*) in the Trapnell data set. Using the MAP pseudotime estimates, differential expression analysis of *ITGAE* over pseudotime attained a q-value of 0.02. However, the gene was only called significant in only 9% of posterior pseudotime samples with a median q-value of 0.32. In contrast, *ID1* - known to be involved in muscle differentiation - had a q-value of 6.6×10^{-11} using the MAP pseudotime estimate, but was also called significant in all the posterior pseudotime estimates having a median q-value of 4.4×10^{-11}. This indicates

that the significance of the temporal expression variability of *ID1* is highly robust whilst the significance *ITGAE* is much more dependent on the exact pseudotemporal ordering chosen.

As a conservative rule of thumb, we designated two sets of putative temporal associations: (i) a *robust* set comprising genes with a q-value less than 5% at the MAP estimate of pseudotime but also identified as significant in 95+% of the posterior pseudotime samples and (ii) an *unstable* set of genes whose q-values are less than 5% using the MAP estimate of pseudotime but is significant in fewer than 95% of the posterior pseudotime samples. Looking across all genes in the the three datasets we found that approximately half of the associations in all three datasets were unstable and whose temporal association depended on the choice of pseudotime estimates (figure 14C).

We performed a Gene Ontology enrichment analysis of the differentially expressed genes using (i) the robust set only, (ii) all DE genes (robust and unstable) and (iii) the unstable set only, using the GOseq package [143] with a 5% FDR significance level, enriching for categories corresponding to biological processes (figure 15). The set of unstable genes give no significant enriched GO categories on their own whilst the robust set gave a similar number of enriched GO categories as using all DE genes despite containing only half the number of genes. In all three datasets, there was a large overlap between the enriched GO categories identified and interestingly a high proportion of GO terms that are only significant from the robust gene set only. This suggests that the inclusion of the unstable gene set, potentially containing nuisance findings, may have reduced power to identify certain GO categories. Overall, our analysis suggests that the robust set of temporally differentially expressed genes identified by taking into account posterior uncertainty is a biologically meaningful gene set and not an arbitrary subset of the set of all DE genes.

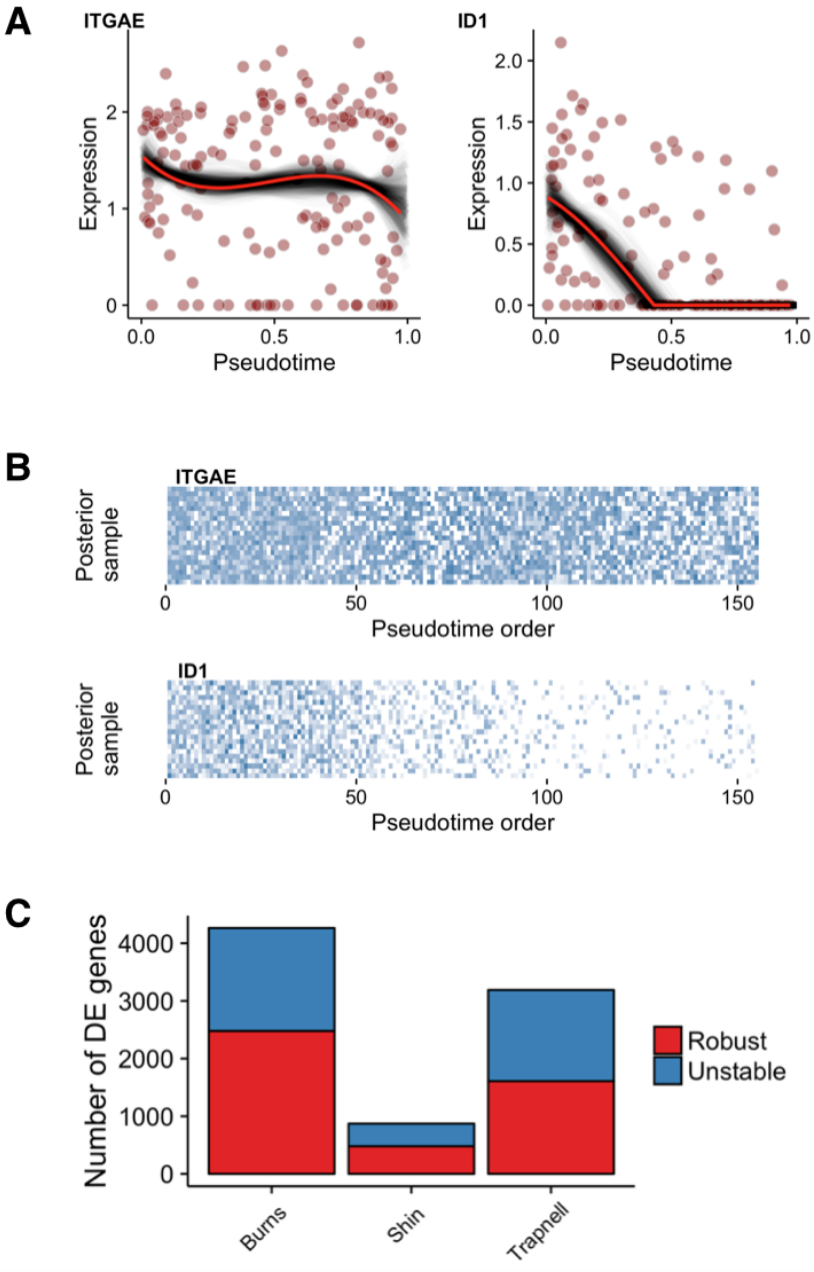


Figure 14: Approximate FDR for differential expression across pseudotime. **A** Gene expression plots across pseudotime, with black traces corresponding to models fitted to pseudotime samples while the red trace corresponds to the point (MAP) estimate for two exemplar genes and **B** corresponding gene expression heatmap for 20 randomly sampled posterior pseudotimes. **C** The number of genes identified as robust and unstable for all three datasets examined.

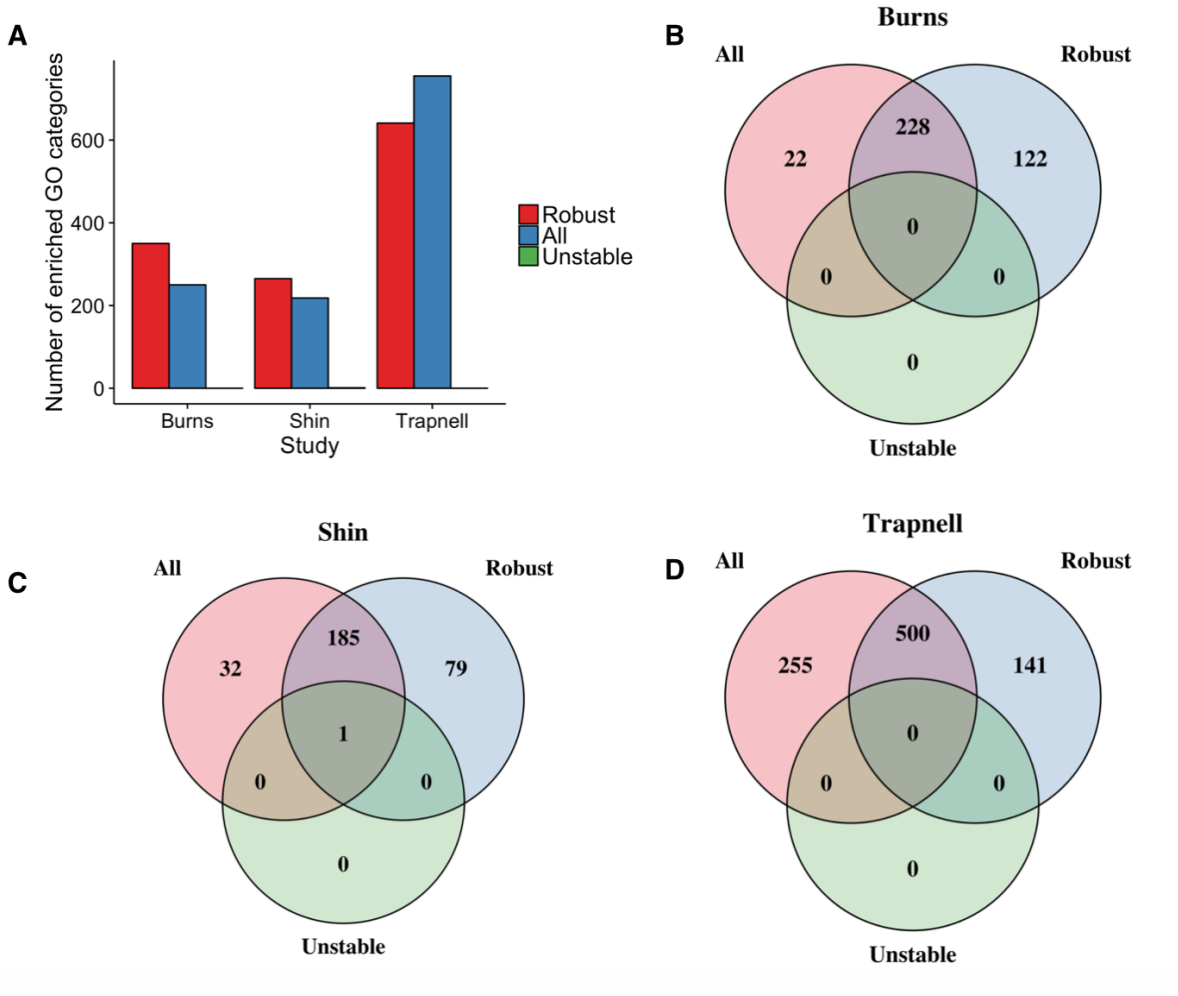


Figure 15: Gene Ontology Enrichment Analysis. **A** Number of enriched GO categories for the three datasets studied. Genes used for enrichment were either those that exhibit *robust* differential expression, *unstable* differential expression or *all*. **B-D** Venn diagrams showing the number of enriched GO terms based on the differential expression categories above.

2.4.3 *Applications of* `switchde` *with probabilistic pseudotime*

In the previous section, we examined differential expression across pseudotime by fitting generalized additive models to the gene expression profiles as done in Trapnell et al. [129]. However, as discussed in 2.2 this model provides a highly flexible but non-specific model of pseudotime dependence that was not suited to the next question we wished to address.

Specifically, we were interested in whether we could identify if two genes switched behaviours at the *same* (or similar) times during the temporal process and therefore an estimate of the time resolution that can be gained from a pseudotime estimation approach. By combining `switchde` with the Bayesian inference of pseudotime we can then infer the resolution to which we can say whether one gene switches on or off before another.

We applied `switchde` to learn patterns of switch-like behaviour of genes in the Trapnell dataset. For each gene we estimated the *activation time* t_0 as well as the *activation strength* k (see section 2.2). We fitted these sigmoidal switching models to all posterior pseudotime samples to approximate the posterior distribution for the time and strength parameters. We uncovered a small set of genes whose median activation strength is distinctly larger than the rest and had low variability across posterior pseudotime samples implying a population of genes that exhibit highly switch-like behaviour (figure 16A). Some genes showed high activation strength for certain pseudotime estimates but low overall median levels across all the posterior samples. We concluded that genes with large credible intervals on the estimates of activation strength do not show robust switch-like behaviour and demonstrate the necessity of using probabilistic methods to infer gene behaviour as opposed to point estimates that might give highly unstable results.

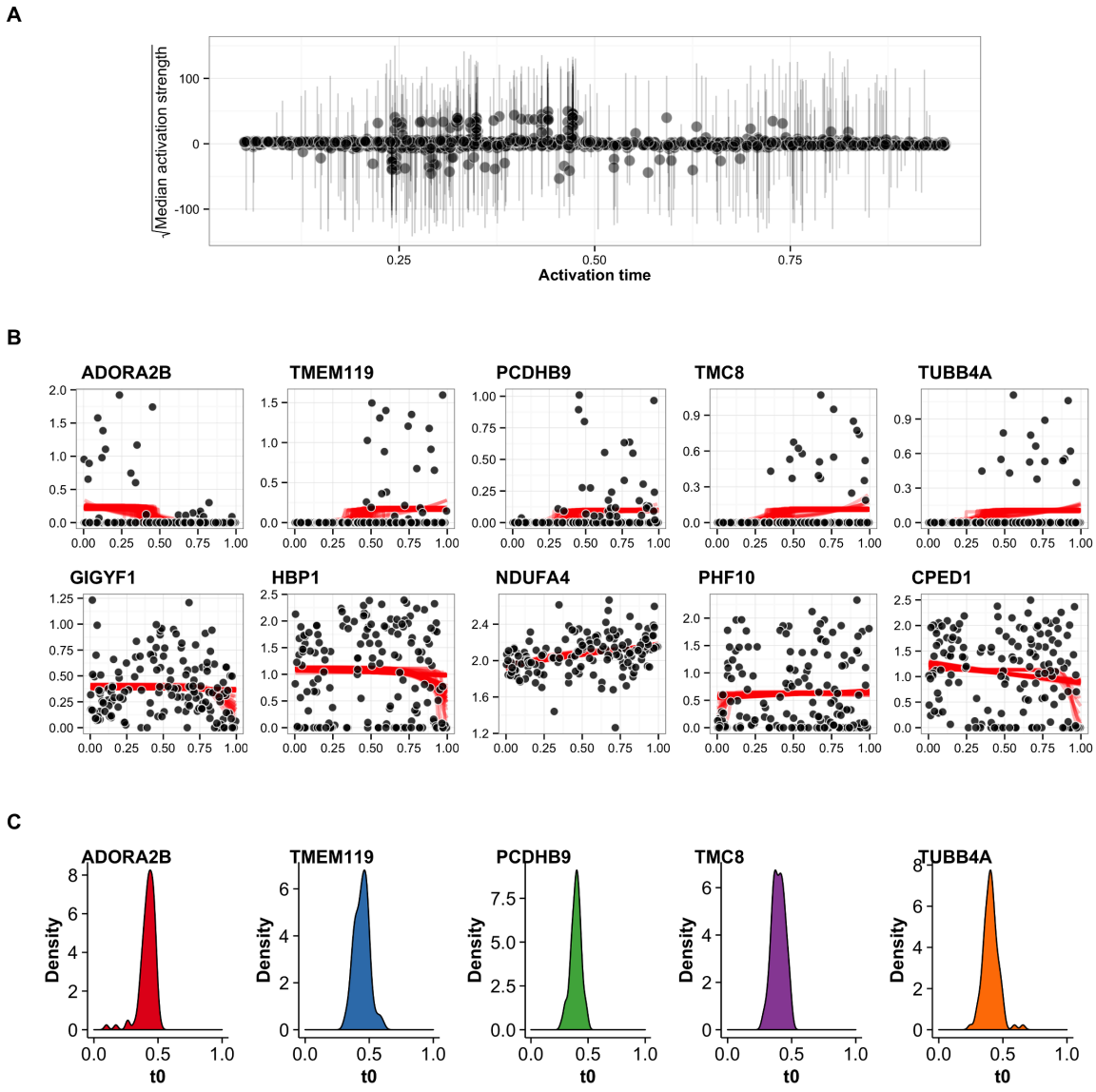


Figure 16: Robust inference of switch-like behaviour in genes across pseudotime. **A** The square-root of the median of the activation strength parameter k across all pseudotime samples as a function of activation time t_0. The error bars show the 95% credible interval, demonstrating that point estimates can severely skew the apparent behaviour of genes and a requirement for a robust Bayesian treatment of gene expression. A distinct population of genes whose median activation strength sits separate from the majority close to the x-axis implies a subset of genes show true switch-like behaviour. **B** Representative examples of genes whose median activation strength is large (top row) compared to small (bottom row). Each black point represents the gene expression of the cell with red lines corresponding to posterior traces of the sigmoidal gene expression model. Genes with a large activation strength show a distinct gene expression pattern compared to those with a small activation strength. **C** A posterior density plot of the activation time for the five genes showing strong activation strength in **B**.

Representative examples of genes with large and small activation strengths showed marked differences in the gene expression patterns corresponding to strong and weak switch-like behaviour as expected (figure 16B). In addition, we examined the posterior density activation time t_0 for the five genes showing strong switching behaviour (figure 16C). Under a point estimate of pseudotime each gene would give a distinct activation time with which these genes can be ordered. However, when pseudotime uncertainty is taken into account, a distribution over possible activation times emerges. In this case, the five genes all have activation times between 0.3 and 0.5 precluding a precise ordering (if one exists) of activation. Visually, this seems sensible since there is considerable cell-to-cell variability in the expression of these genes and not all cells express the genes during the "on" phase. We are therefore unable to determine whether the "on" phase begins when the first cell with high expression is first observed in pseudotime or, if it starts before, and the first few cells simply have null expression (for biological or technical reasons).

We further explore this in figure 17 which shows ten genes identified as having significant switch-like pseudotime dependence but with a range of mean activation times t_0. The switch-like behaviour is stable to the different posterior pseudotime estimates that were sampled from the GPLVM. It is clear that the two genes *RARRES3* and *C1S* are activating at an earlier time compared to the genes *IL20RA* and *APOL4*. However, we cannot be confident of the ordering within the pairs *RARRES3*/*C1S* and *IL20RA*/*APOL4* in pseudotime since the distributions over the activation times are not well-separated and it is impossible to make any definitive statements as to whether one of these genes (in)activates before another. If the probability of a sequence of activation events is required, instead of examining each gene in isolation, we can count the number of posterior samples in which one gene precedes another instead and evidence may

emerge of a possible ordering. These observations suggests a finite temporal resolution limit that can be obtained using pseudotemporal ordering.

2.4.4 *Contribution to pseudotime uncertainty from the reduced dimensional representation*

We next sought to address the impact of the dimensionality reduction that is often applied to single cell gene expression data prior to pseudotime estimation. The choice of dimensionality reduction approach is based on whether the method gives rise to a putative pseudotime trajectory in the reduced dimensionality representation. This is typically conducted with visual inspection followed by confirmational analysis by examining known marker genes with established temporal association. This may lead to a number of possibilities since the same trajectory may exist in a number of reduced dimensionality representations.

We sought to characterise the contribution of the dimensionality reduction process to pseudotime uncertainty. The wide variety of dimensionality reduction methods available and in use for pseudotime estimation precludes a complete investigation here and we choose to focus instead on principal components analysis - a standard technique - and used, for example, in Waterfall [120] or TSCAN [57] or as a preprocessing step used before applying non-linear methods such as t-SNE or GPLVMs.

To do this, we used the differentiating myoblasts data set [129] and performed PCA (using the `R` package `scater` [87]). We then applied GPLVM pseudotime estimation to the two-dimensional principal component representation and identified a set of 1,968 robustly differentially expressed genes from the posterior pseudotime samples. Next, we took 50 random subsets containing 80% of the cells and repeated the above procedure on each subset. As the PCA projection depends on the data itself, the reduced dimensional

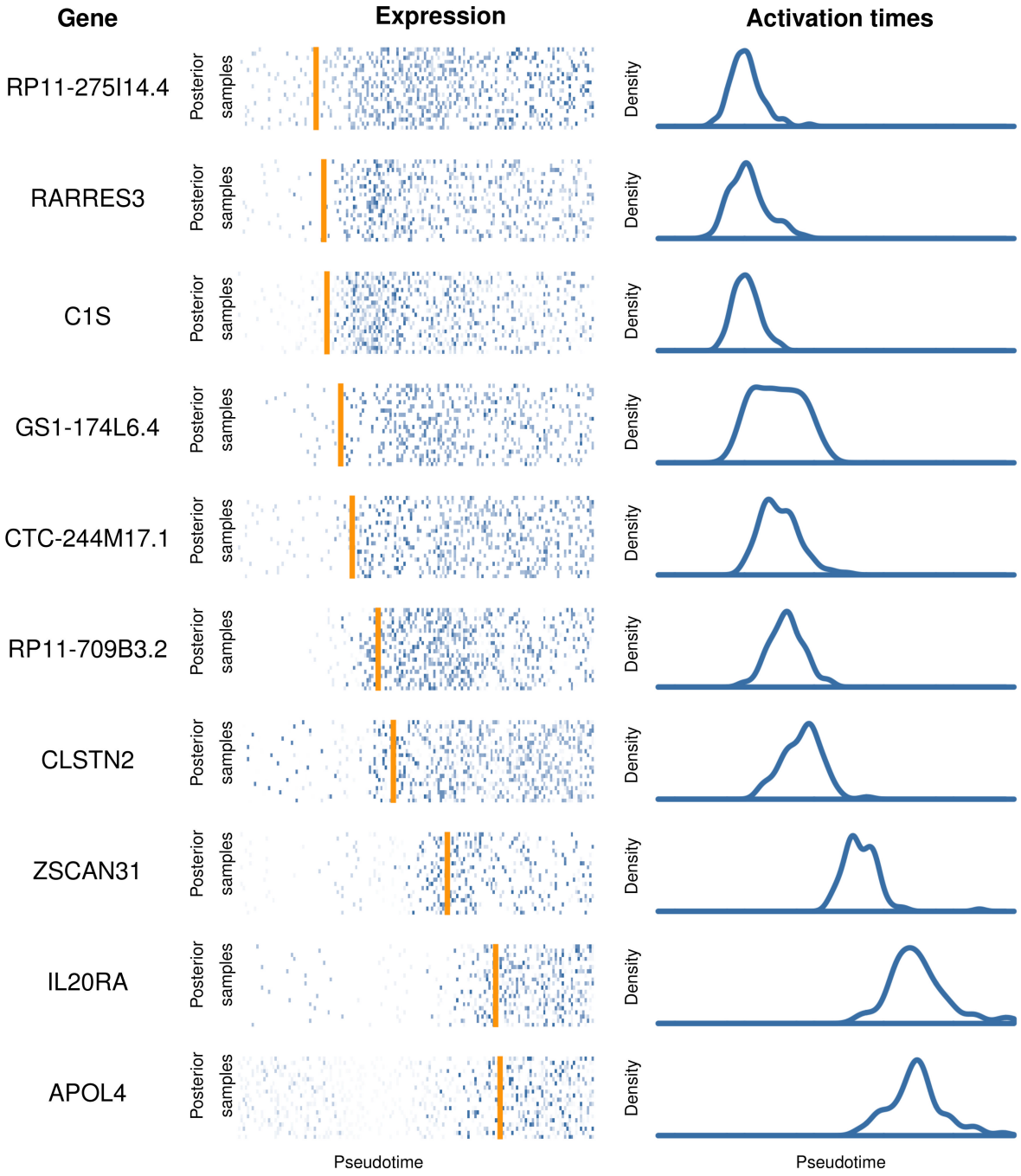


Figure 17: Identifying pseudotime dependent gene activation behaviour. Ten selected genes from [129] found using our sigmoidal gene activation model exhibiting a range of activation times. For each gene, we show the expression levels of each cell (centre) where each row corresponds to an ordering according to a different posterior samples of pseudotime. The orange line corresponds to a point estimate of the activation time. The posterior density of the estimated activation time is also shown (right).

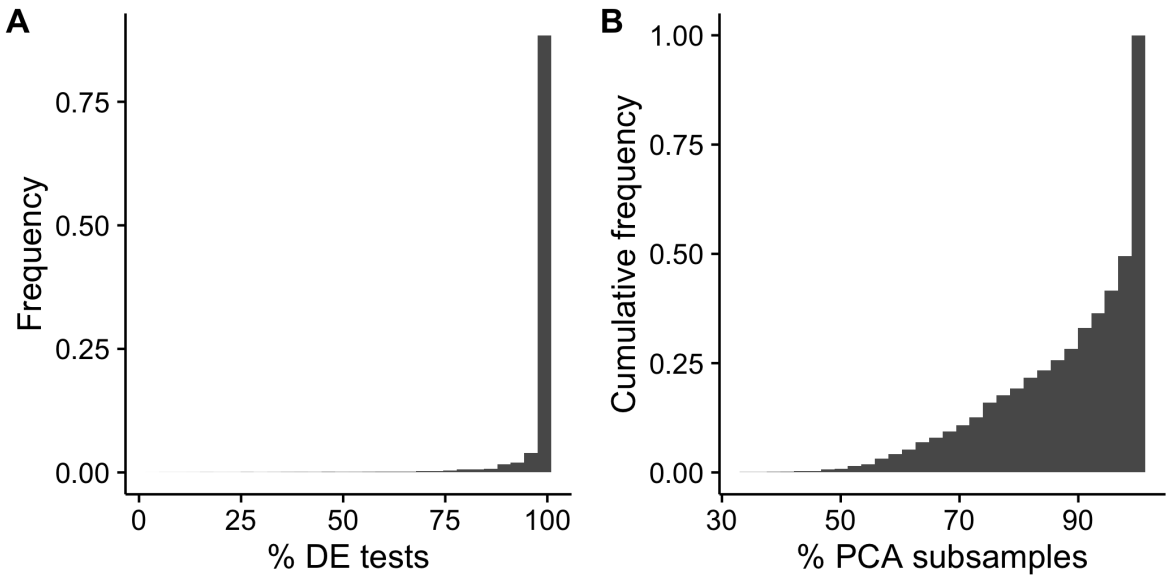


Figure 18: Pseudotime uncertainty arising from reduced dimensional representation. **A** Proportion of all subsampled differential expression tests where the gene was found to be robustly differentially expressed using only those genes robustly differentially expressed when considering all cells. **B** Cumulative frequency of cell subsamples in which a given gene is robustly differentially expressed. Over half of all genes were robustly differentially expressed in all PCA subsamples.

representation derived from each subsample will be different, we wanted to compare the differentially expressed genes identified to those found using the full data to determine if the variation in the reduced dimensionality representation impacts on the downstream differential expression analyses. We examined all genes and PCA subsamples and found that over 90% of differential expression tests showed a robust temporal association (figure 18A). Over half of all genes were robustly differentially expressed in all PCA subsamples and over 80% in at least 40 out of 50 PCA subsamples (figure 18B). These results indicate that the vast majority of genes remained robustly differentially expressed despite the differences in the reduced dimensional space across the random subsets (and the potential loss of detection power due to there being fewer cells).

Our results maybe further explained by the fact that principal components analysis uses a linear, orthogonal transformation that maximises variance in the principal direc-

tions. This will give reduced dimensional representations that are likely to be robust in many instances to variable cell inputs given sufficiently large sample sizes. Highly nonlinear techniques, such as t-SNE, Laplacian Eigenmaps or diffusion maps, maybe more sensitive to the input data and the resultant reduced dimensional representations more variable. A thorough characterisation of the relative contribution of the reduced dimensional representation and the curve fitting to the statistical uncertainty in the pseudotime estimates must be determined through simulations (like the ones detailed here) and conclusions may differ for different dimensionality reduction/curve fitting combinations.

2.4.5 *Inherent uncertainty in point-estimation methods*

To demonstrate that there is inherent uncertainty in pseudotime (i.e. it's not just the consequence of using a probabilistic model) we subsampled cells, recomputed the pseudotime for each subsample and computed the variance across subsamples. In particular, we recomputed the pseudotime estimate using Monocle's MST fitting and the Laplacian Eigenmaps embedding as above for 30 resamples to 80% of the total number of cells (without replacement). For each subsample, the pseudotimes were standardized to lie in $[0, 1)$. Since pseudotimes are equivalent up to a parity transformation, if the correlation between the pseudotimes assigned at each resample and using the entire cell set was less than 0 then the pseudotimes were rescaled to $\tilde{t} = 1 - t$ to 'orient' them in the right direction. Figure 19 shows the 2σ interval[3] across all cells. It can be seen to vary to as much as 0.5, implying there is a large inherent uncertainty in pseudotimes even using point-estimate methods.

3 Approximately equivalent to the 95% CI

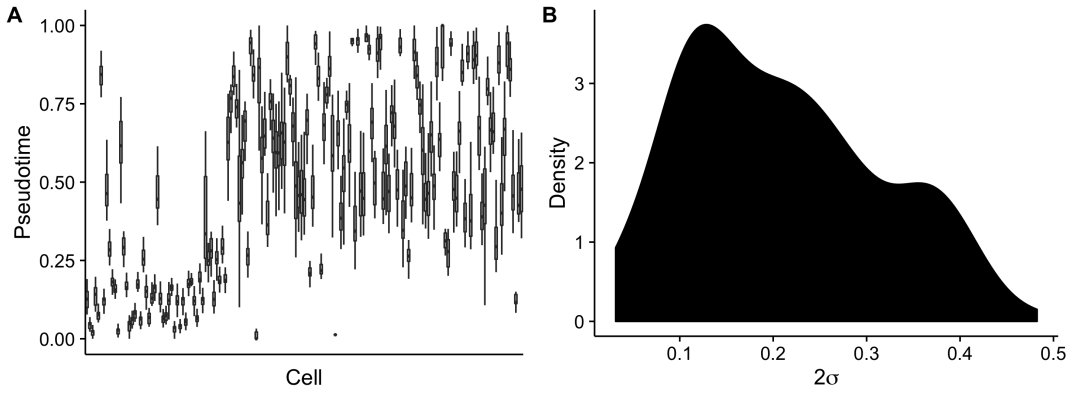


Figure 19: Inherent uncertainty in single-cell pseudotime. **A** The pseudotimes of the cells (ordered by index) with 2σ intervals as error bars. **B** The distribution of 2σ intervals reaches as large as 0.5, covering half the trajectory.

2.4.6 *Consistency with* `Monocle`

To ensure the GPLVM fit is consistent with other methods we compared the MAP pseudotime estimates with those assigned using Monocle [129]. Monocle works by using Independent Component Analysis (ICA) for dimensionality reduction then fits a minimum spanning tree (MST) in the reduced space. The longest path through the minimum spanning tree is taken to be the trajectory and the pseudotime of each cell is the length along this trajectory.

We performed two comparisons: firstly using the entire `Monocle` method (using the 500 most variable genes for dimensionality reduction), and secondly using just the MST fitting using the Laplacian Eigenmaps embedding as described above. The results can be seen in figure 20 with R^2 values of 0.83 and 0.96. Also plotted are the 95% HPD credible intervals, and in general the `Monocle` value is captured within this interval.

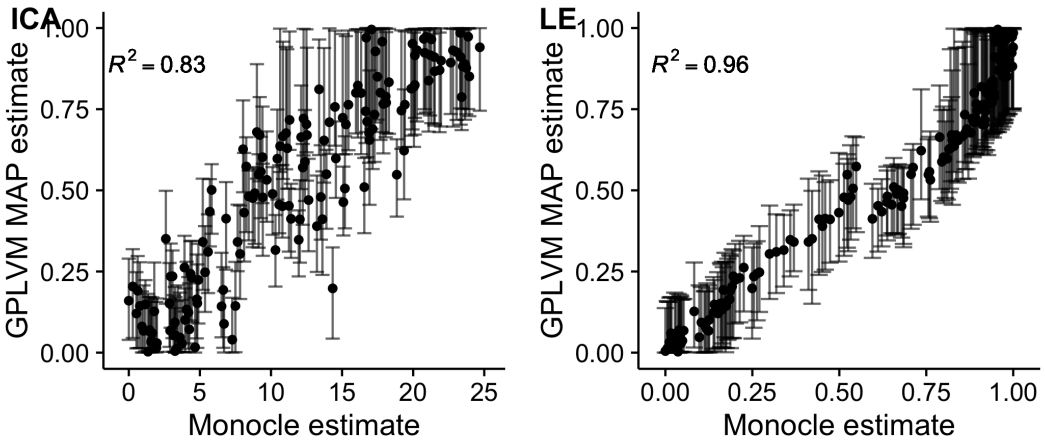


Figure 20: GPLVM pseudotime fits compared to Monocle for both ICA dimensionality reduction (left) and laplacian eigenmaps (right). Error bars show 95% posterior credible interval

2.5 DISCUSSION

Pseudotime estimation from gene expression profiling of single cells provides the ability to extract temporal information about complex biological processes from which *true* time series experimentation may be technically challenging or impossible. In this chapter we have sought to characterise the utility of a probabilistic approach to the single cell pseudotime estimation problem over approaches that only return a single point estimate of pseudotime. Our work is significant since it has so far not been possible to assess the impact of this statistical uncertainty in downstream analyses and to ascertain the level of temporal resolution that can be obtained.

In order to address this we adopted a Gaussian Process Latent Variable modelling framework to perform probabilistic pseudotime estimation within a Bayesian inference setting. The GPLVM allows us to probabilistically explore a range of different pseudotime trajectories within the reduced dimensional space. We showed that in a truly unbiased and unsupervised analysis the properties of the pseudotime trajectory will never purely be a product of the data alone and can heavily depend on prior assumptions

about the smoothness, length scales of the trajectory and noise properties. Using samples drawn from the posterior distribution over pseudotime estimates under the GPLVM we were able to assess if genes that showed a significant pseudotime dependence under a point (MAP) pseudotime estimate would be robust to different possible pseudotime estimates. In two of the three datasets we examined we discovered that, when adjusted for pseudotime uncertainty, the false discovery rate may be significantly larger than the target 5%. Our investigations show that reliance on a single estimate of pseudotime can lead to increased number of false discoveries but that it is possible to assess the impact of such assumptions within a probabilistic framework.

A caveat of the specific methodology adopted in this study is that it is necessarily computationally intensive due to the use of full Markov chain Monte Carlo based Bayesian inference and is dominated by functions of the Gaussian Process covariance matrix that have complexity $\mathcal{O}(N^3)$ where N is the number of cells. Our STAN implementation is able to process 300 cells in around 15 minutes on a standard laptop computer but well-known variational approximations based on inducing point [125] and recent stochastic variational algorithms [49] can reduce the computational burden and improve scaling to $\mathcal{O}(N) > 10^4$. However, this scalability comes at the expense of the reduced ability to fully characterise posterior uncertainty. In this study we have focused purely on best characterising the posterior uncertainty using MCMC algorithms that asymptotically converge to the true posterior distribution. In practice this purest approach may not be necessary but we argue that pseudotime uncertainty should be addressed.

It is important to note that the GPLVM used in our investigations is not intended to be a single, all-encompassing solution for pseudotime modelling problems. For our purposes, it provided a simple and relevant device for tackling the single trajectory pseudotime problem in a probabilistic manner but clearly has limitations when the temporal process under investigation contains bifurcations or heteroscedastic noise processes (as discussed

earlier). Improved and/or alternative probabilistic models are required to address more challenging modelling scenarios but the general procedures we describe are generic and should be applicable to any problem where statistical inference for a probabilistic model can give posterior simulation samples.

We also developed a novel sigmoidal gene expression temporal association model that enabled us to identify genes exhibiting a strong switch-like (in)activation behaviour. For these genes we were then able to estimate the activation times and use these to assess the time resolution that can be attained using pseudotime estimates of single cells. Our investigations show that pseudotime uncertainty prevents precise characterisation of the gene activation time but a probabilistic model can provide a distribution over the possibilities. In application, this uncertainty means that it is challenging to make precise statements about when regulatory factors will turn on or off and if they act in unison. This places an upper limit on the accuracy of dynamic gene regulation models and causal relationships between genes that could be built from the single cell expression data.

In conclusion, single cell genomics has provided a precision tool with which to interrogate complex temporal biological processes. However, as widely reported in recent studies, the properties of single cell gene expression data are complex and highly variable. We have shown that the many sources of variability can contribute to significant uncertainty in statistical inference for pseudotemporal ordering problems. We argue therefore that strong statistical foundations are vital and that probabilistic methods for provide a platform for quantifying uncertainty in pseudotemporal ordering which can be used to more robustly identify genes that are differentially expressed over time. Robust statistical procedures can also temper potentially unrealistic expectations about the level of temporal resolution that can be obtained from computationally-based pseudotime estimation. Ultimately, as the raw input data is not true time series data, pseudotime estimation is only ever an attempt to solve a *missing data* statistical inference problem

that we should remind ourselves involves quantities (pseudotimes) that are *unknown, never can be known.*

3

INFERENCE OF PSEUDOTIMES WITH SWITCH-LIKE MODELS

3.1 INTRODUCTION

A predominant feature of current pseudotime algorithms is that they emphasise an "unsupervised" approach where pseudotimes are learned using no specific prior knowledge of gene behaviour. Typically, the high-dimensional molecular profiles for each cell are projected on to a reduced dimensional space by using a (non)linear transformation provided by the manifold learning algorithms discussed in section 1.3.4. In this reduced dimensional space, it is hoped that any temporal variation is sufficiently strong to cause the cells to align against a trajectory along which pseudotime can be measured. This approach is therefore subject to a number of analysis choices, e.g. the choice of dimensionality reduction technique, the trajectory fitting algorithm, etc., that could lead to considerable variation in the pseudotime estimates obtained. In order to verify that any specific set of pseudotime estimates are biologically plausible, it is typical for investigators to retrospectively examine specific marker genes or proteins to confirm that the predicted (pseudo)temporal behaviour reflects *a priori* beliefs. An iterative "semi-supervised" process maybe therefore be required to concentrate pseudotime algorithms on behaviours that are both consistent with the measured data and compliant with a limited amount of known gene behaviour.

In this chapter we present an orthogonal approach implemented in a latent variable model statistical framework called Ouija that can integrate prior expectations of gene behaviour along trajectories using Bayesian nonlinear factor analysis. Our approach uses known or putative marker genes directly for pseudotime estimation rather than as a device for retrospectively validating pseudotime estimates. In particular, our model

focuses on switch-like expression behaviour and assumes that the marker gene expression follows a noisy switch pattern that corresponds to up- or down-regulation over time. Crucially, we explicitly model when a gene turns on or off as well as how quickly this behaviour occurs. We can then place Bayesian priors on this behaviour that allows us to learn temporal gene behaviours that are consistent with existing biological knowledge.

This chapter proceeds by explicitly framing single-cell pseudotimes as a statistical latent variable model, departing from typical approaches that employ a two stage dimensionality reduction and cell ordering process. We consider some desirable properties of such models including their mean function and mean-variance relationship. We benchmark this method on synthetic data before applying it to a range of single-cell developmental trajectories.

3.2 CONNECTION BETWEEN LATENT VARIABLE MODELLING AND TRAJECTORIES

The aim of pseudotime ordering is to associate a G-dimensional expression measurement $\mathbf{y}_n$ of cell $n \in 1, \ldots, N$ to a latent unobserved univariate pseudotime t_n. Mathematically we can express this as

$$\underbrace{\mathbf{y}_n}_{\text{Expression}} = \underbrace{\mathbf{f}_\Theta}_{\text{Mapping}} \Big(\underbrace{t_n}_{\text{Pseudotime}} \Big) + \underbrace{\epsilon_n}_{\text{Noise}} \tag{33}$$

where the function $\mathbf{f}_\Theta$ maps the one-dimensional pseudotime t_n to the G-dimensional observation space in which the data lies with (possibly output-dimension dependent) parameters Θ. The challenge lies in the fact that the mapping function $\mathbf{f}$, its parameters Θ, and the pseudotimes are all *unknown*. Furthermore, it is hard to derive objective criteria to evaluate potential mapping functions - why is $f(t) = \theta t$ (factor analysis)

or $f(t) =$ nonparametric smooth functions (GPLVM) necessarily "better" than $f(t) =$ const or $f(t) = 1/t$? A key result discussed below is that f corresponds to our *a priori* expectation of how gene expression changes over time.

3.2.1 *Factor analysis assumes linear gene activations*

If the mapping functions $\mathbf{f}$ are restricted to a linear functions of t_n then the generative model for the expression of gene g in cell n is given by a one-dimensional factor analysis model of the form

$$y_{ng} \sim \mathcal{N}(\theta_g t_n, \tau_g^{-1}) \tag{34}$$

where τ_g^{-1} is the measurement precision for gene g and the $G-$dimensional vector $\boldsymbol{\theta}$ acts as the factor loading matrix. Note that the data can always be centred making it redundant to model an intercept[1].

The interpretation of this model is that the value of θ_g regulates the response of the gene expression $\mathbf{y}_g$ to the cell's position along the trajectory t_n. Given this is necessarily linear, factor analysis assumes a linear change in expression for each gene over pseudotime.

The key insight is that the functional form of f necessarily corresponds to our prior expectations for how genes evolve along pseudotemporal trajectories. For linear f, we expect linear behaviour; for nonparametric smooth f (e.g. from GPLVM), we essentially restrict such behaviour to any smooth (mean) function of pseudotemporal progression.

If we modify equation 34 to model a common precision across all genes so $\tau_g = \tau$ then then this further reduces to (probabilistic) principal components analysis [124],

1 This is only true if the expectation of t_n is 0. We could conceive of a model where we place priors on t_n that correspond to physical capture times, in which case we would need to reintroduce gene-specific intercepts.

providing an explicit interpretation for the results of principal component analysis on single-cell data.

Our objective here is to use parametric forms for the mapping function f that will enable relatively fast computations whilst characterising certain gene expression temporal behaviours. Our premise is that prior knowledge might exist for a set of marker genes whose temporal behaviour is known to be approximately switch-like and therefore can be used to infer pseudo-time orderings.

3.3 A GENERATIVE MODEL OF SINGLE-CELL PSEUDOTIME

3.3.1 *Modelling considerations*

3.3.1.1 *Mean function*

If we wish to build a generative statistical latent variable model of pseudotime we must therefore decide on a sensible set of functions f that accurately characterise gene expression over (pseudo-)time. We have so far essentially considered f as a linear change over time (PCA/factor analysis) and f simply as smooth functions (GPLVM).

However, both the linearly-restricted case and the entirely nonparametric cases are unrealistic for different reasons. The linear case assumes that the mean function must change across the entire trajectory (i.e. there can be no parts where the mean value is stationary), making modelling of complex processes where different genes are responsible for different parts of the trajectory impossible. Furthermore, it assumes that as the trajectory progresses the expression values will tend to $\pm\infty$ (which has the additional issue of negative expression values). While the linear case is too restrictive, the non-parametric case is overly-flexible, allowing biologically unrealistic expression patterns such as cubic polynomials or genes with multiple modes over a relatively short trajectory.

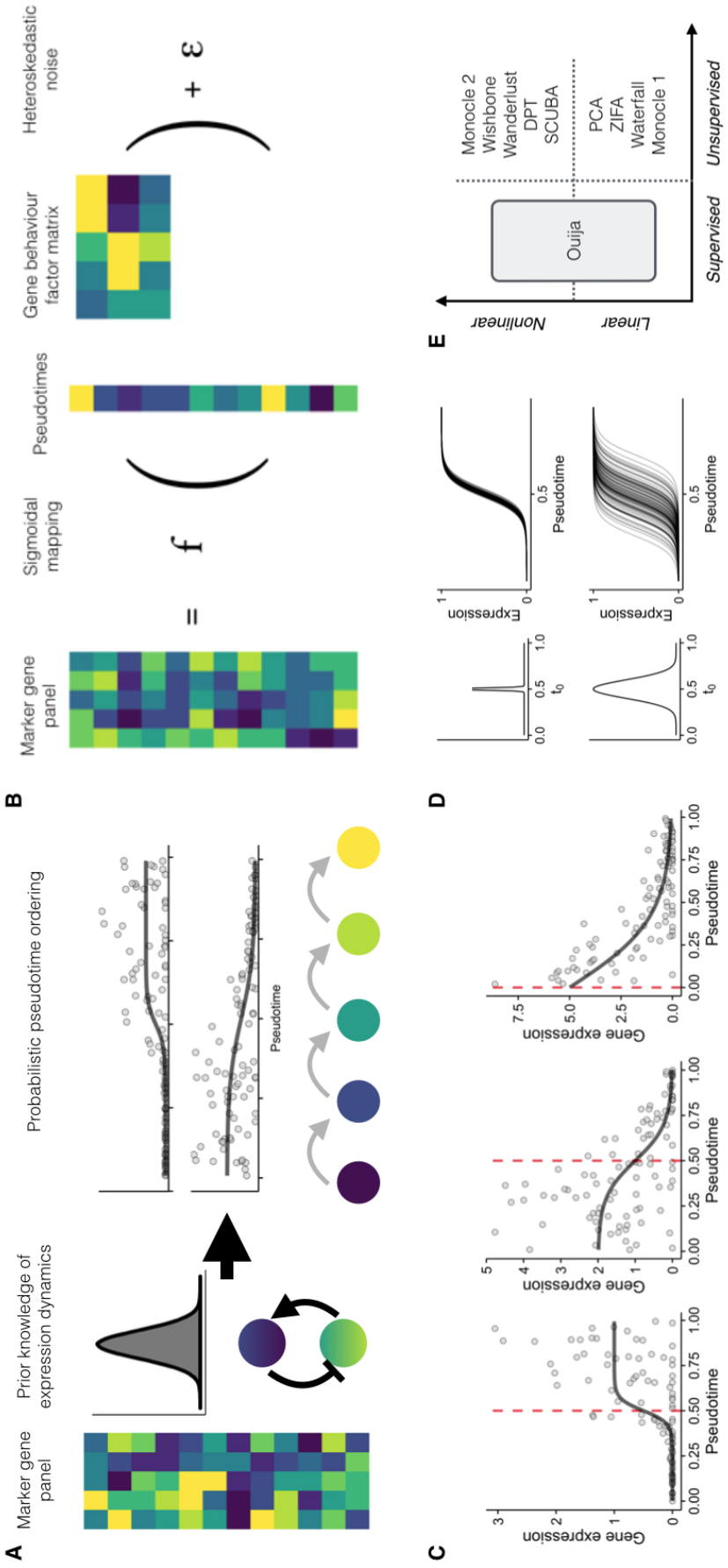


Figure 21: (Caption next page.)

Figure 21: Overview of Ouija. **A** A small panel of genes is chosen with *a priori* knowledge about their expression dynamics, either as cell-type specific marker genes or from pathway interaction databases such as KEGG. The pseudotimes are subsequently inferred using Ouija followed by downstream analyses such as differential expression and clustering of additional genes using standard methods such as those described in [129]. **B** Ouija infers pseudotimes using Bayesian nonlinear factor analysis by decomposing the input gene expression matrix through a sigmoidal mapping function. The latent variables become the pseudotimes of the cells while the factor loading matrix is informative of different types of gene behaviour. A heteroscedastic dispersed noise model with dropout is used to accurately model scRNA-seq data. **C** Examples of sigmoidal gene expression for three different sets of parameters. Points represent simulated gene expression, with the solid black line denoting the sigmoid curve and the red dashed line denoting the activation time. The sigmoid function is parameterised by (i) μ_0 - the average peak expression level, (ii) k - the activation strength and (iii) t_0 - the activation time. *Left* Fast 'switch-on' behaviour with parameters $k = 30$, $\mu_0 = 1$, $t_0 = 0.5$, *Centre* Slower 'switch-off' behaviour with parameters $k = -10$, $\mu_0 = 2$, $t_0 = 0.5$ and *Right* Decaying behaviour with parameters $k = -5$, $\mu_0 = 10$, $t_0 = 0$. **D** The effect of prior expectations on the 'activation time' t_0 parameter. A highly peaked prior (top) corresponds to confident knowledge of where in the trajectory a gene turns on or off, while a more diffuse prior (bottom) indicates more uncertainty as to where in the trajectory the gene behaviour occurs. **E** Comparison of Ouija to alternative pseudotime inference algorithms. Ouija is the only algorithm that explicitly allows incorporation of prior knowledge of gene behaviour.

Instead we opt for a 'goldilocks' mean function, where we assume genes are off, undergo a period of near-linear increase and settle at a constant value (or conversely begin on then turn off)[2]. A good approximation to this is the sigmoid function (considered previously in chapter 2) which parametrises the mean as

$$f(t; \mu_0, k, t_0) = \frac{2\mu_0}{1 + \exp\left(-k(t - t_0)\right)} \tag{35}$$

where t corresponds to the pseudotime of a cell, μ_0 corresponds to the mean expression of the gene, k is the *activation strength* and corresponds to how fast the gene turns on or off and t_0 corresponds to the *activation time* - where along the trajectory the gene behaviour occurs. The key result here is that the activation times and strengths are highly interpretable quantities for which realistic Bayesian priors may be available. For example, a researcher may know how quickly a given gene is up or downregulated over pseudotime, providing the sign and magnitude of k. Furthermore, they may know whether such regulation occurs early or late in the trajectory, providing prior information for t_0. Crucially, by specifying a variance on the prior knowledge the researcher can encode the perceived certainty of the prior information, which leads to differing prior function draws (figure 21D).

The approach we adopt is therefore a form of latent variable model implemented as *non-linear parametric factor analysis* where the factors correspond to the pseudo-times and the factor loadings correspond to the parameters of the sigmoidal function which provides the interpretable non-linearity (figure 21B). Overall, this positions our model apart from previous pseudotime approaches that all emphasise linear and nonlinear *unsupervised* learning of the trajectories (figure 21E).

2 In practice this restricts us to the class of monotonically increasing and decreasing gene expression signatures

3.3.1.2 *Mean-variance relation*

RNA-seq counts are commonly assumed to follow the negative binomial distribution [2, 74, 115] which for gene g in sample n implies a mean-variance relationship of the form

$$\sigma^2_{ng} = \mu_{ng}(1 + \phi_g \mu_{ng}) \tag{36}$$

where ϕ_g is a gene-specific dispersion factor. Such strong parametric forms are required since for low sample sizes the estimates of the variance can be very unstable.

However, typically in single-cell data there are enough measurements (i.e. cells) to allow robust estimation of both the mean and variance for each gene [35]. The exception to this is in pseudotime analyses, where we are assuming each cell represents a unique time point, and therefore the mean μ_{cg} and variance σ^2_{cg} are effectively measured only once. Consequently we must consider a strong mean-variance relationship since assuming a constant variance per gene is akin to under-fitting while it would be impossible to fit a variance for each cell and each gene (since there is only one measurement).

As a solution to this we examine the mean-variance relationship for the genes across all cells and assume the same relationship approximately holds for cells as they progress along pseudotime trajectories. The relationship in 36 applies to the original untransformed data (e.g. TPM or scaled counts) while we wish to model the $\log_2(\text{TPM} + 1)$ transformed relationship directly. Therefore we must examine the mean-variance relationship for the $\log_2$ data, since in general the mean-variance relationship of the log-transformed data isn't the same as the log of the mean-variance relationship on the untransformed data.

The mean-variance relationship of log count data has been examined both in Limma Voom [68] and sleuth [103] that conclude there is no closed form mean-variance relationship for logged data. Their solution is to fit a LOESS curve between the empirical

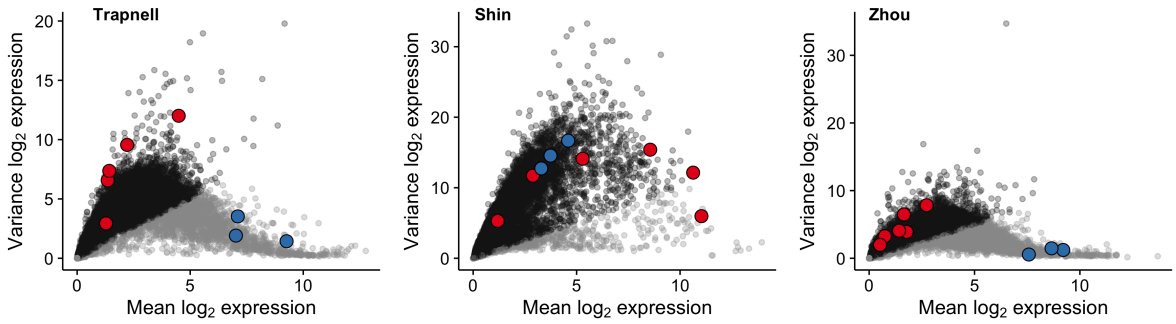


Figure 22: The mean-variance relationship in $\log_2(\text{TPM}+1)$ single-cell data for the three datasets. Red denotes the marker genes identified in the original publications while blue corresponds to three house-keeping genes (*LDHA*, *NONO*, *PGK1*) or their mouse equivalents. Both the Trapnell and Zhou datasets show consistent evidence that pseudotemporal marker genes exist on the 'leading edge' of the data with medium mean expression but high variance. This suggests a linear relationship between mean and variance in $\log_2$ space. In contrast, the housekeeping genes all sit in the 'tails' with high mean expression but very low variance.

mean and variance (or standard deviation) of each gene, and use this to shrink variance estimates towards a common value given the mean expression. However, if possible we would like a fully Bayesian model that fits all parameters simultaneously, rather than pre-fitting variance estimates with LOESS. We therefore consider a parametric approximation.

We examined the mean-variance relationship of the logged data for three datasets, as seen in figure 22. The pseudotemporal marker genes identified in the original text are shown in red. For both the Trapnell and Zhou datasets these lie on the 'leading edge' of the relationship, in areas of moderate mean expression but high variance. In comparison, the housekeeping genes (shown in blue) lie in the tails in regions of high mean expression but low overall variance. This makes intuitive sense, as we expect the marker genes to turn on across the trajectory, giving them a mean of around half the maximal value but maximum variance. In contrast, we expect housekeeping genes to have maximal expression across the trajectory but very low variance in keeping with the

constancy of their expression. We therefore assumed that any genes we wish to model as pseudotemporal marker genes follow the same linear mean-variance relationship.

3.3.1.3 *Single-cell dropout*

As previously discussed in section 1.1.5.2 single-cell RNA-seq data is known to exhibit substantial dropout, whereby a failure to reverse transcribe lowly expressed transcripts results in zeros in the count matrix. Note that this is distinct from *true zeros* where no mRNA transcripts are truly present, due to either bursty or non-expression.

To account for this we assume that the probability of a dropout is logistic on the latent gene expression mean in a similar approach to Kharchenko et al. [59]. The difference to previous approaches is that (1) we are working in $\log_2(\text{TPM}+1)$ space and (2) we assume a unique mean μ_{cg} for every gene in every cell, giving a per-gene per-cell dropout probability $p(\mu_{cg})$ of

$$\log\left[\frac{p(\mu_{cg})}{1-p(\mu_{cg})}\right] = \beta_0 + \beta_1\mu_{cg}. \tag{37}$$

The dropout parameters β_0 and β_1 are constant across all cells as modelling gene specific dropout parameters[3] would require extreme shrinkage across genes in order to make the parameters identifiable.

3.3.2 *Statistical model specification*

We now detail the overall specification of our nonlinear factor analysis pseudotime model. As usual we index N cells by $n \in 1,\ldots,N$ and G genes by $g \in 1,\ldots,G$ and let $y_{ng} = [\mathbf{Y}]_{ng}$ denote the (log-transformed and suitably normalised) observed cell-by-gene expression matrix. In order to make the strength parameters comparable between

3 Which is presumably closer to the true model due to gene-specific biases such as GC content.

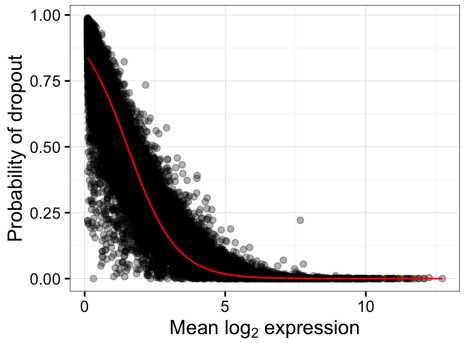


Figure 23: The probability of a dropout against the mean $\log_2$ expression in the Trapnell et al. dataset. The red solid line shows the logistic regression fit.

genes we normalise the gene expression so the approximate half-peak expression is 1 through the transformation $\mathbf{y}_g \rightarrow \mathbf{y}_g / s_g$, in which we define s_g as a gene-specific size factor that approximates the half-peak expression of gene g:

$$s_g = \frac{1}{|\mathcal{Y}_g^*|} \sum_{y_{ng}^* \in \mathcal{Y}_g^*} y_{ng}^* \tag{38}$$

where $\mathcal{Y}_g^* = \{y_{ng} : y_{ng} > 0\}$ is the set of non-zero measurements for g

Our statistical model can be specified as a Bayesian hierarchical model where the likelihood is given by a bimodal distribution formed from a mixture of zero-component (dropout) and an non-zero expressing cell population. Let $\mathbf{t}$ be an N-length pseudotime vector (one for each cell) and let π_{ng} denote the probability of observing a dropout in cell n and gene g, then the gene expression is modelled as

$$\boldsymbol{\beta} \sim \mathcal{N}(0, 0.1) \tag{39}$$

$$\pi_{ng} \sim \text{Bernoulli}(\text{logit}^{-1}(\beta_0 + \beta_1 \mu_{ng})) \tag{40}$$

$$p(y_{ng} | \pi_{ng}, \mu_{ng}, \sigma_{ng}^2) = \pi_{ng} \delta(y_{ng}) + (1 - \pi_{ng}) \text{Student}(y_{ng} | \mu_{ng}, \sigma_{ng}^2) \tag{41}$$

where $\delta(y)$ is the Dirac-delta function centred at 0.

The relationship between dropout rate and expression level is expressed as a logistic regression model [59]. Furthermore, we impose a mean-variance relationship of the form

$$\sigma_{ng}^2 = (1+\phi)\mu_{ng} + \epsilon \tag{42}$$

with the hierarchical prior structure

$$t_n \sim \mathcal{N}(0.5, 1), \tag{43}$$
$$\mu_{ng} = \mu_g^{(0)} f(t_n, k_g, t^{(0)}), \tag{44}$$
$$\phi \sim \text{Gamma}(\alpha_\phi, \beta_\phi) \tag{45}$$

where ϕ is a dispersion parameter.

We define the sigmoid function as

$$f(t_c; k_g, t_g^{(0)}) = \frac{2}{1+\exp\left(-k_g(t_c - t_g^{(0)})\right)}, \tag{46}$$

where k_g and $t_g^{(0)}$ denote the activation strength and activation time parameters for each gene and $\mu_g^{(0)}$ the average peak expression with default priors

$$\mu_g^{(0)} \sim \text{Gamma}(\delta/2, 1/2), \tag{47}$$
$$k_g \sim \mathcal{N}(\mu_g^{(k)}, 1/\tau_g^{(k)}), \tag{48}$$
$$t_g^{(0)} \sim \mathcal{N}(\mu_g^{(t)}, 1/\tau_g^{(t)}), \tag{49}$$

If available, user-supplied prior beliefs can be encoded in these priors by specifying the parameters $\mu_g^{(k)}, \tau_g^{(k)}, \mu_g^{(t)}, \tau_g^{(t)}$. Otherwise, inference can be performed using uninformative hyperpriors on these parameters. Specifying $\mu_g^{(k)}$ encodes a prior belief in the

strength and direction of the activation of gene g along the trajectory with $\tau_g^{(k)}$ (inversely) representing the strength of this belief. Similarly, specifying $\mu_g^{(t)}$ encodes a prior belief of where in the trajectory gene g exhibits behaviour (either turning on or off) with $\tau_g^{(t)}$ encoding the strength of this belief.

3.3.3 *Inference*

We performed posterior inference using Markov Chain Monte Carlo (MCMC) stochastic simulation algorithms, specifically the No U-Turn Hamiltonian Monte Carlo approach [53] implemented in the STAN probabilistic programming language [24] (see section 2.3.3). The parameter $\epsilon = 0.01$ is used to avoid numerical issues in MCMC computation. For larger marker gene panels, such as in the cell cycle analysis (section 3.8), we used the automatic differentiation variational inference (ADVI) implemented in STAN to perform approximate Bayesian inference. The STAN code can be found in appendix C.1.

3.3.3.1 *Convergence diagnostics*

Since Ouija relies on MCMC simulation for inference it is crucial to assess model convergence before drawing statistical conclusions. The Ouija `R` package contains a convenience function `plot(ouija_fit, what = "diagnostic")` to plot both the posterior log-likelihood and posterior log-likelihood autocorrelation, an example of which is in figure 24A. In general we found it practical to run MCMC inference on only a single chain for most models. Therefore, to check both good mixing of the chains and that the models aren't getting stuck in local maxima, we tested the models for both the Trapnell and Shin datasets across multiple chains. We then examined several diagnostics such as the log-posterior mixing, effective sample size, Gelman-Rubin statistic and ratio of

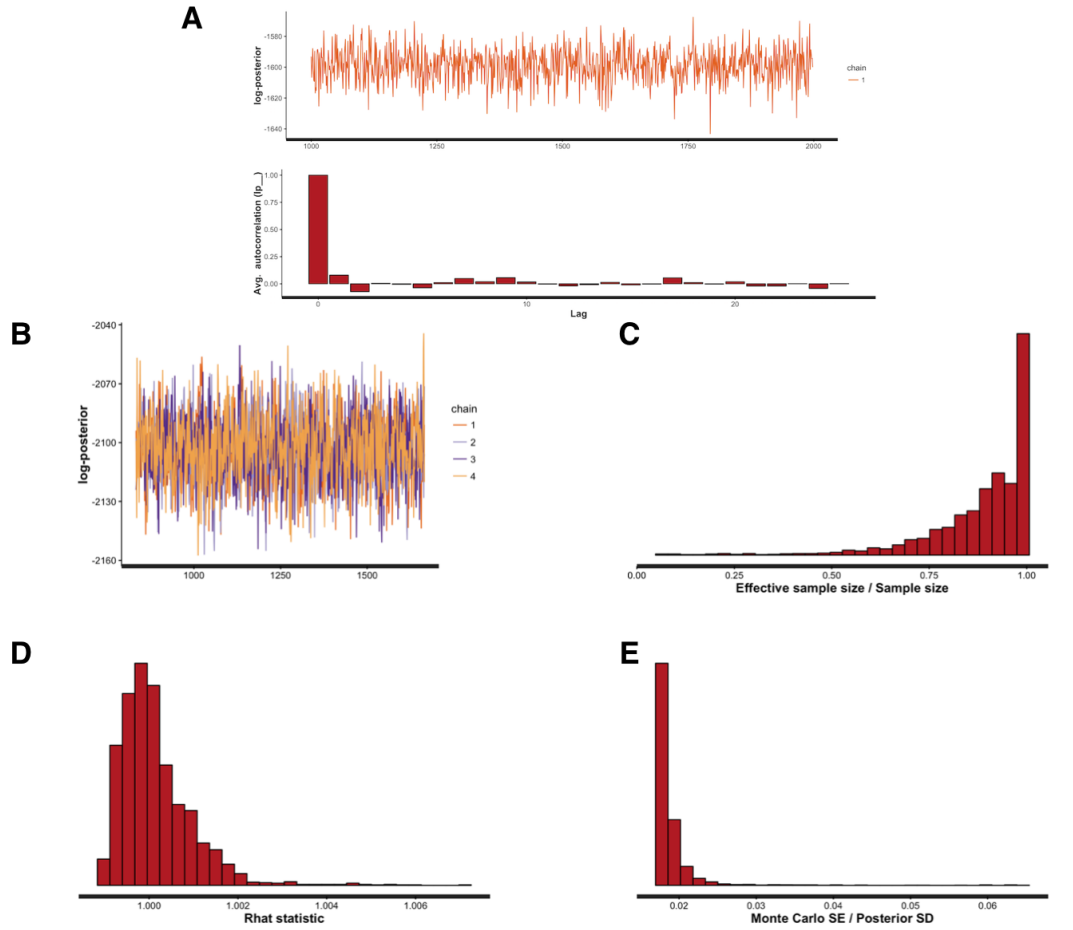


Figure 24: MCMC convergence diagnostics. **A** For a single chain Ouija contains a convenience function to plot both the trace of the log-likelihood and the log-likelihood auto-correlation. **B-E** MCMC diagnostic statistics such as posterior log-likelihood trace, effective sample size, Gelman-Rubin statistic and ratio of monte carlo standard error to posterior standard deviation show good mixing across chains.

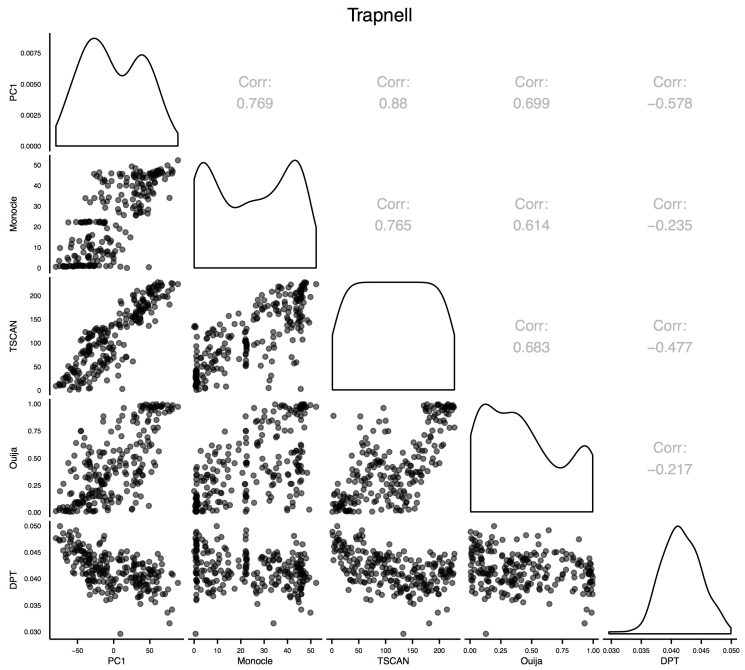
monte carlo standard error to posterior standard deviation (figures 24B-E), all of which fell inside the acceptable range.

3.4 COMPARISON OF MARKER AND WHOLE-TRANSCRIPTOME PSEUDOTIMES

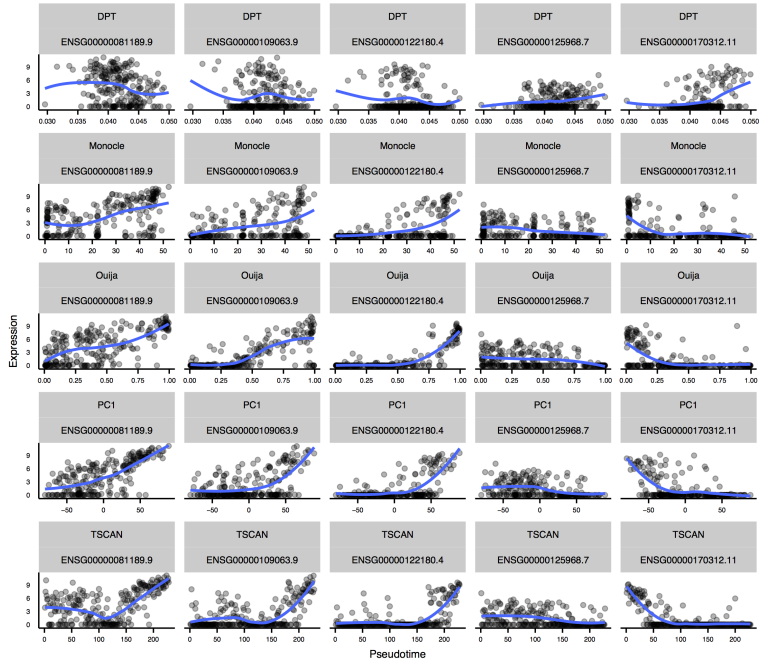
We investigated the use of limited marker gene panels for pseudotime inference using three single cell expression datasets: (i) adult hippocampal quiescent neural stem cells (Shin) [120], (ii) differentiating myoblasts (Trapnell) [129] and (iii) embryonic precursor cells into haematopoietic stem cells (Zhou) [148]. We began by computing pseudotimes using a selection of pseudotime algorithms for each dataset using the small set of 5-6 marker genes identified in the respective studies. We compared marker-based pseudotimes estimated using Ouija, a reference pseudotime based on the first principal component (PC1), TSCAN [57], Monocle 2 [107], and DPT [46]. Monocle and TSCAN were run with default parameter settings. For DPT, white noise with standard deviation of 10^{-6} of each gene's standard deviation was added to the expression data to avoid errors concerning cells with identical expression.

Figure 25(a) shows the correlation between pseudotimes obtained for the Trapnell data set whilst equivalent results for the Shin and Zhou data sets can be found in appendix C figures 65(a) and 66(a) respectively. Our analysis shows that pseudotime estimates obtained from marker gene panels differed between pseudotime algorithms with many showing only weak correlation between inferred pseudotimes. However, qualitatively, all algorithms produced plausible gene expression variation over the inferred pseudotimes (figures 25(b), appendix C 65(b), and 66(b)).

We investigated further by examining the relationship between pseudotimes obtained from marker gene panels, whole transcriptomes and gene set sizes falling between these two extrema. We computed a transcriptome-wide pseudotime where genes retained had



(a) Pseudotimes reported by the five algorithms do not exhibit strong correlation.



(b) Expression level fits reported by each algorithm qualitatively reflect their intrinsic modelling assumptions. Blue line shows a smoothing curve fitted to the pseudotemporally ordered expression values.

Figure 25: Comparison of marker gene-based pseudotime estimates across five algorithms.

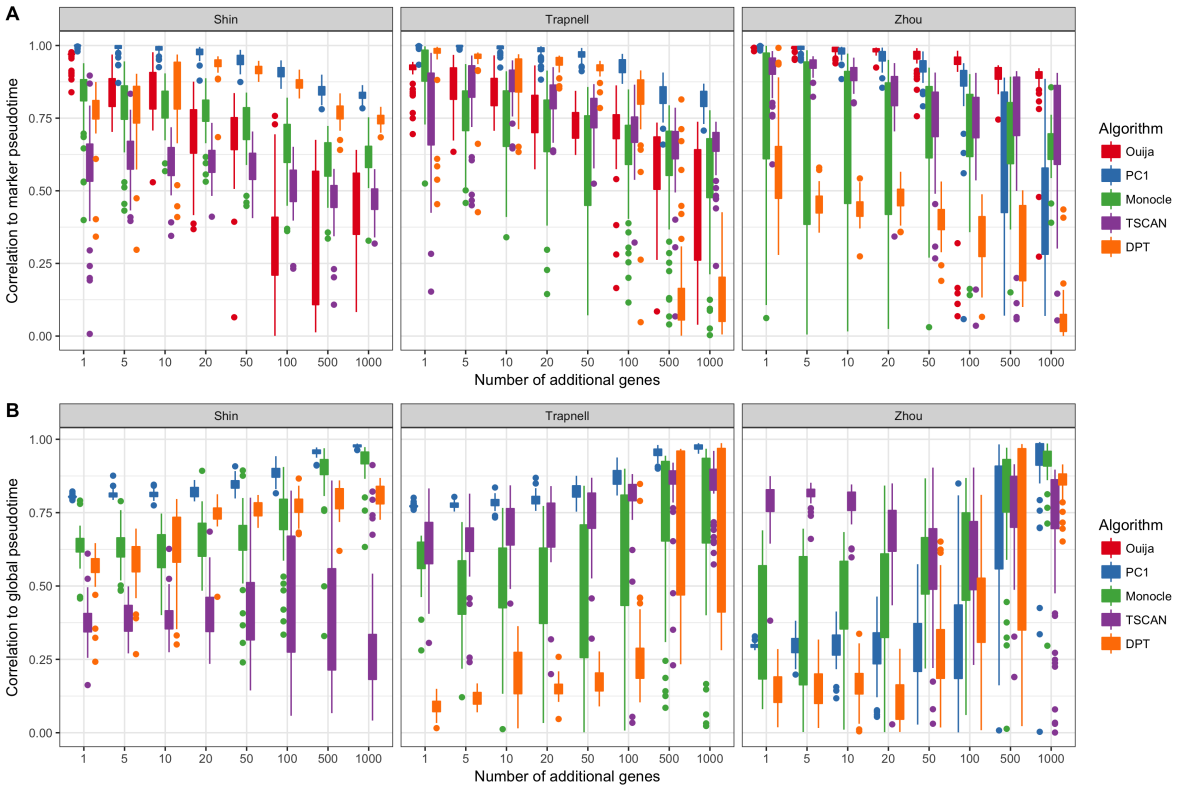


Figure 26: Comparison of pseudotime methods across three scRNA-seq data sets. We compared pseudotime estimates from three studies [120, 129, 148] using 5 different approaches. **A** Boxplots of the correlation between marker-based pseudotimes with those obtained from random gene-sets of varying (including the marker gene set) across 50 replicates. **B** Boxplots of correlation with whole transcriptome (global) pseudotime obtained from all genes across 50 replicates.

a variance in log-expression (TPM/FPKM) greater than 1 and were expressed in at least one cell at log-expression greater than 1. Furthermore, for each dataset and algorithm combination, we generated additional gene sets that consisted of the marker genes and $N = 1, 5, 10, 20, 50, 100, 500, 1000$ randomly chosen genes from the full transcriptome-wide gene set. In doing so we were able to investigate the continuum between marker-based and whole-transcriptome pseudotime estimation. This was performed for 50 replicates for each N. We then compared the results by considering (Pearson) correlation to the marker pseudotimes and correlation to the global pseudotimes.

Our major observation is the striking variability in pseudotime estimates across data sets, gene sets and replicates for all algorithms (figure 26). As the number of additional genes included in each data set increases, the correlation of the estimated pseudotimes to the reference marker-based pseudotimes typically decreases (figure 26A) whilst the correlation with global pseudotime increases respectively (figure 26B). Note that Ouija is not included in the latter analysis since it is not intended to be applied to whole transcriptome analyses. This behaviour is expected since the forms of variation encoded in the larger gene set are likely to differ from that encoded within the marker genes. However, it is evident that depending on the dataset and genes chosen there is large variability in the consistency of pseudotime algorithm fitting. In other words, there is no one global solution that consistently fits the same pseudotime that will recapitulate the desired behaviour of marker genes.

We finally compared the consistency of differential expression across pseudotime using a recently published method for detecting which genes vary across pseudotime [18]. All p-values were corrected using Benjamini-Hochberg and an FDR threshold of 5% used to define significance. For each algorithm and number of additional genes we computed the proportion of genes deemed significant found in (a) the differential expression test

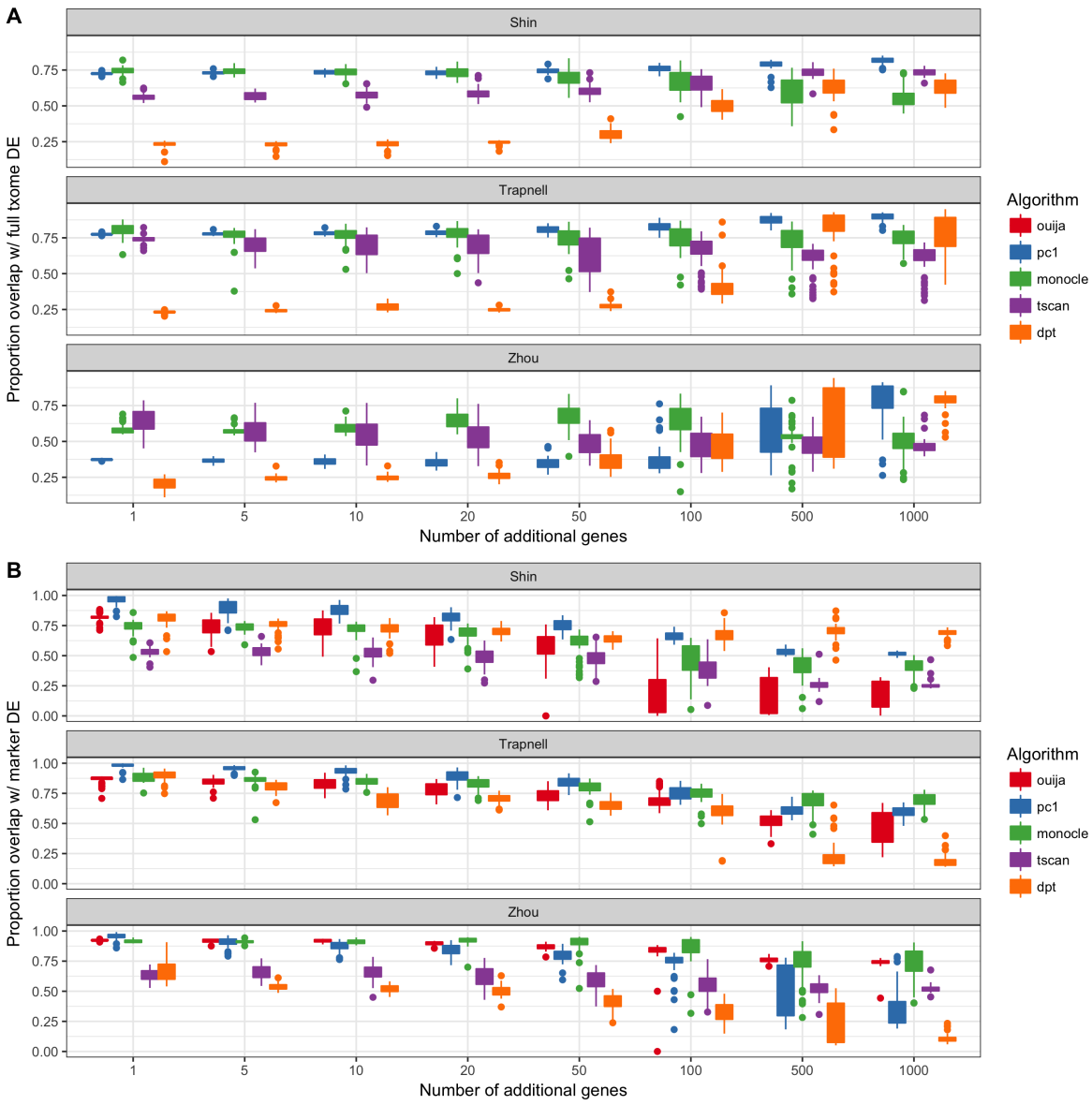


Figure 27: Consistency of differential expression analyses. For each dataset and algorithm, we added a set number of randomly chosen genes (1, 5, 10, 20, 100, 500, 1000) to the set of marker genes, recomputed the pseudotimes and performed differential expression analysis. We then compared the proportion overlap of differentially expressed genes to those found at the full transcriptome pseudotimes (**A**) and at the marker-based pseudotimes (**B**).

using the transcriptome-wide pseudotime and (b) the differential expression test using the marker-only pseudotime.

It can be see that as the number of markers included increases the rate of overlap to the transcriptome-wide DE calls also generally increases (figure 27A). Notably, the rate is very low for DPT when the number of included markers is small, which is consistent with the low correlation to marker pseudotimes exhibited in figure 26. In contrast, as the number of included marker genes increases, the rate of overlap to the marker DE calls decreases (figure 27B) as expected.

3.5 INCORPORATING PRIOR INFORMATION IMPROVES PSEUDOTIME INFERENCE

3.5.1 *Simulating switch-like pseudotime regulation*

We performed large scale simulations to assess both the utility of prior information in pseudotime inference and the robustness of Ouija to model misspecification in how the genes are regulated. Pseudotimes were simulated for $N = 100$ cells from a Unif(0,1) distribution. One of four mean functions (see below) was used to simulate the regulation of expression across pseudotime at various levels of misspecification with respect to the Ouija model.

Subsequently, an identical noise model was placed on top of the mean function motivated by empirical observations of single-cell RNA-seq data. If μ_{ng} is the mean expression value for cell n and gene g under a given mean function, a variance $\sigma^2_{ng} = \gamma\mu_{ng}$ is calculated in line with previous observations, where we take $\gamma = 3.5$ as the result of regressing the variance on mean expression in key marker genes from [129]. Following this a latent expression value $y^*_{ng} \sim \mathcal{N}(\mu_{ng}, \sigma^2_{ng})$ is generated, from which the observed

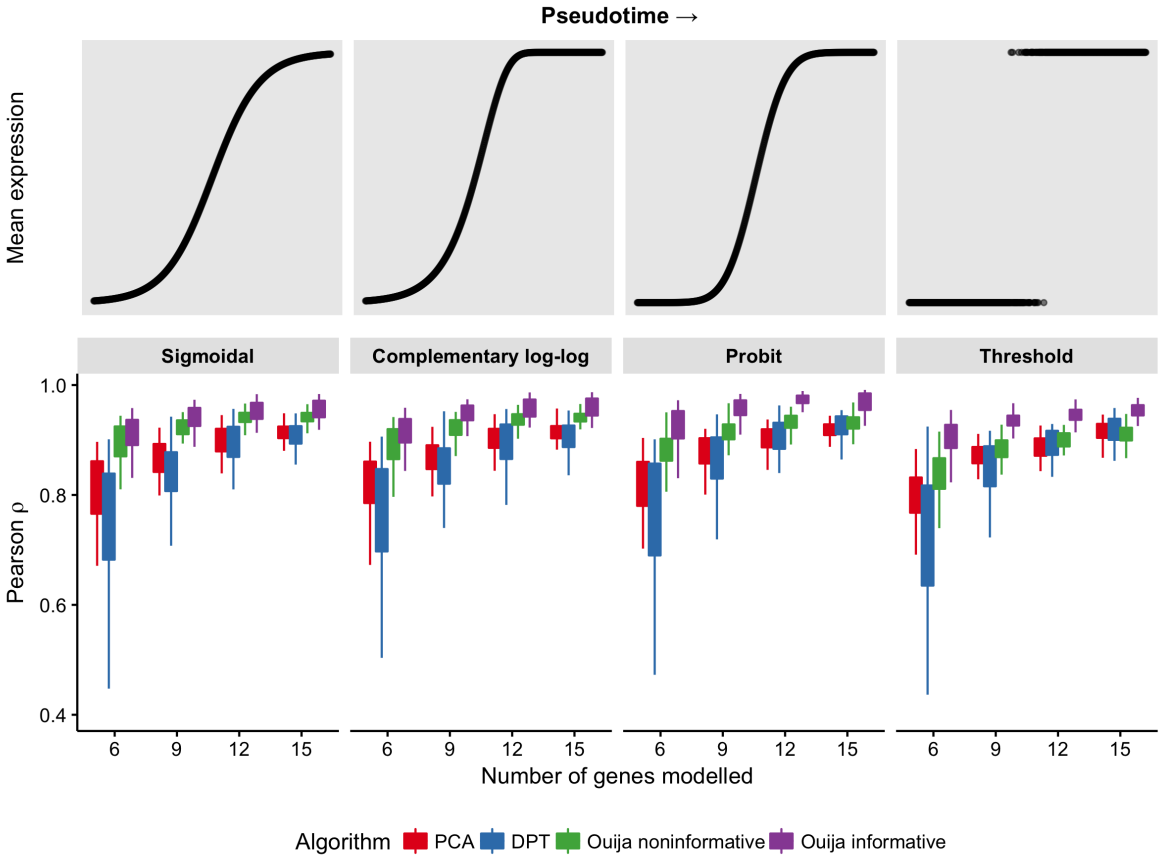


Figure 28: Incorporating prior information improves pseudotime inference of switch-like genes. Four mean functions were considered in simulations (top), each corresponding to different switch-like behaviour across pseudotime. We then reinferred the pseudotimes for four different gene set sizes with 40 replications each, for PC1, DPT, and Ouija with and without informative priors. In each case, increasing the number of genes improves pseudotime inference, as does incorporating prior information.

expression values are constructed. Firstly, any values of $y^* < 0$ are set to 0 (though these are typically small in number as the variance of the observations approaches zero as the latent values do so). Secondly, a zero inflation step is applied to y^* to simulate dropout ubiquitous in single-cell RNA-seq data. For each observation a dropout probability $p_{ng}^{\text{dropout}} = \text{Logit}^{-1}(\beta_0 + \beta_1 \mu_{ng})$ is calculated, where $\beta_0 = 1.76$ and $\beta_1 = -1.16$ are calculated using the same dataset as before. Each observation is subsequently set to zero with probability p_{ng}^{dropout}.

3.5.1.1 *Mean functions*

We sought to simulate switch-like behaviour from three link functions commonly used in generalized linear regression, along with a modified version that leads to further misspecification.

SIGMOIDAL The sigmoidal mean function corresponds to that used by Ouija. Given the pseudotime t_n the mean is calculated via

$$\mu_{ng} = \frac{2\mu_g^{(0)}}{1 + \exp(-k_g(t_n - t_g^{(0)}))} \tag{50}$$

for which we draw $\mu_g^{(0)} \sim \text{Unif}(3, 4)$, $t_g^{(0)} \sim \text{Unif}(0.1, 0.9)$ and $k_g \sim \text{Unif}(5, 20)$ and negated with probability $\frac{1}{2}$. Given an assumed scale of $\log_2(\text{TPM} + 1)$ this selection of parameters, combined with the noise model, provides a reasonable range of observed expression values.

COMPLEMENTARY LOG-LOG The complementary log-log (*cloglog*) acts as a link function in logistic regression, modelling the probability of success π in terms of regres-

sors $\mathbf{x}$ and coefficients $\boldsymbol{\beta}$ via $\log(-\log(1-\pi_n)) = \mathbf{x}_n^T\boldsymbol{\beta}$. Therefore, we use it to generate a mean via

$$\mu_{ng} = 2\mu_g^{(0)}\left(1 - \exp(-e^{k_g(t_n - t_g^{(0)})})\right) \tag{51}$$

for which we draw the parameters identically to the sigmoidal case.

PROBIT The probit link models the probability of success as $\pi_n = \Phi(\mathbf{x}_n^T\boldsymbol{\beta})$, where Φ is the cumulative distribution function of the standard normal distribution. Thus we use

$$\mu_{ng} = 2\mu_g^{(0)}\Phi\left(k_g(t_n - t_g^{(0)})\right). \tag{52}$$

$\mu_g^{(0)}$ and $t_g^{(0)}$ are drawn as before, while $k_g \sim \text{Unif}(10, 50)$ to take into account the differing curvature compared to sigmoid and cloglog (and negated with probability $\frac{1}{2}$ as before).

THRESHOLD We sought to create a further switch-like mean function that would be maximally mis-specified with respect to Ouija. This follows a two part process and requires a sign variable $|k_g|$ that describes whether the gene is up or down regulated along pseudotime. Firstly, a probit variable is generated by drawing $m_g \sim |k_g| \times \text{Unif}(1, 2)$ and $c_g = -m_g t_g^{(0)}$. Then draw $\mu_{ng}^* \sim \mathcal{N}(m_g t_n + c_g, 0.1)$. The mean function is given by

$$\mu_{ng} = \begin{cases} 2\mu_g^{(0)} & \text{if } \mu_{ng}^* > 0 \\ 0 & \text{otherwise} \end{cases} \tag{53}$$

In this situation we draw $\mu_g^{(0)}$ and $t_g^{(0)}$ as before and $|k_g|$ is positive or negative with equal probability.

Examples for the four different mean functions can be seen in figure 28.

3.5.2 *Pseudotime inference*

For each mean function we re-inferred the pseudotimes with PCA (first principal component), diffusion pseudotime, and Ouija in two configurations - noninformative and informative. For the noninformative case, default settings are used which consists of $k \sim \mathcal{N}(0, 1)$ and $t^{(0)} \sim \mathcal{N}(0.5, 1)$. For the informative case, the prior mean on $t_g^{(0)}$ was set to its true value and the prior standard deviation 0.1.

For the informative case the prior on k varied slightly depending on the simulation condition, but in each case the standard deviation was reduced to 0.1. For the sigmoidal and cloglog mean function regimes, the prior means were set to the true values of the generated data. For the probit and threshold datasets, the prior was set to 50 multiplied by the sign of the true k. In other words, we declare the direction of regulation, and that strong switch-like behaviour is exhibited, but nothing more.

We included a further two settings for Ouija on the sigmoidal dataset only. In the "switch midpoint" setting, the prior mean on $t^{(0)}$ is set to 0.5, while in "switch uncertainty" the prior mean on $t^{(0)}$ is set to the true value plus a $\mathcal{N}(0, 0.1)$ random variable. In both cases, all other parameters are the same as the Ouija informative setting.

This data was simulated for $G = 6, 9, 12, 15$ "marker" genes with 40 replications per gene set and mean function.

3.5.3 *Results*

The results can be seen in figure 28, in which a few trends are obvious. Across all algorithms, correlation with the "true" pseudotimes increases as the number of genes increases, which makes sense as we expect more data to increase the accuracy of inference. In general, Ouija with noninformative priors has better concordance with the ground truth pseudotimes, which again makes sense as Ouija explicitly models switch-like expression in comparison to PCA which models linear changes and DPT that models smooth changes. The exception to this is the threshold model, where Ouija with noninformative priors performs comparatively with PCA and DPT. This is perhaps the most challenging situation as there is virtually no information in the gene expression pre- and post- switch point to guide inference.

A further trend is that incorporating prior information improves pseudotime inference across all mean functions. For example, with a probit mean function and $G = 12$, the average pearson correlation without prior information is $\rho = 0.91$ while incorporating prior information gives $\rho = 0.97$. The results also implies that Ouija's performance is agnostic to the precise form of the switching function, with no obvious degradation in accuracy in moving from sigmoidal to alternative mean parameterisations. Intruigingly competitive performance is maintained using the threshold mean function, which contains virtually no pseudotime information pre- and post- switch point for each gene.

We subsequently sought to test the dependency of Ouija's accuracy on how the priors on the switching times are specified. As mentioned above, Ouija was run in two additional modes for the sigmoidal mean function - "switch midpoint", where the prior means on the switch times are set to 0.5, and "switch uncertainty", where the prior means on the switch times are set to the true values plus a $\mathcal{N}(0, 0.1)$ variable. The results of this can be seen in figure 29. It can be seen that incorporating any information on the switch

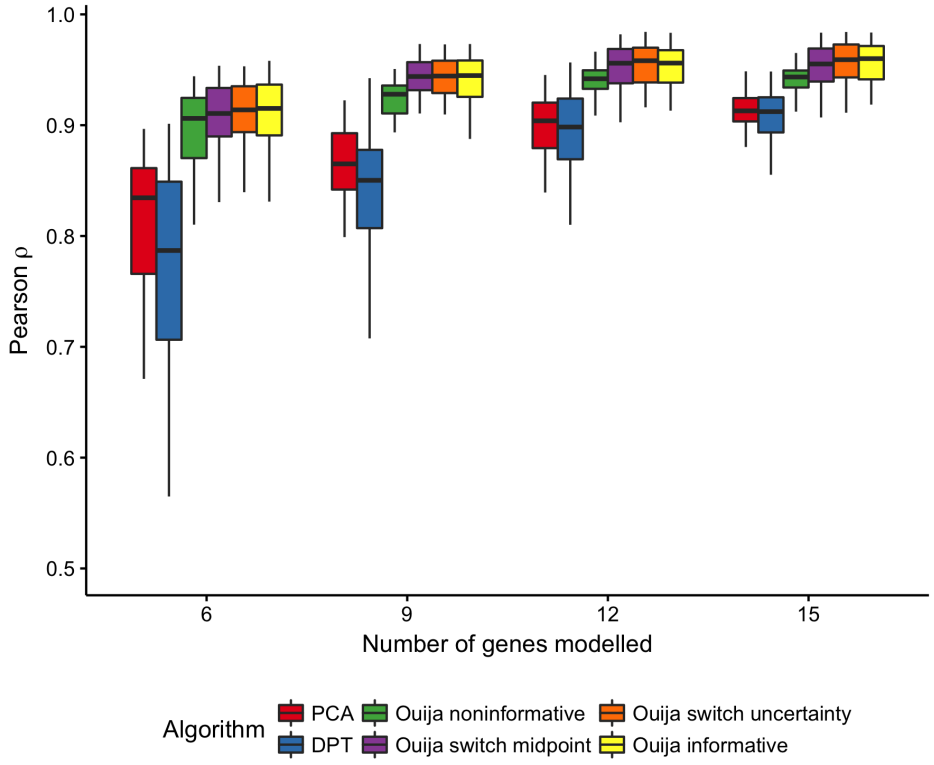


Figure 29: Correlation to true pseudotime across four gene set sizes and 40 replications for the sigmoidal mean function. Ouija was benchmarked with four configurations: noninformative priors, noninformative on the switch time but informative on activation strength (*switch midpoint*), informative on activation strength and weakly informative on switch time (*switch uncertainty*), and informative on both parameters. As the priors converge to the true values the accuracy of pseudotime inference increases.

strength (k) improves the correlation with the true pseudotimes, even if the prior on $t^{(0)}$ is noninformative. In other words, the incorporation of prior information is fairly robust to uncertainty in prior knowledge of the switch times. Furthermore, as the prior knowledge in the switch times approaches the "truth" (i.e. as we move from specifying the switch time to be the midpoint of pseudotime, to near the true values, to the true values), the correlation to the true pseudotimes increases.

3.6 ROBUSTNESS TO TRANSIENT GENE BEHAVIOUR

Finally, a potential limitation of our switch behaviour model is that it assumes that all selected marker genes follow a strict monotonically increasing (or decreasing) behaviour and there exists a smooth, non-transient transition from an initial cell expression state to a final resting state at the end of the pseudotemporal period. In selecting a marker gene panel the investigator may not always be fully certain of the quantitative behaviour of all genes and some may indeed exhibit a transient rather than switch behaviour. In order to test of the impact of such genes on Ouija we performed a simulation study to assess the effect of genes exhibiting transient behaviour on pseudotime estimation. We did this by simulating panels of single cell expression values that mimic the zero-inflated properties observed in real data for a variety of genes containing mixed numbers of switch-like and transient gene behaviours (figure 30A). Specifically we considered two scenarios, the first in which the numbers of switch-like and transient genes are equal and the second in which three-quarters of the gene panels were switch-like genes and the reminder transient (Figure 30B). Our simulation study showed that if the switch-like genes remained the dominant class of genes in the simulated marker gene panels, it remained possible to accurately infer pseudotime in the presence of transient genes (figure 30C). Furthermore, as the size of the panel increases, the absolute number of switch genes was a greater determinant of pseudotime estimation accuracy than the proportion of the marker gene panel that was truly switch-like.

3.7 OUIJA CLUSTERS CELL TYPES BASED ON PSEUDOTIME CONTINUITY

We further investigated the single cell expression data from a study tracking the differentiation of embryonic precursor cells into haematopoietic stem cells (HSCs) [148]. The

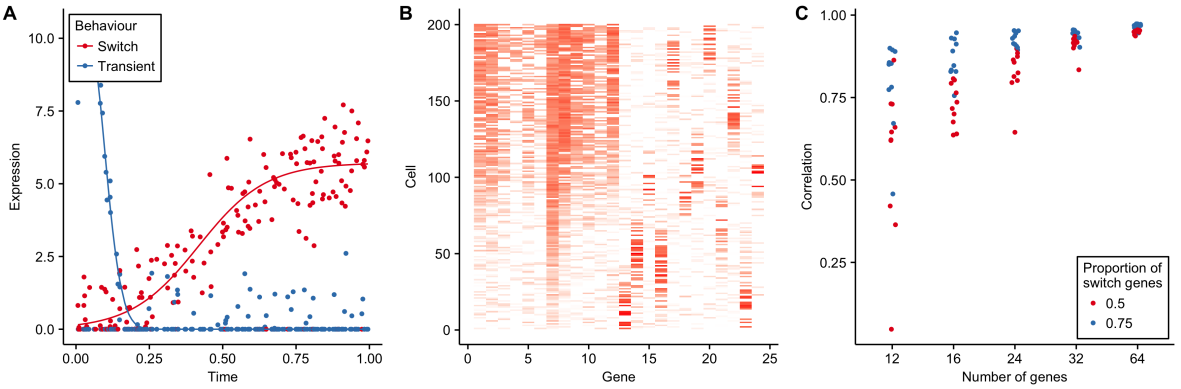


Figure 30: Impact of marker genes exhibiting transient rather than switch-like gene behaviours. **A** Examples of simulated switch and transient expression genes. **B** Example 24-gene panel with 12 switch and 12 transient genes. **C** Correlation between true and estimated pseudotime as a function of the gene panel size and proportion of switch genes for ten replicates.

cells begin as haemogenic endothelial cells (ECs) before successively transforming into pre-HSC and finally HSC cells. The authors identified six marker genes that would be down-regulated along the differentiation trajectory, with early down-regulation of *Nrp2* and *Nr2f2* as the cells transform from ECs into pre-HSCs, and late down-regulation of *Nrp1*, *Hey1*, *Efnb2* and *Ephb4* as the cells emerge from pre-HSCs to become HSCs. The study investigated a number of distinct cell types at different stages of differentiation: EC cells, T1 cells ($CDK45^{-}$ pre-HSCs), T2 cells ($CDK45^{+}$ pre-HSCs) and HSC cells at the E12 and E14 developmental stages.

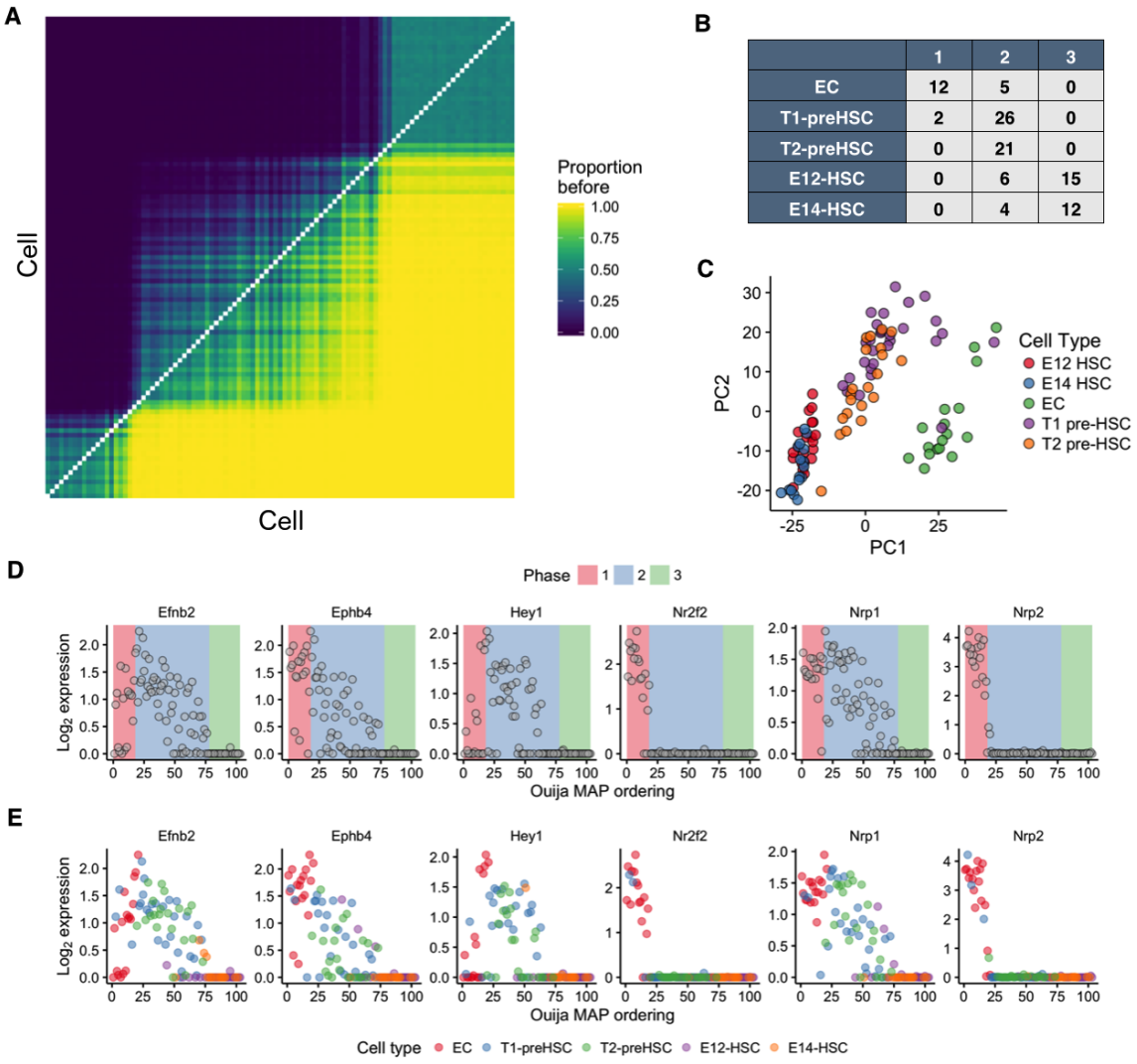
We conducted a pseudotime analysis using Ouija on the 105 cells featured in the original experiment to investigate whether the inferred pseudotime progression could recapitulate the known cell types in the study and their known relationships in an unsupervised analysis using just these six marker genes alone. As Ouija uses a probabilistic model and inference we were able to obtain a posterior ordering matrix (figure 31A) where the entry in the i^{th} row and j^{th} column is the proportion of times cell i was ordered before cell j in the MCMC posterior traces (thus giving a posterior probability under the joint model that $p(t_i < t_j)$. When cells are ordered by the expected pseudo-

time, this posterior matrix contained three metastable groups of cells corresponding to endothelial, pre-HSCs and HSCs respectively (figure 31B). Misclassifications within cell types (T1/T2 and E12/E14 cells) could be explained by examining a principal components analysis of the global expression profiles (figure 31C) which suggests that these cell types are not completely distinct in terms of expression.

When examining the inferred pseudotime progression of each marker gene (figure 31D), these three metastable states corresponded to the activation of all genes at the beginning of pseudotime time, the complete inactivation of all the marker genes at the end of the pseudotime and a intervening transitory period as each marker gene turns off. Each metastable state clearly associates with a particular cell type (figure 31E). As expected, *Nrp2* and *Nr2f2* exhibited early down-regulation and *Nrp1*, *Hey1*, *Efnb2* and *Ephb4* all exhibited late down-regulation. Using this HSC formation system as a proof-of-principle it is evident that, if a small number of switch-like marker genes are known, it is possible to recover signatures of temporal progression using Ouija and that these trajectories are compatible with real biology.

3.8 CELL CYCLE PREDICTION AS A PSEUDOTIME ESTIMATION PROBLEM

We wanted to consider a study composed of a large panel of marker genes and identified a single-cell RNA-seq study [66] that examined variation between individual hematopoietic stem and progenitor cells from two mouse strains (C57BL/6 and DBA/2) as they age. Principal component analysis for each cell type and age showed a striking association of the top principal components with cell cycle-related genes (figure 32A), indicating that transcriptional heterogeneity was dominated by cell cycle status. They scored each cell for its likely cell cycle phase using signatures based on functional annotations [40] and profiles from synchronized HeLa cells [140] for the G1/S, S, G2, and G2/M phases.



	1	2	3
EC	12	5	0
T1-preHSC	2	26	0
T2-preHSC	0	21	0
E12-HSC	0	6	15
E14-HSC	0	4	12

Figure 31: Pseudotime ordering and cell type identification of haematopoeietic stem cell differentiation. **A** Consistency matrix of pseudotime ordering. Entry in the i^{th} row and j^{th} column is the proportion of times cell i was ordered before cell j in the MCMC posterior traces. Gaussian mixture modelling on the first principal component of the matrix identified three clusters that are evident in the heatmap. **B** Confusion matrix for cell types identified in original study (rows) and Ouija inferred (columns). Ouija inferred cluster 1 largely corresponds to EC cells, cluster 2 corresponds to pre-HSC cells while cluster 3 corresponds to HSC cells. **C** PCA plot similar to the original publication [148] suggests supports the existence of three distinct cell types in the data. **D** HSC gene expression as a function of pseudotime ordering for six marker genes. Background colour denotes the maximum likelihood estimate for the Ouija inferred cell type in that region of pseudotime. **E** HSC gene expression as a function of pseudotime ordering for six marker genes with cells coloured by known cell type.

We investigated if Ouija could be used to identify cell cycle phase, treating the inferential problem as a continuous pseudotime process. We applied Ouija to 1,008 C57Bl/6 HSCs using 374 GO cell cycle genes that satisfied gene selection criteria used in the original study. The estimated pseudotime progression given by Ouija recapitulates the trajectory observed in principal component space. The estimated pseudotime distribution correlates well with the cell cycle phase categorisation given in the original study (figure 32C). Furthermore, we identified 88 genes with large (negative) activation strengths indicating strong switch (off) behaviour (figure 32D) ordering the cells according to pseudotime and then ordering by activation time shows a cascade of expression inactivation across these 88 genes over cell cycle progression with the quiescent (G_0) indicated by complete inactivation of all 88 genes (figure 32E,F). The explicit parametric model assumed by Ouija makes this gene selection and ordering process simple and *quantitative* compared to a non-parametric approach that would require some retrospective analysis or visual inspection.

This investigation indicates that although Ouija makes strong assumptions about gene-specific expression behaviour, its utility is not limited to small marker gene panels that strictly obey its switch-like behavioural assumptions.

3.9 DISCUSSION

We have developed a novel approach for pseudotime estimation based on modelling switching expression behaviour over time for marker genes. Our strategy provides an orthogonal and complimentary approach to unsupervised, whole transcriptome methods that do not explicitly model any gene-specific behaviours and do not readily permit the inclusion of prior knowledge.

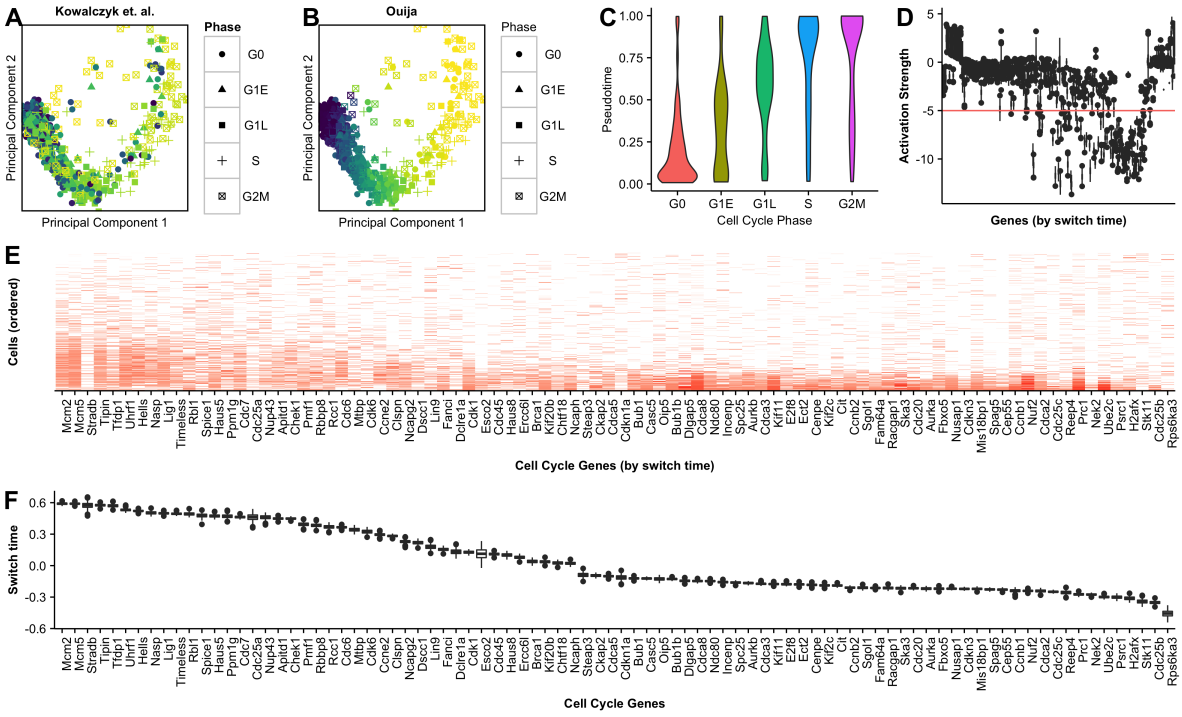


Figure 32: Cell cycle phase prediction. Principal component representation of hematopoietic stem cells coloured according to **A** the original cell cycle progression score [66] and **B** Ouija - cell cycle classes indicated are based on original study classifications. **C** Distribution of Ouija inferred pseudotime versus the original cell cycle classifications. **D** Estimated activation strengths for the 374 cell cycle gene panels. **E** Gene expression profile for 88 switch-like genes with cells ordered by pseudotime and **F** genes ordered by activation time.

We demonstrate that the selection of a few marker genes allows comparable pseudotime estimates to whole transcriptome methods on real single cell data sets. Furthermore, using a parametric gene behaviour model and full Bayesian inference we are able to recover posterior uncertainty information about key parameters, such as the gene activation time, that allows us to explicitly determine a potential ordering of gene (de)activation events over (pseudo)time. The posterior ordering uncertainty can also be used to identify homogeneous phases of transcriptional activity that might correspond to transient, but discrete, cell states.

Although our focus was on switching expression behaviour, alternative parametric functions could be used to capture other gene behaviours. However, it is critical to recognise that in a latent variable modelling framework such as this, prior information has a strong influence over the final outcome. Therefore any constraints should match *a priori* knowledge of the marker genes under investigation.

In summary, Ouija provides a novel contribution to the increasing plethora of pseudotime estimation methods available for single cell gene expression data.

4

MODELLING BIFURCATIONS WITH A BAYESIAN MIXTURE OF FACTOR ANALYSERS

4.1 INTRODUCTION

So far we have focussed on the *single-trajectory* case, where cells have a single fate and progress uniformly towards it. Chapter 2 considered probabilistic inference of trajectories in reduced-dimension space, while chapter 3 extended such ideas to generatively model the expression of a small set of marker genes.

However, often cells undergo some fate decision and their expression programmes bifurcate into two or more end points. Examples include progenitor cells that differentiate into distinct cell types or stressed cells that may either recover or trigger apoptosis.

Several methods have been proposed to infer bifurcation structure from single-cell data. Wishbone [117] constructs a k-nearest neighbour graph and uses shortest paths from a *root* cell to define pseudotimes, using inconsistencies over multiple paths to detect bifurcations. Diffusion Pseudotime (DPT) [46] similarly constructs a transition matrix where each entry may be interpreted as a diffusion distance between two cells. Bifurcations are inferred by identifying the anticorrelation structure of random walks from both a root cell and its maximally distant cell. While DPT arguably has a probabilistic interpretation, neither method specifies a fully generative model that incorporates measurement noise, while both infer bifurcations after constructing pseudotimes. A further algorithm Monocle [108] learns pseudotimes based on dimensionality reduction using the DDRTree algorithm [84] and provides post-hoc inference of genes involved in the bifurcation process using generalised linear models.

Consequently, we we propose a Bayesian hierarchical mixture of factor analysers for inferring bifurcations from single-cell data. Since developmental bifurcations involve two related processes it is therefore natural to extend such models to involve a mixture of two factor analysers in a Bayesian hierarchical setting that relates expression patterns between branches.

The model we propose is unique compared to existing bifurcation inference methods in the following: (1) by specifying a fully generative probabilistic model we incorporate measurement noise into inference and provide full uncertainty estimates for all parameters, (2) we simultaneously infer cell "pseudotimes" and branching structure as opposed to post-hoc branching inference as is typically performed, and (3) our hierarchical shrinkage prior structure automatically detects which features are involved in the bifurcation, providing statistical support for detecting which genes drive fate decisions.

In this chapter, we introduce our model and apply it to both a synthetic datasets and demonstrate its consistency with existing algorithms on real single-cell data. We further propose a zero-inflated Empirical-Bayes-like variant that takes into account zero-inflation and quantify the levels of dropout at which such models are beneficial. We highlight the multiple natural solutions to bifurcation inference when using gene expression data alone and finally discuss both the merits and drawbacks of using such a unified probabilistic model.

4.2 METHODS

4.2.1 *Statistical model*

We begin with an $N \times G$ matrix of suitably normalised[1] gene expression measurements for N cells and G genes, where $\mathbf{y}_n$ denotes the n^{th} row vector corresponding to the expression measurement of cell n. We assign a pseudotime t_n to each cell along with a binary variable γ_n indicating to which of B branches cell n belongs:

$$\gamma_n = b \text{ if cell } n \text{ on branch } b \tag{54}$$

with $b \in 1, \ldots, B$.

The pseudotime t_n is a surrogate measure of a cell's progression along a trajectory while it is the behaviour of the genes - given by the factor loading matrix - that changes between the branches. We therefore introduce B factor loading matrices $\Lambda_b = [\mathbf{c}_b \ \mathbf{k}_b], \ b \in 1, \ldots, B$ for each branch modelled. Here, $\mathbf{c}_b$ represents the gene-specific intercepts of expression while $\mathbf{k}_b$ can be thought of as the gene-specific gradient along pseudotime.

The likelihood of a given cell's gene expression measurement conditional on all the parameters is then given by

$$\mathbf{y}_n | \gamma_n, \Lambda_{\gamma_n}, t_n, \boldsymbol{\tau} \sim \mathcal{N}(\mathbf{c}_{\gamma_n} + \mathbf{k}_{\gamma_n} t_n, \boldsymbol{\Sigma}) \tag{55}$$

where $\boldsymbol{\Sigma} = \text{diag}(\tau_1, \ldots, \tau_G)$ and $\boldsymbol{\tau}$ is a G-length vector of measurement precisions.

We seek a prior structure on the loading matrix that (a) encourages the behaviour of genes to be identical across branches while (b) identifying the subset of genes that

1 Such as $\log(\text{TPM} + 1)$ or $\log(\text{FPKM} + 1)$.

deviate from this and are differentially regulated across the branches. It is therefore reasonable that the factor loading gradients $\mathbf{k}_\gamma$ should be similar to each other unless the data suggests otherwise. We therefore place a prior of the form

$$[\mathbf{k}_{\gamma_n}]_g \sim \mathcal{N}(\theta_g, \chi_g^{-1}) \tag{56}$$

where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_g)$ denotes a common factor gradient across branches. This has similar elements to Automatic Relevance Determination (ARD) models with the difference that rather than shrinking regression coefficients to zero to induce sparsity we shrink factor loading gradients towards a common value to induce similar behaviour between mixture components. We can then inspect the posterior precision to identify genes involved in the bifurcation: if χ_g is very large then the model is sure that $k_{0g} \approx k_{1g}$ and gene g is not involved in the bifurcation; however, if χ_g is relatively small then $|k_{0g} - k_{1g}| \gg 0$ and the model indicates that g is involved in the bifurcation[2].

2 Note that we only place this prior structure on k as the intercepts c can be identical but the genes still exhibit differential regulation through different ks.

With these considerations the overall model becomes

$$
\begin{aligned}
\boldsymbol{\omega} &\sim \text{Dirichlet}(1/B, \ldots, 1/B) \\
\gamma_n &\sim \text{Categorical}(\boldsymbol{\omega}) \\
\eta &\sim \mathcal{N}(\tilde{\eta}, \tau_\eta^{-1}) \\
\theta_g &\sim \mathcal{N}(\tilde{\theta}, \tau_\theta^{-1}) \\
\chi_g &\sim \text{Gamma}(\alpha_\chi, \beta_\chi) \\
\mathbf{c}_{\gamma_n} &\sim \mathcal{N}(\eta, \tau_c^{-1}) \\
\mathbf{k}_{\gamma_n} &\sim \mathcal{N}(\boldsymbol{\theta}, \boldsymbol{\chi}^{-1}\mathbb{1}_G) \\
t_n &\sim \mathcal{N}(0, 1) \\
\boldsymbol{\tau} &\sim \text{Gamma}(\alpha, \beta) \\
\mathbf{y}_n &\sim \mathcal{N}(\mathbf{c}_{\gamma_n} + \mathbf{k}_{\gamma_n} t_n, \boldsymbol{\tau}^{-1}\mathbb{1}_G)
\end{aligned}
\tag{57}
$$

where $\tilde{\eta}$, $\tilde{\theta}$, τ_η, τ_θ, τ_c, α_χ, β_χ, α and β are hyperparameters fixed by the user. By default we set the noninformative prior $\alpha_\chi = \beta_\chi = 10^{-2}$ to maximise how informative the posterior of $\boldsymbol{\chi}$ is in identifying genes that show differential expression across the branches.

4.2.2 *Inference*

As the model exhibits complete conditional conjugacy, inference was performed using Gibbs sampling. Briefly, Gibbs sampling is a Markov Chain Monte-Carlo (MCMC) algorithm for drawing a sequence of T samples $\boldsymbol{\theta}^1, \ldots, \boldsymbol{\theta}^T$ that approximate some target distribution $p(\boldsymbol{\theta}|x)$ that is analytically intractable, i.e. for which we cannot write down an exact expression. Given the parameter vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_P)$, Gibbs sampling requires analytical expressions (or at least the ability to quickly generate samples from)

the conditional distributions $p(\theta_p|\boldsymbol{\theta}_{-p}, x)$ where $\boldsymbol{\theta}_{-p}$ is the set of parameters other than p. Gibbs sampling then proceeds by consecutively drawing

$$\theta_p^{t+1} \sim p(\theta_p|\boldsymbol{\theta}_{-p}^t, x) \tag{58}$$

for all variables $p = 1, \ldots, P$ up to some predefined number of iterations T. The resulting samples $\boldsymbol{\theta}^1, \ldots, \boldsymbol{\theta}^T$ asymptotically converge in the limit $T \to \infty$ to the target distribution $p(\boldsymbol{\theta}|x)$.

Analytical expression for the conditional Gibbs updates for our `mfa` model are given in appendix D. This is implemented in the `R` package `mfa` available at `http://www.github.com/kieranrcampbell/mfa`. `R` is particularly slow for calculating such Gibbs updates as frequent subsetting of assignment vectors occurs. Therefore, computation of the updates was implemented in `C++` and linked to the `R` package using `Rcpp`, leading to orders of magnitude speed up in the calculation of some quantities.

4.2.3 *Modelling zero-inflation*

Single-cell RNA-seq data is known to exhibit *dropout*, where lowly expressed genes register as zero counts. Several computational methods attempt to correct for this by modelling the observed expression as a mixture of a point-mass at zero representing dropout and an *amplified* component representing true expression. However, the probability of measuring a zero is not constant but is inversely proportional to the latent expression, as the smaller the input quantity of mRNA the higher the probability of a failure of reverse-transcription.

As a solution to this, several methods model a dropout probability dependent on the latent expression. For example, SCDE [59] is a statistical model for single-cell differential

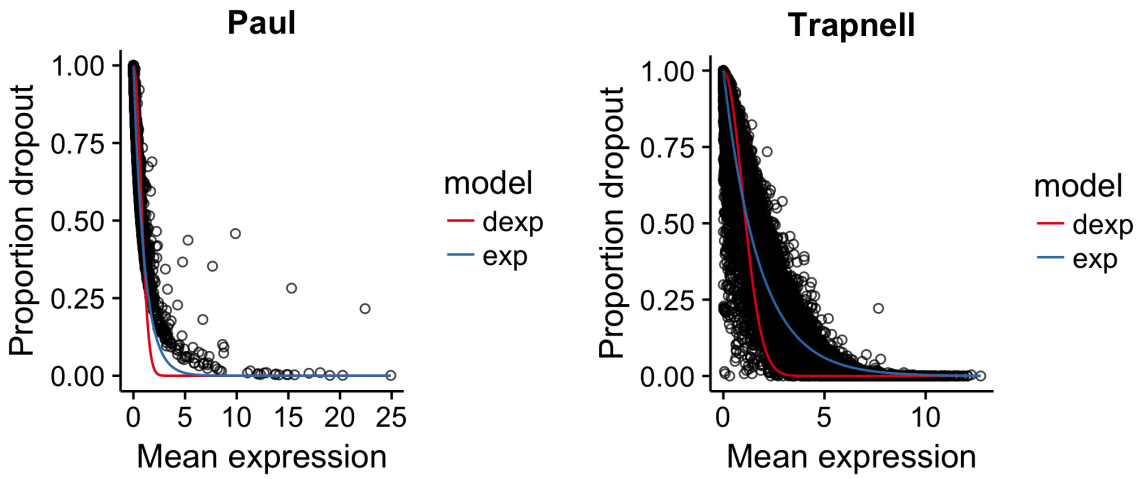


Figure 33: Dropout relationships in single-cell RNA-seq data for two scRNA-seq datasets. An exponential dropout model (blue) empirically fits the data better than double exponential (red).

expression which models the dropout probability of a gene as a logistic regression on the latent expression as part of a mixture-of-experts model. In a further example, ZIFA [102] proposes a double-exponential dropout model where $p(\text{dropout}) \propto \exp(-\lambda x^2)$ and x is the latent expression and λ is a constant dropout parameter.

Therefore, accounting for zero-inflation in our mixture-of-factor analysers model is equivalent to modifying the likelihood to a mixture of an amplified component and dropout component depending on the latent expression. However, the difficulty here is that we must sample from the conditional distribution $p(\lambda|\cdot)$ (where $\cdot$ is all parameters and data other than λ), which to the best of our knowledge does not exist analytically.

As a solution to this we propose an Empirical-Bayes like procedure to estimate λ globally then infer the latent expression x through further Gibbs updates. First we note that a single exponential dropout empirically fits the dropout relations in single-cell RNA-seq datasets better than the double exponential dropout (figure 33).

We subsequently modify the likelihood to give a per-gene dropout probability of

$$p(\text{dropout in gene } g) = \exp(-\lambda \sum_{n=1}^{N} x_{ng}) \tag{59}$$

which depends on the mean latent expression level of the gene. While this is of course an approximation and we expect the probability of a dropout to be specific to each gene in each cell depending on the latent expression, this allows us to estimate λ by fitting the maximum likelihood exponential curve of the proportion of cells a gene is expressed in against the mean expression level (similar to figure 33) using the `R` function `nls`. Thus the modified likelihood becomes

$$\begin{aligned}
\mathbf{x}_n &\sim \mathcal{N}(\mathbf{c}_{\gamma_n} + \mathbf{k}_{\gamma_n} t_n, \mathbf{\Sigma}) \\
h_{ng} &\sim \text{Bernoulli}(\exp\left(-\frac{\lambda}{N}\sum_{n'} x_{n'g}\right)) \\
y_{ng} &= \begin{cases} x_{ng} & \text{if } h_{ng} = 0 \\ 0 & \text{if } h_{ng} = 1 \end{cases}
\end{aligned} \tag{60}$$

Note that since h_{ng} is effectively observed we only need to Gibbs sample x_{ng} for which $h_{ng} = 1$. The conditional distribution for x_{ng} is then given by

$$x_{ng}|\cdot \sim \mathcal{N}\left(\mu_{ng} - \frac{\lambda}{N\tau_g}, \tau_g^{-1}\right) \tag{61}$$

where μ_{ng} is defined as above.

While incorporating zero-inflation in the likelihood leads to a less-mispecified model we must perform inference on an additional N_0 parameters, where N_0 is the number of zero measurements in the expression matrix. For single-cell RNA-seq data this can be as high as 90% of all measurements leading to a significant additional computational burden.

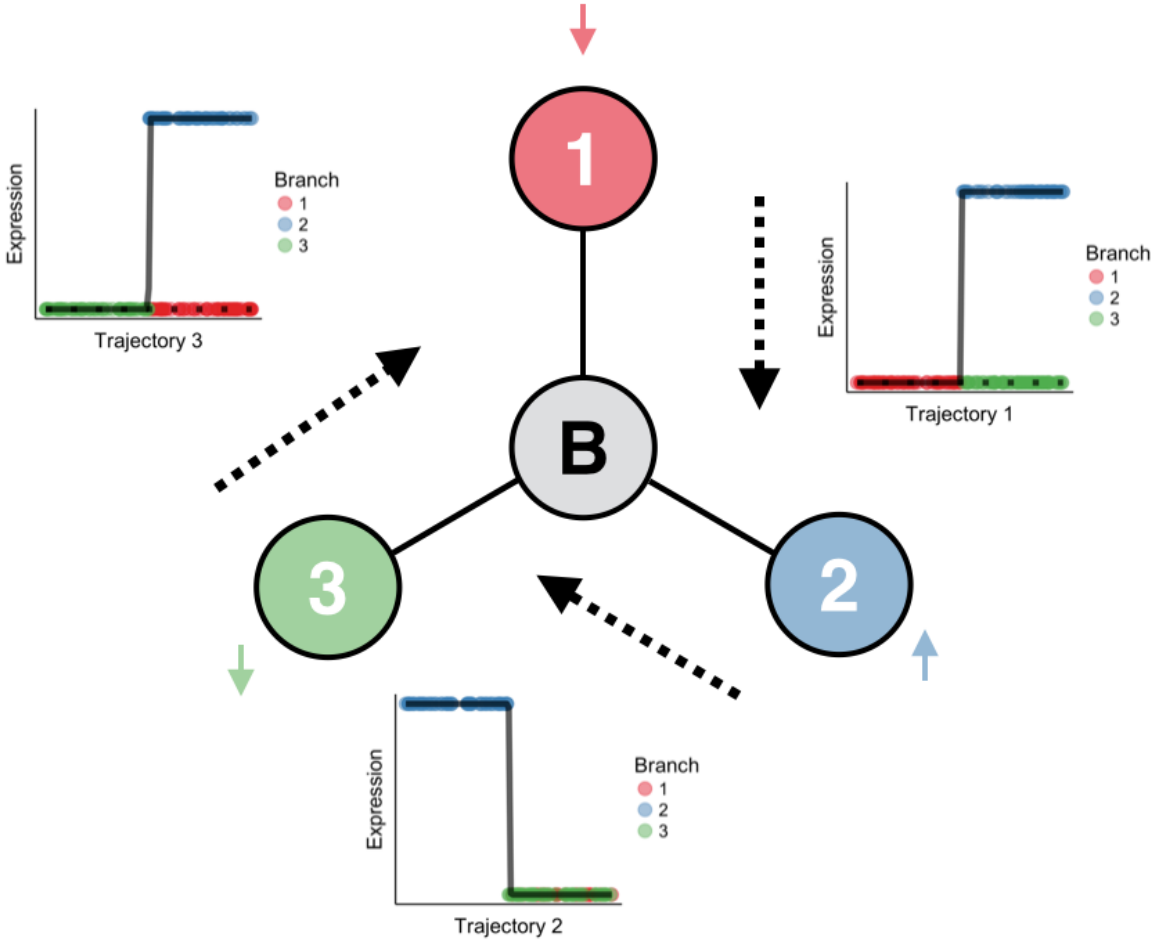


Figure 34: Multiple solutions to bifurcation inference. Starting with three cell states we would like to infer a bifurcation process from one to the other two. If a single gene is upregulated in one of the states yet downregulated in the other two then clearly any state may act as the beginning of the trajectory. For example, if we start in state 1 then the gene is upregulated along state 2 and stays constant in state 3; if we start in state 2 then the gene is downregulated in states 1 & 3; if we start in state 3 then the gene is upregulated in state 2 and remains downregulated in state 1. However, due to the nonidentifiability this is true if we add additional genes that are upregulated in one or two of the cell states. The equivalent geometric argument is that we can build the transcriptomic profiles across all genes by spinning the figure about **B** (with possible inversion) and "adding" that gene. No matter how many additional genes we add, any one of the three states can act as the root state or beginning of pseudotime. Therefore, in the absence of any additional information there are always three equally valid solutions to bifurcation inference from gene expression data alone.

4.3 MULTIPLE SOLUTIONS TO BIFURCATION INFERENCE

It is common in bifurcation inference methods to specify additional information aside to gene expression data alone. For example, Wishbone requires the specification of a *root* cell that signifies the beginning of pseudotime. DPT also allows for the specification of a root cell or picks the furthest from a random cell if unspecified. Monocle equivalently allows re-fitting of the pseudotimes with the constraint that one of the inferred "states" is the initial or root state.

We argue that such requirements are necessary due to a fundamental invariance in the gene expression of bifurcating cells. Figure 34 shows a conceptual model of three end-states (1-3) and a gene which is expressed in one end state (2) but not the others. We can envisage three possible bifurcation routes here: state 1 is the initial state that bifurcates to 2 & 3 ($1 \rightarrow 2, 3$), or equivalently $3 \rightarrow 1, 2$ or $2 \rightarrow 1, 3$. If 1 or 3 is the initial state then the gene exhibits differential expression across the branches, while if we start at 2 the gene exhibits concordant expression across the branches. Note that for a bifurcation we require some genes that show differential expression between the branches and some that show concordant expression - lacking the former would give a non-branching trajectory and lacking the latter would give separate cell types.

The above reasons that in a single-gene case the initial state is indistinguishable from the gene expression alone. We can easily generalise this to the multiple-gene case due to the fact that the labels in figure 34 are statistically nonidentifiable. The equivalent geometric argument is that you can 'spin' figure 34 about $\mathbf{B}$ for each gene (and optionally invert the expression to give two states of non-zero expression).

While in algorithms such as Wishbone and DPT this non-identifiability is solved by setting an initial cell or state, the equivalent in our model is the correct initialisation of the pseudotimes. Principal component analysis is applied to the data before inference

and the principal component that best corresponds to the trajectory based on the expression of known genes is used to initialise the pseudotimes. Such trajectories correspond to local modes in the posterior space that are sufficiently narrow the probability of the Gibbs sampler moving to another is negligible. A future extension that would solve this non-identifiability would involve placing priors on the behaviour of certain genes across the branches, which combined with more efficient inference would pick out the 'true' trajectory.

4.4 RESULTS

4.4.1 *Synthetic datasets*

4.4.1.1 *Generating of synthetic datasets*

Synthetic datasets were generated for various simulations throughout the analysis. Rather than simply generating data from the model we attempted to create synthetic data that was as close to real single-cell data as feasible, meaning the synthetic data is severely misspecified with respect to our model.

The first consideration is the functional form of gene expression along pseudotime. A linear assumption is fundamentally unrealistic as the gene expression cannot go to $\pm\infty$ as pseudotime progresses. Consequently we adopt sigmoidal expression across pseudotime (previously suggested in [18, 22] and discussed in chapters 2-3), parameterised[3] by the half-peak expression ϕ, the *switch-time* δ and the switch-strength k:

$$\text{Sigmoid}(t, k, \phi, \delta) = \frac{2\phi}{1 + \exp(-k(t - \delta))}. \tag{62}$$

3 We depart from the previous notation here to avoid index hell when referencing multiple branches.

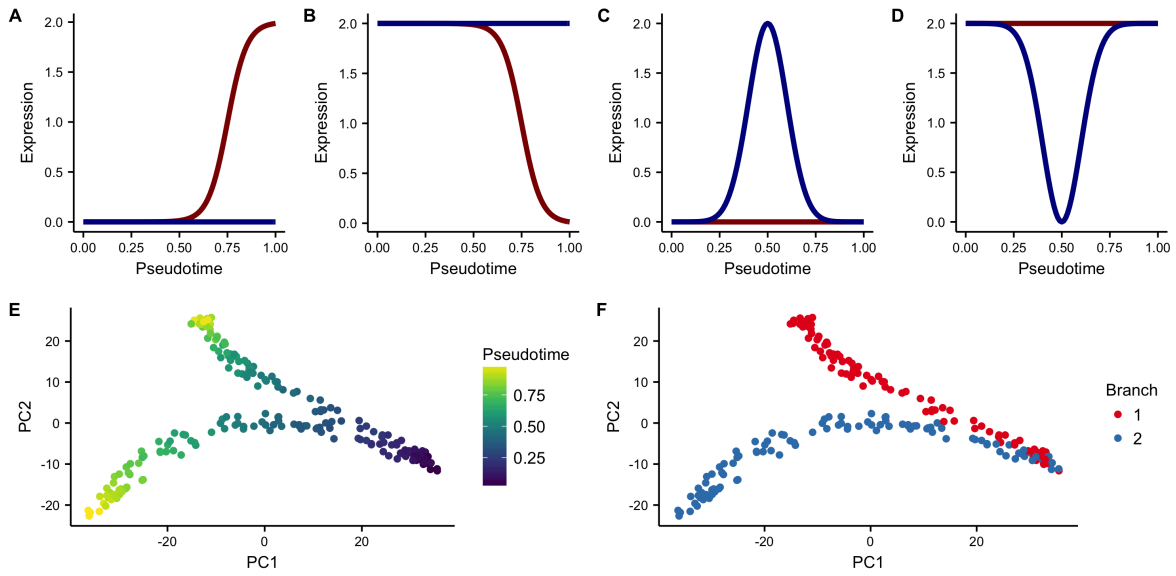


Figure 35: Generation of synthetic data to test `mfa`. Examples of diverging monotonic behaviour across branches (**A** - **B**) and diverging transient behaviour (**C** - **D**), with resulting PCA representations coloured by both pseudotime (**E**) and branch allocation (**F**).

For half of the G genes we assume the expression along the two branches is the same, thus we model common k, ϕ and δ parameters. For the second half we assume the expression diverges, and in particular for each gene $k = 0$ for one of the branches, which we can call b_0, and b_1 for the branch for which $k \neq 0$. If k on the other branch is positive (ie $k_{gb_1} > 0$) then we set the half-peak expression on b_0 to zero, as the genes must both start at 0, and if one turns on the other must remain off (figure 35A). Alternatively, if $k_{gb_1} < 0$ then the genes must begin in an *on* state and switch off for cells on b_1 (figure 35B). Thus we set ϕ_{gb_0} to twice its original value. For any gene that shows divergent behaviour across branches we set δ to be in the second half of the trajectory. We then construct the mean, using the sigmoid function, and generate the data from a Gaussian noise model ensuring any negative values are set to zero. This gives the characteristic bifurcation pattern in PCA space as seen in figures 35E&F.

We may also wish to generate transiently expressed genes to test the limits of the monotonicity assumptions in our model. Transient behaviour can either be across both

branches or exhibit divergent behaviour (transient on one branch only). To simulate transient genes we swap out the sigmoidal mean function for a Gaussian function centred around the mid-point of the trajectory:

$$\text{Transient}(t, l, s) = \exp(-\frac{1}{2s}(t-l)^2) \tag{63}$$

We additionally ensure the behaviour is constrained to be identical on each branch at the beginning and end of the trajectory. Examples of such behaviour may be seen in figures 35C&D.

If we would like to incorporate zero-inflation, we calculate a dropout probability for each measurement via $p_{ng} = \exp(-\lambda x_{ng})$, where λ is set to a reasonable value based on observations of real datasets. Note that our model is mis-specified with respect to this as it models a per-gene dropout probability p_g.

4.4.1.2 *Performance on toy dataset*

We first demonstrate our method on a synthetic 'toy' dataset of 300 bifurcating cells and 60 genes, half of which exhibit differential behaviour across the bifurcation and half of which show similar behaviour. Pseudotimes were inferred using Gibbs sampling for 10^5 iterations. PCA representations of the synthetic data can be seen in figures 36A&B showing the characteristic Y shape associated with bifurcating data, coloured by both maximum aposteriori (MAP) pseudotime and branch assignment estimates respectively. We compared the Pearson correlation of the estimated pseudotimes to the true pseudotimes (figure 36C) for both MFA, PC1 (the first principal component of the data), Monocle and Diffusion Pseudotime, giving values of 0.98, 0.98, 0.98 and 0.99 (to 2 s.f.) respectively. Broad benchmarking of pseudotime algorithms to "ground-truth" data is difficult due to the inherent assumptions that are necessary about how

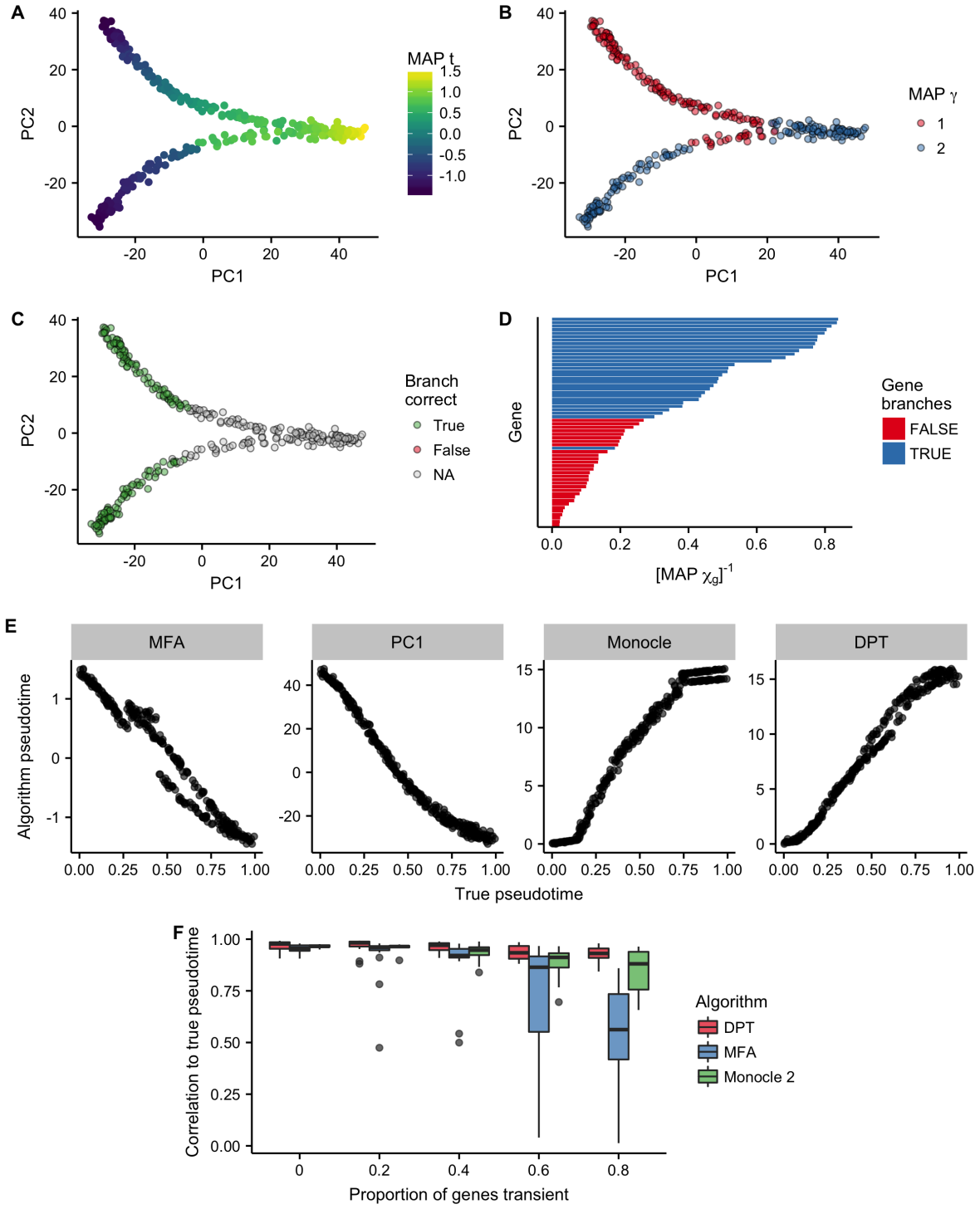


Figure 36: Caption next page.

Figure 36: Probabilistic inference of bifurcations in synthetic data. **A** PCA representation of a toy dataset for 300 cells and 60 genes, coloured by the maximum aposteriori (MAP) pseudotime estimates. **B** Equivalent representation as **(A)** colour by the MAP branch estimate. **C** Equivalent representation showing whether each branch was assigned correctly. Due to the non-identifiability of mixture components, we map component indices from true to inferred such that the agreement is maximised. **D** The inverse MAP estimates of χ largely identify which genes in the dataset exhibit different behaviour across the two branches. **E** Comparison of different pseudotime inference algorithms to the ground truth pseudotime on this particular dataset. The algorithms MFA, PC1 (principal component 1), Moncole and DPT had correlations of 0.98, 0.98, 0.98, 0.99 (to 1 s.f.) respectively. **F** The correlation of inferred pseudotimes to ground truth depending on the proportion of genes in the dataset exhibiting transient behaviour. MFA shows competitive performance up to around 40% of genes begin transient despite an inherent linear assumption.

genes expression evolves along trajectories. However, such toy examples demonstrates the consistency of multiple algorithms on our toy dataset.

4.4.1.3 *Impact of transient expression*

One weakness of our model is that it assumes gene expression changes as a linear function of time. This allows us to perform fast conjugate Gibbs sampling but is highly unrealistic for real data. The synthetic data generated is based on sigmoidal changes across pseudotime, which being nonlinear is already mildly mis-specified with respect to our model. However, genes may also exhibit transient behaviour, in which they are briefly down- or up-regulated before returning to their initial state. We sought to quantify the robustness of MFA to transient gene expression by performing extensive simulations. Specifically, we generated synthetic datasets with 0%, 20%, ..., 80% of genes exhibiting transient expression, and inferred the pseudotimes using DPT, MFA and Monocle 2. This was repeated 20 times for each percentage of transient genes. The results can be seen in figure 36F. The performance of MFA remains competitive up to around 40% of genes

exhibiting transient expression, after which DPT and Monocle 2 perform significantly better. However, MFA is highly consistent with DPT and Monocle 2 on the two real datasets examined (figures 38 & 39) implying the occurrence of transient expression is limited enough in practice for the linearity assumption to be feasible.

One notable difference between MFA and existing bifurcation inference algorithms is in the pre-bifurcation branch assignment. Algorithms such as Wishbone and DPT will assign a separate branch to cells preceding the bifurcation. However, MFA will typically assign pre-bifurcation cells to one of the two branches modelled, with the other branch beginning at the bifurcation. A bifurcation process consists of two temporal processes that have a common origin but differing end points. Thus, due to non-identifiability, cells pre-bifurcation can equally be said to be on one branch with the second beginning at the bifurcation point. Importantly, no matter how we assign the branches under this regime the observed behaviour of genes as a function of both pseudotime and branch assignment will be consistent, which is necessary for biological insight.

4.4.2 *Benefits of modelling zero-inflation*

Single-cell RNA-seq data is known to exhibit *dropout*, where a failure to reverse transcribe lowly-expressed mRNA results in zero counts. However, a zero count for a particular gene in a particular cell may also be a *true zero* where no mRNA in the cell is present, which we expect to be useful for pseudotime inference. Figure 37A shows a conceptual model where a gene is upregulated along pseudotime with two cells exhibiting dropout. The true zeros (in blue) help pseudotime inference as the low-expression implies they are at the beginning of pseudotime. However, the cells exhibiting dropout (in red) would potentially impede pseudotime inference as MFA would order them with the true zero cells at the beginning of the trajectory.

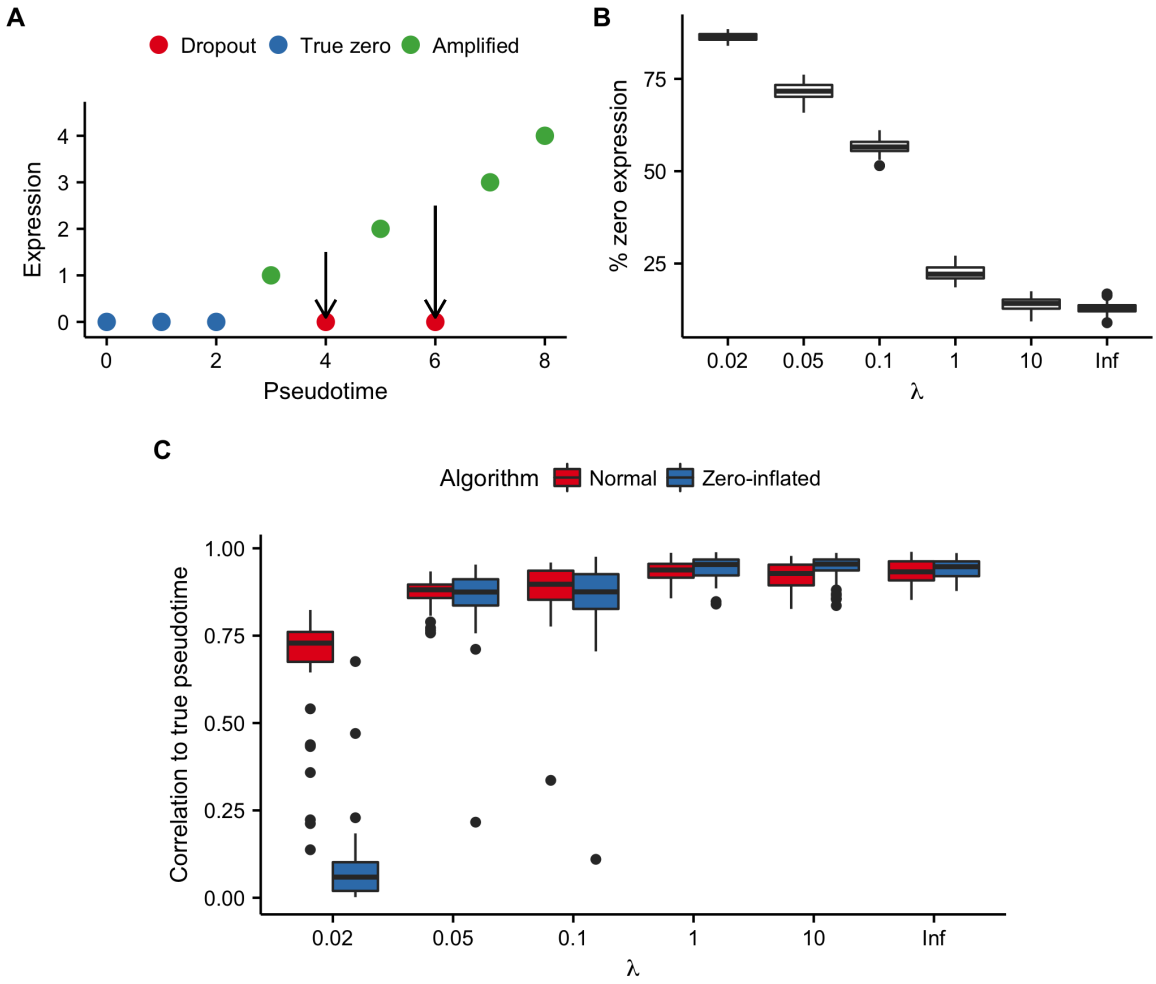


Figure 37: The effects of modelling zero-inflation. **A** Zero counts observed in single-cell RNA-seq data may be attributed to either *true zeros* where no mRNA of a given gene is produced in a cell, or *dropout* where there is a failure to reverse-transcribe the low levels of starting material. Alternatively, a count is registered and the gene is *amplified*. In theory not accounting for dropouts will reduce the accuracy of pseudotime inference - the two red counts at pseudotimes of 4 and 6 would be ordered with the blue counts. However, in practice it is impossible to distinguish between *dropouts* and *true zeros*. **B** The percentage of counts with zero expression across 50 replicates for each value of λ used in dropout simulations. **C** The pearson correlation to true pseudotime using both the non-zero-inflated and zero-inflated variants of MFA as a function of λ used to generate the dataset. Accounting for zero-inflation shows marginal benefits if only a small percentage counts are dropouts. However, for high dropout percentages ($> 80\%$) the algorithm has to "impute" such a large percentage of the data that correlations to the true pseudotime reduce to near-zero.

Accounting for such dropouts involves modifying the model so that zero counts are likely if the underlying latent expression is low. Therefore, the red dropout cells in figure 37A would be effectively imputed (via Gibbs updates) upwards towards the mean expression line, increasing the accuracy of pseudotime inference. However, as there is no way to distinguish between true zeros and dropouts, we also "impute" the expression of the true zeros, which may itself decrease the accuracy of pseudotime inference.

We sought to quantify the benefits of modelling zero inflation against the drawbacks of losing the information contained in "true zeros". We created multiple synthetic datasets while varying the dropout parameter $\lambda \in \{0.02, 0.05, 0.1, 1, 10, \infty\}$, where $\lambda = 0.02$ has the largest levels of dropout while $\lambda = \infty$ has no dropout, only true zeros. This was repeated 50 times for each λ, and the proportion of zero counts in each dataset can be seen in figure 37B. We subsequently re-inferred the pseudotimes using MFA with both the zero-inflated and standard variants.

The resulting correlations with the true pseudotimes across the range of λ and MFA variants can be seen in figure 37C. At very high levels of dropout ($\lambda = 0.02$, where $> 80\%$ of counts are zeros) the zero-inflated variant performs considerably worse than the non-zero-inflated variant, with virtually no correspondence to the true pseudotimes compared to $\rho \approx 0.75$. We suggest this is due to the inference procedure effectively imputing such a large proportion of the data that there are too many degrees of freedom to effectively infer the trajectory. For the remaining values of λ the zero-inflated variant infers pseudotimes largely comparable to those of the non-zero inflated version, with marginal improvements in accuracy when there is moderate dropout ($\lambda = 1, 10$). We conclude that incorporating zero-inflation into such pseudotime inference is possible, but the variable quality across the (unknown in practice) dropout range along with considerable additional computational cost render it unnecessary for practical purposes.

4.4.3 *Application to single-cell RNA-seq data*

We next applied our method to previously published single-cell RNA-seq data of 4,423 Hematopoietic progenitor/stem cells (HSPCs) differentiating into myeloid and erythroid precursors [99]. To reduce the dataset to a computationally feasible size we used only genes expressed in at least 20% of cells with a variance in normalised expression greater than 5. We performed Gibbs sampling for 4×10^4 iterations using default hyperparameter values except for $\tau_\theta = \tau_\eta = 1$ and initialised the pseudotimes to the second principal component of the data. The results can be seen in figure 38(A-B). The MAP pseudotime estimates clearly recapitulates the trajectory in the data as shown using a tSNE representation from [117], while the MAP estimates of γ_n detects the branching structure in the data, consistent with previous methods.

We went on to analyse the genes suggested by the model to be involved in the bifurcation process. Figure 38C shows the inverse posterior mean of χ_g, with larger values indicating more evidence that gene g is involved in the bifurcation process. For illustration purposes, we plot the expression of *ELANE* and *CAR2*, which the model suggests will show differential behaviour across the bifurcation, along with *RPL26*, which the model suggests will show common behaviour (figure 38D).

We next sought to compare the performance of MFA to existing bifurcation inference algorithms, in particular Wishbone, DPT and Monocle (v2), along with the second principal component of the data (PC2), which we noted from exploratory analyses was highly correlated with the existing Wishbone values. We subsampled down to 1000 cells for Monocle comparisons for computational convenience and used the previously published results for Wishbone (from [117]). The root cell for DPT was selected as the cell with the minimum value for the second principal component and similarly the root

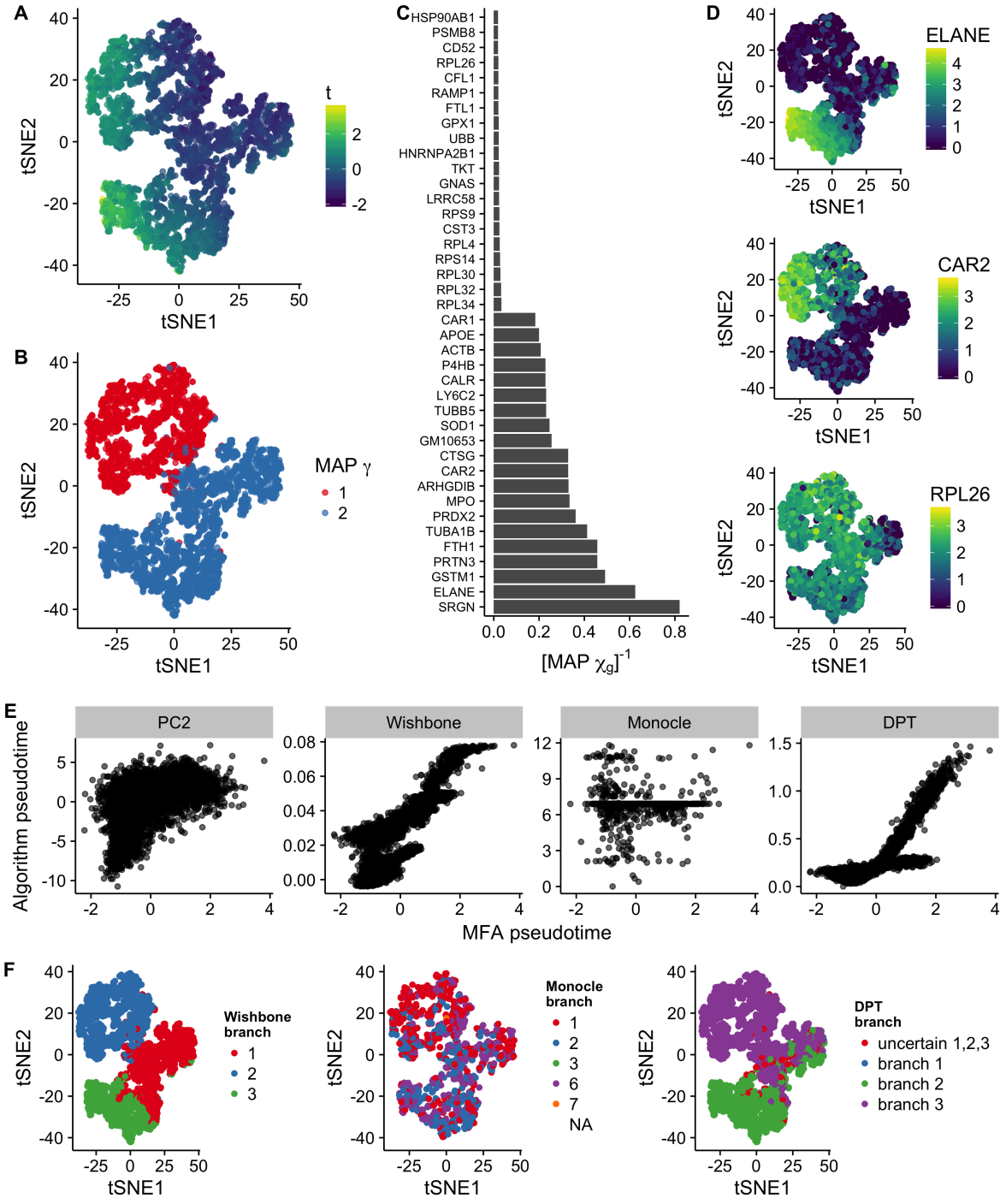


Figure 38: Caption next page.

Figure 38: Inference of bifurcations in scRNA-seq data of 4,423 Hematopoietic progenitor/stem cells (HSPCs) differentiating into myeloid and erythroid precursors. **A** tSNE representation from [117] coloured by the maximum aposteriori (MAP) pseudotime. **B** Equivalent plot as **A** coloured by MAP γ (branch assignment). **C** Inverse map χ showing both the 20 largest and 20 smallest values indicating which genes do and do not show differential behaviour across the bifurcation. **D** tSNE representation of the dataset coloured by gene expression. Both *ELANE* and *CAR2* were predicted by the inverse χ values to show differing expression across the branches, while *RPL26* was predicted to show similar expression. **E** Scatter plots of pseudotime values compared to those inferred by PC2, Wishbone, Monocle, and DPT. These had pearson correlations of 0.54, 0.83, 0.01, and 0.78 respectively. **F** tSNE representations of the dataset coloured by branch allocation of alternative algorithms shows good agreement with Wishbone and DPT.

state for Monocle was chosen such that it contained that cell. Otherwise, algorithms were run with default parameters.

The comparison of the inferred pseudotimes with that of MFA can be seen in figure 38E. There is high correlations with PC2 ($\rho = 0.54$), Wishbone ($\rho = 0.83$), and DPT ($\rho = 0.78$). However, there is virtually no correlation with Monocle ($\rho = 0.01$), though as this low correlation only occurs with Monocle we assume it is not an issue with MFA. We also sought to compare branch allocations across the algorithms across the algorithms which is difficult due to the non-identifiability of the statistical models involved. Figure 38F shows a tSNE representation of the cells coloured by branch allocation for each of Wishbone, Monocle and DPT. We see that MFA is largely consistent with Wishbone and DPT, detecting a bifurcation at the "pinch" in the tSNE plot, but as with the pseudotimes there is barely any correspondence in branch allocations with Monocle (which, as of version 2, does not allow pre-specification of the number of branches to model).

4.4.4 *Application to single-cell mass-cytometry data*

We next applied MFA to single-cell mass cytometry data tracking the differentiation of 22,850 monocytes and erythrocytes from hematopoietic stem and progenitor cells across 12 markers as published in [7] and previously analysed in [117]. For computational convenience with all algorithms we subsampled the data down to 2,000 randomly chosen cells, with the exception of Monocle which we subsequently subsampled further down to 1,000 cells. We found that due to the small number of proteins measured there was too much freedom for the MFA model to infer mixtures using the default parameter settings. We therefore had to encourage large levels of similarity across the two branches by setting $\alpha_\chi = 5 \times 10^3$ and $\beta_\chi = 1$.

The results can be seen in figure 39. Figure 39A shows a tSNE representation (as published in [117]) showing the inferred MAP pseudotimes correctly following the left-right trajectory, while figure 39B correctly shows the MAP γ values identifying a bifurcation at the "pinch" in the plot.

We subsequently compared the inferred pseudotimes and branching to those found using the alternative algorithms. We found good correspondence to all other methods (figure 39C), with pearson correlations of 0.84, 0.86, 0.80 and 0.69 for PC2, Wishbone, Monocle, and DPT respectively. We further compared the branch assignment of MFA to those of the alternative algorithms (figure 39D). As of version 2, Monocle does not allow for the number of branches to be selected *a priori* and typically returns a large number. For the convenience of visualisation we therefore only display the 30% most frequent states and group the remaining infrequent ones into "Other". We find good agreement between MFA and Monocle and DPT, and similarities with the Monocle assignments (MFA branch 2 loosely corresponds to Monocle branch 17).

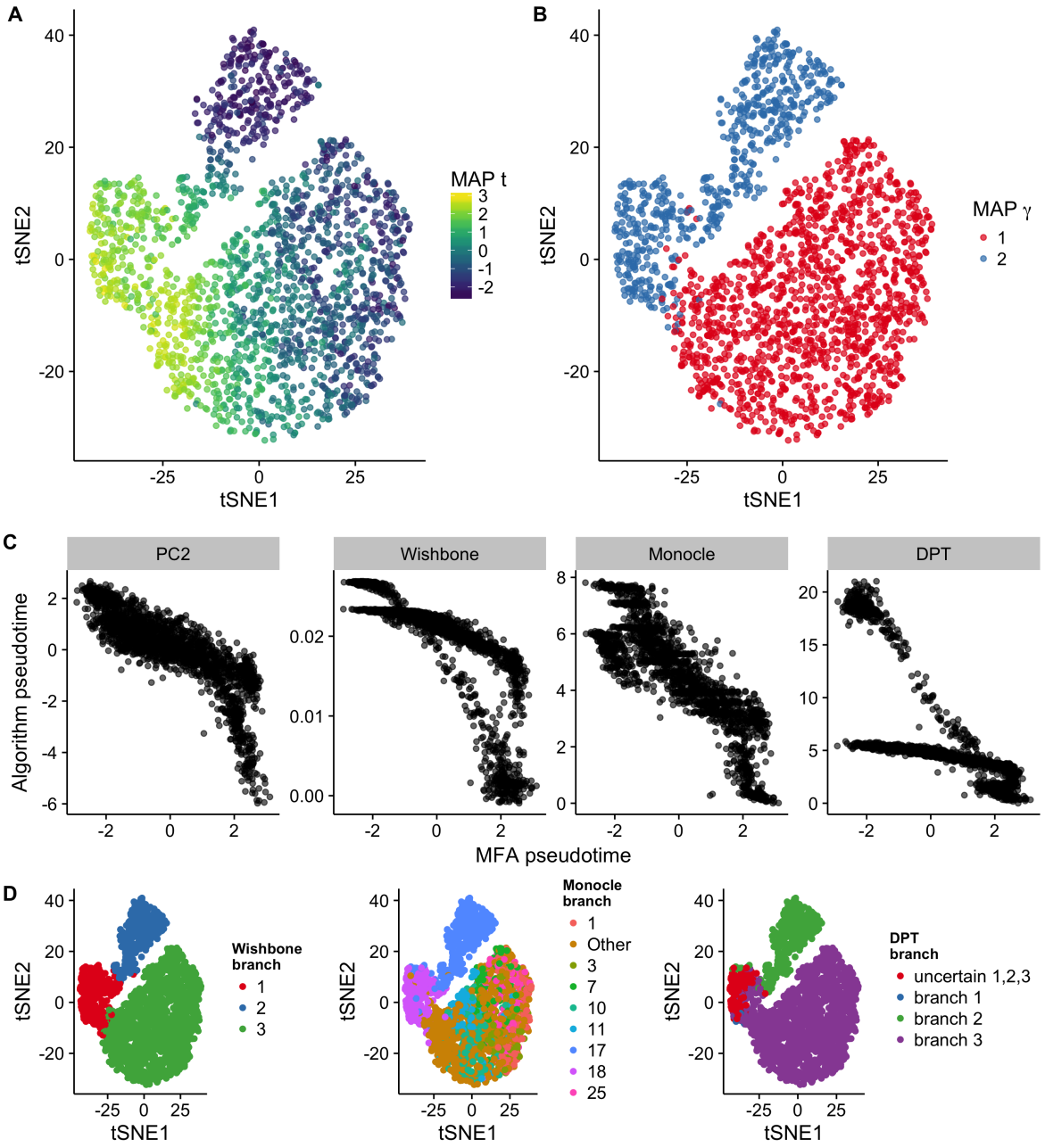


Figure 39: Caption next page.

Figure 39: Inference of bifurcations in single-cell mass cytometry data of a subsample of 2,000 HPSCs differentiating into monocyte and erythrocyte progenitors. **A** tSNE representation from [117] coloured by the maximum aposteriori (MAP) pseudotime. **B** Equivalent plot as **A** coloured by MAP γ (branch assignment). **C** Scatter plots of MFA pseudotime compared to PC2, Wishbone, Monocle, and DPT, with pearson correlations of 0.84, 0.86, 0.80 and 0.69 respectively. **D** tSNE representation coloured by branch assignment of Wishbone, Monocle, and DPT. As of version 2, Monocle does not allow for the number of branches to be selected *a priori* and typically returns a large number. For the convenience of visualisation we therefore only display the 30% most frequent states and group the remaining infrequent ones into "Other". The figures suggest a good agreement of branch assignment of MFA with Wishbone and DPT, and moderate agreement with Monocle.

4.5 DISCUSSION

In this chapter we have presented a Bayesian hierarchical mixture of factor analysers for inference of bifurcating trajectories in single-cell data. Our model is unique compared to existing efforts in that it (a) is fully generative, incorporating measurement noise into inference, (b) jointly infers both the pseudotimes and branches compared to post-hoc inference of branch detection, and (c) jointly infers which genes are differentially regulated across the branches. We also proposed an extension that accounts for the high levels of zero-inflation present in single-cell RNA-seq data. We applied our model to a range of synthetic and real datasets and demonstrated it performs competitively with existing methods.

4.5.1 *Trade off between model expressivity and practicality*

There is a natural trade-off in designing such models between flexibility and practicality. The implicit assumption of MFA that gene expression develops linearly across pseudo-

time allows for fast MCMC sampling and joint inference of branch structure. However, it is highly mis-specified: the predicted expression can easily become negative leading to erroneous inference. A solution to this would be to not explicitly assume a strongly parametric form of gene expression and consider nonparametric methods. However, such methods are often overly flexible, requiring either additional capture information to correctly infer pseudotimes [112] or hard-setting the pseudotimes prior to inferring the branching structure [73]. As such there is a natural trade-off between the expressivity of such models and being able to perform valid statistical inference that fully incorporates parameter variation without additional constraints or *tweaking*.

4.5.2 *Scalable inference*

While our current inference procedure performs well on large single-cell RNA-seq datasets there are scalable extensions that help as the number of single cells sequenced increases. The conditionally conjugate nature of the model makes it amenable to co-ordinate ascent variational inference (CAVI)[4], where a variational approximation transforms inference into an optimisation procedure. Once CAVI updates have been derived, an easy next step is stochastic variational inference (SVI, [54]), which subsamples data points for highly scalable inference. Such procedures will become necessary in single-cell statistical analysis as new technologies such as DropSeq [82] and 10x genomics [32] produce expression profiles for $> \mathcal{O}(10^6)$ cells.

4 Ideas we explore further in chapter 5.

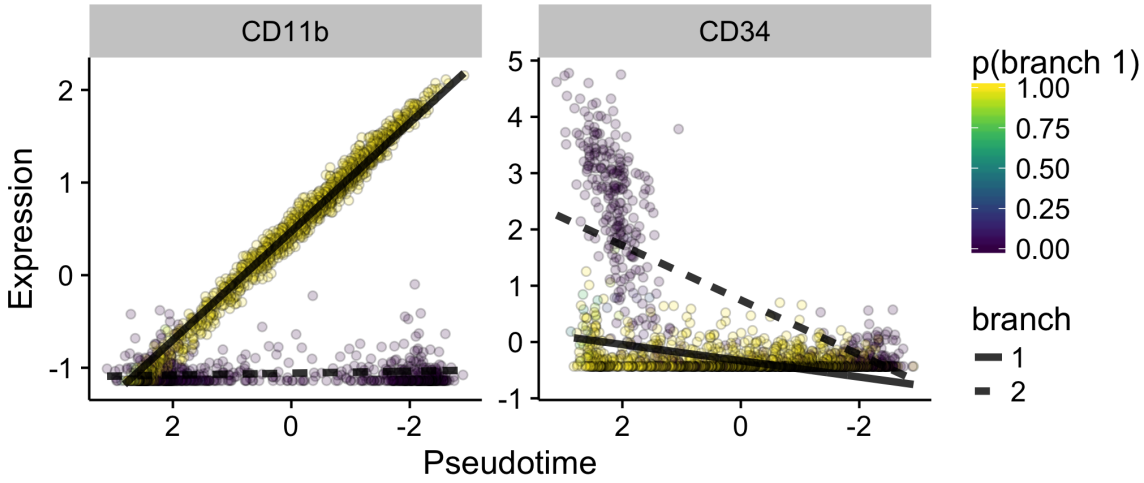


Figure 40: The limits of linear latent variable models for inferring bifurcations from single-cell data. *CD11b* is correctly identified as exhibiting bifurcating behaviour. However, due to model mis-specification the cells with nonzero expression are assigned to branch 2, when in fact they should be assigned to both branches 1 & 2, meaning *CD34* is incorrectly identified as differentially regulated across the bifurcation.

4.5.3 *Limits of linear latent variable models*

In general we expect linear latent variables to be highly mis-specified with respect to real gene expression data - gene expression will rarely evolve linearly as a function of time, nor even necessarily monotonically. A surprising result is that such an assumption is in practice sufficient to recapitulate the results on real data of algorithms that specifically account for nonlinearities in the data. We can therefore assume that the majority of genes in real datasets behave approximately linearly.

Such mis-specification - though conducive to fast full MCMC inference - does come at a cost. An example is given in figure 40 of the mass cytometry data (where the pseudotimes are reversed so time runs from left to right). The gene *CD11b* displays differential regulation across the branches - up-regulation on branch 1 and constant expression on branch 2. The model fits two values of k for this as can be seen by the

black line fits, and this the value of χ is sufficiently small that we correctly designate it as a gene that bifurcates.

However, the gene *CD34* is incorrectly designated as one that should bifurcate when in fact it doesn't. Due to model mis-specification the cells with non-zero expression at the beginning are "hard-assigned" to branch 2, when in fact they should be equally assigned to both branches 1 & 2. Consequently, $|k_2| \gg |k_1|$ and the value of χ indicates that the gene is involved in the bifurcation, when in fact it isn't. Such incorrect inferences can easily be checked visually on a reduced-dimension representation.

4.5.4 *Choosing the number of branches*

A further limitation of our model is the requirement to specify the number of branches B *a priori*. While at time of writing no single-cell datasets are known to have more than a single bifurcation point, the increasing resolution of single-cell technologies make such situations likely in the future. Here we propose two extensions for selecting the number of branches, by viewing it as a model selection problem or by modifying the model to be a nonparametric (i.e. infinite) mixture of factor analysers.

4.5.4.1 *As a model selection problem*

One method to choose the number of branches is to construct a finite set of (non-zero) integers and choose the number that best explains the data by some measure. We can view this as the classic Bayesian model selection problem where we wish to compute the marginal likelihood of the data given a pre-specified number of branches.

Without loss of generality consider two numbers of branches, B_1 and B_2. We can compute the marginal likelihood of the data $\mathbf{Y}$ given the number of branches B via

$$p(\mathbf{Y}|B) = \int_{\boldsymbol{\Theta}} p(\mathbf{Y}|\boldsymbol{\Theta}, B)p(\boldsymbol{\Theta}|B)d\boldsymbol{\Theta} \tag{64}$$

where $\boldsymbol{\Theta}$ is the entire set of model parameters. We then compute the *Bayes factor* comparing the two branches B_1 and B_2 as

$$\text{Bayes factor} = \frac{p(\mathbf{Y}|B_1)}{p(\mathbf{Y}|B_2)} \tag{65}$$

which provides the relative evidence for B_1 over B_2.

4.5.4.2 *Nonparametric mixture of factor analysers*

An alternative approach to choosing the number of branches would be to use a nonparametric mixture of factor analysers through the use of dirichlet process (DP) priors on the loading matrix. To define a DP we require a *base distribution* H defined on $\boldsymbol{\theta}$ and a positive number α. Then G is a DP ($G \sim \text{DP}(\alpha, H)$) if for any finite partition $\theta_1, \ldots, \theta_n$ of $\boldsymbol{\theta}$ we have

$$(G(\theta_1), \ldots, G(\theta_n)) \sim \text{Dirichlet}(\alpha H(\theta_1), \ldots, \alpha H(\theta_n)). \tag{66}$$

DPs can easily be sampled from by drawing a sample x_1 from the base distribution H and for the remaining $2, \ldots, N$ samples resampling from H with probability $\frac{\alpha}{\alpha+n-1}$ and sampling x_n from the existing $x_1, \ldots, x_{n-1}$ otherwise. In other words, α can be thought of as controlling the probability of adding a new value, which in the case of a mixture of factor analysers would add an additional branch. An example draws of a DP for a standard normal distribution for varying values of α can be seen in figure 41.

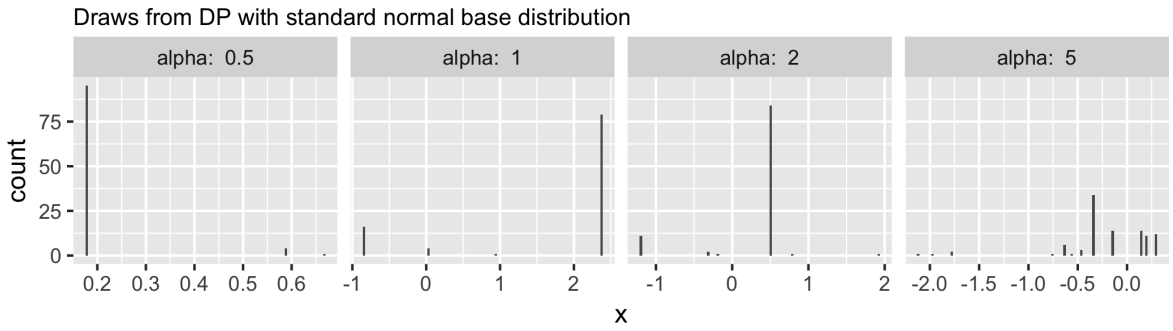


Figure 41: 100 draws from a dirichlet process with $\alpha = 0.5, 1, 2, 5$ with a standard normal base distribution.

Nonparametric mixtures of factor analysers have previously been considered for compressed sensing [25] and applied to NMR metabolomic data [93]. Such models typically consider a nonparametric prior on the number of latent dimensions allowing the model to "automatically decide" the dimensionality of the latent space, though in pseudotemporal applications this can be fixed to 1. A nonparametric mixture of overlapping Gaussian Processes was considered in [73], though they found that in practice this typically overestimated the number of branches.

The use of DP priors in this context has other drawbacks too. Though often sold as "inferring the number of components [of a mixture model] automatically" there is still the requirement to tune the parameter α that affects the eventual number of components. Furthermore, it is known that DP mixture models typically overestimate the number of components and are in fact inconsistent for the number of components [90]. In other words, even if we had an infinite amount of data a DP mixture model will not estimate the correct number of mixture components. Therefore, other approaches (such as model selection above) may be more suitable for inferring the number of branches in such settings.

4.5.5 *Accounting for technical effects*

Single-cell RNA-seq data are known to substantially suffer from technical batch effects [51, 131], particularly if multiple cells are sequenced across different plates or microfluidic devices. To account for this our model could be modified to take the form of a linear mixed model as

$$\mathbf{y}_n = \boldsymbol{\alpha} x_n + \mathbf{c}_{\gamma_n} + \mathbf{k}_{\gamma_n} t_n + \boldsymbol{\epsilon}_n \tag{67}$$

where x_n indicates to which batch cell n belongs and $\boldsymbol{\alpha}$ is a vector of gene-specific intercepts that account for global expression shifts due to technical effects.

5

COVARIATE-ADJUSTED LATENT VARIABLE MODELS

5.1 INTRODUCTION

So far in this thesis - and indeed in all published pseudotime algorithms to date - we have assumed that all cells or samples evolve along each trajectory identically. However, this assumption could easily be violated. For example, gene expression may change along the trajectory in a manner dependent on a cell's genetic background or perhaps upon a stimulant the cell has been exposed to.

The intuition for this problem is shown in figure 42. We can imagine the association between gene expression and the latent trajectory depending on some additional covariate (here given by the colour of the line). In the case of "red" samples, expression increases along the trajectory, while in the case of blue samples expression decreases. The same idea can easily be extended to the case in which the covariate is continuous.

Applying current pseudotime algorithms to such situations would confound inference. In the example in figure 42 there is no change in gene expression along the trajectory if the covariate is ignored, meaning there is no information contained in the gene that could be used to infer the trajectory unless the covariate is somehow incorporated. Furthermore, such interactions would not exist for all genes, so it would be advantageous if a model could pick out these interactions as they would be informative of the underlying biology.

As a solution to such issues this chapter introduces a general class of statistical models termed *covariate-adjusted latent variable models* that allows an externally measured set of covariates to perturb the change of features along the latent space. We proceed by deriving a scalable variational inference algorithm for inferring such models and their

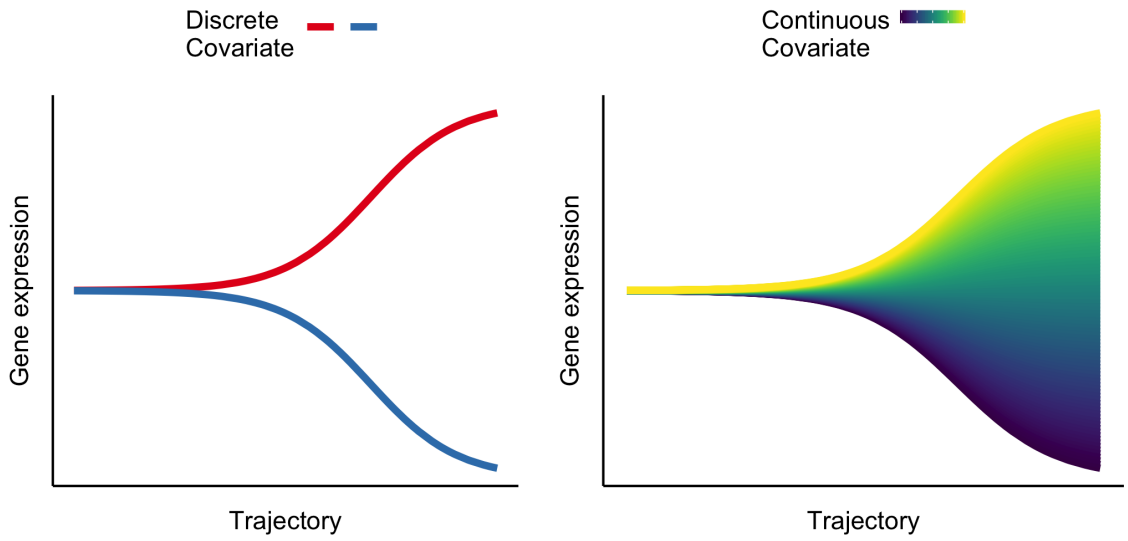


Figure 42: The behaviour of gene expression along trajectories may be affected by externally measured covariates. Such covariates may be discrete (left) or continuous (right).

interactions. This is applied to single-cell expression data in section 5.3.1 but also bulk RNA-seq data in sections 5.3.2 and 5.3.3, where we show that such trajectories correspond to the activation of biological pathways. Next, the case of the external covariate being a censored survival time is considered, with an application to population-level breast cancer studies. Finally, a non-parametric extension similar to GPLVM - termed *Covariate-adjusted Gaussian Process Latent Variable Models* - is proposed.

This chapter includes a minor change in notation - latent variables are now z_n rather than t_n as they may no longer represent physical time processes but more abstract notions of biological pathway activation.

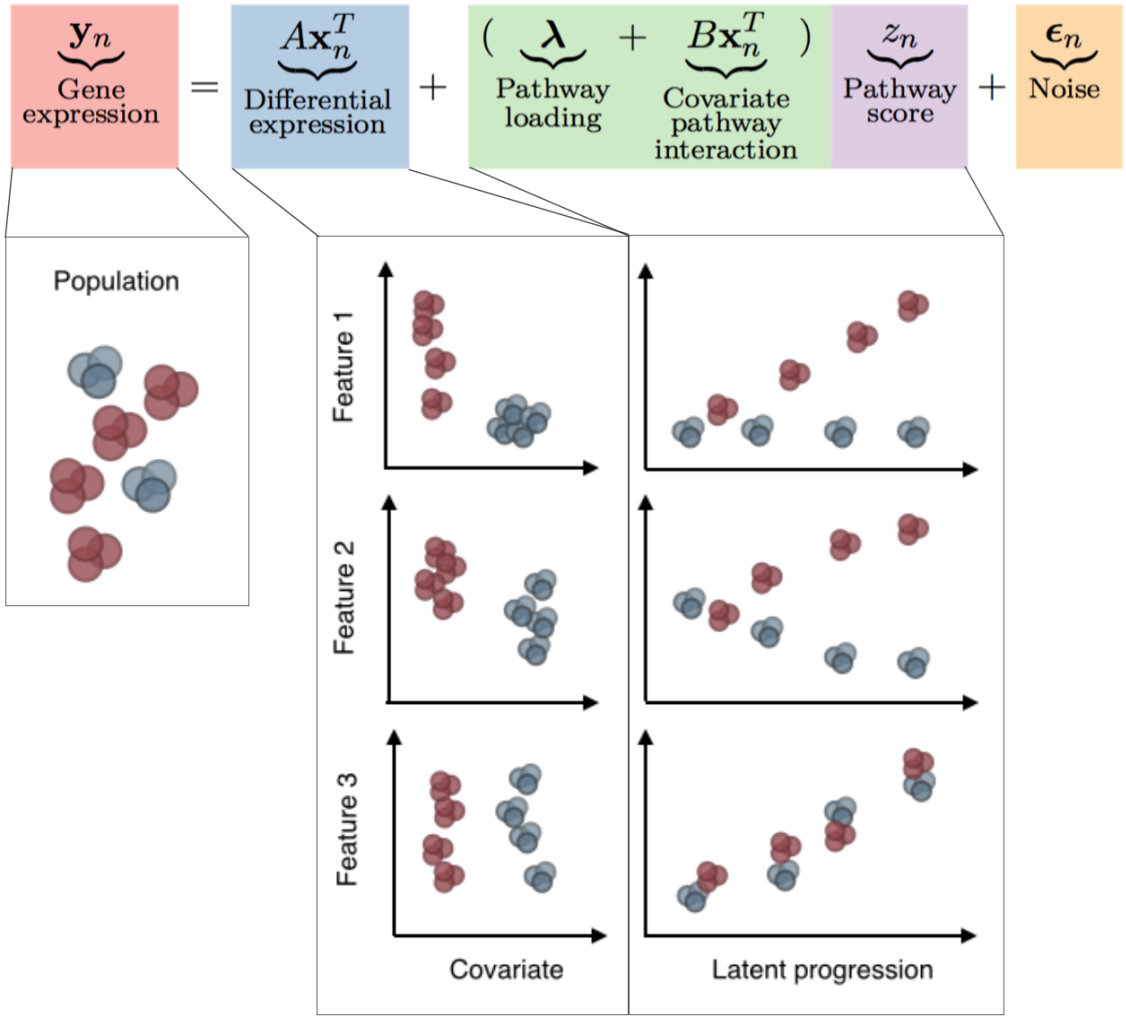


Figure 43: Covariate-adjusted latent variable models consider observed expression as a combination of standard differential expression (DE) and pathway effects, including covariate-pathway interactions.

5.2 COVARIATE-ADJUSTED LATENT VARIABLE MODELS

5.2.1 *Statistical model*

We begin as usual with an $N \times G$ matrix of gene expression $\mathbf{Y}$ with row vectors $\mathbf{y}_n$ for N samples and G genes. A standard factor analysis model would infer a Q-dimensional embedding $\mathbf{z}_n$ for each sample $n = 1, \ldots, N$ where $Q \ll G$ via a model of the form

$$\begin{aligned} \mathbf{z}_n &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\ \mathbf{y}_n &\sim \mathcal{N}(\mathbf{\Lambda}\mathbf{z}_n, \mathbf{\Sigma}) \end{aligned} \tag{68}$$

where $\mathbf{\Lambda}$ is a $G \times Q$ factor loading matrix with column vectors $\boldsymbol{\lambda}_q$ and $\mathbf{\Sigma} = \text{diag}(\sigma_1^2, \ldots, \sigma_G^2)$ is a diagonal covariance matrix of measurement noise. As noted before, if $Q = 1$ then z_n can be interpreted as the "pseudotime" of each cell in which case the factor loading matrix becomes a $G-$length factor loading vector $\boldsymbol{\lambda}$ and the likelihood of $\mathbf{y}_n$ becomes $\mathbf{y}_n \sim \mathcal{N}(\boldsymbol{\lambda} z_n, \mathbf{\Sigma})$.

We now consider the case mentioned in the introduction that we have an $N \times P$ matrix $\mathbf{X}$ that represents P covariates for each of the N samples. In a population wide study such covariates might represent phenotypic variables such as age or sex, while in a single-cell setting such covariates might represent genetic background or cell stimulus.

We would like these covariates to perturb the change in expression of each gene along the trajectory. To do this we introduce an additional $G \times P$ matrix $\mathbf{B}$ whose entries β_{pg} represent the effect of covariate p on the change of gene g along the trajectory. Thus the factor loading for gene g becomes

$$\lambda_g \to \lambda'_{ng} = \lambda_g + \sum_{p=1}^{P} \beta_{pg} x_{np} \tag{69}$$

or in vector notation $\boldsymbol{\lambda} \to \boldsymbol{\lambda}'_n = \boldsymbol{\lambda} + \mathbf{B}\mathbf{x}_n$. In other words, each sample has a unique loading for each gene that depends on a common loading vector $\boldsymbol{\lambda}$ and modulation by the sample-specific covariates.

In factor analysis models it is typical to standardize the data so that the marginal mean is 0 which simultaneously enforces the constraint $y = 0$ when $z = 0$. However in this case that constrains the data to "swing" around the origin based on the covariate which is overly restrictive. To solve this we introduce an additional $N \times P$ matrix $\mathbf{A}$ whose entries α_{pg} account for the global shift in expression of gene g in response to covariate p. Therefore, the generative *covariate-adjusted latent variable model* takes the form

$$\begin{aligned} \mathbf{z}_n &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\ \mathbf{y}_n &\sim \mathcal{N}\left(\mathbf{A}\mathbf{x}_n + (\boldsymbol{\lambda} + \mathbf{B}\mathbf{x}_n)z_n, \boldsymbol{\Sigma}\right). \end{aligned} \tag{70}$$

In a genomics context α_{pg} can be thought of as mediating differential expression (figure 43) while $\boldsymbol{\lambda}$ can be thought of as the change in expression along the latent trajectory regardless of covariates. Note that if we multiply out the bracket in 70 we see that this is a form of linear mixed model with interactions between the random and fixed effects, though to our knowledge such a model has not been proposed before.

In practice we restrict ourselves to $Q = 1$ dimensional latent spaces (that roughly correspond to "pseudotimes" or "trajectories"). However, this can be readily extended to the $Q > 1$ case by making $\mathbf{B}$ a $Q \times P \times G$ factor loading tensor[1] whose entries β_{qpg} quantify the interaction between covariate p and gene g in latent dimension q. The mean μ_{ng} for observation y_{ng} is then given by

$$\mu_{ng} = \sum_{q=1}^{Q} \left(\lambda_{qg} z_{nq} + \sum_{p=1}^{P} (\alpha_{qpg} x_{np} + \beta_{qpg} x_{np} z_{nq}) \right) \tag{71}$$

1 Technically an array rather than a tensor in the true sense.

In general we expect trajectory-covariate interactions to be rare we place an automatic relevance determination (ARD) prior on them (previously discussed in chapter 4). In the $Q = 1$ dimensional case this takes the form $\beta_g \sim \mathcal{N}(0, \chi_g^{-1})$, $\chi_g \overset{iid}{\sim} \text{Gam}(a_\beta, b_\beta)$. We set $a_\beta = b_\beta = 0.01$ which places the prior precision close to zero but has high variance. The overall generative model then takes the form

$$\begin{aligned}
\alpha_{pg} &\sim \mathcal{N}(0, \tau_\alpha^{-1}) \\
\lambda_g &\sim \mathcal{N}(0, \tau_\lambda^{-1}) \\
z_n &\sim \mathcal{N}(q_n, \tau_q^{-1}) \\
\beta_{pg} &\sim \mathcal{N}(0, \chi_{pg}^{-1}) \\
\chi_{pg}^{-1} &\sim \text{Gamma}(a_\beta, b_\beta) \\
\tau_g^{-1} &\sim \text{Gamma}(a, b) \\
\epsilon_{ng} &\sim \mathcal{N}(0, \tau_g^{-1}) \\
y_{ng} &= \mu_g + \sum_p \alpha_{pg} x_{np} + \left(\lambda_g + \sum_p \beta_{pg} x_{np}\right) z_n + \epsilon_{ig}
\end{aligned} \tag{72}$$

where τ_α, τ_λ, a, b, a_β, b_β, τ_q are fixed hyperparameters and q_n encodes prior information about z_n if available but typically $q_n = 0\ \forall n$ in the uninformative case. We refer to this statistical model applied to genomics data and the accompanying `R` package for inference as "PhenoPath".

5.2.2 *Inference*

5.2.2.1 *Gibbs sampling*

The conditionally conjugate nature of the model in equation 135 makes Gibbs sampling once more possible, as was explored in section 4.2.2. The Gibbs updates for this are given

in appendix E.2. However, Gibbs sampling becomes slow as the number of parameters increases, which needs particular care in this model given the number of parameters scales as $G \times P$ for G genes and P covariates.

5.2.2.2 *Co-ordinate ascent variational inference*

Instead we turn to variational inference which recasts Bayesian inference as an optimisation problem, rather than the sampling approaches previously used. Below, variational inference is briefly introduced along with the strategy for deriving updates for covariate-adjusted latent variable models, which may be found in E.3.

In general Bayesian inference is concerned with inferring the posterior distribution $p(\boldsymbol{\theta}|\mathbf{X}) = \frac{p(\mathbf{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{X})}$ for some parameters $\boldsymbol{\Theta}$ and data $\mathbf{X}$. This is difficult in general as the marginal likelihood $p(\mathbf{X})$ is often intractable and hard to compute.

Variational inference posits an approximating or *variational* distribution $q(\boldsymbol{\theta}|\boldsymbol{\lambda})$ with variational parameters $\boldsymbol{\lambda}$ and the objective of inference is to choose optimal $\boldsymbol{\lambda}$ so that $q(\boldsymbol{\theta}|\boldsymbol{\lambda})$ is as "close" to $p(\boldsymbol{\theta}|\mathbf{X})$ as possible. This closeness is normally defined with respect to some divergence measure such as the Kullback-Leibler (KL) divergence which is a measure of the non-symmetric difference between two probability distributions. Thus Bayesian inference is transformed into an optimisation procedure that attempts to minimise

$$\mathrm{KL}\left(q(\boldsymbol{\theta}|\boldsymbol{\lambda}) \parallel p(\boldsymbol{\theta}|\mathbf{X})\right) = \int d\boldsymbol{\theta} q(\boldsymbol{\theta}|\boldsymbol{\lambda}) \log\left[\frac{q(\boldsymbol{\theta}|\boldsymbol{\lambda})}{p(\boldsymbol{\theta}|\mathbf{X})}\right]. \tag{73}$$

A little algebra shows that minimising the KL-divergence in equation 73 is equivalent to maximising the evidence lower-bound (ELBO), defined as

$$\mathrm{ELBO}(\boldsymbol{\lambda}; \mathbf{X}, \boldsymbol{\theta}) = \mathbf{E}_{q(\boldsymbol{\theta}|\boldsymbol{\lambda})}\left[\log q(\boldsymbol{\theta}|\boldsymbol{\lambda}) - \log p(\mathbf{X}, \boldsymbol{\theta})\right]. \tag{74}$$

Interestingly, because $\mathrm{KL}\left(q \parallel p\right) \geq 0$ for any q and p it is easy to show that the ELBO acts as a lower bound on the log marginal likelihood, i.e. $\log p(\mathbf{X}) \geq \mathrm{ELBO}(\boldsymbol{\lambda}; \mathbf{X}, \boldsymbol{\theta})$. Therefore, variational inference can be viewed either as choosing $\boldsymbol{\lambda}$ so that q is as similar to p as possible, or as choosing $\boldsymbol{\lambda}$ so that a lower bound on the marginal likelihood is maximised.

For our model in equation 135 we make what is known as a mean-field variational approximation where the variational approximation factorises across all the parameters so $q(\boldsymbol{\theta}|\boldsymbol{\lambda}) = \prod_i q(\theta_i|\lambda_i)$. For co-ordinate ascent variational inference the approximating distributions should be of the family as each conditional distribution. The variational distribution for our model is therefore given by

$$\begin{aligned} &q\left(\{z_n\}_{n=1}^N, \{\mu_g\}_{g=1}^G, \{\tau_g\}_{g=1}^G, \{\lambda_g\}_{g=1}^G, \{\alpha_{pg}\}_{g=1,p=1}^{G,P}, \{\beta_{pg}\}_{g=1,p=1}^{G,P}, \{\chi_{pg}\}_{g=1,p=1}^{G,P}\right) \\ &= \prod_{n=1}^{N} \underbrace{q_z(z_n)}_{\text{Normal}} \prod_{g=1}^{G} \underbrace{q_\mu(\mu_g)}_{\text{Normal}} \underbrace{q_\tau(\tau_g)}_{\text{Gamma}} \underbrace{q_\lambda(\lambda_g)}_{\text{Normal}} \prod_{p=1}^{P} \underbrace{q_\alpha(\alpha_{pg})}_{\text{Normal}} \underbrace{q_\beta(\beta_{pg})}_{\text{Normal}} \underbrace{q_\chi(\chi_{pg})}_{\text{Gamma}} \end{aligned} \tag{75}$$

For conditionally conjugate models it is possible to derive updates that maximise the ELBO for each parameter, conditioned on all other parameters being held constant. For a general model let θ_j denote the j^{th} parameter and $\boldsymbol{\theta}_{-j}$ the vector of all parameters other than j. The update that provides the distribution for θ_j that maximises the ELBO is given by

$$q_j^*(\theta_j) \propto \exp\left\{\mathbf{E}_{-j}\left[\log p(\theta_j|\boldsymbol{\theta}_{-j}, \mathbf{Y})\right]\right\}, \tag{76}$$

where the expectation is taken with respect to the approximating distributions for all parameters other than j. Successively computing updates for each variable until the change in the ELBO falls below some pre-set threshold is known as co-ordinate ascent variational inference (CAVI). In our model all parameters have conditional distribu-

tions that are either normally distributed or gamma distributed. We can derive general updates for these two cases and the specific updates may be found in appendix E.3.

Suppose $p(\theta_j|\theta_{-j}, \mathbf{X}) \sim \mathcal{N}(\mu_{\theta_j}, \tau_{\theta_j}^{-1})$ where both the mean and precision are dependent on the conditioning variables $\boldsymbol{\theta}_{-j}$ and the data $\mathbf{X}$. It follows that

$$q(\theta_j) = \mathcal{N}\left(\theta_j|m_{\theta_j} = \frac{\mathrm{E}_{-j}[\mu_{\theta_j}\tau_{\theta_j}]}{\mathrm{E}_{-j}[\tau_{\theta_j}]}, s^2_{\theta_j} = \mathrm{E}_{-j}[\tau_{\theta_j}]^{-1}\right) \tag{77}$$

where the expectations are computed with respect to the variational distributions in 75.

If instead $p(\theta_j|\theta_{-j}, \mathbf{X}) = \mathrm{Gamma}(\theta_j|a_{\theta_j}, b_{\theta_j})$ where again a_{θ_j} and b_{θ_j} are functions of the data $\mathbf{X}$ and all parameters other than θ_j, then the CAVI update is given by

$$q(\theta_j) = \mathrm{Gamma}\left(\theta_j|\mathrm{E}_{-j}[a_{\theta_j}], \mathrm{E}_{-j}[b_{\theta_j}]\right). \tag{78}$$

Variational inference for non-conditionally conjugate models is considered in 5.4.

5.2.3 *Bayesian significance testing of interactions*

It is possible to construct a Bayesian significance test for the existence of covariate-pathway interactions (in other words, whether β_g is "significant" under some criterion) using posterior credible intervals and a region of practical interest. The variational approximation for each β_g is given by $q(\beta_g) \sim \mathcal{N}(m_{\beta_g}, s^2_{\beta_g})$. Therefore, if $\hat{m}_{\beta_g}$ and $\hat{s}_{\beta_g}$ are

the estimates of m_{β_g} and s_{β_g} respectively after a sufficient number of iterations that we consider the ELBO to be converged, then we define g to have a *significant interaction* if

$$\begin{aligned} &\hat{m} + k\hat{s} < 0, \quad \text{or} \\ &\hat{m} - k\hat{s} > 0 \end{aligned} \tag{79}$$

for some k. In other words, g is significant if the $k\hat{s}_{\beta_g}$ posterior interval of β_g falls outside of zero. In practice, minimising the $\mathrm{KL}\left(q \parallel p\right)$ divergence typically underestimates posterior variances [9] so we choose a conservative $k = 3$.

5.2.4 *Inference of convergence points*

In PhenoPath we model gene expression evolving along the trajectories separately for each phenotype (or covariate) considered. Unless the gradient of change along the trajectory is exactly equal for both phenotypes (i.e. $\beta = 0$ exactly), the gene expression will cross at a given point in the trajectory.

Inference of this point would allow us to identify sections of the trajectory not affected by the covariate and consequently sections of the trajectory that are. This is important as if the crossover point occurs towards the beginning of the trajectory, it would mean gene expression is similar at the beginning but diverges as we move along the trajectory. Similarly, if the crossover points occur towards the end of the trajectory, it would imply the expression profiles for the two phenotypes are different at the beginning of the trajectory, but converge as the trajectory progresses. An interpretation of this would be that the effect on expression from the trajectory slowly dominates over the effect of phenotypes on the trajectory.

It is important to note that the latent trajectory values loosely follow a $\mathrm{N}(0,1)$ distribution. This means the 'middle' of the trajectory is any value around zero, values of -1 or less could be thought of as the 'beginning' while values greater than 1 may be thought of as the 'end'. Crucially, we can derive an analytical expression from the PhenoPath parameters for the crossover point z^*. The condition for the crossover point is that the predicted expression for each phenotype is identical. Therefore if we have phenotypes $+$ and $-$ we require

$$y_g^+(z_g^*) = y_g^-(z_g^*) \tag{80}$$

which leads to the condition

$$\alpha_g x_+ + (c_g + \beta_g x_+)\, z_g^* = \alpha_g x_- + (c_g + \beta_g x_-)\, z_g^* \tag{81}$$

which is in turn solved by $z_g^* = -\frac{\alpha_g}{\beta_g}$.

5.2.5 *Benchmarking through simulations*

We first performed a simulation study to demonstrate the value of modelling covariate-pathway interactions and to show that such effects are missed by standard differential expression analyses. Specifically, we sought to compare differentially expressed genes identified by Limma Voom [68], one of the leading RNA-seq differential expression methods, to the β interactions from PhenoPath. For $N = 200$ samples we assigned each to one of two categories given by the x values $x = -1, 1$, and assigned a pseudotime z through draws from a standard normal distribution. For each sample $n = 1, \ldots, N$

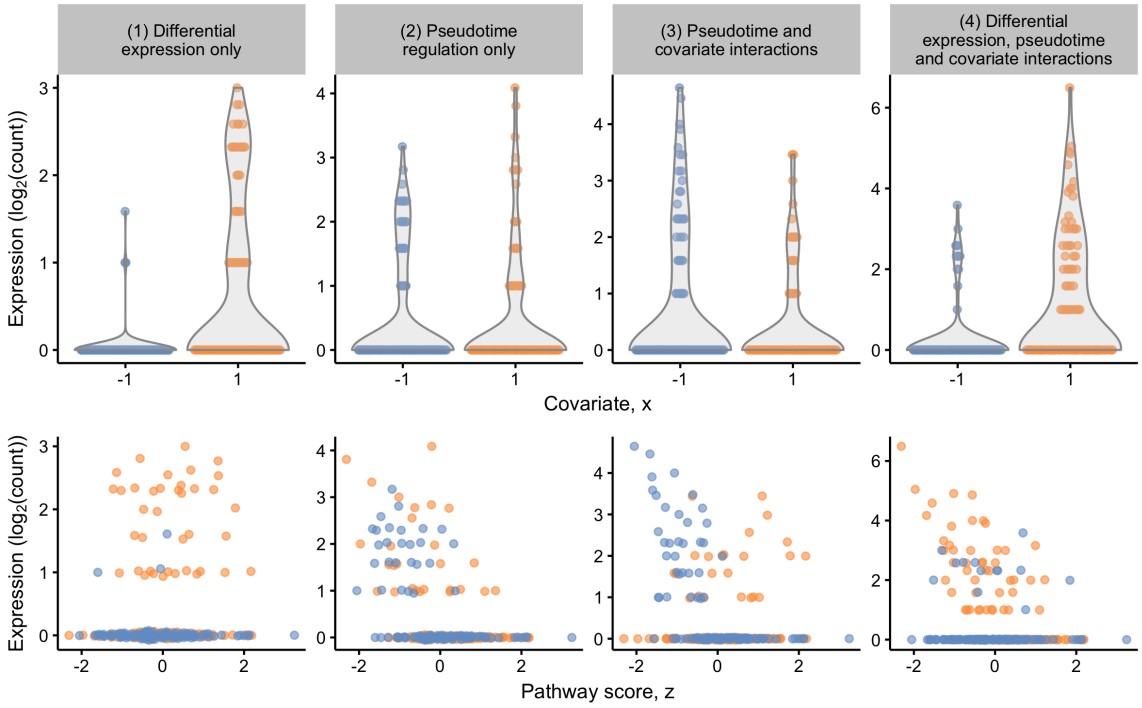


Figure 44: Four gene expression simulation scenarios were used: (1) differential expression only where the overall expression level for groups -1 and 1 differed but there is no dependence on pseudotime or pathway score, (2) pseudotime regulation only where the overall marginal distribution of expression values is identical between groups but expression changes with latent pathway score, (3) pseudotime and covariate interactions where the trajectory for each group differs over pathway score and (4) a complex scenario where differential expression and covariate-pseudotime interactions all exist.

and gene $g = 1, \ldots, G$ we then generated a mean value through the PhenoPath mean function

$$\mu_{ng} = \alpha_g x_n + (\lambda_g + \beta_g x_n) z_n \tag{82}$$

The gene-specific parameters $(\alpha_g, \lambda_g, \beta_g)$ were sampled in equal proportions from one of four classes (figure 44):

1. *Differential expression only* where $\alpha_g = 1$ or -1 with equal probability and $\lambda_g = \beta_g = 0$
2. *Pseudotime regulation only* where $\lambda_g = 1$ or -1 with equal probability and $\alpha_g = \beta_g = 0$
3. *Pseudotime and covariate interactions* where λ_g and β_g are set to 1 or -1 with equal probability and $\alpha_g = 0$
4. *Differential expression, pseudotime and covariate interactions* where all parameters take on values of -1 or 1 with equal probabilities

Examples of expression from these four simulation regimes can be seen in figure 44.

In order to generate RNA-seq reads we need positive count values. In the spirit of general linear models, we then used $g(x) = 2^x$ as a link function and generated a matrix of positive means

$$\tilde{\mu}_{ng} = 2^{\mu_{ng}} \tag{83}$$

We subsequently simulated a count matrix c_{ng} by sampling for each entry from a negative binomial distribution with mean $\tilde{\mu}_{ng}$ and size parameter $\tilde{\mu}_{ng}/3$. While this could be used as input to PhenoPath (suitable log transformed), we sought to make

Algorithm	True positive rate	False positive rate	False discovery rate
Limma Voom	0.82	0.09	0.18
PhenoPath	0.97	0.02	0.03

Table 3: A comparison of true positive, false positive, and false discovery rates for Limma Voom detecting differential expression and PhenoPath detecting covariate-pseudotime interactions on synthetic data.

our simulation as realistic as possible including quantification errors. We subsequently simulated FASTA files using the Bioconductor package `polyester` [37] using the first 400 transcripts of the reference transcriptome of the 22nd human chromosome. FASTA files were then converted to FASTQ files and quantified into TPM and count estimates using Kallisto [12]. The $\log_2(\text{TPM}+1)$ values were then used for input to PhenoPath while the raw count values were used for input to Limma Voom.

An exploratory PCA analysis of the data reveals a *wishbone*-like pattern similar to bifurcations (figure 45A). Given the first principal component of the data is often a good estimator of a one-dimensional trajectory, we compared PC1 to the true pseudotimes (figure 45A). This demonstrates that the distinct phenotypic or covariate classes do confound inference of the trajectory as PC1 has a Spearman correlation of $\rho = 0.85$ with the true values compared to $\rho = 0.97$ with inference from our model when the covariates are explicitly taken into account.

Algorithm(s)	n
Both	47
PhenoPath only	16
Limma only	12
Neither	25

Table 4: Number of interactions discovered as significant under the *Differential expression, pseudotime and covariate interactions* regime.

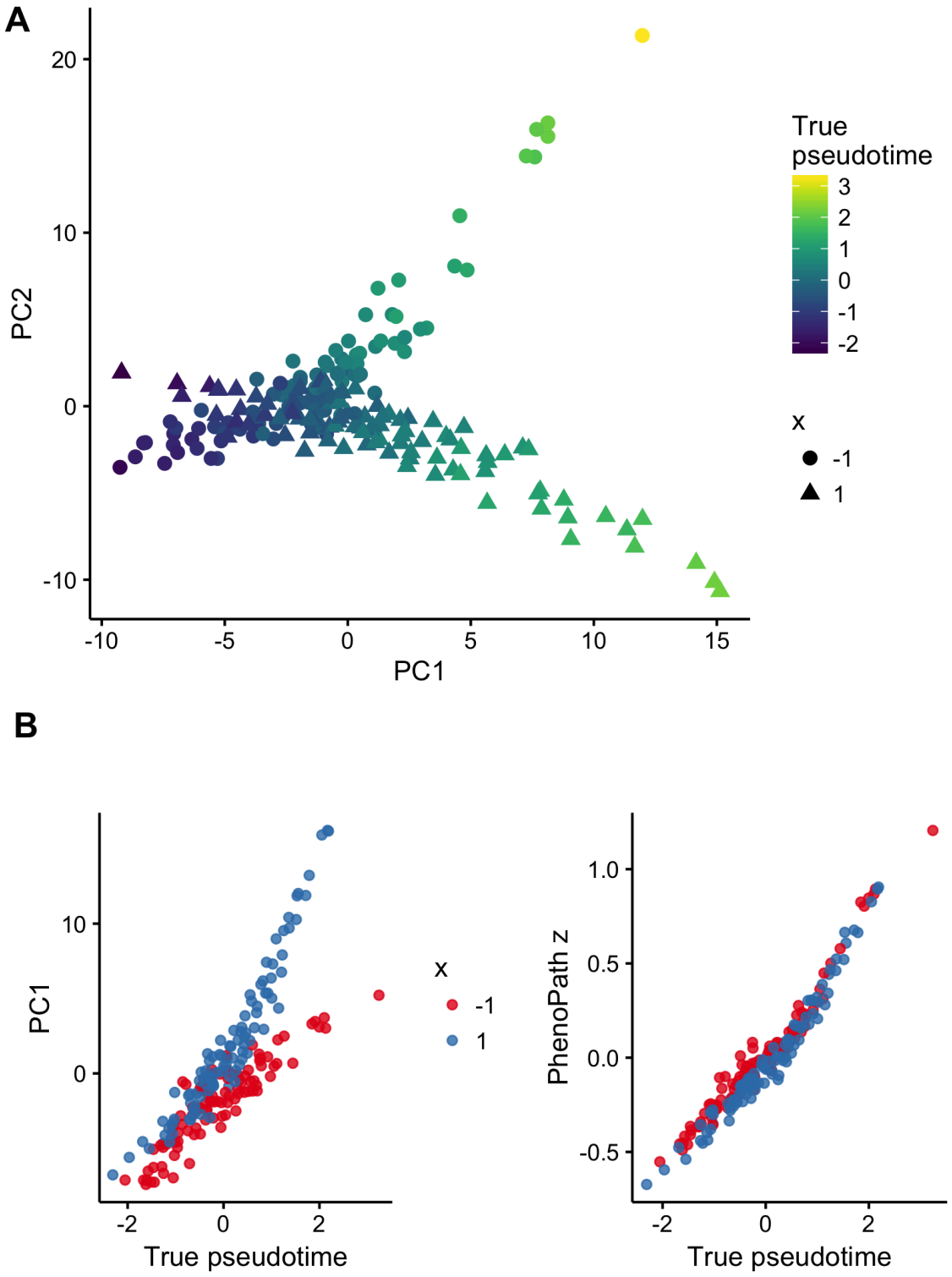


Figure 45: Simulations of RNA-seq data with covariate pseudotime interactions for 200 samples and 400 genes using the R/Bioconductor package `polyester`. **A** A PCA representation of the data coloured by pseudotime shows a clear splitting of trajectories between covariate status. **B** Comparison of the true pseudotime to both PC1 and PhenoPath pathway score z with correlations of 0.85 and 0.97 respectively.

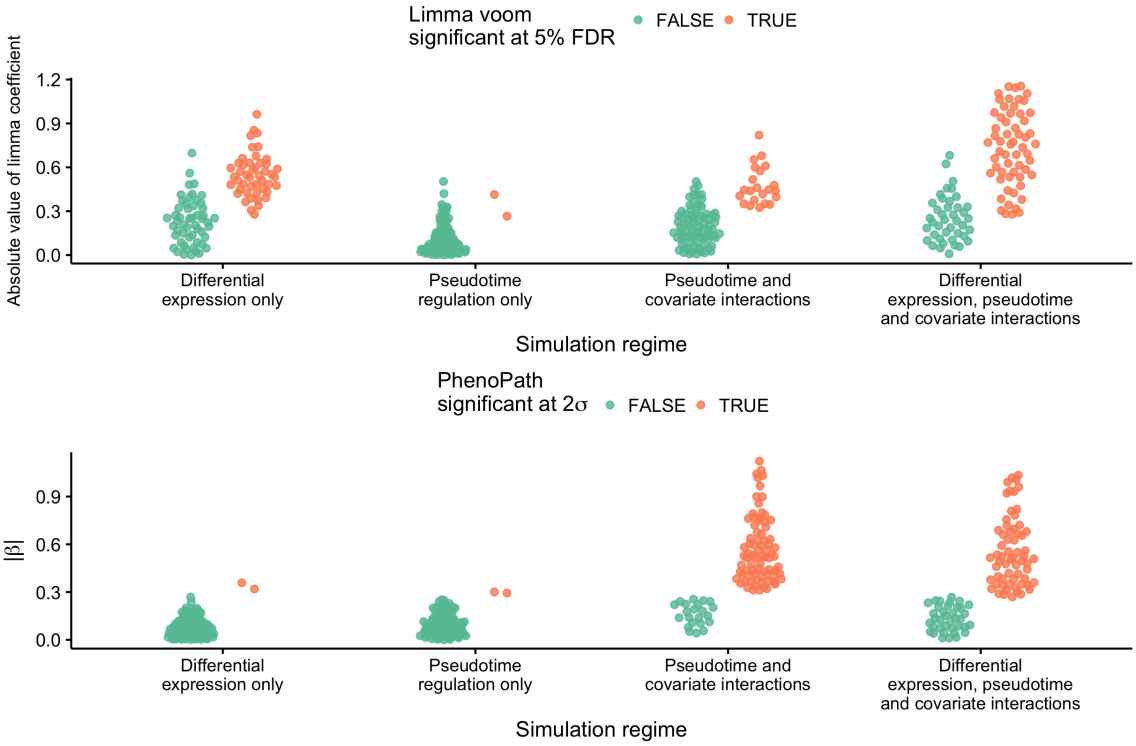


Figure 46: A comparison of the effect sizes and number of genes identified as differentially regulated across the four simulation regimes for both Limma Voom (top) and PhenoPath (bottom).

We then benchmarked the ability of our model to identify such interactions and whether they confounded standard differential expression analyses. Our model exhibited high specificity and sensitivity by classifying only a small number of simulated genes (2%) as exhibiting interaction effects in cases 1-2 where there are no covariate-pathway interactions but identifies 78% and 63% of genes as exhibiting significant covariate-pathway interactions in cases 3 and 4 respectively (tables 3 and 4 and figure 46). For comparison, a standard DE analysis using Limma-Voom identified 47% and 59% of genes as differentially expressed in cases 1 and 4 respectively. In case 2 only 2% of genes are identified as DE as expected but, in case 3, 22% of genes are identified as DE where Limma-Voom would not be expected to report any differentially expressed genes.

In our simulation study, Limma Voom "only" detects 47% of the genes simulated as differentially expressed. Such power to detect differential expression is dependent on effect sizes and measurement noise, and so such a figure is in no way unreasonable given the parameters used. While a more comprehensive simulation study could examine detection rates across entire distributions over effect sizes and measurement noise, we simply sought to perform a simulation that demonstrated that PhenoPath identifies a subset of differential expression and that standard differential expression misses some interactions across a consistent effect size and noise regime.

5.3 APPLICATIONS OF CONDITIONALLY CONJUGATE MODEL

5.3.1 *Single-cell RNA-seq*

We next examined a time-series single-cell RNA-seq dataset of bone marrow derived dendritic cells responding to particular stimuli [119]. Cells were exposed to LPS, a component of Gram-negative bacteria, and PAM, a synthetic mimic of bacterial lipopeptides,

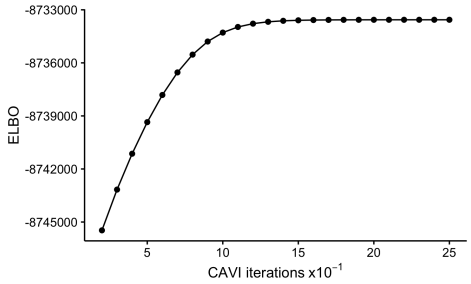


Figure 47: Evidence lower bound (ELBO) as a function of CAVI iterations for the Shalek et al. dataset.

and expression quantification performed at 0, 1, 2, 4 and 6 hours after stimulation. Despite the time-series measurement, previous studies have suggested this dataset is more suited to a "pseudotime" analysis as the cells respond asynchronously and heterogeneity exists within the cellular populations at each time point [112]. To-date pseudotime inference algorithms would typically assume a common trajectory across all experimental conditions or a pseudotime analysis performed separately for each stimulant. This might give a loss of statistical power and artefacts introduced by confounding effects. Using PhenoPath we can encode the stimulant to which the cells were exposed as a covariate and allow gene expression to evolve along pseudotime differently for either LPS or PAM exposure. This allows us to learn a single trajectory for all cells regardless of stimulant applied yet simultaneously infer which genes are differentially regulated in response. We applied this to the 820 cells exposed to LPS and PAM in the time points 1, 2, 4, and 6 hours after stimulation using the 7,533 genes whose variance in normalised log-expression exceeded a pre-set threshold (see appendix B.4). PhenoPath was run for approximately 250 CAVI iterations until convergence of the ELBO (figure 47).

We inferred a covariate-perturbed trajectory using PhenoPath and uncovered a landscape of pseudotime-stimulant interactions (figure 48A), unveiling genes whose regulation along pseudotime is modulated by the application of LPS or PAM. The trajectory inferred largely recapitulated the true time-series measurement (figure 48B, $R^2 = 0.64$), despite no explicit temporal information being provided to the algorithm, though tran-

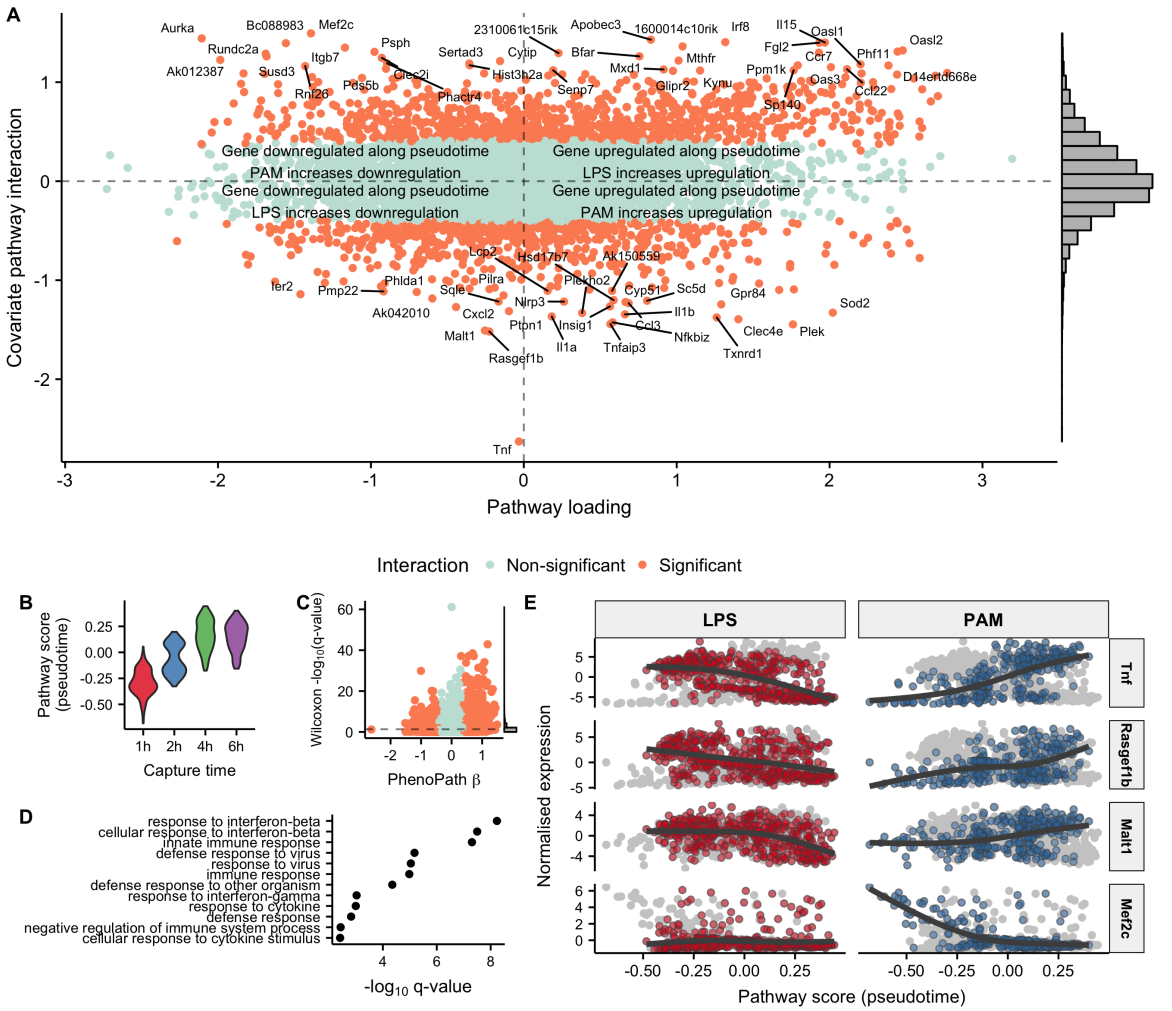


Figure 48: Stimulant-immune interactions in time-series single-cell RNA-sequencing data. **A** PhenoPath applied to the Shalek et al. dataset uncovers genes differentially regulated along pseudotime depending on the stimulant (LPS or PAM) applied. **B** PhenoPath infers pseudotimes (z) consistent with the physical capture times. **C** A comparison of p-values obtained through a nonparametric statistical test for differential expression between LPS and PAM stimulation shows no particular relation with the interaction parameters β inferred with PhenoPath. **D** A GO enrichment analysis of the genes upregulated along pseudotime whose upregulation was increased by LPS stimulation showed enrichment for immune system processes. **E** Expression of the four genes with the largest interaction effect sizes along over pseudotime, stratified by stimulant applied. Strikingly, *Tnf* is upregulated under PAM exposure yet downregulated under LPS stimulation.

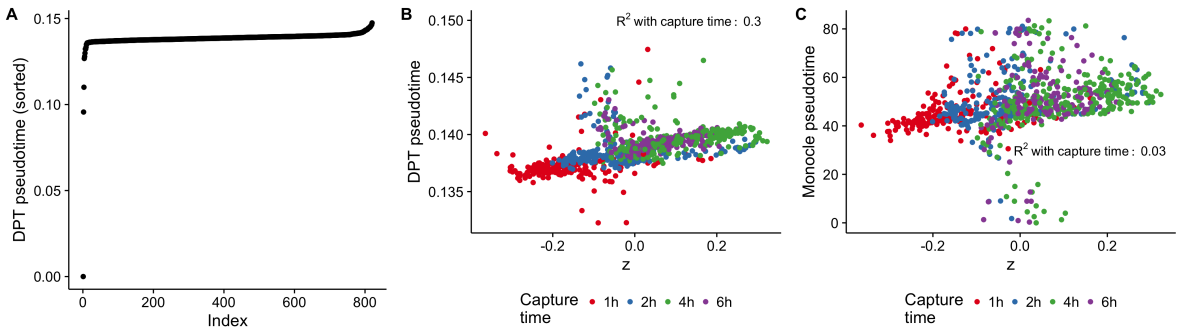


Figure 49: Performance of DPT and Monocle 2 on Shalek et al dataset. **A** Sorted DPT pseudotimes by index identifies three outlier cells. **B** Comparison of DPT pseudotimes to PhenoPath pathway score z. **C** Comparison of Monocle 2 pseudotimes to PhenoPath pathway score z.

scriptional heterogeneity at each time point is still evident. We also compared this to two commonly-used pseudotime algorithms and found that the pseudotimes inferred using PhenoPath had the best agreement with the capture times (figure 49).

Using PhenoPath we found a large number of stimulant-modulated interactions masked by standard differential-expression analysis (figure 48C). A GO analysis revealed genes whose upregulation along the common trajectory was increased by LPS exposure (as opposed to PAM) were highly enriched for immune response (figure 48D), which recapitulates previous results [112, 119] that suggest a "core" module of antiviral genes upregulated at later timepoints in LPS cells but in an entirely unsupervised, integrated manner. We finally examined the individual genes most perturbed by LPS or PAM along the trajectory (figure 48E), which identifies as yet uncharacterised expression patterns associated with LPS and PAM. Most notably, the tumour necrosis factor *Tnf* had around twice the interaction effect size of any other gene, and decreases under LPS stimulation but increases under PAM. Further genes exhibit differential regulation according to stimulant, such as *Mef2c* that has constant expression over pseudotime under LPS stimulation yet shows downregulation under PAM stimulation. These results com-

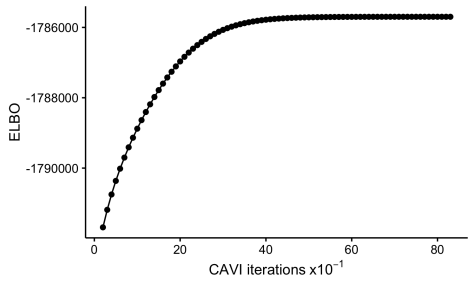


Figure 50: Evidence lower bound (ELBO) as a function of CAVI iterations for the COAD RNA-seq dataset.

plement previously discovered gene differences such as that of *Tnf*, but in a systematic, transcriptome-wide approach.

The Shalek et al. dataset of time-series dendritic cells was previously used in a pseudotime analysis where the capture times were explicitly used as priors on the latent space [112]. However, in PhenoPath we provide no explicit temporal information, so sought to perform a brief comparison to two popular pseudotime algorithms, Monocle 2 [108] and DPT [46]. For both methods we provided the same normalised log expression (see appendix B.4) and ran the algorithms with the default parameters. Performance of each algorithm was assessed by regressing the inferred pseudotimes on the capture times using the `R` function `lm` and computing the R^2. Oddly DPT has several "outlying" pseudotimes at the beginning of the trajectory (figure 49) that we removed for both visualisation purposes and to compute a comparable R^2. The results gave an R^2 of 0.30 and 0.03 with the true capture time for DPT and Monocle 2 respectively, meaning PhenoPaths $R^2 = 0.64$ is significantly larger, perhaps because it accounts for differing stimulants applied.

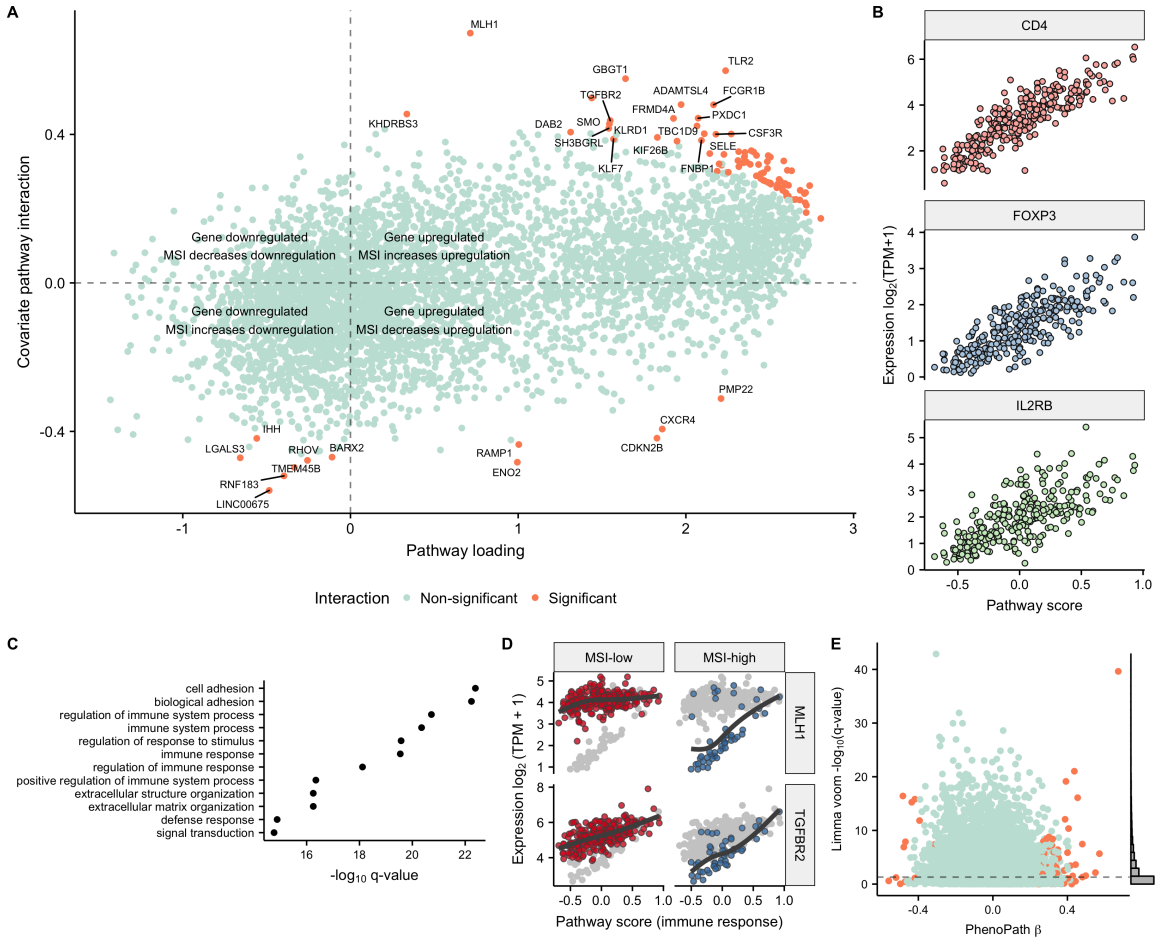


Figure 51: Immune-microsatellite instability interactions uncovered in colorectal adenocarcinoma. **A** PhenoPath applied to colorectal adenocarcinoma (COAD) RNA-seq expression data uncovers a landscape of interactions between the inferred immune trajectory and microsatellite instability status (MSI). **B** Expression of three T regulatory cell markers demonstrates that our pseudotime corresponds to activation of immune response pathways. **C** A comparison to the FDR-corrected q-values reported by Limma Voom demonstrates genes found interacting with MSI status and the immune pathway are found to be both DE and non-DE in standard analyses. **D** A GO enrichment analysis of upregulated genes implies the latent trajectory encodes immune pathway activation in each tumour. **E** The tumour suppressor genes *MLH1* and *TGFBR2* were identified by our method as being significantly perturbed along the immune trajectory by MSI status. *MLH1* shows no interaction with immune pathway activation in the MSI-low regime yet is highly correlated with immune pathway activation in the MSI-high regime.

5.3.2 *Colorectal cancer bulk RNA-seq*

We next applied our model to the RNA-seq gene expression data from the TCGA colorectal adenocarcinoma (COAD) cohort [96] using microsatellite instability status (MSI) as a phenotypic covariate. MSI is genetic hypermutability that is present in around 15% of colorectal tumours and is associated with differential response to chemotherapeutics and marginally improved prognosis [10]. Due to a significant technical effect in the dataset we removed around half of samples (see appendix B.4), applying PhenoPath to 4,801 highly variable genes across 284 samples to identify a pseudotemporal trajectory through the tumours. PhenoPath was run for approximately 800 CAVI iterations until convergence of the ELBO (figure 50).

This analysis uncovered a landscape of 92 pathway-MSI interactions including known tumour suppressor genes (figure 51A). Patients further advanced along the trajectory exhibited higher expression of T regulatory cell (Tregs) immune markers (figure 51B) likely due to increasing T regulatory cell infiltration of the tumour. This led us to hypothesise that the inferred pathway corresponds to immune response activation in the tumours, further supported by a Gene Ontology (GO) enrichment analysis for genes upregulated along the trajectory (figure 51C). Tumour-infiltrating Tregs are potent immunosuppressive cells of the immune system that promote progression of cancer through their ability to limit antitumuor immunity and promote angiogenesis and often associated with a poor clinical outcome [33]. A standard differential expression analysis using Limma Voom [68] (figure 51D) demonstrates that PhenoPath is required to uncover such interactions as a gene being differentially expressed does not imply a pathway-MSI interaction, while such interactions do not require differential expression.

The most striking interaction discovered for this dataset was the *MLH1* gene whose interaction effect size was far larger than any other gene. *MLH1* is a DNA mismatch

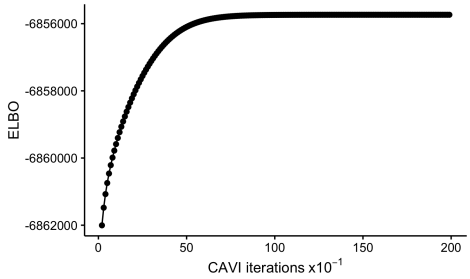


Figure 52: Evidence lower bound (ELBO) as a function of CAVI iterations for the BRCA RNA-seq dataset.

repair gene, germline mutations of which are causal for hereditary non-polyposis colorectal cancer [11, 41]. By applying PhenoPath we correctly identified that in patients with low or absent levels of microsatellite instability there is no relationship between *MLH1* expression and immune pathway interaction, with *MLH1* expressed at an approximately constant level (figure 51E). However, when MSI occurs in a tumour, *MLH1* expression is highly correlated with immune response, showing almost no expression when the immune pathway is inactive and gradually being upregulated with immune pathway response [89].

5.3.3 *Breast cancer bulk RNA-seq*

We next performed a pseudotemporal analysis of the TCGA breast cancer cohort using estrogen receptor (ER) status as a phenotypic covariate. Approximately 60% of breast cancers are estrogen receptor positive [31], which is typically associated with improved prognosis and a longer time to recurrence [98]. We applied PhenoPath to 1,135 samples post-QC and 4,579 highly variable genes. PhenoPath was run for approximately 2000 CAVI iterations until convergence of the ELBO (figure 52). Using our Bayesian significance testing criterion (section 5.2.3) we found 1,932 genes (42%) affected by an interaction between the pseudotemporal trajectory and ER receptor status (figure 53A).

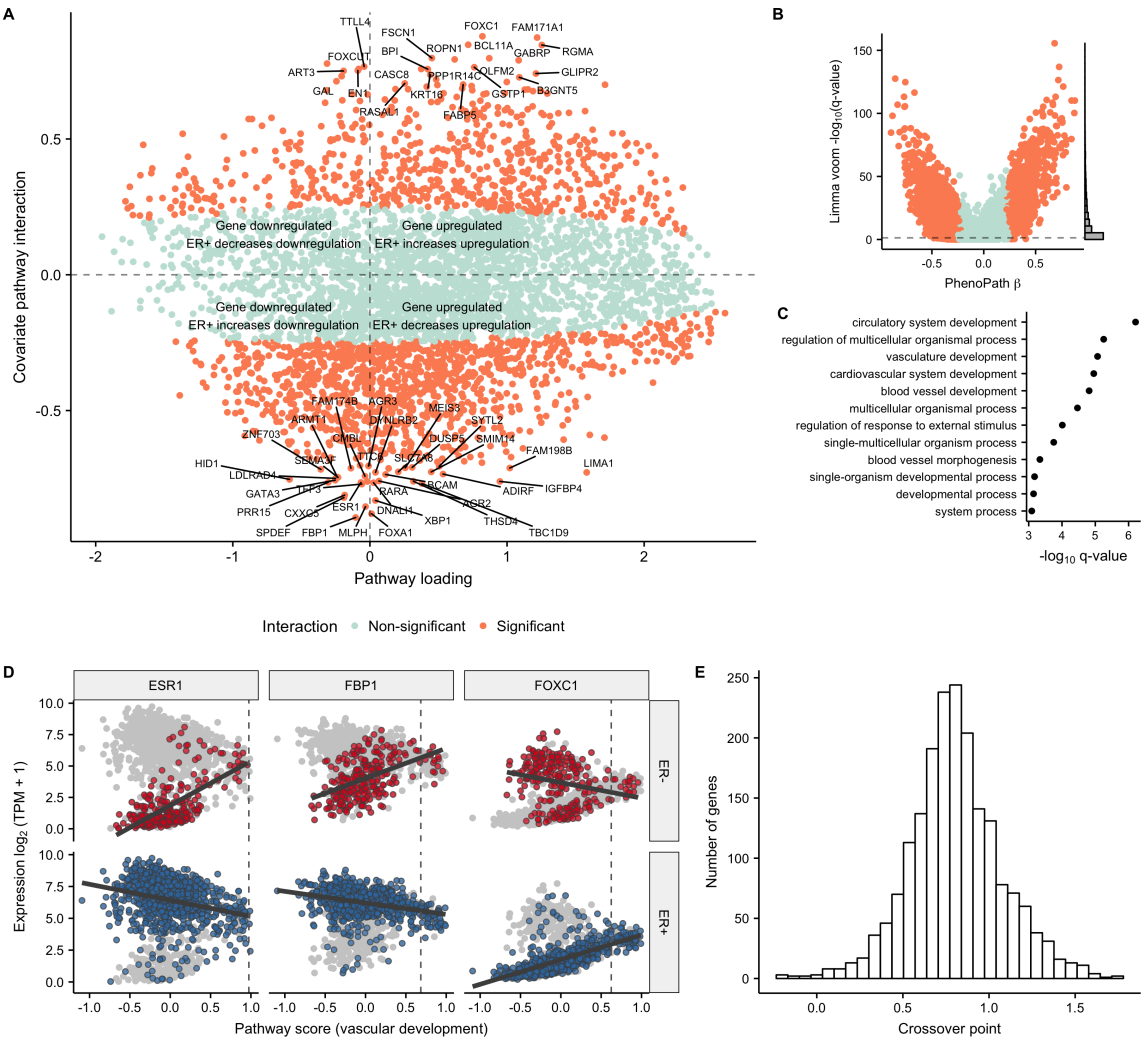


Figure 53: Vascular growth-ER status interactions uncovered by PhenoPath in breast cancer. **A** PhenoPath applied to Breast Cancer (BRCA) RNA-seq expression data uncovers a landscape of interactions between the inferred angiogenesis trajectory and estrogen receptor (ER) status. **B** A comparison to the FDR-corrected q-values reported by Limma Voom identifies a significant number of DE genes display an interaction with ER status and the angiogenic pathway. **C** A GO enrichment analysis of upregulated genes implies the latent trajectory encodes angiogenesis pathway activation in each tumour. **D** Four example genes *ESR1*, *FBP1*, and *FOXC1* were identified by PhenoPath as significantly perturbed along the angiogenesis trajectory by ER status. The vertical dashed line signifies the calculated crossover point, demonstrating the expression profiles of these genes converge towards the end of the trajectory. **E** A histogram of the crossover points of all genes whose trajectory-covariate interactions were significant. The vast majority of crossover points are at the end of the trajectory (around 0.5, where the "middle" pathway score is 0) implying a convergence of gene expression as the trajectory progresses.

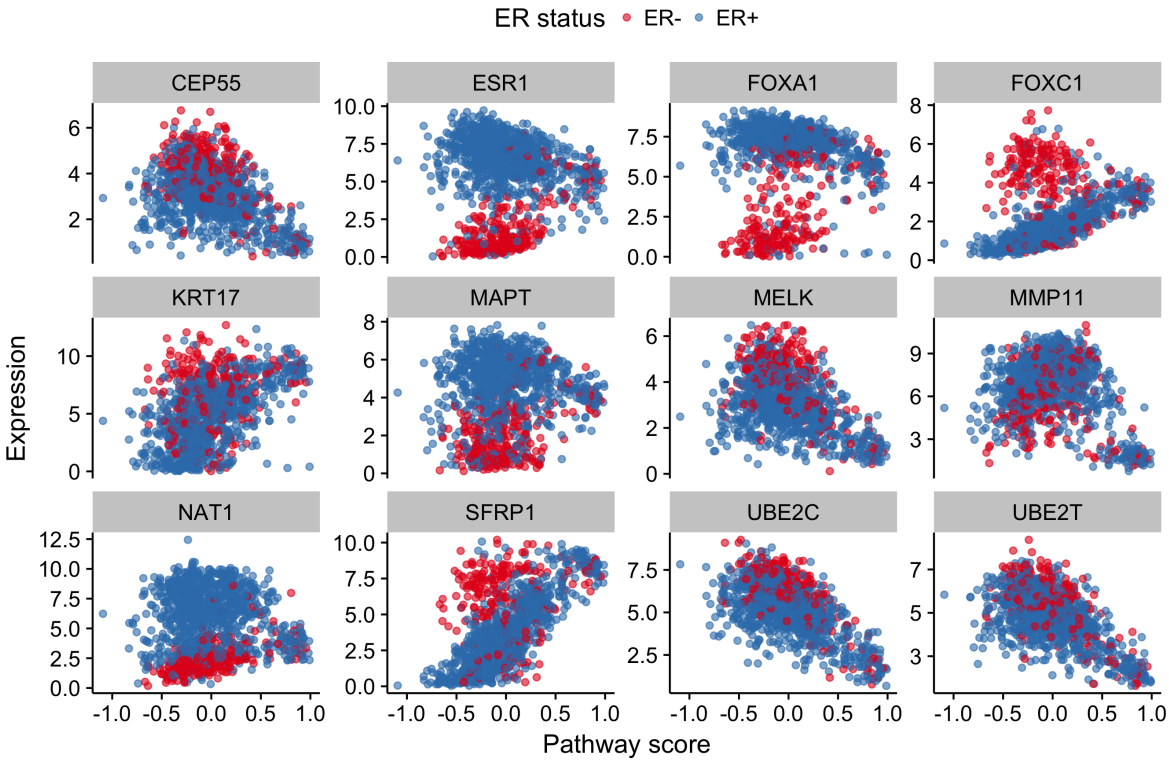


Figure 54: Pseudotemporally ordered gene expression trajectories for the TCGA Breast Cancer data for 12 breast cancer-associated genes.

There was a correlation between the pathway interaction strength and the p-value reported through standard differential expression (figure 53B), though there remained some genes that exhibited pathway interaction and no differential expression.

A GO enrichment analysis indicated that the inferred pseudotemporal trajectory corresponded to vascular growth pathways or *angiogenesis* (figure 53C) – a well-known and uncontroversial hallmark of cancer development [34, 139]. We confirmed this finding by specifically examining the expression of known angiogenesis inducing genes (figure 55). We found increasing fibroblast growth factor-2 (*FGF-2*) and vascular endothelial growth factors C and D (*VEGF-C/D*) expression along the trajectory whose behaviours were independent of ER status.

We finally sought to examine some genes PhenoPath identified as being most affected by the interaction between angiogenesis and estrogen receptor status. Importantly, this

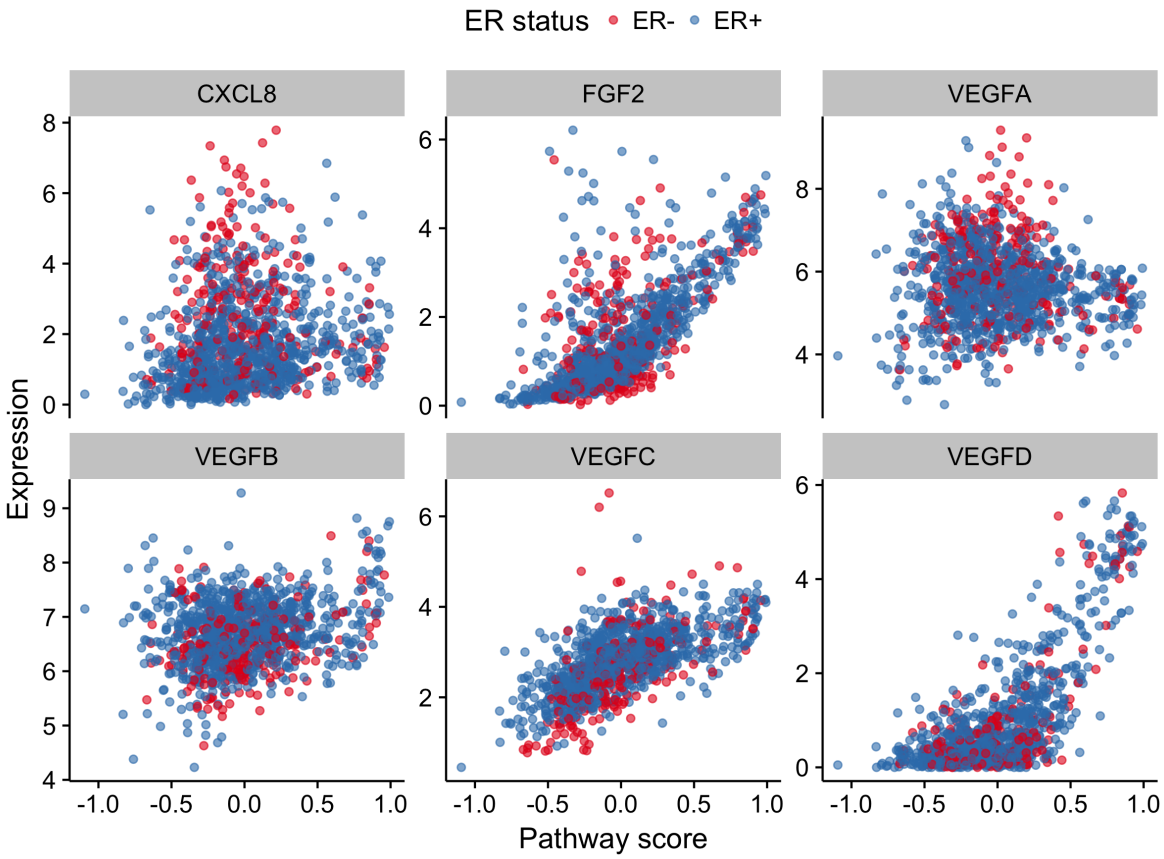


Figure 55: Pseudotemporally ordered gene expression trajectories for the TCGA Breast Cancer data for six angiogenesis-associated genes.

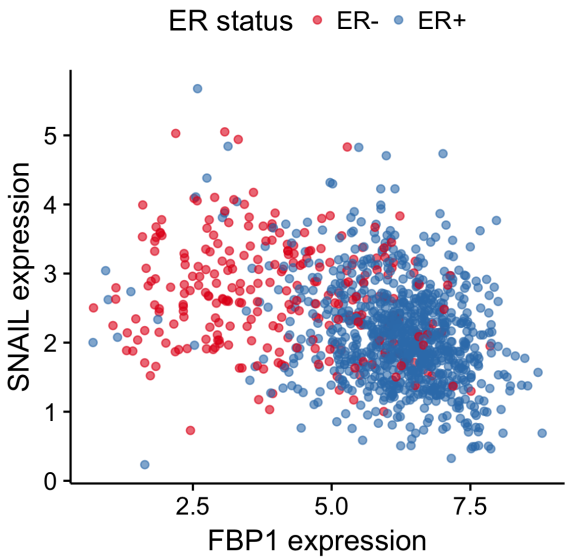


Figure 56: *FBP1* expression is inversely correlated with Snail in ER- breast cancers but shows no dependence in ER+ breast cancers.

set included the Estrogen Receptor 1 (*ESR1*) gene as well as the forkhead transcription factors *FOXA1* and *FOXC1* which are known to be involved with ERα mediated action in breast cancer [67, 144] (figures 53D and 54). Figure 53D shows how the ructose-1,6-biphosphatase (*FBP1*) and *FOXC1* genes evolve along the angiogenesis pathway dependent on ER status. In the ER- regime, *FBP1* is upregulated along the trajectory while in the ER+ regime it is downregulated. Intriguingly, *FBP1* has been identified as a marker to distinguish ER+ from ER- subtypes and its expression has been shown to be negatively correlated with *SNAIL* as the Snail-G9a-Dnmt1 complex, is critical for E-cadherin promoter silencing, and required for the promoter methylation of FBP1 in basal-like breast cancer (figure 56) [30]. Similarly, *FOXC1* shows no regulation in the ER- regime yet is strongly upregulated in the ER+ case.

We noted that these genes represent a convergence - they have markedly different expression at the beginning of the trajectory based on ER status yet converge towards the end. Using our previous formula for the crossover point (section 5.2.4), we found the vast majority converge towards the end of the trajectory (figure 53E), implying a common

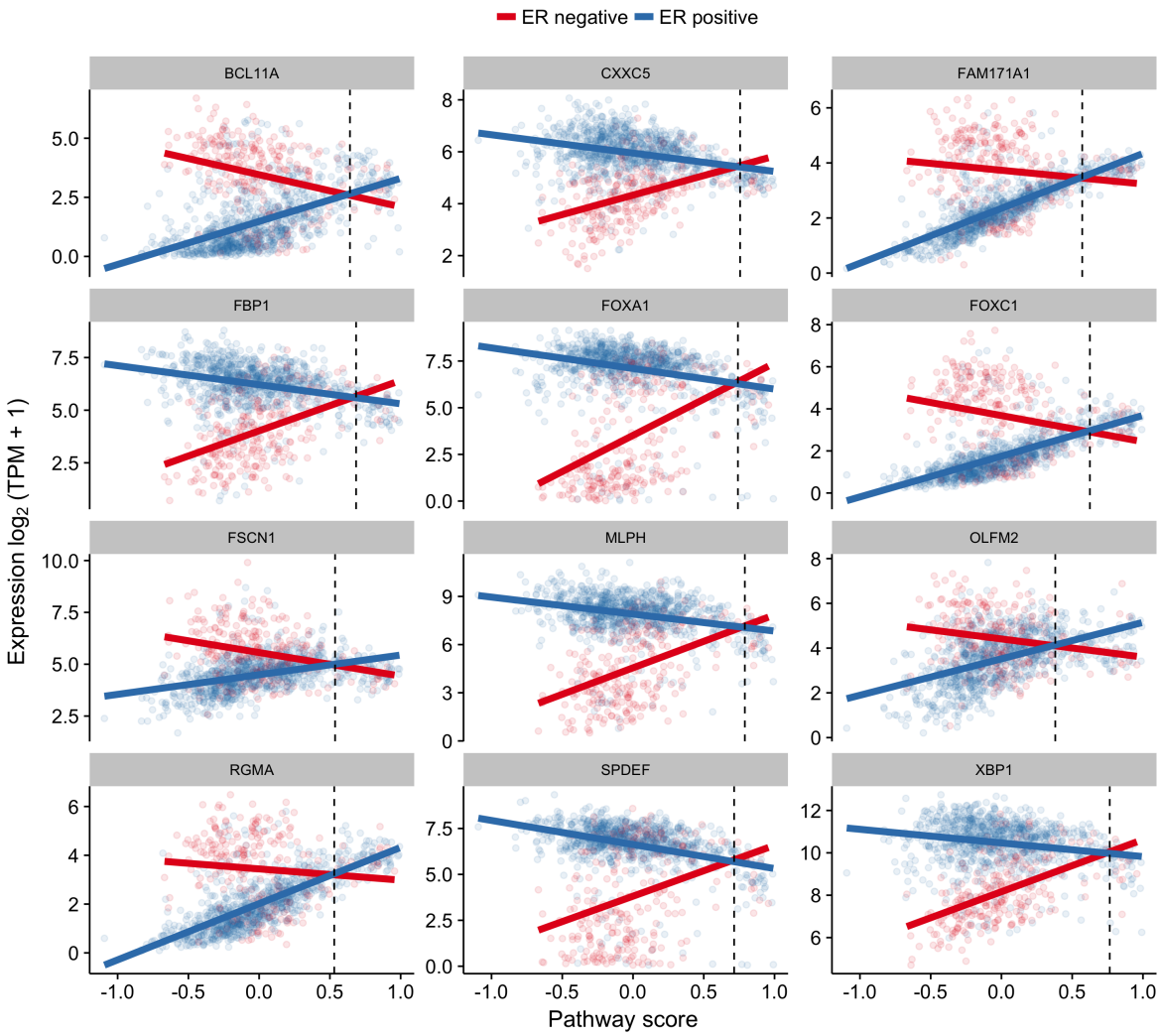


Figure 57: Expression of 20 genes with the largest interaction effects along the inferred pseudotemporal trajectory coloured by estrogen receptor status with linear fits as solid lines. The vertical dashed line indicates the crossover point.

end-point in vascular development for both ER+ and ER- cancer subtypes. This effect can be seen in the example expression plots in figures 53D and 57, where the vertical dashed line represents the convergence point always at the end of the trajectory. This suggests that while there exists low levels of angiogenesis pathway activation, ER status dominates gene expression while as angiogenesis pathway activation increases it comes to dominate expression patterns over ER status. This finding might have implications for the application of angiogenesis inhibitors in breast cancer treatment.

5.4 PERTURBATIONS BY CENSORED SURVIVAL TIMES

5.4.1 *Modified statistical model*

We now suppose that $\mathbf{x}_n$ are not fully observed but represent partially observed survival times. In particular for each sample we observe the tuple $(o_n, \delta_n),\ n = 1, \ldots, N$ where o_n is a survival time and δ_n is the censoring status, where $\delta_n = 1$ if o_n is observed and $\delta_n = 0$ if o_n is censored. We also introduce the true survival times $t_n,\ n = 1, \ldots, N$ where $t_n = o_n$ if $\delta_n = 1$ and $t_n > o_n$ if $\delta_n = 0$. In this $P = 1$ form let $\boldsymbol{\alpha} \equiv \mathbf{A}_{:,1}$ and $\boldsymbol{\beta} \equiv \mathbf{B}_{:,1}$ be G-length vectors. The survival-adjusted latent variable model then takes the form

$$\mathbf{y}_n \sim \mathcal{N}\left(\boldsymbol{\alpha} h(t_n) + (\boldsymbol{\lambda} + \boldsymbol{\beta} h(t_n)) z_n, \boldsymbol{\Sigma}\right). \tag{84}$$

where $h(t)$ is any desirable mapping function such as one that induces a variance-stabilizing transformation.

If all the survival times t_n were uncensored then such a model reduces to the form in equation 70 and inference can proceed as before. However, in many biomedical settings the majority of observations are censored, rendering such a problem intractable. We therefore turn our attention to creating a generative model of the censored true survival times that makes use of both the observed survival times and censoring times.

A classic choice for modelling survival times is the Weibull distribution parametrised by $\boldsymbol{\theta}_W = (k, p)$ with probability density function

$$f_{\text{Weibull}}(t; k, p) = pkt^{k-1} \exp(-pt^k) \tag{85}$$

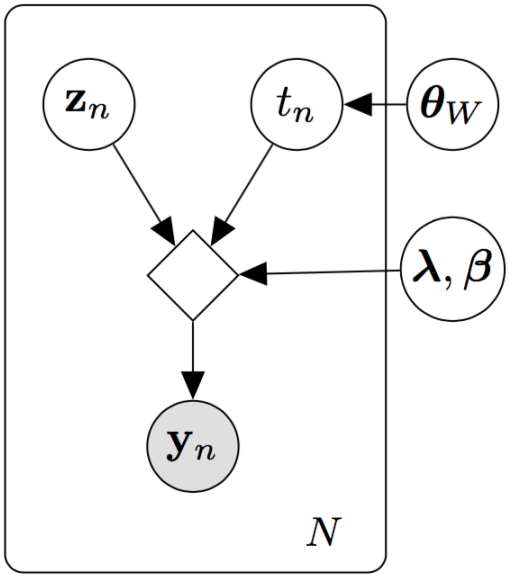


Figure 58: Survival-adjusted latent variable model, with data $\mathbf{y}_n$, latent variables z_n, survival times t_n, and global parameters $\boldsymbol{\theta}_W$, $\boldsymbol{\beta}$, and $\boldsymbol{\lambda}$. The diamond represents the deterministic transform of the parameters to produce the observed data with sampling noise.

and survival function $S_{\text{Weibull}}(t; k, p) = \int_t^\infty f(t'; k, p)dt' = \exp(-pt^k)$. We can then write the probability density of the true survival times as

$$f(t_n; o_n, k, p) = \begin{cases} f_{\text{Weibull}}(t_n; k, p), & \text{if } \delta_n = 1 \\ \frac{f_{\text{Weibull}}(t_n; k, p)}{S_{\text{Weibull}}(o_n; k, p)}, & \text{if } \delta_n = 0 \end{cases} \tag{86}$$

where observed survival times are drawn from a Weibull distribution while censored survival times are drawn from a Weibull distribution truncated below at the censoring times (i.e. $p(t_n \leq o_n) = 0$ if $\delta_n = 0$).

The overall generative model is represented in figure 58. We place a further Sparse Bayesian Learning prior $\alpha_g \sim \mathcal{N}(0, \eta_g^{-1})$, $\alpha_g \overset{iid}{\sim} \text{Gam}(a_\alpha, b_\alpha)$ and set $a_\alpha = b_\alpha = a_\beta = b_\beta = 10^{-2}$.

5.4.2 *Inference*

Our goal is Bayesian inference of the posterior $p(\mathbf{z}, \mathbf{t}^{cens}, \mathbf{A}, \mathbf{B}, \boldsymbol{\lambda}, \theta_W | \mathbf{Y}, \mathbf{o}, \mathbf{t}^{obs}, \Psi)$ where Ψ is the complete set of fixed hyperparameters. If we have $N_{\text{obs}} = \sum_{n=1}^{N} \delta_n$ observed survival times and $N_{\text{cens}} = N - N_{\text{obs}}$ censored observations, we may reorder $\mathbf{t}$, $\mathbf{o}$, $\boldsymbol{\delta}$, $\mathbf{z}$ and the rows of $\mathbf{Y}$ so that the first N_{obs} entries of $\boldsymbol{\delta}$ are 1 and the remaining N_{cens} are 0. Then define $\mathbf{t}^{\text{cens}}$ and $\mathbf{o}^{\text{cens}}$ as the vector made of the first N_{cens} entries of $\mathbf{t}$ and $\mathbf{o}$ respectively, and define $\mathbf{t}^{\text{obs}}$ as the vector of the final N_{obs} entries of $\mathbf{t}$. Then the posterior factorises in the form

$$\begin{aligned} p(\mathbf{z}, \mathbf{t}^{\text{cens}}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda}, \theta_W, \boldsymbol{\sigma}^2, \boldsymbol{\chi} | \mathbf{Y}, \mathbf{o}, \mathbf{t}^{\text{obs}}, \Psi) \propto\ & p(\mathbf{Y} | \mathbf{z}, \mathbf{t}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\sigma}^2) \\ & \times p(\mathbf{t}^{\text{obs}} | \boldsymbol{\theta}_W) p(\mathbf{t}^{\text{cens}} | \mathbf{o}^{\text{cens}}, \boldsymbol{\theta}_W) \\ & \times p(\boldsymbol{\alpha} | \boldsymbol{\eta}) p(\boldsymbol{\beta} | \boldsymbol{\chi}) p(\boldsymbol{\eta}) p(\boldsymbol{\chi}) p(\boldsymbol{\lambda}) \\ & \times p(\mathbf{z}) p(\boldsymbol{\theta}_W) p(\boldsymbol{\sigma}) \end{aligned} \tag{87}$$

where we have omitted any dependency on hyperparameters.

5.4.2.1 *Black-box variational inference*

To infer posterior distributions over the entire set of latent variables

$$\boldsymbol{\Theta} = \{\mathbf{z}, \mathbf{t}^{\text{cens}}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda}, \theta_W, \boldsymbol{\sigma}^2, \boldsymbol{\chi}\} \tag{88}$$

we posit a set of approximating distributions $q(\boldsymbol{\Theta} | \boldsymbol{\Phi})$ with variational parameters $\boldsymbol{\Phi}$. Inference then proceeds by finding estimates of $\boldsymbol{\Phi}$ by minimising

$$\text{KL}\left(q(\boldsymbol{\Theta} | \boldsymbol{\Phi}) \parallel p(\mathbf{Y}, \mathbf{o}^{\text{cens}}, \mathbf{t}^{\text{obs}}, \boldsymbol{\Theta}) \right), \tag{89}$$

which is equivalent to maximising the Evidence Lower Bound (ELBO) defined as $\text{ELBO}(\boldsymbol{\Theta}) = \mathbf{E}_{q(\boldsymbol{\Theta})}\left[p(\boldsymbol{\Theta}, \mathbf{Y}, \mathbf{o}^{\text{cens}}, \mathbf{t}^{\text{obs}}) - q(\boldsymbol{\Theta}|\boldsymbol{\Phi}))\right]$, with respect to the variational parameters $\boldsymbol{\Phi}$ for which we make a fully factorised mean-field approximation $q(\boldsymbol{\Theta}|\boldsymbol{\Phi}) = \prod_i q_i(\theta_i|\phi_i)$. Inference of this model was implemented using the probabilistic programming language `Edward` [127] that allows for black-box variational inference of non-conditionally-conjugate models. The approximating distributions have reparameterizations (section 5.4.2.2), allowing us to sample from them via $\epsilon \sim q(\epsilon),\ \theta_i = g_i(\epsilon, \phi_i)$ where $q(\epsilon)$ is a distribution independent of the variational parameters and g_i is a deterministic transform. This allows us to use the reparametrization trick [60] to compute noisy estimates of the gradient via

$$\nabla_{\boldsymbol{\Phi}}\text{ELBO}(\boldsymbol{\Phi}) = \mathbf{E}_{q(\boldsymbol{\epsilon})}\left[\nabla_{\boldsymbol{\Phi}}\left(\log p(\mathbf{g}(\boldsymbol{\epsilon}, \boldsymbol{\Phi}), \mathbf{Y}, \mathbf{o}^{\text{cens}}, \mathbf{t}^{\text{obs}}) - \log q(\mathbf{g}(\boldsymbol{\epsilon}, \boldsymbol{\Phi})|\boldsymbol{\Phi})\right)\right] \quad (90)$$

where the expectation is computed via Monte-Carlo sampling.

5.4.2.2 *Choice of approximating distributions*

For $\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{z}$, and $\boldsymbol{\lambda}$ we posit variational approximations of the form $q(\theta_i) \sim \mathcal{N}(\mu_i, \sigma_i^2)$ which admits reparameterization via $\epsilon \sim \mathcal{N}(0, 1),\ \theta_i = \mu_i + \sigma_i\epsilon$. The variables $\boldsymbol{\chi}$, $\boldsymbol{\sigma}^2$, and p are constrained to be positive, so we choose log-normal approximating distributions which is equivalent to the reparametrization $\epsilon \sim \mathcal{N}(0, 1),\ \theta_i = \exp(\mu_i + \sigma_i\epsilon)$.

The censored true survival times $\mathbf{t}^{\text{cens}}$ are bounded from below by the observed survival times, ie $t_n^{\text{cens}} > o_n$. However, in practice they are also upper bounded by the patient's maximum possible months remaining alive r_n. We can estimate r_n for each patient by calculating the time from their date of diagnosis to the maximum possible human

lifespan, which we take to be 115 years. We subsequently construct a reparametrised variational approximation with variational parameters $(\mu_{t_n}, \sigma_{t_n})$:

$$\begin{aligned} \epsilon &\sim \mathcal{N}(0,1) \\ t_n &= g_t(\epsilon|\mu_{t_n}, \sigma_{t_n}) = \frac{o_n + r_n e^{\mu_{t_n} + \sigma_{t_n}\epsilon}}{1 + e^{\mu_{t_n} + \sigma_{t_n}\epsilon}} \end{aligned} \tag{91}$$

where $g_t(\epsilon) \in [o_n, r_n)$. Evaluation of $q(t_n|\mu_{t_n}, \sigma_{t_n}, o_n, r_n)$ is then made via

$$q(t_n|\mu_{t_n}, \sigma_{t_n}, o_n, r_n) = \mathcal{N}(\epsilon|0,1) \left| \frac{\partial g_t^{-1}(\epsilon|\mu_{t_n}, \sigma_{t_n})}{\partial \epsilon} \right|. \tag{92}$$

The variable k parametrizing the Weibull distribution is constrained to be greater than zero. It further has an intuitive interpretation: for $k < 1$ the failure rate decreases over time, for $k = 1$ the failure rate is constant, while for $k > 1$ the survival rate increases over time. For most biomedical applications it is known the survival rate increases over time, so we may constrain $k > 1$. However, k appears in the log-likelihood via $\sum_n t_n^k$, where t_n is both the observed and censored true survival times. Thus in settings where t_n may exceed orders of 10^2 and k sampled from a log-normal distribution, t_n^k leads to highly unstable estimates of the log-likelihood and can easily overflow the machine precision. Thus we constrain k to be on the interval $[a, b)$ via the same reparametrization as in equation 91 and set $a = 1$ and $b = 3$.

5.4.3 *Application to breast cancer bulk RNA-seq*

We applied our model to RNA sequencing data and survival times from breast cancer samples in the Cancer Genome Atlas [123, 137]. This resource includes 1137 patients of which 180 (16%) have recorded survival times and the remaining 957 have censored survival times. Gene expression estimates from RNA sequencing counts have been shown

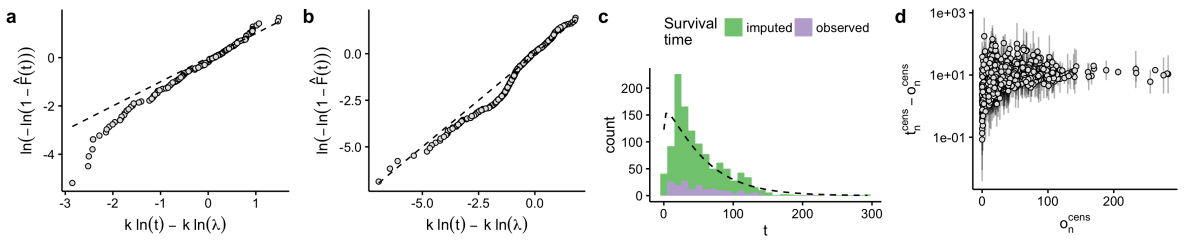


Figure 59: Imputation of survival times. Weibull plots of (**a**) observed and (**b**) imputed survival times t_n, where $\hat{F}$ is the empirical cumulative distribution function for the MAP estimates of k and $\lambda = p^{-k}$. If the survival times were truly drawn from the fitted Weibull distribution the points would fall along the diagonal $y = x$ (dashed line). **(c)** Histograms of the observed and imputed survival times t_n along with the unnormalised Weibull density (dashed line) at the MAP estimates of $\boldsymbol{\theta}_W$ shown by the dashed line. **(d)** Deviations of the imputed survival times t_n^{cens} compared to their censoring times o_n^{cens} as a function of censoring times. Error bars show the posterior 2σ interval.

to follow a log-normal distribution [68, 103], so our input data consisted of transforming the raw transcripts-per-million x via $\tilde{x} = \log(x+1)$. The survival times (both observed and censored) followed a highly right-skewed distribution so we chose $h(t) = \log(t+1)$ as a variance-stabilising transformation.

5.4.3.1 *Imputation of survival times*

We sought to assess the suitability of the Weibull distribution for modelling survival times compared to nonparametric alternatives such as Cox proportional hazards model using a *Weibull plot*. Here, if the survival times are truly drawn from the Weibull distribution then the graph of $\log(-\log(1-\hat{F}(t)))$ against $k\log(t) - k\log(\lambda)$ will fall along the diagonal $y = x$, where $\hat{F}$ is the empirical distribution function and $\lambda = p^{-k}$.

Weibull plots for both $\mathbf{t}^{\text{obs}}$ and $\mathbf{t}^{\text{cens}}$ are shown in figures 59 (**a**) and (**b**) respectively for the MAP estimates of k and p. The data fit the Weibull distribution well falling close to the dashed diagonal $y = x$ line, except in the case of the imputed survival times for low t_n where we see deviations. An further alternative representation is shown by the

histogram in figure 59(**c**).We also examined the MAP estimates of the imputed survival times t_n^{cens} compared to the censoring times o_n^{cens} as seen in figure 59(d). The imputed t_n^{cens} typically lie close to the observed o_n^{cens}, which is expected given the log-normal approximating distribution $q(t_n)$ constrained below at o_n.

5.4.3.2 *Patient trajectory interactions*

We first compared our inferred latent variables z_n to the first principal component of the data. Setting $\alpha_g = \beta_g = 0 \; \forall g$ (which is encouraged by the ARD prior) will recover a rank-one factor analysis in the model, which in the limit of isotropic measurement variance reduces to (probabilistic) principal component analysis (PCA). Therefore, such a comparison serves to qualitatively identify the effect to which the covariates perturb the trajectory.

The comparison can be seen in figure 60(**a**) coloured by the logarithm of the true survival time. We can visually identify that for longest survival times the latent trajectory largely follows that of principal component analysis, while for short survival times it deviates substantially. In case such a result was driven by a few outlying samples we removed the patients whose event times (survival or censoring) were less than 1 (38/1137) and repeated the analysis, finding good correlation between the values of $\boldsymbol{\beta}$ ($\rho_{\text{pearson}} = 0.96$).

We subsequently performed a gene ontology (GO) enrichment analysis to identify the biological characteristics of the inferred trajectory. Many genes in the human genome are annotated with *ontologies* that describe biological processes the genes are involved in. We can therefore perform statistical tests [143] that identify which ontologies are over-represented in a given set compared to a background population alone. Performing such a test on genes whose Pearson correlation of expression with the inferred trajectory exceeded 0.5 identified processes related to vascular growth, also known as angiogenesis.

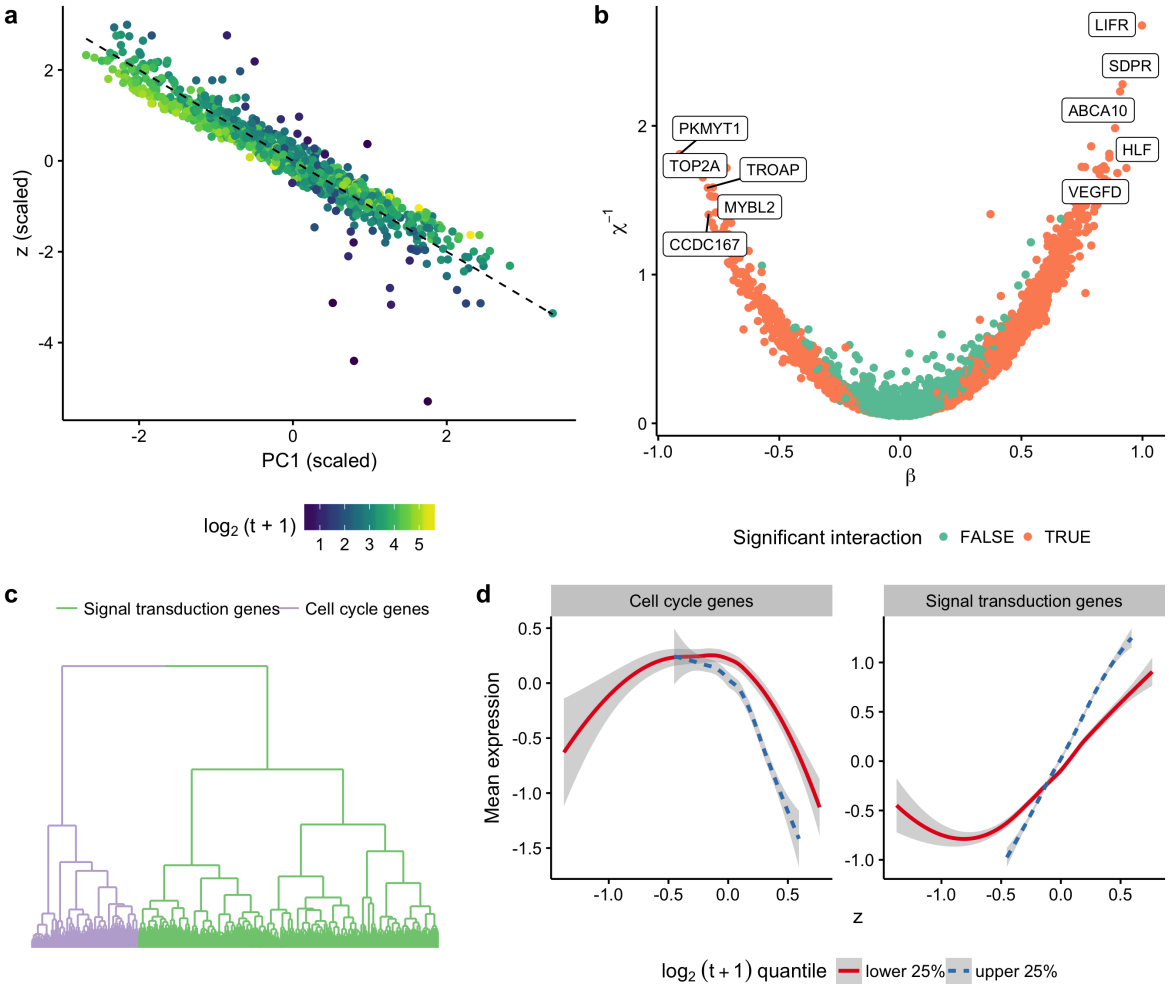


Figure 60: Results on breast cancer gene expression data. (**a**) Comparison of the inferred latent variables z_n to the first principal component of the data. Different trajectories are visible for long survival times (which largely mimics PC1) to short survival times, which departs from PC1 on a separate trajectory. (**b**) A plot of MAP χ_g^{-1} versus β_g identifies significant interactions between the inferred latent variables and survival times. (**c**) A hierarchical clustering of expression of genes designated significant from (**b**) reveals two main classes of expression programme over along z_n, segregating into those involved in signal transduction and cell cycle dependent genes. (**d**) LOESS fits of mean standardized expression of each gene type from (**c**) stratified into patients in the lowest quantile of survival times compared to the highest quantile of survival times.

Therefore, the trajectory z_n loosely corresponds to a continuous score of angiogenesis pathway activation in each of the breast cancer tumours.

Figure 60(**b**) shows the posterior estimates of χ_g^{-1} against the posterior estimates of β_g coloured by whether the interaction is identified as significant. A total of 2,371 genes (42% of those included) exhibited a significant interaction between the inferred angiogenesis trajectory and the survival times of the patients. Many of the genes with the largest β effect sizes are previously implicated in cancer progression: *SDPR* acts as a metastasis suppressor [97], *PRMT1* promotes mitosis in cancer cells [28] and modulates the epithelial-mesenchymal transition and cellular senescence in breast cancer cells [38], while *TOP2A* is a predictive marker of chemotherapy efficacy [135].

We next performed hierarchical clustering on the expression profiles of the 2,371 genes that showed significant interactions between angiogensis and survival time, as shown in figure 60(**c**). This identified two main classes of genes whose expression programmes followed correlated expression patterns. A further GO analysis individually on each gene set revealed their biological relevance. The first contains genes involved in signal transduction - the cell to cell communication through the transmission of signalling molecules - dysregulation of which is a major cause of cancer progression [118]. The second set contains genes associated with cell cycle - the sequence of events leading to cell division that leads to cancerous growth when uncontrolled. We subsequently formed smoothed expression profiles of each gene set along the vascular growth trajectory, stratified by those patients in the lowest quantile of survival times and those in the highest (figure 60(**d**)). For patients with the shortest survival times the signal transduction pathway exhibits near constant expression for low z_n before gradual upregulation, while for the longest survival times the gene set is expressed at later z_n but experiences quicker upregulation. Similarly, for the cell cycle genes patients with the shortest survival times

experience transient expression across z_n with slow downregulation, while patients with the longest survival times experience quick downregulation along z_n.

5.5 COVARIATE-ADJUSTED GAUSSIAN PROCESS LATENT VARIABLE MODELS

5.5.1 *Marginalising over the mapping*

One limitation of our proposed covariate-adjusted latent variable models is their linear nature, making inferred latent variables similar to those from factor analysis. We therefore propose a nonlinear, nonparametric extension similar to Gaussian Process Latent Variable Models [70] that were previously discussed in section 1.3.5.

We wish to marginalise over the mapping $\{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda}\}$ with priors $\alpha_g \sim \mathcal{N}(0, \sigma^2_{\alpha,g})$, $\beta_g \sim \mathcal{N}(0, \sigma^2_{\beta,g})$, $\lambda_g \sim \mathcal{N}(0, \sigma^2_{\lambda,g})$. This leads to a multivariate normal likelihood for each $\mathbf{y}_g$ (where $\mathbf{y}_g$ is the g^{th} column vector of $\mathbf{Y}$) given by

$$\mathbf{y}_g|\cdot \sim \mathcal{N}\left(\mathbf{0}, \sigma^2_{\alpha,g}\mathbf{x}\mathbf{x}^T + \sigma^2_{\lambda,g}\mathbf{z}\mathbf{z}^T + \sigma^2_{\beta,g}(\mathbf{x}\odot\mathbf{z})(\mathbf{x}\odot\mathbf{z})^T + \sigma^2_g\mathbf{I}_N\right) \tag{93}$$

where $\odot$ denotes element-wise multiplication ($[\mathbf{x}\odot\mathbf{y}]_i = x_i y_i$), $\mathbf{I}_N$ is the $N \times N$ identity matrix and σ_g is the residual (measurement) variance not explained by the model.

The trick to derive a Gaussian Process Latent Variable-like model from here is two-fold. Firstly, Lawrence (2005) [70] noted that entries in the low-rank covariance structure $\mathbf{x}\mathbf{x}^T$ loosely represent "similarity" between samples and can thus be replaced by any kernel $K(x, x')$. Secondly, the sum of kernels is also a valid kernel so if $K_1(x, x')$ and $K_2(x, x')$ are two kernels then $\alpha K_1(x, x') + \beta K_1(x, x')$ also represents a valid kernel for some

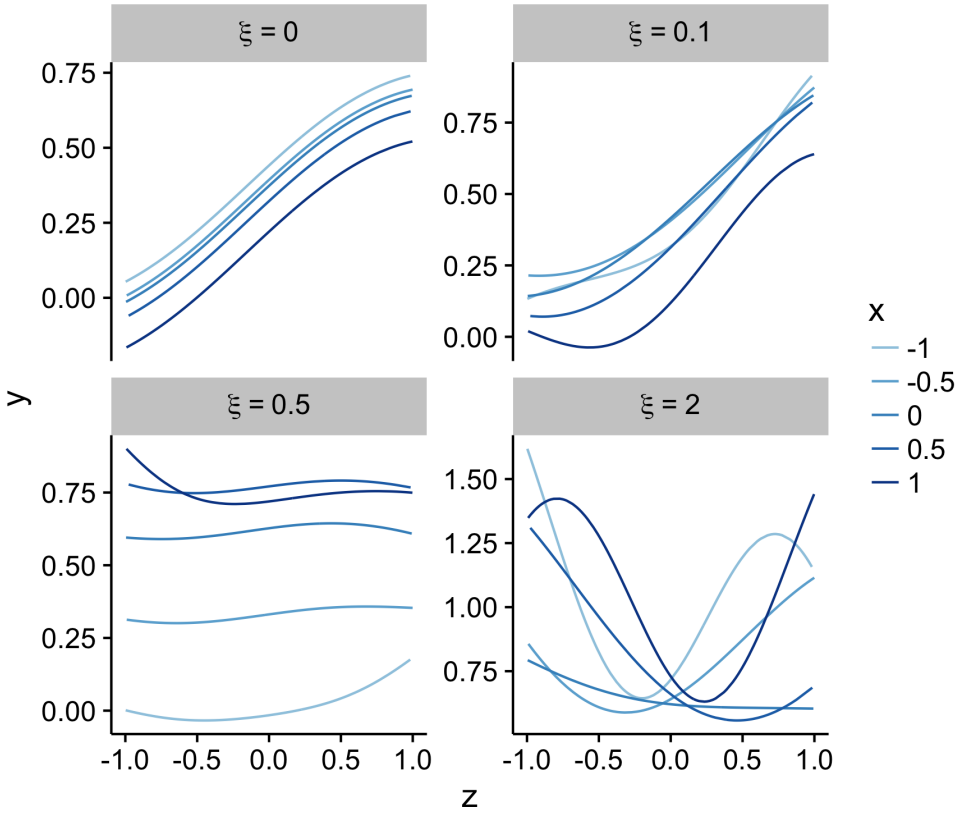


Figure 61: Draws from a CGPLVM prior with $\lambda = 0.5$, $\gamma = \nu = \eta = \delta = 0.2$ and $\xi \in \{0, 0.1, 0.5, 1\}$. For small ξ an identical trajectory is taken regardless of x but as ξ increases differing trajectories become apparent.

scalars $\alpha, \beta > 0$. We therefore introduce the concept of *Covariate-adjusted Gaussian Process Latent Variable Models* (CGPLVMs) that have kernels of the form

$$K\left(\{\mathbf{x}, \mathbf{z}\}, \{\mathbf{x}', \mathbf{z}'\}\right) \propto K(\mathbf{x}, \mathbf{x}') + K(\mathbf{z}, \mathbf{z}') + K(\mathbf{x} \odot \mathbf{z}, \mathbf{x}' \odot \mathbf{z}') \tag{94}$$

for some choice of kernel function K such as the squared exponential kernel $k_{\text{SQE}}(x, x') = \sigma_f^2 \exp\left(-\frac{1}{2l^2}(x - x')^2\right)$. In this case the kernel for the g^{th} output dimension (typically genes in a biological context) is given by

$$\begin{aligned} K_g(x, z; x', z') = \delta_g \exp(-\eta_g(x - x')^2) + \nu_g \exp(-\gamma_g(z - z')^2) \\ + \xi_g \exp(-\lambda_g(xz - x'z')^2) \end{aligned} \tag{95}$$

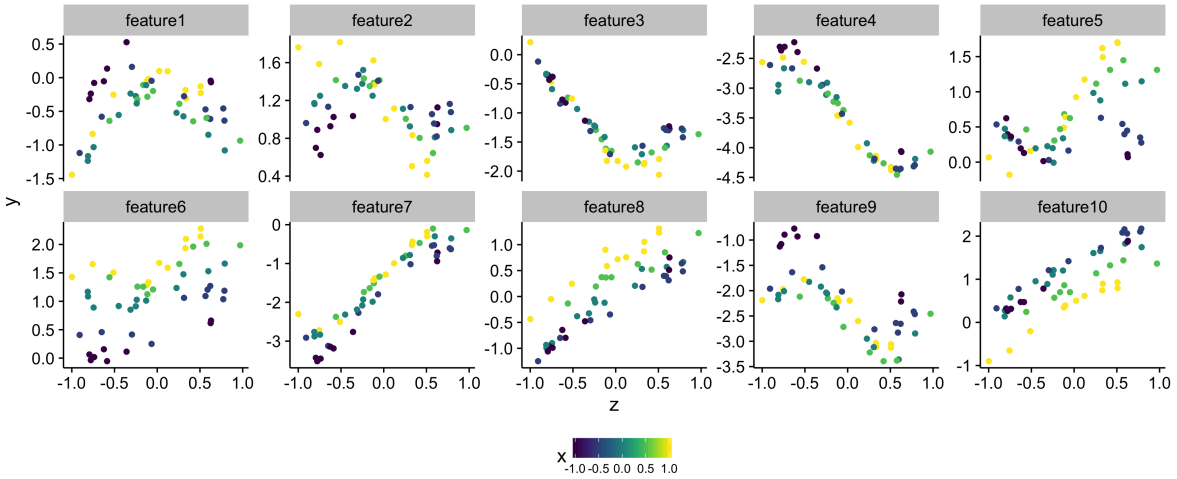


Figure 62: Synthetic data used to test CGPLVM for $N = 50$ samples and $G = 10$ features, with kernel parameters $\eta = 0.2$, $\gamma\nu = \delta = \xi = 1$, and $\sigma^2 = 0.01$. The first 5 features had $\lambda = 0.2$ and the final 5 had $\lambda = 0$.

where $\{\boldsymbol{\delta}, \boldsymbol{\eta}, \boldsymbol{\nu}, \boldsymbol{\gamma}, \boldsymbol{\xi}, \boldsymbol{\lambda}\}$ are the complete set of kernel parameters. Example draws from the GP priors of such kernels can be seen in figure 61.

Note that the interactions between the latent variables and the covariates appear linearly (through the element-wise multiplication) because this was the formulation in the original model. These interactions may therefore be made nonlinear by replacing the third term in the covariance of equation 94 with $K(\mathbf{f}(\mathbf{x}, \mathbf{z}), \mathbf{f}(\mathbf{x}', \mathbf{z}'))$ where $\mathbf{f}$ is any function that obeys $[\mathbf{f}(\mathbf{x}, \mathbf{z})]_i = f(x_i, z_i)$ (i.e. $\mathbf{f}$ acts element-wise along the vectors $\mathbf{z}$ and $\mathbf{x}$.)

5.5.2 *Black-box inference*

As a small proof-of-concept we created a synthetic dataset (figure 62) for $N = 50$ samples and $G = 10$ features, with z drawn from Unif(-1, 1) and the covariates sampled randomly from $\{-1, -0.5, 0, 0.5, 1\}$. The kernel parameters were fixed to $\eta = 0.2$, $\gamma = \nu = \delta = \xi = 1$, and $\sigma^2 = 0.01$. The first 5 features had $\lambda = 0.2$ and the final 5 had $\lambda = 0$ so that half the features exhibited covariate-latent variable interactions. The model was implemented

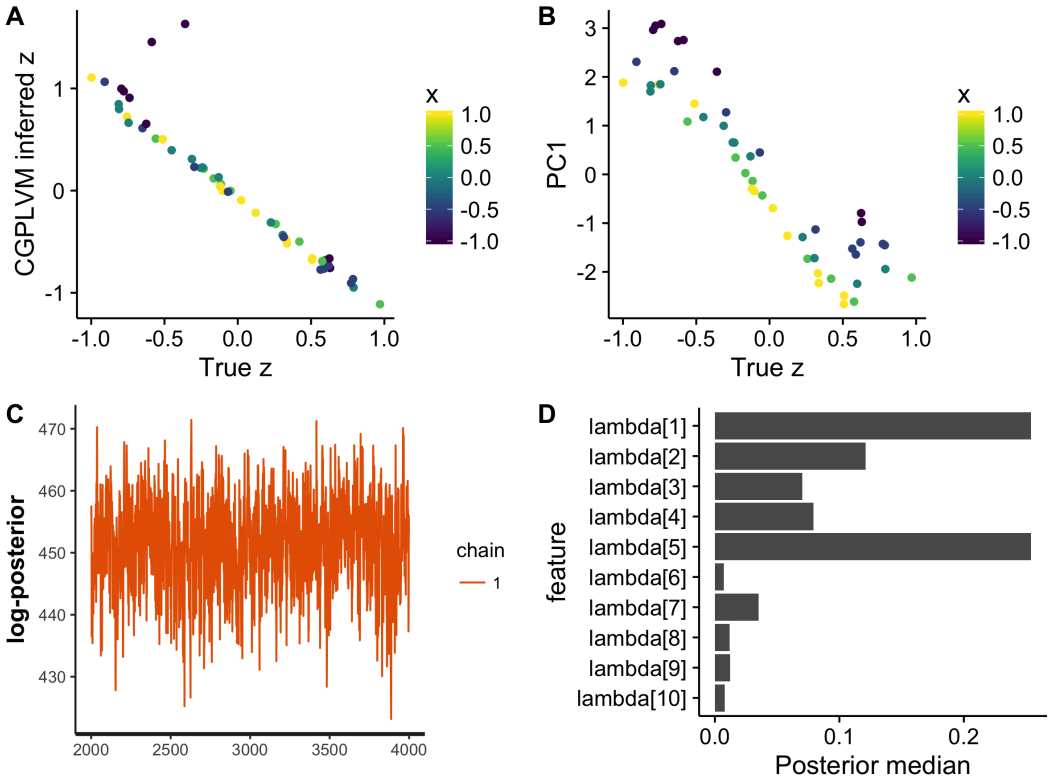


Figure 63: Black-box inference for CGPLVM on synthetic data. **A** Comparison of posterior mean inferred z from CGPLVM vs. true z. **B** Same comparison with first principal component of the data. **C** Traceplot of the posterior log-likelihood suggests good convergence of the model. **D** Posterior boxplot of $\boldsymbol{\lambda}$ identifies interactions for the odd-numbered features.

in the probabilistic programming language `Stan` with hierarchical priors $\lambda_g \sim \chi^2(1)$, $z_n \sim \mathcal{N}(0,1)$ and $1/\sigma^2 \sim \text{Gamma}(0.1, 0.1)$. `Stan` was run for 4000 iterations (2000 burn-in) with the latent variables initialised to the first principal component of the data.

The results can be seen in figure 63. Figure 63A compares the true latent variable values z to those inferred by the model, demonstrating high correlation (with the exception of two points). Figure 63B compares the first principal component of the data to the true values, showing obvious correlation but also confounding due to x. An initial analysis suggests the HMC algorithm has converged (figure 63C) due to good mixing of the log-posterior. Also of interest was the ability of the model to detect interactions

between x and z in the Gaussian Process model. To do this we an examine boxplots of the posterior of $\boldsymbol{\lambda}$. Figure 63D demonstrates that the first 5 features (that were designed with interactions in the synthetic data) have higher posterior mean values than the second five, demonstrating such models are able to detect interactions.

5.6 DISCUSSION

One weakness of our CAVI implementation of PhenoPath is the requirement of a full sweep through the data for every iteration (and indeed for every local parameter). While in practice it is unlikely for the number of features to increase[2] the reducing cost of genomics means the number of samples - particularly in the single cell case - could easily exceed one million. In order to make this scalable, CAVI can be replaced with stochastic variational inference (SVI) [54] where the data is subsampled at each iteration. Given the conditionally conjugate nature of the model it is relatively easy to derive an SVI algorithm from CAVI.

A further issue is that the number of interaction parameters grows as PG for P covariates and G featues. Given G is typically $\mathcal{O}(10^3)$, the introduction of many covariates leads to large model complexity and potentially cumbersome inference. Under the assumption that the covariates P are in some way correlated, we may reduce the number of parameters in the model by introducing an intermediate factor analysis step with an additional $R \times P$ matrix $\boldsymbol{\xi}$ for $R \ll P$ where the interaction term involved in the mean becomes

$$y_{ng} \propto \sum_{p=1}^{P} \sum_{r=1}^{R} \beta_{rg} \xi_{rp} x_{np} \tag{96}$$

2 The transciptome is naturally fixed at around 20,000 genes, of which typically 5,000 to 10,000 will exhibit “interesting” behaviour in a dataset, typically defined as being high-variance.

reducing the number of interaction parameters to $R(G+P)$.

There are several issues with and extensions of the covariate-adjusted Gaussian Process Latent Variable Model. When we performed inference we assumed the kernel hyperparameters were fixed and known, which is an unlikely assumption in practice (though often made). As discussed in chapter 2, GPLVM is often too flexible without strong priors on the kernel parameters. This issue persists in CGPLVM, where the introduction of additional kernel parameters and scale dependent interaction parameters will worsen the non-identifiability. A further issue is the $\mathcal{O}(N^3)$ inversion of the covariance matrix required in order to evaluate the likelihood to perform inference. More scalable inference may be performed by combining SVI with inducing points [126].

6

CONCLUSION

In this thesis we have considered inferring "trajectories" through gene expression space as a statistical latent variable problem. Such an approach is feasible when the data is gene expression from single-cells that undergo a physical time process where it is difficult to truly measure time and population-level cancer studies where such trajectories correspond to the activation of biological pathways.

As such quantities are estimated from noisy biological data it is important that a full probabilistic treatment taking uncertainty into account is considered as this has a large impact on downstream differential expression testing. We further demonstrated that contrary to existing approaches, which typically employ dimensionality reduction and cell ordering steps, it is possible to learn pseudotimes from a small set of marker genes using a simple parametric model. In such settings if the genes exhibit "switch-like" changes in expression over pseudotime then encoding this information as informative Bayesian priors typically improves inference.

We further considered the case of bifurcations in single-cell data, where cells proceed along some developmental trajectory before undergoing a fate decision that leads to two or more possible outcomes. All existing approaches to date treat this as a two step procedure that first learn a pseudotime for each cell then identify branching post-hoc assuming the pseudotimes are fixed. This has obvious issues in terms of treating the pseudotimes as fixed quantities and assumes the major source of variation comes from the pseudotemporal progress rather than the branch structure. We proposed that a Bayesian mixture of factor analysers is a solution to jointly generatively model both the trajectory and bifurcation event. By imposing a unique hierarchical structure between

the loading matrices of each mixture we were able to automatically identify genes that bifurcated, though this was limited to an extent by the linear assumptions of the model.

Finally we considered the case where such trajectories exist against a heterogeneous genetic or environmental background. We proposed a novel type of latent variable model that allows the behaviour of features to vary along the trajectory differently depending on some externally measured quantity, such as mutational burden or stimulant exposure. We further applied this to population wide cancer gene expression data, demonstrating that such "trajectories" correspond to biological pathway activation and identified interactions between such pathways and externally measured covariates. A fast variational inference algorithm was derived in order to infer such trajectories and interactions for thousands of genes and samples relatively quickly. We further proposed an extension for the case that the externally measured covariate is a possibly-censored survival time. Finally, we derived a non-parametric extension termed a covariate-adjusted Gaussian process latent variable model and demonstrated its utility on some synthetic data.

There are several ideas to be considered that form extensions to this work and the basis for new research. A major theme in our discussions so far and indeed among the current single-cell community is scaling up inference. This is particularly pertinent as new technologies such as Drop-seq and 10x genomics scale up sequencing to hundreds of thousands and millions of cells[1].

A major advantage of phrasing the pseudotime estimation problem as a (Bayesian) statistical latent variable one is that once we have written down a model we can "borrow" scalable inference procedures from elsewhere in the field. This is in contrast to what you could term the more algorithmic approach to pseudotime inference - that often relies on some tree or graph-building procedure - where a hand-crafted computational considerations are required for each algorithm and modification.

1 Gone are the early-PhD days of emailing yourself a single-cell dataset and pretending to everyone else that you work on "Big Data".

An obvious future direction for such research is to construct Bayesian models that use (black-box) variational inference[2] and stochastic variational inference (SVI). The major advantage behind SVI is that it can subsample observations typically in batches in order to climb noisy estimates of the ELBO, meaning it scales well as we vastly increase the number of cells in our datasets. An interesting observation is that statistical modelling of gene expression has typically been a $P \gg N$ problem, as we have $P \approx 20,000$ genes but $N \sim \mathcal{O}(10)$ samples. However, as the single-cell library preparation technology improved, we now face the opposite situation: P will always be fixed[3] at $\approx 20,000$, while N is unbounded in practice.

The scalability issue becomes worse if we work with Gaussian processes, where just evaluating the likelihood requires an expensive $\mathcal{O}(N^3)$ matrix inversion. A promising future direction in the field would then look at scalable inference for GPs in single-cell transcriptomics using techniques such as inducing points.

A prominent theme running through chapters 2 to 4 is the idea of *a priori* smoothness assumptions and model flexibility. Several methods have been proposed to overcome this, such as strong priors on the kernel parameters or priors in the latent space centred around physical capture times. On the other hand, we have shown that it's possible to learn such trajectories by making strongly parametric assumptions about how gene expression is regulated over pseudotime that require no tuning but are limited in the gene behaviour the can model (under this framework we cannot model transient expression). Therefore, there is a natural trade-off between model expressivity and practicality / accuracy. This opens several exciting research avenues: for example, is it possible to constrain the parameters of a GPLVM in terms of the expected number of fluctuations

2 To quote David Blei, "Variational inference is that thing you implement while waiting for your Gibbs sampler to converge."

3 Alternatively if you look at expression of individual transcripts rather than genes then $P \approx 170,000$ but as before this is fundamentally fixed.

over pseudotime? Can we impose additional structural constraints on GPLVMs through kernel design such as enforcing monotonicity or other interpretable gene behaviour?

A more theoretical avenue to be explored is the connection between existing "algorithmic" approaches (dimensionality reduction followed by MST fitting) to latent variable approaches such as nonlinear factor analysis and GPLVM. MST based approaches such as Monocle and TSCAN find the longest single path through the MST in the reduced space. Intuitively, this should be *very* similar to the first principal component of the data[4] as we find a one dimensional path through a maximal number of data points that is similar to finding a one dimensional projection that maximises the variance in the reduced space. Indeed, if the initial dimensionality reduction is linear (as is often a case), then the only nonlinearity induced will be by the longest path through the MST curving back on itself, which is very rare. Therefore we can conclude that ordering cells using MSTs in linear dimensionality reduction spaces will be very similar to principal component analysis.

A future research area related to single-cell trajectories that has the potential to become popular is unsupervised multiview learning. As it becomes feasible to make multiple "omic" measurements alongside expression from a single-cell at once - such as epigenetic modifications and DNA sequence - we should in theory be able to learn pseudotimes from the different data sources simultaneously and improve inference as a result. Such models would include a common underlying pseudotime z_n for each cell but could model a different likelihood $L_d(\mathbf{X}_d|\mathbf{z}, \boldsymbol{\theta}_d)$ for each $d = 1, \ldots, D$ data source. The challenge here is showing what you might gain from modelling more than one data source simultaneously as ultimately if they follow a common trajectory the sources are likely to be heavily correlated.

Finally, there is a paradox that runs through the entire single-cell pseudotime field. It is impossible to truly benchmark such algorithms because current technology can-

4 Indeed the output of many pseudotime algorithms is extremely similar to PC1.

not track a single-cell over time while performing transcriptome-wide gene expression quantification. This has led researchers to use a variety of heuristics such as smoothness criteria or consistency with capture times[5] to compare their algorithms to others. However, the moment we develop transcriptome-wide time series expression measurement technology pseudotime will no longer be required as we will have something infinitely better - real time. In other words, as soon as we are able to correctly benchmark and compare such algorithms we have rendered them obsolete.

5 Though this is dubious since the whole point behind pseudotime algorithms is that the cells progress asynchronously meaning capture times do not reflect true biological progress.

A

MODEL INFERENCE FOR SWITCHDE

A.1 MAXIMUM LIKELIHOOD MODEL FITTING

We begin with a $N \times G$ expression matrix $\mathbf{Y}$ for G genes and N cells with column vector $\mathbf{y}_g, g \in 1, \ldots, G$, that is non-negative and represents gene expression in a form comparable to $\log(\text{TPM} + 1)$. If the sigmoid function is defined as

$$f(t_n; \mu_g^{(0)}, k_g, t_g^{(0)}) = \frac{2\mu_g^{(0)}}{1 + \exp\left(-k_g(t_n - t_g^{(0)})\right)} \tag{97}$$

then the likelihood of the data given the parameters is

$$L(\mathbf{y}_g, \boldsymbol{t}; \mu_g^{(0)}, k_g, t_g^{(0)}) = \prod_{n=1}^{N} \mathcal{N}\left(y_{ng} | f(t_n; \mu_g^{(0)}, k_g, t_g^{(0)}), \sigma_g^2\right) \tag{98}$$

We infer maximum likelihood estimates of the parameters using L-BFGS-B optimisation [16] using the `R` function `optim`. This allows fast inference by passing analytical gradients as well as handling constraints on bounded variables. All parameters are defined on $\mathbb{R}$ except for μ_0 and σ^2 which are optimised on $\mathbb{R}^+$. The parameters are initialised as follows: μ_0 is set to $\frac{1}{N}\sum_n y_n$, $t_0 = \text{median}_n\ [t_n]$, $\sigma^2 = \text{Var}_n[y_n]$. We initialise k by using the gradient of the regression of $\mathbf{y}$ off $\mathbf{t}$ to ensure the sign is correct.

We next need to compute the gradients for all parameters. If we consider the function (dropping g subscripts)

$$f(t_n; k, \mu_0, t_0) = \frac{2\mu_0}{1 + \exp\left(-k(t_n - t^{(0)})\right)} \tag{99}$$

(which we may write succinctly as $f(t_n;\Theta)$ where $\Theta = \{\mu_0, k, t^{(0)}\}$) then the partial derivatives are given by

$$\begin{aligned}
\frac{\partial f}{\partial \mu_0} &= \frac{2}{1+\exp\left(-k(t_n - t^{(0)})\right)} \\
\frac{\partial f}{\partial k} &= \frac{f(t_n;\Theta)(t_n - t^{(0)})}{1+e^{k(t_n - t^{(0)})}} \\
\frac{\partial f}{\partial t^{(0)}} &= \frac{-kf(t_n;\Theta)}{1+e^{k(t_n - t^{(0)})}}
\end{aligned} \tag{100}$$

To find the maximum likelihood estimate of $\Theta \equiv (\mu_0, k, t^{(0)})$ we wish to minimise the negative log-likelihood, given by

$$\begin{aligned}
\mathcal{L} = -\log L(\mathbf{y}, \mathbf{t}; \Theta) &= -\sum_{n=1}^{N} \log \mathcal{N}\left(y_n | f(t_n;\Theta), \sigma^2\right) \\
&= -\sum_n \left[\log \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2}\left(y_n - f(t_n;\Theta)\right)^2\right] \\
&= N \log \sqrt{2\pi\sigma^2} + \frac{1}{2\sigma^2}\sum_n \left(y_n - f(t_n;\Theta)\right)^2
\end{aligned} \tag{101}$$

Then to compute the gradient of $\mathcal{L}$ with respect to a given parameter $\theta \in \Theta$ it follows that

$$\frac{d\mathcal{L}}{d\theta} = \frac{1}{2\sigma^2}\sum_n \left[-2\left(y_n - f(t_n;\Theta)\right)\frac{\partial f(t_n;\Theta)}{\partial \theta}\right] \tag{102}$$

Note that this is general to any iid Gaussian measurements with a parametric mean function.

Further, since we have

$$\mathcal{L} \propto \frac{N}{2}\log \sigma^2 + \frac{1}{2\sigma^2}\sum_n \left(y_n - f(t_n;\Theta)\right)^2 \tag{103}$$

it follows that

$$\frac{d\mathcal{L}}{d\sigma^2} = \frac{N}{2\sigma^2} - \frac{1}{2\sigma^4}\sum_n \left(y_n - f(t_n;\Theta)\right)^2 . \tag{104}$$

A.2 EXPECTATION-MAXIMISATION FOR ZERO INFLATION

Single-cell RNA-seq data is known to exhibit an over-representation of zeros known as "dropouts". To account for this we propose a model that incorporates dropouts in a similar style to [102]. Our model becomes

$$\begin{aligned}
\mu(t_n,\theta) &= \frac{2\mu_0}{1+\exp\left(-k(t_n - t^{(0)})\right)} \\
x_n &\sim \mathcal{N}(\mu(t_n,\theta),\sigma^2) \\
h_n|x_n &\sim \text{Bernoulli}(\exp(-\lambda x_n^2)) \\
y_n &= \begin{cases} x_n, & \text{if } h_n = 0 \\ 0, & \text{if } h_n = 1 \end{cases}
\end{aligned} \tag{105}$$

Where we have replaced $f \to \mu$ to avoid cluttering notation later. We essentially introduce a latent variable x_n for each gene expression measurement but must now perform inference using the Expectation-Maximisation (EM) algorithm due to the intractability of directly maximising the log-likelihood. The secondary latent variable h_n is a binary indicator for whether the expression measurement in cell c is dropout or not. In the following we derive the EM algorithm which follows a similar derivation to [102] but with some differences, so it is provided in full below.

In the following let $\Theta = \{\mu_0, k, t^{(0)}, \sigma^2\}$ and consider the complete-data likelihood:

$$\begin{aligned} p(\mathbf{y}, \mathbf{x}, \mathbf{h}, \Theta) &= \prod_n p(y_n, x_n, h_n, \Theta), \\ &= \prod_n p(y_n|x_n, h_n) p(h_n|x_n, \Theta) p(x_n|\Theta), \\ p(y_n, x_n, h_n, \Theta) &= \begin{cases} (1 - e^{-\lambda x_n^2}) p(y_n|x_n, h_n = 0) p(x_n|\Theta), & h = 0, \\ e^{-\lambda x_n^2} p(y_n|x_n, h_n = 1) p(x_n|\Theta), & h = 1. \end{cases} \end{aligned} \tag{106}$$

We then use the same trick as [102]: if $y_n = 0$ then we know necessarily that $h_n = 1$ as $h_n = 0$ with zero probability. Similarly, if $y_n > 0$ then we observe $x_n = y_n$ and know that $h_n = 0$. We therefore split the product up into terms involving $y_n = 0$ and those involving $y_n > 0$, and now consider the log of the complete data likelihood with the shorthand notation $\mu_n \equiv \mu(t_n, \theta)$:

$$\begin{aligned} \mathcal{L}(\mathbf{y}, \mathbf{x}, \mathbf{h}, \Theta) &= \sum_{c:y_n=0} \left[\log \mathcal{N}(x_n|\mu_n, \sigma^2) - \lambda x_n^2\right] + \sum_{c:y_n>0} \left[\log \mathcal{N}(y_n|\mu_n, \sigma^2) + \log(1 - e^{-\lambda y_n^2})\right] \\ &= \frac{-N}{2} \log(2\pi\sigma^2) + \sum_{c:y_n=0} \left[\frac{(x_n - \mu_n)^2}{2\sigma^2} - \lambda x_n^2\right] \\ &+ \sum_{c:y_n>0} \left[\frac{(y_n - \mu_n)^2}{2\sigma^2} + \log(1 - e^{-\lambda y_n^2})\right] \\ &= \frac{-N}{2} \log(2\pi\sigma^2) + \sum_{c:y_n=0} \left[-(\frac{1}{2\sigma^2} + \lambda) x_n^2 + \frac{\mu_n}{\sigma^2} x_n - \frac{\mu_n^2}{2\sigma^2}\right] \\ &+ \sum_{c:y_n>0} \left[\frac{(y_n - \mu_n)^2}{2\sigma^2} + \log(1 - e^{-\lambda y_n^2})\right] \end{aligned} \tag{107}$$

In order to perform EM we need to calculate the expected value of this log likelihood, conditional on the data $\mathbf{y}$ and a previous estimate $\Theta^{(t)}$:

$$Q(\Theta|\Theta^{(t)}) = \mathbb{E}_{\mathbf{x}|\mathbf{y},\Theta^{(t)}}[\mathcal{L}(\mathbf{y}, \mathbf{x}, \mathbf{h}, \Theta)] \tag{108}$$

In order to calculate this it is obvious from equation 107 we must calculate $\mathbb{E}_{\mathbf{x}|\mathbf{y},\Theta^{(t)}}[x_n]$ and $\mathbb{E}_{\mathbf{x}|\mathbf{y},\Theta^{(t)}}[x_n^2]$. Notice we only care about $c : y_n = 0$ since for $c : y_n > 0$ we know x_n exactly. Note that in all the following all the parameters are assumed fixed at the previous iteration, e.g. $\mu_n \equiv \mu_n^{(t)}$. If we consider a conditional density of the form

$$f(\mathbf{x}|\mathbf{y},\Theta^{(t)}) = \prod_n f(x_n|y_n,\Theta^{(t)}) \tag{109}$$

then

$$\begin{aligned} f(x_n|y_n,\Theta^{(t)}) &= \frac{f(y_n|x_n,\Theta^{(t)})f(x_n|\Theta^{(t)})}{\int dx_n f(y_n|x_n,\Theta^{(t)})f(x_n|\Theta^{(t)})} \\ &= \frac{e^{-\lambda x_n^2}\mathcal{N}(x_n|\mu_n,\sigma^2)}{\int dx_n e^{-\lambda x_n^2}\mathcal{N}(x_n|\mu_n,\sigma^2)} \end{aligned} \tag{110}$$

Some algebra later and we arrive at

$$f(x_n|y_n,\Theta^{(t)}) = \mathcal{N}\left(x_n|\alpha(t_n,\Theta^{(t)}),\beta(\Theta^{(t)})\right) \tag{111}$$

where

$$\begin{aligned} \alpha(t_n,\Theta^{(t)}) &= \frac{\mu_n}{2\sigma^2\lambda+1} \\ \beta(t_n) &= \frac{\sigma^2}{2\sigma^2\lambda+1} \end{aligned} \tag{112}$$

and so

$$\begin{aligned} \mathbb{E}_{\mathbf{x}|\mathbf{y},\Theta^{(t)}}[x_n] &= \alpha(t_n,\Theta^{(t)}) \\ \mathbb{E}_{\mathbf{x}|\mathbf{y},\Theta^{(t)}}[x_n^2] &= \alpha(t_n,\Theta^{(t)})^2 + \beta(\Theta^{(t)}). \end{aligned} \tag{113}$$

We need to maximise

$$\begin{aligned} Q(\Theta|\Theta^{(t)}) = &\frac{-N}{2}\log(2\pi\sigma^2) \\ &+ \sum_{c:y_n=0}\left[-(\frac{1}{2\sigma^2}+\lambda)\mathbb{E}_{\mathbf{x}|\mathbf{y},\Theta^{(t)}}[x_n^2] + \frac{\mu_n}{\sigma^2}\mathbb{E}_{\mathbf{x}|\mathbf{y},\Theta^{(t)}}[x_n] - \frac{\mu_n^2}{2\sigma^2}\right] \\ &+ \sum_{c:y_n>0}\left[\frac{(y_n-\mu_n)^2}{2\sigma^2} + \log(1-e^{-\lambda y_n^2})\right] \end{aligned} \tag{114}$$

with respect to $\theta = \{\mu_0, k, t_0\}$, σ^2 and λ, recalling $\mu_n \equiv \mu_n(\theta, t_n)$. We wish to use gradient-based optimisation and so require the gradients. Note that

$$\begin{aligned} \frac{dQ}{d\theta} &= \sum_n \frac{\partial Q}{\partial \mu_n}\frac{d\mu_n}{d\theta} \\ &= \frac{1}{\sigma^2}\left[\sum_{c:y_n=0}\left(\mathbb{E}_{\mathbf{x}|\mathbf{y},\Theta^{(t)}}[x_n] - \mu_n\right)\frac{d\mu_n}{d\theta} + \sum_{c:y_n>0}(y_n-\mu_n)\frac{d\mu_n}{d\theta}\right] \end{aligned} \tag{115}$$

where the derivatives $\frac{d\mu_n}{d\theta}$ are the same as those given in equation 100. Finally we require the partial derivatives with respect to λ and σ^2 which are given by

$$\frac{dQ}{d\lambda} = -\sum_{c:y_n=0}\mathbb{E}_{\mathbf{x}|\mathbf{y},\Theta^{(t)}}[x_n] + \sum_{c:y_n>0}\frac{y_n^2 e^{-\lambda y_n^2}}{1-e^{-\lambda y_n^2}} \tag{116}$$

and

$$\frac{dQ}{d\sigma^2} = \frac{-N}{2\sigma^2} + \frac{1}{2\sigma^4}\left[\sum_{c:y_n=0}\left(\mathbb{E}_{\mathbf{x}|\mathbf{y},\Theta^{(t)}}[x_n^2] - 2\mu_n\mathbb{E}_{\mathbf{x}|\mathbf{y},\Theta^{(t)}}[x_n] + \mu_n^2\right) + \sum_{c:y_n>0}(y_n-\mu_n)^2\right] \tag{117}$$

Note that σ^2 has an analytical maximum by setting $\frac{dQ}{d\sigma^2} = 0$, but since this depends on θ and vice versa we instead numerically optimise all simultaneously.

B

DATA ANALYSIS

B.1 CHAPTER 2

All analyses are available either as Rmarkdown notebooks or as `R` scripts at http://github.com/kieranrcampbell/pseudogp-paper/

B.1.1 *Trapnell et al.*

The data was imported from the `HSMMSingleCell` package[1] and $\log_{10}$ of the FPKM values with a pseudo-count of 1. Then, using genes with a transformed expression greater than 0.3 a generalized linear model was fit with $\text{CV}^2 \sim a10^{-k\mu}$ where CV^2 is the square of the coefficient of variation and μ is the mean for a particular gene. Genes whose measured coefficient of variation was greater than four times that predicted by the model were then used for dimensionality reduction, with the reasoning that those that vary greatest will contribute most to pseudotemporal processes.

The Laplacian Eigenmaps embedding was then found using the `R` package `embeddr`[2] using the default parameters. The Trapnell dataset contains a differentiation trajectory of differentiating myoblasts as well as contaminating interstitial mesenchymal cells. These mesenchymal cells were discovered using Gaussian Mixture Model clustering in the reduced space (using $k = 3$ components) and subseqently removed. A further four cells were removed as outliers on the manifold. This reduced the original 271 cell dataset to 155 cells. The PCA and t-SNE representations were found using the `R` package `scater`[3]

1 http://bioconductor.org/packages/release/data/experiment/html/HSMMSingleCell.html
2 http://github.com/kieranrcampbell/embeddr/
3 http://github.com/davismcc/scater

using the default gene set (the 500 most variable genes) and the reduced set of 155 cells as discovered above. For the t-SNE representation a perplexity of 3 was used.

The Bayesian GPLVM pseudotime trajectory was fitted on the Laplacian Eigenmaps embedding using the `R` package `pseudogp`[4] using the default MCMC parameters and smoothing hyperparameters $\gamma_\alpha = 30$, $\gamma_\beta = 5$ which were chosen empirically based on the quality of the fit.

B.1.2 *Burns et al.*

The data were downloaded from the Gene Expression Omnibus accession number 71982 (http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE71982) as the pre-processed TPM count matrix. The transformation applied to the TPM counts was performed identically to the original paper: genes with TPM expression less than 1 are set to 0 while those with TPM expression greater than 1 are set to $\log_2$ of that expression level. The original publication focussed pseudotime analyses on Utrichular cells rather than Cochlear ones. Using "Ute_P1" in the column names of the cells in the data as designating Utrichular cells, we took only those forward for analysis.

Then a PCA plot (similar to the original paper) was constructed once more using `scater`. The original publication states the "top" 195 genes were used for PCA analysis and we emailed the corresponding author asking the definition of "top", but received no reply so assumed it refers to "most variable". Using this definition we found a similarly-shaped PCA plot to that in the publication. The pseudotime trajectory identified in the paper involves only a subset of cells that represent a supporting cell to hair cell transition. We identified these cells in the PCA by plotting the intensity of the fluorescent markers

4 http://kieranrcampbell.github.com/pseudogp

measured as well as three marker genes (*LFNG*, *CDH4* and *SOX2*) and computationally isolated these using k-means clustering.

The Laplacian Eigenmaps embedding of just the trajectory cells was found using the `embeddr R` package with 35 nearest-neighbours and the entire gene-set. Two cells were removed from the analysis as outliers. A principal curve[5] was fitted and the behaviour of several marker genes observed to be identical to that shown in the original publication, implying we had recovered the same cell ordering. For the PCA representation of the differentiation the `scater` defaults were used but with `scale_features = FALSE`. For the t-SNE representation, a range of perplexities were examined to find one that most gave a trajectory like structure in the reduced embedding. A perplexity of 2 was subsequently chosen, with all other parameters those of the defaults in `scater` except `scale_features = FALSE`.

The GPLVM was fitted on the Laplacian Eigenmaps embedding using `pseudogp`. The curve was initialised from a principal curve and smoothing hyperparameters $\gamma_\alpha = 20$, $\gamma_\beta = 2$ were used.

B.1.3 *Shin et al.*

The data were downloaded as supplementary data from http://www.cell.com/cms/attachment/2038326541/2052521610/mmc7.xlsx. This included the `Waterfall` pseudotime assignment, allowing us to compare as a sanity check. The PCA representation was found using the top 195 most variable genes and `scale_features = FALSE` in the `scater` package. The shape closely resembled that from the original publication and colouring by the pseudotime assigned by `Waterfall` clearly showed this was the case. The t-SNE representation was found again using `scater` with 195 most variable genes,

5 Which can be thought of as representing the MAP of a 1D GPLVM

`scale_features = FALSE` and a perplexity of 3. The Laplacian Eigenmaps embedding was found with `embeddr`, again using the top 195 most variable genes, a euclidean distance metric and 30 nearest neighbours.

The GPLVM was fitted on the PCA embedding (as in original publication) using `pseudogp`. The MCMC chain was initialised from the first component of the PCA of the representation and smoothing hyperparameters of $\gamma_\alpha = 8$, $\gamma_\beta = 2$.

B.2 CHAPTER 3

B.2.1 *Trapnell et al.*

Data were retrieved as described in section B.1.1. The genes *CDK1*, *ID1*, *MYOG*, *MEF2C*, and *MYH3* were used as "marker genes" as identified in the original publication. The prior means of activation were given by the direction of regulation from the original publication via $\boldsymbol{\mu}^{(k)} = [-10, -10, 10, 10, 10]$.

B.2.2 *Shin et al.*

Data were retreived as described in section B.1.3. The genes *Sox11*, *Eomes*, *Stmn1*, *Apoe*, *Aldoc*, and *Gfap* were used as "marker genes" as identified in the original publication. The prior means of activation were given by the direction of regulation from the original publication via $\boldsymbol{\mu}^{(k)} = [10, 10, 10, -10, -10, -10]$.

B.2.3 *Zhou et al.*

FPKM values were downloaded from the gene expression ombnibus entry `GSE67120`. Cells belonging to the original publication's clusters `Adult_HSC` and `E11.0_T1CD201neg` were removed. The genes *Nrp1*, *Hey1*, *Efnb2*, *Ephb4*, *Nrp2*, and *Nr2f2* were chosen as "marker genes" according to the original publication. As all marker genes should be downregulated, the prior mean vector was set to $\boldsymbol{\mu}^{(k)} = [-10, -10, -10, -10, -10, -10]$.

B.3 CHAPTER 4

B.3.1 *Paul et al.*

Data from [99] were downloaded as post-processed from the analysis from [117] from `http://www.c2b2.columbia.edu/danapeerlab/html/wishbone-data.html`. Genes expressed in at least 20% of cells were retained. For inference with MFA, genes whose variance exceeds 5 were used.

B.3.2 *Bendall et al.*

Data from [7] were downloaded as post-processed from the analysis from [117] from `http://www.c2b2.columbia.edu/danapeerlab/html/wishbone-data.html`. The entire panel of 12 marker genes were used.

B.4 CHAPTER 5

B.4.1 *Shalek et al.*

Preprocessed TPM values for all cells were retrieved from the Gene Expression Omnibus (`GSE48968`). We retained cells treated by LPS and PAM at time points 1h, 2h, 4h, and 6h, resulting in 820 cells (479 LPS and 341 PAM). We retained the 7533 genes whose variance in $\log_2(\text{TPM}+1)$ expression was greater than 2. The first principal component of the data showed a strong dependency on the number of features expressed - previously been implicated in technical effects [50] - which we subsequently removed using the `normalizeExprs` function in `Scater` [87].

B.4.2 *TCGA studies*

For both COAD and BRCA studies, TPM matrices were retrieved from a recent transcript-level quantification of the entire TCGA study [123]. Clinical metadata, including the phenotypic covariates used in PhenoPath, were retrieved using the RTCGA `R` package [65]. Transcript level expression estimates were combined to gene level expression estimates using `Scater` [87].

B.4.2.1 *COAD*

A PCA visualisation of the COAD dataset (figure 64A) showed two distinct clusters based on the plate of sequencing. Rather than try to correct such a large batch effect, we retained samples with a PC1 score of less than 0 and a PC3 score greater than -10, and removed any "normal" tumour types. For input to PhenoPath we used the 4801

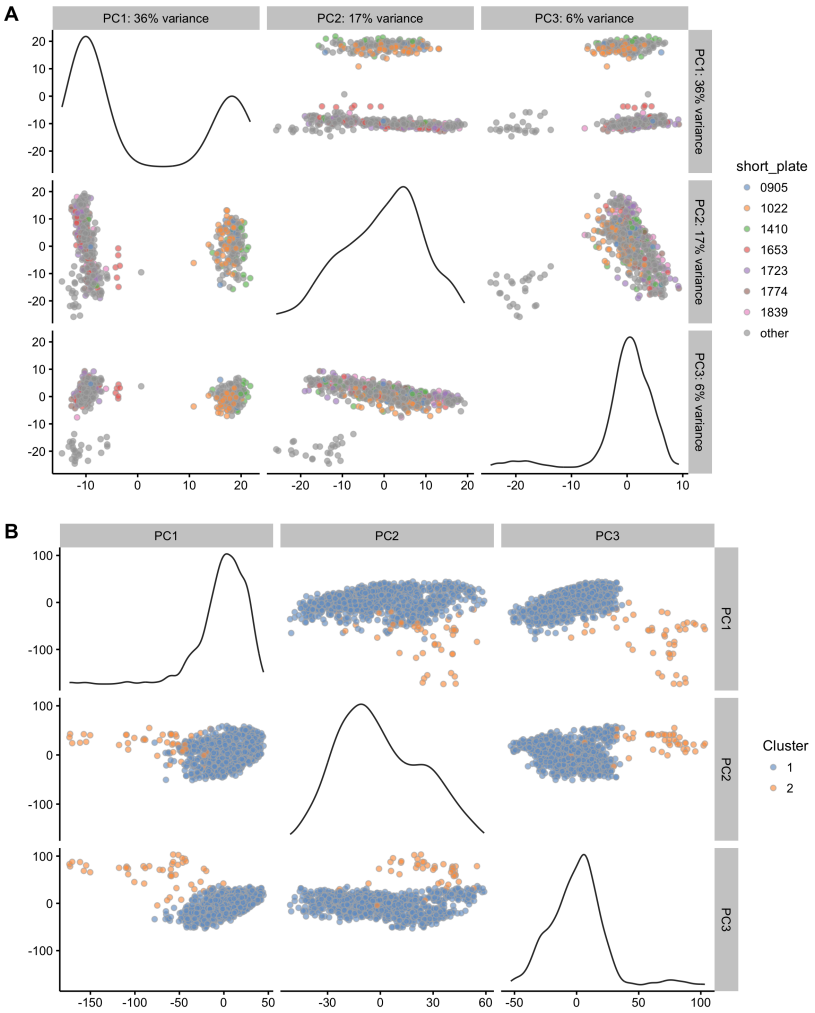


Figure 64: PCA representations of the COAD (**A**) and BRCA (**B**) datasets, coloured by sequenced plate and GMM cluster assignment respectively.

genes whose median absolute deviation in $\log(\text{TPM} + 1)$ expression was greater than $\sqrt{\frac{1}{2}}$.

B.4.2.2 *BRCA*

A PCA visualisation of the BRCA dataset (figure 64B) showed a loosely dispersed outlier population that separated on the first and third principal components. We performed Gaussian mixture model clustering using the `R` package `mclust`[36], and removed samples designated as cluster 2 in figure 64B. For input to PhenoPath we used the 4579 genes whose variance in $\log(\text{TPM} + 1)$ expression was greater than 1 and whose median absolute deviation was greater than 0.

C

ADDITIONAL MATERIALS FOR SWITCH-LIKE PSEUDOTIME

C.1 STAN CODE FOR OUIJA

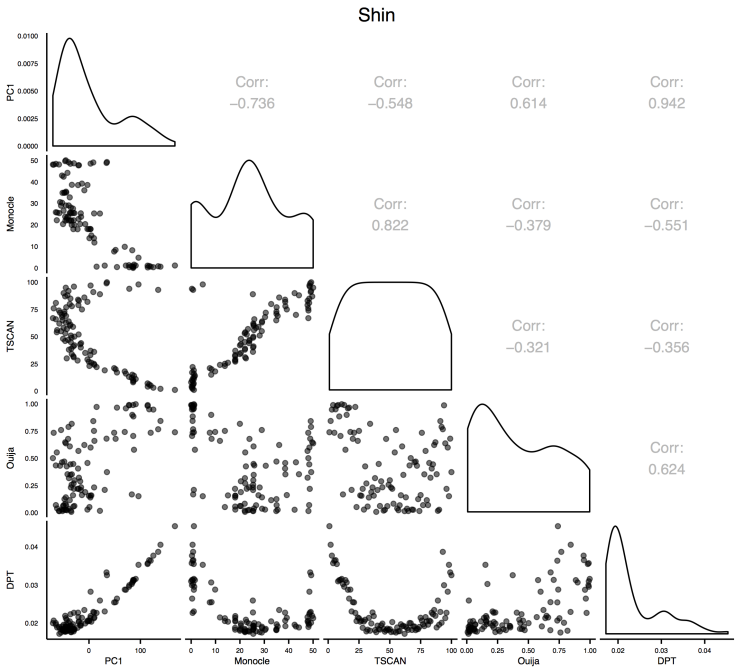
```
data {
  int<lower = 2> N; // number of cells
  int<lower = 2> G; // number of genes

  vector<lower = 0>[N] Y[G]; // matrix of gene expression values

  real k_means[G]; // mean parameters for k provided by user
  real k_sd[G]; // standard deviation parameters for k provided
     by user

  real t0_means[G]; // mean parameters for t0 provided by user
  real t0_sd[G]; // standard deviation parameters for t0
     provided by user

  real student_df;
}

parameters {
  // parameters we'll let stan infer
  real<lower = 0> mu0[G];
```

```
  real<lower = 0> phi; // mean-variance "overdispersion"
     parameter

  // parameters with user-defined priors
  real k[G];
  real<lower = 0, upper = 1> t0[G];

  real<lower = 0> mu_hyper;

  real<lower = 0, upper = 1> t[N]; // pseudotime of each cell

  real beta[2];
}

transformed parameters {
  vector[N] mu[G]; // mean for cell i gene g
  vector<lower = 0>[N] ysd[G];

  for(g in 1:G) {
    for(i in 1:N) {
      mu[g][i] = 2 * mu0[g] / (1 + exp(-k[g] * (t[i] - t0[g])));
      ysd[g][i] = sqrt( (1 + phi) * mu[g][i] + 0.01);
    }
  }
}

model {
  // user defined priors
```

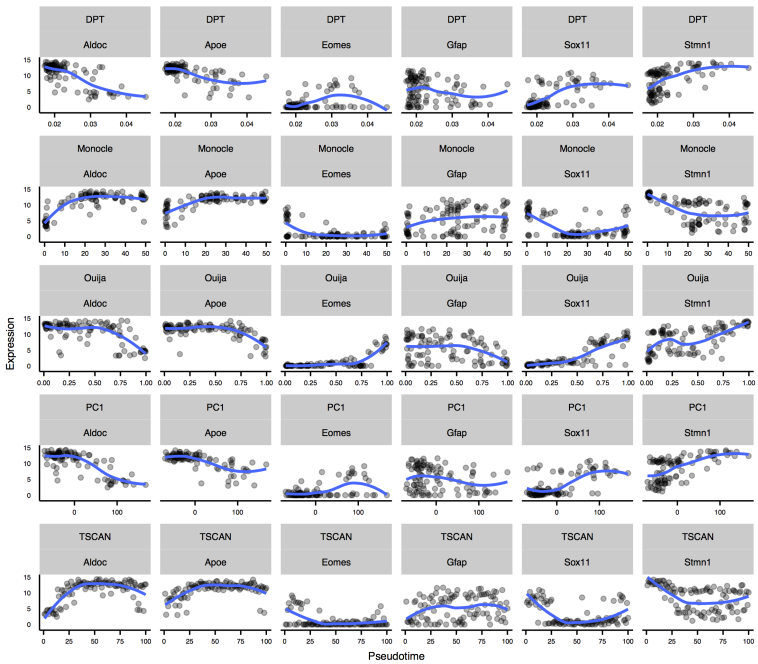
```
k ~ normal(k_means, k_sd);
t0 ~ normal(t0_means, t0_sd);

// model priors
mu0 ~ gamma(mu_hyper / 2, 0.5);

phi ~ gamma(12, 4);

t ~ normal(0.5, 1);

beta ~ normal(0, 0.1);

// Zero inflation per-mean
for(g in 1:G) {
  for(i in 1:N) {
    if(Y[g][i] == 0) {
      target += log_sum_exp(bernoulli_logit_lpmf(1 | beta[1] +
          beta[2] * mu[g][i]),
                            bernoulli_logit_lpmf(0 | beta[1] +
                                beta[2] * mu[g][i]) +
                            student_t_lpdf(Y[g][i] |
                               student_df, mu[g][i], ysd[g][i
                               ]));
    } else {
      target += bernoulli_logit_lpmf(0 | beta[1] + beta[2] *
         mu[g][i]) +
      student_t_lpdf(Y[g][i] | student_df, mu[g][i], ysd[g][i
         ]);
```

```
        }
      }
    }
  }
```

C.2 PSEUDOTIME COMPARISON FIGURES

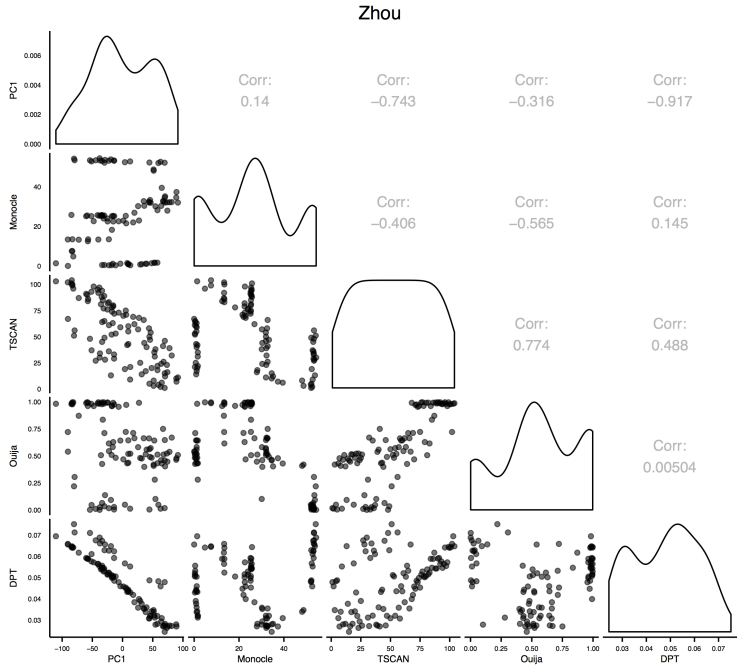


(a) Pseudotime correlations across the five algorithms.

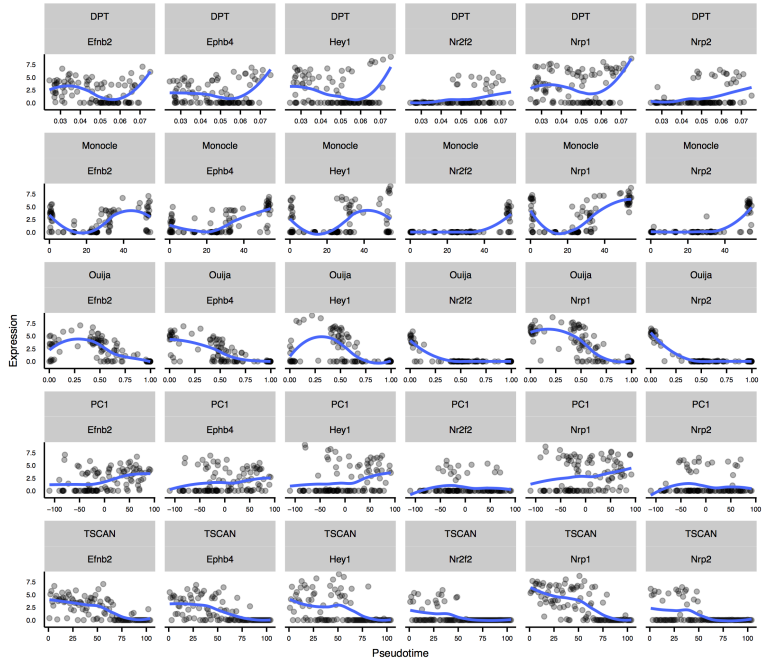


(b) Expression level fits reported by each algorithm.

Figure 65: Comparison of marker gene-based pseudotime estimates across five algorithms for the Shin et al. dataset.



(a) Pseudotime correlations across the five algorithms.



(b) Expression level fits reported by each algorithm.

Figure 66: Comparison of marker gene-based pseudotime estimates across five algorithms for the Zhou et al. dataset.

D

GIBBS UPDATES FOR MFA

D.1 GIBBS UPDATES

The full model is specified by

$$
\begin{aligned}
\boldsymbol{\omega} &\sim \text{Dirichlet}(1/B, \ldots, 1/B) \\
\gamma_n &\sim \text{Categorical}(\boldsymbol{\omega}) \\
\eta &\sim \mathcal{N}(\tilde{\eta}, \tau_\eta^{-1}) \\
\theta_g &\sim \mathcal{N}(\tilde{\theta}, \tau_\theta^{-1}) \\
\chi_g &\sim \text{Gamma}(\alpha_\chi, \beta_\chi) \\
\mathbf{c}_{\gamma_n} &\sim \mathcal{N}(\eta_{\gamma_n}, \tau_c^{-1}) \\
\mathbf{k}_{\gamma_n} &\sim \mathcal{N}(\boldsymbol{\theta}, \boldsymbol{\chi}^{-1}\mathbb{1}_G) \\
t_n &\sim \mathcal{N}(0, 1) \\
\boldsymbol{\tau} &\sim \text{Gamma}(\alpha, \beta) \\
\mathbf{y}_n &\sim \mathcal{N}(\mathbf{c}_{\gamma_n} + \mathbf{k}_{\gamma_n} t_n, \boldsymbol{\tau}^{-1}\mathbb{1}_G)
\end{aligned}
\tag{118}
$$

where $\mathbf{y}_n$ is a G-length vector of expression in cell n, B is the number of branches modelled and we define $\Lambda_\gamma = [\mathbf{c}_\gamma \ \mathbf{k}_\gamma]$, $\gamma \in 1, \ldots, B$.

UPDATE FOR γ Defining $\pi_{n,b} = p(\gamma_n = b)$ (the probability that cell n is on branch b) then

$$
p(\gamma_n | \mathbf{y}_n, t_n, \{\Lambda_b\}_{b=1}^B, \boldsymbol{\omega}, \boldsymbol{\tau}) = \text{Categorical}(\boldsymbol{\pi}_n) \tag{119}
$$

where

$$\pi_{n,b} = \frac{\omega_b \mathcal{N}(\mathbf{y}_n | \mathbf{c}_b + \mathbf{k}_b t_n, \boldsymbol{\tau}^{-1})}{\sum_{b'=1}^{B} \omega_{b'} \mathcal{N}(\mathbf{y}_n | \mathbf{c}_{b'} + \mathbf{k}_{b'} t_n, \boldsymbol{\tau}^{-1})} \tag{120}$$

UPDATE FOR k

$$k_{gb} | \boldsymbol{\gamma}, c_{gb}, \mathbf{Y}, \boldsymbol{\tau}, \mathbf{t} \sim \mathcal{N}(\nu_{gb}^k, 1/\lambda_{gb}^k) \tag{121}$$

where

$$\begin{aligned} \nu_{gb}^k &= \frac{\tau_g \sum_{n:\gamma_n=b} t_n (y_{ng} - c_{gb})}{\tau_k + \tau_g \sum_{n:\gamma_n=b} t_n^2} \\ \lambda_{gb}^k &= \tau_k + \tau_g \sum_{n:\gamma_n=b} t_n^2 \end{aligned} \tag{122}$$

UPDATE FOR c

$$c_{gb} | \boldsymbol{\gamma}, k_{gb}, \mathbf{Y}, \boldsymbol{\tau}, \mathbf{t} \sim \mathcal{N}(\nu_{gb}^c, 1/\lambda_{gb}^c) \tag{123}$$

where

$$\begin{aligned} \nu_{gb}^c &= \frac{\tau_g \sum_{n:\gamma_n=b} (y_{ng} - k_{gb} t_n)}{\tau_c + N_b \tau_g} \\ \lambda_{gb}^c &= \tau_c + N_b \tau_g \end{aligned} \tag{124}$$

and N_b is the number of cells assigned to branch b at that iteration.

UPDATE FOR ω

$$\boldsymbol{\omega} | \{N_b\}_{b=1}^B, B \sim \text{Dirichlet}(1/B + N_1, \ldots, 1/B + N_B) \tag{125}$$

UPDATE FOR t

$$t_n|\boldsymbol{\gamma}, \{\Lambda_b\}_{b=1}^B, \boldsymbol{\tau} \sim \mathcal{N}(\nu_n^t, 1/\lambda_n^t) \tag{126}$$

where

$$\begin{aligned} \nu_n^t &= \frac{\sum_g \tau_g k_{g\gamma_n}(y_{ng} - c_{g\gamma_n})}{1 + \sum_g \tau_g k_{g\gamma_n}^2} \\ \lambda_n^t &= 1 + \sum_g \tau_g k_{g\gamma_n}^2 \end{aligned} \tag{127}$$

UPDATE FOR $\boldsymbol{\tau}$

$$\tau_g|\{\Lambda_b\}_{b=1}^B, \mathbf{t}, \boldsymbol{\gamma} \sim \text{Gamma}\left(\alpha + N/2, \beta + \sum_{n=1}^N \frac{(y_{ng} - \mu_{ng})^2}{2}\right) \tag{128}$$

where $\mu_{ng} = c_{g\gamma_n} + k_{g\gamma_n} t_n$.

UPDATE FOR η

$$\eta|\tau_c, \{\mathbf{c}_b\}_{b=1}^B, \tau_\eta, \tilde{\eta} \sim \mathcal{N}(\nu^\eta, 1/\lambda^\eta) \tag{129}$$

where

$$\begin{aligned} \nu^\eta &= \frac{\tau_c \sum_{b,g} c_{gb} + \tau_\eta \tilde{\eta}}{BG\tau_c + \tau_\eta} \\ \lambda^\eta &= BG\tau_c + \tau_\eta \end{aligned} \tag{130}$$

UPDATE FOR θ

$$\theta_g|\boldsymbol{\chi}, \{\mathbf{k}_b\}_{b=1}^B, \tau_\theta, \tilde{\theta} \sim \mathcal{N}(\nu_g^\theta, 1/\lambda_g^\eta) \tag{131}$$

where

$$\begin{aligned}\nu_g^\theta &= \frac{\chi_g \sum_b k_{gb} + \tau_\theta \tilde{\theta}}{B\chi_g + \tau_\theta} \\ \lambda_g^\theta &= B\chi_g + \tau_\theta\end{aligned} \tag{132}$$

UPDATE FOR χ

$$\chi_g | \{\mathbf{k}_b\}_{b=1}^B, \theta_g, \alpha_\chi, \beta_\chi \sim \text{Gamma}\left(\alpha_\chi + \frac{B}{2}, \beta_\chi + \frac{\sum_b (k_{gb} - \theta_g)^2}{2}\right) \tag{133}$$

D.2 VALIDATION OF UPDATES

All Gibbs updates were checked numerically using the Gwecke test (see e.g. [42]). This exploits the identity

$$\frac{p(\theta = x|\Theta, \mathcal{D})}{p(\theta = x'|\Theta, \mathcal{D})} = \frac{p(\theta = x, \Theta|\mathcal{D})}{p(\theta = x', \Theta|\mathcal{D})} \tag{134}$$

which should hold up to the numerical precision of the computer used. We can therefore use the function that computes the likelihood (which we have implemented to monitor convergence) to ensure the conditional updates are correct, and vice versa.

E

INFERENCE FOR COVARIATE-ADJUSTED LATENT VARIABLE MODELS

E.1 OVERVIEW

For both Gibbs sampling and variational inference we require the conditional densities $p(\theta_i|\theta_{-i}, \mathbf{Y})$.

Recall we have data in the form of an $N \times G$ matrix $\mathbf{Y}$ and an $N \times P$ matrix $\mathbf{X}$ for N samples, G features and P covariates. Our model is

$$\begin{aligned}
\alpha_{pg} &\sim \mathcal{N}(0, \tau_\alpha^{-1}) \\
\lambda_g &\sim \mathcal{N}(0, \tau_\lambda^{-1}) \\
z_n &\sim \mathcal{N}(q_n, \tau_q^{-1}) \\
\beta_{pg} &\sim \mathcal{N}(0, \chi_{pg}^{-1}) \\
\chi_{pg}^{-1} &\sim \text{Gamma}(a_\beta, b_\beta) \\
\tau_g^{-1} &\sim \text{Gamma}(a, b) \\
\epsilon_{ng} &\sim \mathcal{N}(0, \tau_g^{-1}) \\
y_{ng} &= \mu_g + \sum_p \alpha_{pg} x_{np} + \left(\lambda_g + \sum_p \beta_{pg} x_{np}\right) z_n + \epsilon_{ng}
\end{aligned} \tag{135}$$

where τ_α, τ_λ, a, b, a_β, b_β, τ_q are fixed hyperparameters and q_n encodes prior information about z_n if available but typically $q_n = 0 \; \forall n$ in the uninformative case.

The motivation to reintroduce μ_g even if the columns of $\mathbf{Y}$ are centred is as follows: if we consider (in a frequentist setting) the factor loadings to be fixed and Y, X and Z as random variables, then we have

$$E[Y_{ng}] \propto \lambda_g E[Z] + \sum_p \beta_{pg} E[X_{pg} Z_n] \tag{136}$$

So if we wish to place an informative prior on Z (ie $E[Z] \neq 0$) then $E[Y] \neq 0$ and we have to introduce gene-specific intercepts. This also demonstrates why it is advantageous to have the covariates as uncorrelated with the desired "pseudotimes" as possible: covariance here takes the marginal expectation of Y away from zero, biasing modelling assumptions.

E.2 GIBBS UPDATES

The conditional distributions are given below (where $\theta|\cdot$ can be interpreted as the conditional distribution of variable θ conditioned on *all* other variables and the data). For simplicity we assume the summation is obvious from the variable (ie $\sum_p \equiv \sum_{p=1}^P$, etc).

CONDITIONAL DISTRIBUTION OF $\mathbf{z}$

$$z_n|\cdot \sim \mathcal{N}\left(\frac{\sum_g \tau_g k_{ng}(y_{ng} - \mu_g - \sum_p \alpha_{pg} x_{np}) + \tau_q q_n}{\sum_g \tau_g k_{ng}^2 + \tau_q}, [\sum_g \tau_g k_{ng}^2 + \tau_q]^{-1}\right) \tag{137}$$

where $k_{ng} = \lambda_g + \sum_p \beta_{pg} x_{np}$.

CONDITIONAL DISTRIBUTION OF α_{pg}

$$\alpha_{pg}|\cdot \sim \mathcal{N}\left(\frac{\tau_g \sum_n (y_{ng} - \tilde{\mu}_{ng}^{\alpha_p}) x_{np}}{\tau_g \sum_n x_{np}^2 + \tau_\alpha}, [\tau_g \sum_n x_{np}^2 + \tau_\alpha]^{-1}\right) \tag{138}$$

where

$$\tilde{\mu}_{ng}^{\alpha_p} = \mu_g + t_n(\lambda_g + \sum_{p'} \beta_{p'g} x_{np'}) + \sum_{p' \neq p} \alpha_{p'g} x_{np'} \tag{139}$$

in which $\sum_{p' \neq p}$ denotes the summation over 1 to P excluding p.

CONDITIONAL DISTRIBUTION OF β_{pg}

$$\beta_{pg}|\cdot \sim \mathcal{N}\left(\frac{\tau_g \sum_n (y_{ng} - \tilde{\mu}_{ng}^{\beta_p}) x_{np} z_n}{\tau_g \sum_n z_n^2 x_{np}^2 + \chi_{pg}}, [\tau_g \sum_n z_n^2 x_{np}^2 + \chi_{pg}]^{-1}\right) \tag{140}$$

where

$$\tilde{\mu}_{ng}^{\beta_p} = \mu_g + z_n \lambda_g + \sum_{p'} \alpha_{p'g} x_{np'} + z_n \sum_{p' \neq p} \beta_{p'g} x_{np'} \tag{141}$$

CONDITIONAL DISTRIBUTION OF τ_g

$$\tau_g|\cdot \sim \text{Gamma}\left(a + \frac{N}{2}, b + \sum_n \frac{(y_{ng} - \tilde{\mu}_{ng}^{\tau})^2}{2}\right) \tag{142}$$

where

$$\tilde{\mu}_{ng}^{\tau} = \mu_g + \sum_p \alpha_{pg} x_{np} + \left(\lambda_g + \sum_p \beta_{pg} x_{np}\right) \lambda_n. \tag{143}$$

CONDITIONAL DISTRIBUTION OF χ_{pg}

$$\chi_{pg}|\cdot \sim \text{Gamma}\left(a_\beta + \frac{1}{2}, b_\beta + \frac{\beta_{pg}^2}{2}\right) \tag{144}$$

CONDITIONAL DISTRIBUTION OF λ_g

$$\lambda_g|\cdot \sim \mathcal{N}\left(\frac{\tau_g \sum_n z_n(y_{ng} - \tilde{\mu}^{\lambda}_{ng})}{\tau_g \sum_n z_n^2 + \tau_\lambda}, [\tau_g \sum_n t_n^2 + \tau_\lambda]^{-1}\right) \tag{145}$$

where

$$\tilde{\mu}^{\lambda}_{ng} = \mu_g + \sum_p \alpha_{pg} x_{np} + \left(\sum_p \beta_{pg} x_{np}\right) z_n \tag{146}$$

CONDITIONAL DISTRIBUTION OF μ_g

$$\mu_g|\cdot \sim \mathcal{N}\left(\frac{\tau_g \sum_n (y_{ng} - \nu_{ng})}{N\tau_g + \tau_\mu}, [N\tau_g + \tau_\mu]^{-1}\right) \tag{147}$$

where

$$\nu_{ng} = \sum_p \alpha_{pg} x_{np} + \left(\lambda_g + \sum_p \beta_{pg} x_{np}\right) z_n \tag{148}$$

E.3 VARIATIONAL INFERENCE

E.3.1 *Conditional expectations*

CONDITIONAL EXPECTATION OF $\mathbf{z}$

$$\begin{aligned}
\mathbb{E}_{-z_n}[\mu_{z_n}\tau_{z_n}] &= \sum_g \left[\frac{a_{\tau_g}}{b_{\tau_g}}\left(m_{\lambda_g} + \sum_p m_{\beta_{pg}} x_{np}\right)\right. \\
&\quad \left.\times \left(y_{ng} - m_{\mu_g} - \sum_p m_{\alpha_{pg}} x_{np}\right)\right] + \tau_q q_n \\
\mathbb{E}_{-z_n}[\tau_{z_n}] &= \sum_g \frac{a_{\tau_g}}{b_{\tau_g}}\left(m^2_{\lambda_g} + s^2_{\lambda_g} + 2m_{\lambda_g}\sum_p m_{\beta_{pg}} x_{np}\right. \\
&\quad \left. + \sum_p (m^2_{\beta_{pg}} + s^2_{\beta_{pg}}) x^2_{np} + \sum_{p,p':p\neq p'} m_{\beta_{pg}} m_{\beta_{p'g}} x_{np} x_{np'}\right) + \tau_q
\end{aligned} \tag{149}$$

Where we have used the fact that

$$\mathbb{E}_{-z_n}[(\lambda_g + \sum_p \beta_{pg} x_{np})^2] = \Big(m_{\lambda_g}^2 + s_{\lambda_g}^2 + 2m_{\lambda_g} \sum_p m_{\beta_{pg}} x_{np} + \sum_p (m_{\beta_{pg}}^2 + s_{\beta_{pg}}^2) x_{np} + \sum_{p,p':p\neq p'} m_{\beta_{pg}} m_{\beta_{p'g}} x_{np} x_{np'}\Big) \tag{150}$$

and for several variables that $\mathbb{E}_{-z_n}[\theta^2] = \mathrm{Var}_{-z_n}[\theta] + \mathbb{E}_{-z_n}[\theta]^2$.

CONDITIONAL EXPECTATION OF α_{pg}

$$\begin{aligned}
\mathbb{E}_{-\alpha_{pg}}[\mu_{pg}\tau_{pg}] &= \frac{a_{\tau_g}}{b_{\tau_g}} \sum_n \Big(y_{ng} - m_{\mu_g} - m_{z_n}(m_{\lambda_g} + \sum_{p'} m_{\beta_{p'g}} x_{np}) \\
&\quad - \sum_{p'\neq p} m_{\alpha_{p'g}} x_{np'}\Big) x_{np} \\
\mathbb{E}_{-\alpha_{pg}}[\tau_{\alpha_{pg}}] &= \frac{a_{\tau_g}}{b_{\tau_g}} \sum_n x_{np}^2 + \tau_\alpha
\end{aligned} \tag{151}$$

CONDITIONAL EXPECTATION OF β_{pg}

$$\begin{aligned}
\mathbb{E}_{-\beta_{pg}}[\mu_{\beta_{pg}}\tau_{\beta_{pg}}] &= \frac{a_{\tau_g}}{b_{\tau_g}} \sum_n \Big[y_{ng} - m_{\mu_g} - \frac{m_{z_n}^2 + s_{z_n}^2}{m_{z_n}} m_{\lambda_g} \\
&\quad - \sum_{p'} m_{\alpha_{p'g}} x_{np'} - \frac{m_{z_n}^2 + s_{z_n}^2}{m_{z_n}} \sum_{p'\neq p} m_{\beta_{p'g}} x_{np}\Big] m_{z_n} x_{np} \\
\mathbb{E}_{-\beta_{pg}}[\tau_{\beta_{pg}}] &= \frac{a_{\tau_g}}{b_{\tau_g}} \sum_n (m_{z_n}^2 + s_{z_n}^2) x_{np}^2 + \frac{a_{\chi_{pg}}}{b_{\chi_{pg}}}
\end{aligned} \tag{152}$$

Where in both cases we have used the fact that $\mathbb{E}_{-\beta_{pg}}[z_n^2] = \mathrm{Var}_{-\beta_{pg}}[z_n] + \mathbb{E}_{-\beta_{pg}}[z_n]^2$.

CONDITIONAL EXPECTATION OF τ_g

$$\begin{aligned}
\mathbb{E}_{-\tau_g}[a_{\tau_g}] &= a + \frac{N}{2} \\
\mathbb{E}_{-\tau_g}[b_{\tau_g}] &= b + \frac{1}{2} \sum_n f_{ng}
\end{aligned} \tag{153}$$

Where

$$
\begin{aligned}
f_{ng} &= \mathbb{E}_{-\tau_g}\left[\left(y_{ng} - \mu_g - \sum_p \alpha_{pg}x_{np} - \left(\lambda_g + \sum_p \beta_{pg}x_{np}\right) z_n\right)^2\right] \\
&= \mathbb{E}_{-\tau_g}\Bigg[\mu_g^2 + 2\mu_g \sum_p \alpha_{pg}x_{np} 2\mu_g z_n \lambda_g \\
&+ 2\mu_g z_n \sum_p \beta_{pg}x_{np} - 2y_{ng}\mu_g + + (\sum_p \alpha_{pg}x_{np})^2 \\
&2z_n\lambda_g \sum_p \alpha_{pg}x_{np} + 2z_n(\sum_p \alpha_{pg}x_{np})(\sum_p \beta_{pg}x_{np}) \\
&- 2y_{ng}\sum_p \alpha_{pg}x_{np} + z_n^2\lambda_g^2 + 2\lambda_g z_n^2 \sum_p \beta_{pg}x_{np} \\
&- 2\lambda_g z_n y_{ng} + z_n^2(\sum_p \beta_{pg}x_{np})^2 \\
&- 2y_{ng}z_n\sum_p \beta_{pg}x_{np} + y_{ng}^2\Bigg]
\end{aligned}
\tag{154}
$$

For this we require the identities

$$
\begin{aligned}
\mathbb{E}[\theta^2] &= \mathrm{Var}[\theta] + \mathbb{E}[\theta]^2 \\
\mathbb{E}[(\sum_p \gamma_{pg}x_{np})^2] &= \sum_p x_{np}^2 \mathbb{E}[\gamma_{pg}^2] + \sum_{p,p':p\neq p'} x_{np}x_{np'}\mathbb{E}[\gamma_{pg}\gamma_{p'g}]
\end{aligned}
\tag{155}
$$

This gives

$$
\begin{aligned}
f_{ng} = & \, m_{\mu_g}^2 + s_{\mu_g}^2 + \\
& + 2m_{\mu_g} \sum_p m_{\alpha_{pg}} x_{np} \\
& + 2m_{\mu_g} m_{z_n} m_{\lambda_g} \\
& + 2m_{\mu_g} m_{z_n} \sum_p m_{\beta_{pg}} x_{np} \\
& - 2y_{ng} m_{\mu_g} \\
& + \sum_p (m_{\alpha_{pg}}^2 + s_{\alpha_{pg}}^2) x_{np}^2 + \sum_{p,p':p \neq p'} m_{\alpha_{pg}} m_{\alpha_{p'g}} x_{np} x_{np'} \\
& + 2m_{z_n} m_{\lambda_g} \sum_p m_{\alpha_{pg}} x_{np} \\
& + 2m_{z_n} (\sum_p m_{\alpha_{pg}} x_{np}) (\sum_p m_{\beta_{pg}} x_{np}) \\
& - 2y_{ng} \sum_p m_{\alpha_{pg}} x_{np} \\
& + (m_{z_n}^2 + s_{z_n}^2)(m_{\lambda_g}^2 + s_{\lambda_g}^2) \\
& + 2(m_{z_n}^2 + s_{z_n}^2) m_{\lambda_g} \sum_p m_{\beta_{pg}} x_{np} \\
& - 2m_{\lambda_g} m_{z_n} y_{ng} \\
& + (m_{z_n}^2 + s_{z_n}^2) \Big[\sum_p (m_{\beta_{pg}}^2 + s_{\beta_{pg}}^2) x_{np}^2 \\
& + \sum_{p,p':p \neq p'} m_{\beta_{pg}} m_{\beta_{p'g}} x_{np} x_{np'} \Big] \\
& - 2m_{z_n} y_{ng} \sum_p m_{\beta_{pg}} x_{np}) \\
& + y_{ng}^2
\end{aligned}
\tag{156}
$$

CONDITIONAL EXPECTATION OF χ_{pg}

$$
\begin{aligned}
\mathbb{E}_{-\chi_{pg}}[a_{\chi_{pg}}] &= a_\beta + \frac{1}{2} \\
\mathbb{E}_{-\chi_{pg}}[b_{\chi_{pg}}] &= b_\beta + \frac{1}{2} \left(m_{\beta_{pg}}^2 + s_{\beta_{pg}}^2 \right)
\end{aligned}
\tag{157}
$$

where again we have used the fact that $\mathbb{E}_{-\chi_{pg}}[\beta_{pg}^2] = \text{Var}_{-\chi_{pg}}[\beta_{pg}] + \mathbb{E}_{-\chi_{pg}}[\beta_{pg}]^2$.

CONDITIONAL EXPECTATION OF λ_g

$$\begin{aligned}\mathbb{E}_{-\lambda_g}[\mu_{\lambda_g}\tau_{\lambda_g}] &= \frac{a_{\tau_g}}{b_{\tau_g}}\sum_n m_{z_n}\Bigg(y_{ng} - m_{\mu_g} \\ &\quad - \sum_{p'} m_{\alpha_{p'g}} x_{np'} - \frac{m_{z_n}^2 + s_{z_n}^2}{m_{z_n}}\Big(\sum_{p'} m_{\beta_{p'g}} x_{np'}\Big)\Bigg) \\ \mathbb{E}_{-\lambda_g}[\tau_{\lambda_g}] &= \frac{a_{\tau_g}}{b_{\tau_g}}\sum_n (m_{z_n}^2 + s_{z_n}^2) + \tau_\lambda\end{aligned} \tag{158}$$

CONDITIONAL EXPECTATION OF μ_g

$$\begin{aligned}\mathbb{E}_{-\mu_g}[\mu_{\mu_g}\tau_{\mu_g}] &= \frac{a_{\tau_g}}{b_{\tau_g}}\sum_n \Bigg(y_{ng} - \sum_{p'} m_{\alpha_{p'g}} x_{np'} \\ &\quad - m_{z_n}\big(m_{\lambda_g} + \sum_{p'} m_{\beta_{p'g}} x_{np'}\big)\Bigg) \\ \mathbb{E}_{-\mu_g}[\tau_{\mu_g}] &= \frac{a_{\tau_g}}{b_{\tau_g}} N + \tau_\mu\end{aligned} \tag{159}$$

E.3.2 *Calculating the ELBO*

To assess convergence of the CAVI algorithm we need to calculate the evidence lower bound (ELBO) at every iteration (or every i^{th} iteration). The ELBO is given by

$$\text{ELBO} = \mathbb{E}[\log p(\mathbf{Y}|\Theta)] + \mathbb{E}[\log p(\Theta)] - \mathbb{E}[\log q(\Theta)] \tag{160}$$

where all expectations are taken with respect to the approximating distribution $Q(\cdot)$ and Θ denotes the full parameter set. Note that we are implicitly conditioning on the data wherever appropriate, so $p(\mathbf{Y}|\Theta) \equiv p(\mathbf{Y}|\Theta, \mathbf{X})$. For this we require the result

that if $\theta \sim \text{Gamma}(a, b)$ then $\mathbb{E}[\log \theta] = \phi(a) - \log b$ where ϕ is the digamma function $\phi(x) = \frac{\Gamma'(x)}{\Gamma(x)}$.

DERIVATION OF $\mathbb{E}[\log p(\mathbf{Y}|\Theta)]$ We have

$$\log p(\mathbf{Y}|\Theta) = \sum_n \sum_g \log \mathcal{N}(y_{ng}|\mu_{ng}, \tau_g^{-1}) \tag{161}$$

where $\mu_{ng} = \mu_g + \sum_p \alpha_{pg} x_{np} + z_n \left(\lambda_g + \sum_p \beta_{pg} x_{np}\right)$. Then

$$\begin{aligned} \mathbb{E}[\log p(\mathbf{Y}|\Theta)] &\propto \sum_g \left[\frac{N}{2} \mathbb{E}[\log \tau_g] - \frac{\mathbb{E}[\tau_g]}{2} \sum_n \mathbb{E}[(y_{ng} - \mu_{ng})^2] \right] \\ &= \sum_g \left[\frac{N}{2} (\phi(a_{\tau_g}) - \log b_{\tau_g}) - \frac{a_{\tau_g}}{2 b_{\tau_g}} \sum_n f_{ng} \right] \end{aligned} \tag{162}$$

where f_{ng} is defined as above and we have dropped additive terms since we are only concerned by changes in the ELBO.

DERIVATION OF $\mathbb{E}[\log p(\Theta)]$ We consider $\mathbb{E}[\log p(z_n)]$ which generalises to all parameters with Gaussian priors. We have

$$\begin{aligned} \mathbb{E}[\log p(z_n)] &= \mathbb{E}\left[\frac{1}{2} \log \frac{\tau_q}{2\pi} - \frac{\tau_q}{2} (z_n - q_n)^2 \right] \\ &= \frac{1}{2} \log \frac{\tau_q}{2\pi} - \frac{\tau_q}{2} \mathbb{E}[z_n^2 - 2 z_n q_n + q_n^2] \\ &= \frac{1}{2} \log \frac{\tau_q}{2\pi} - \frac{\tau_q}{2} \left(m_{z_n}^2 + s_{z_n}^2 - 2 m_{z_n} q_n + q_n^2 \right) \end{aligned} \tag{163}$$

Next consider $\mathbb{E}[\log p(\tau_g)]$ which generalises to all parameters with Gamma priors. We have

$$\begin{aligned}\mathbb{E}[\log p(\tau_g)] &\propto \mathbb{E}[\log(\tau_g^{a-1}e^{-\tau_g b})] \\ &= (a-1)\mathbb{E}[\log \tau_g] - b\mathbb{E}[\tau_g] \\ &= (a-1)(\phi(a_{\tau_g}) - \log b_{\tau_g}) - \frac{a_{\tau_g}}{b_{\tau_g}}b\end{aligned} \tag{164}$$

Thus the expression across all parameters up to a constant value is given by

$$\begin{aligned}\mathbb{E}[\log p(\Theta)] \propto &-\frac{\tau_q}{2}\sum_n (m_{z_n}^2 + s_{z_n}^2 - 2m_{z_n}q_n) \\ &- \frac{\tau_\mu}{2}\sum_g (m_{\mu_g}^2 + s_{\mu_g}^2) - \frac{\tau_\lambda}{2}\sum_g (m_{\lambda_g}^2 + s_{\lambda_g}^2) \\ &+ \sum_g \left[(a-1)(\phi(a_{\tau_g}) - \log b_{\tau_g}) - \frac{a_{\tau_g}}{b_{\tau_g}}b\right] \\ &- \sum_p \sum_g \left[\frac{\tau_\alpha}{2}(m_{\alpha_{pg}}^2 + s_{\alpha_{pg}}^2) + \frac{a_{\chi_{pg}}}{2b_{\chi_{pg}}}(m_{\beta_{pg}}^2 + s_{\beta_{pg}}^2)\right. \\ &\left. - (a_\beta - 1)(\phi(a_{\chi_{pg}}) - \log b_{\chi_{pg}}) - \frac{a_{\chi_{pg}}}{b_{\chi_{pg}}}b_\beta\right]\end{aligned} \tag{165}$$

DERIVATION OF $\mathbb{E}[\log q(\Theta)]$ We consider $\mathbb{E}[\log q_z(z_n)]$ which naturally generalises to all parameters whose approximating distributions are Gaussian. We have

$$\begin{aligned}\mathbb{E}[\log q_z(z_n)] &\propto \mathbb{E}\left[-\frac{1}{2}\log s_{z_n}^2 - \frac{1}{2s_{z_n}^2}(z_n - m_{z_n})^2\right] \\ &= -\frac{1}{2}\log s_{z_n}^2 - \frac{1}{2s_{z_n}^2}\mathbb{E}[z_n^2 - 2z_n m_{z_n} + m_{z_n}^2] \\ &= -\frac{1}{2}\log s_{z_n}^2 - \frac{1}{2s_{z_n}^2}\mathbb{E}[m_{z_n}^2 + s_{z_n}^2 - 2m_{z_n}^2 + m_{z_n}^2] \\ &\propto -\frac{1}{2}\log s_{z_n}^2\end{aligned} \tag{166}$$

Similarly we consider $\mathbb{E}[\log q_\tau(\tau_g)]$ which generalises to all parameters whose approximating distribution is Gamma. We have

$$\begin{aligned}\mathbb{E}[\log q_\tau(\tau_g)] &= \mathbb{E}[a_{\tau_g}\log b_{\tau_g} + (a_{\tau_g}-1)\log\tau_g - \tau_g b_{\tau_g} - \log\Gamma(a_{\tau_g})] \\ &= a_{\tau_g}\log b_{\tau_g} + (a_{\tau_g}-1)(\phi(a_{\tau_g}) - \log b_{\tau_g}) - a_{\tau_g} - \log\Gamma(a_{\tau_g})\end{aligned} \tag{167}$$

Summing this across all parameters gives

$$\begin{aligned}\mathbb{E}[\log q(\Theta)] = &-\frac{1}{2}\sum_n s^2_{z_n} \\ &+ \sum_g \Big(-\frac{1}{2}s^2_{\mu_g} - \frac{1}{2}s^2_{\lambda_g} + a_{\tau_g}\log b_{\tau_g} + (a_{\tau_g}-1)(\phi(a_{\tau_g}) - \log b_{\tau_g}) \\ &- a_{\tau_g} - \log\Gamma(a_{\tau_g})\Big) \\ &+ \sum_g\sum_p \Big(-\frac{1}{2}s^2_{\alpha_{pg}} - \frac{1}{2}s^2_{\beta_{pg}} \\ &+ a_{\chi_{pg}}\log b_{\chi_{pg}} + (a_{\chi_{pg}}-1)(\phi(a_{\chi_{pg}}) - \log b_{\chi_{pg}}) - a_{\chi_{pg}} - \log\Gamma(a_{\chi_{pg}})\Big)\end{aligned} \tag{168}$$

E.4 STAN CODE FOR CGPLVM

```
data {
  int N; // number of samples
  int G; // number of features
  vector [N] Y[G]; // gene expression input
  real x[N]; // covariate measurements
  real z_prior[N]; // priors on z
  real<lower = 0> z_sd;
```

```
  // CGPLVM kernel parameters (fixed)
  real <lower = 0> delta;
  real <lower = 0> eta;
  real <lower = 0> nu;
  real <lower = 0> gamma;
  real <lower = 0> xi;
}

transformed data {
  vector[N] mu;
  for(i in 1:N) mu[i] = 0;
}

parameters {
  real<lower = 0> sigma2[G]; // variance for each gene
  real<lower = 0> lambda[G]; // interaction dependence
  real z[N]; // pseudotimes
}

model {
  matrix[N, N] Sigma[G];

  for(i in 1:(N-1)) {
    for(j in (i+1):N) {
      for(g in 1:G) {
        Sigma[g,i,j] = nu * exp(-gamma * pow(z[i] - z[j], 2)) +
        xi * exp(-lambda[g] * pow(z[i] * x[i] - z[j] * x[j], 2))
            +
```

```
          delta * exp(-eta * pow(x[i] - x[j], 2));

          Sigma[g,j,i] = Sigma[g,i,j];
        }
      }
    }
    for(i in 1:N) {
      for(g in 1:G) {
        Sigma[g,i,i] = delta + nu + xi + sigma2[g] + 1e-6;
      }
    }

    for(i in 1:N) {
      z[i] ~ normal(z_prior[i], z_sd);
    }

    for(g in 1:G) {
      lambda[g] ~ chi_square(1.0);
      1 / sigma2[g] ~ gamma(0.1, 0.1);
      Y[g] ~ multi_normal(mu, Sigma[g]);
    }
}
```

BIBLIOGRAPHY

[1] El-ad David Amir, Kara L Davis, Michelle D Tadmor, Erin F Simonds, Jacob H Levine, Sean C Bendall, Daniel K Shenfeld, Smita Krishnaswamy, Garry P Nolan, and Dana Pe'er. "viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia." In: *Nature biotechnology* 31.6 (June 2013), pp. 545–52. ISSN: 1546-1696. DOI: 10.1038/nbt.2594. URL: http://www.ncbi.nlm.nih.gov/pubmed/23685480.

[2] Simon Anders and Wolfgang Huber. "Differential expression analysis for sequence count data." In: *Genome biology* 11.10 (Jan. 2010), R106. ISSN: 1465-6914. DOI: 10.1186/gb-2010-11-10-r106. URL: http://genomebiology.com/2010/11/10/R106.

[3] T S Andrews and M Hemberg. "Modelling dropouts allows for unbiased identification of marker genes in scRNASeq experiments." In: *bioRxiv* (2016).

[4] Christof Angermueller et al. "Parallel single-cell sequencing links transcriptional and epigenetic heterogeneity." en. In: *Nat. Methods* 13.3 (Mar. 2016), pp. 229–232.

[5] Rhonda Bacher, Li-Fang Chu, Ning Leng, Audrey P Gasch, James A Thomson, Ron M Stewart, Michael Newton, and Christina Kendziorski. "SCnorm: robust normalization of single-cell RNA-seq data." en. In: *Nat. Methods* 14.6 (June 2017), pp. 584–586.

[6] Mikhail Belkin and Partha Niyogi. "Laplacian Eigenmaps for Dimensionality Reduction and Data." In: 1396 (2003), pp. 1373–1396.

[7] Sean C Bendall, Erin F Simonds, Peng Qiu, D Amir El-ad, Peter O Krutzik, Rachel Finck, Robert V Bruggner, Rachel Melamed, Angelica Trejo, Olga I Ornatsky, et al. "Single-cell mass cytometry of differential immune and drug responses across a human hematopoietic continuum." In: *Science* 332.6030 (2011), pp. 687–696.

[8] Sean C Bendall, Kara L Davis, El-Ad David Amir, Michelle D Tadmor, Erin F Simonds, Tiffany J Chen, Daniel K Shenfeld, Garry P Nolan, and Dana Pe'er. "Single-cell trajectory detection uncovers progression and regulatory coordination in human B cell development." en. In: *Cell* 157.3 (2014), pp. 714–725.

[9] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. "Variational inference: A review for statisticians." In: *arXiv preprint arXiv:1601.00670* (2016).

[10] C R Boland and A Goel. "Microsatellite instability in colorectal cancer." In: *Gastroenterology* (2010).

[11] Valérie Bonadona et al. "Cancer risks associated with germline mutations in MLH1, MSH2, and MSH6 genes in Lynch syndrome." en. In: *JAMA* 305.22 (2011), pp. 2304–2310.

[12] Nicolas L Bray, Harold Pimentel, Páll Melsted, and Lior Pachter. "Near-optimal probabilistic RNA-seq quantification." en. In: 34.5 (May 2016), pp. 525–527.

[13] Philip Brennecke et al. "Accounting for technical noise in single-cell RNA-seq experiments." en. In: *Nat. Methods* 10.11 (Nov. 2013), pp. 1093–1095.

[14] Florian Buettner and Fabian J Theis. "A novel approach for resolving differences in single-cell gene expression patterns from zygote to blastocyst." In: *Bioinformatics* 28.18 (2012), pp. i626–i632.

[15] Joseph C. Burns, Michael C. Kelly, Michael Hoa, Robert J. Morell, and Matthew W. Kelley. "Single-cell RNA-Seq resolves cellular complexity in sensory organs

from the neonatal inner ear." In: *Nature Communications* 6 (2015), p. 8557. ISSN: 2041-1723. DOI: 10.1038/ncomms9557. URL: http://www.nature.com/doifinder/10.1038/ncomms9557.

[16] Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. "A limited memory algorithm for bound constrained optimization." In: *SIAM Journal on Scientific Computing* 16.5 (1995), pp. 1190–1208.

[17] K Campbell and C Yau. "Ouija: Incorporating prior knowledge in single-cell trajectory learning using Bayesian nonlinear factor analysis." In: *bioRxiv* (2016).

[18] Kieran R Campbell and Christopher Yau. "switchde: Inference of switch-like differential expression along single-cell trajectories." In: *Bioinformatics* (2016), btw798.

[19] Kieran R Campbell and Christopher Yau. "Probabilistic modeling of bifurcations in single-cell gene expression data using a Bayesian mixture of factor analyzers." In: *Wellcome Open Research* 2 (2017).

[20] Kieran Campbell, Chris P Ponting, and Caleb Webber. "Laplacian eigenmaps and principal curves for high resolution pseudotemporal ordering of single-cell RNA-seq profiles." In: *bioRxiv* (2015), p. 027219.

[21] Kieran Campbell and Christopher Yau. "Bayesian Gaussian Process Latent Variable Models for pseudotime inference in single-cell RNA-seq data." In: *bioRxiv* (2015), p. 026872.

[22] Kieran Campbell and Christopher Yau. "Order under uncertainty: robust differential expression analysis using probabilistic models for pseudotime inference." In: *bioRxiv* 1 (2016), p. 047365.

[23] Robrecht Cannoodt, Wouter Saelens, Dorine Sichien, Simon Tavernier, Sophie Janssens, Martin Guilliams, Bart N Lambrecht, Katleen De Preter, and Yvan

Saeys. “SCORPIUS improves trajectory inference and identifies novel modules in dendritic cell development.” In: *bioRxiv* (2016), p. 079509.

[24] Bob Carpenter, Andrew Gelman, Matt Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus A Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. “Stan: a probabilistic programming language.” In: *Journal of Statistical Software* (2015).

[25] Minhua Chen, Jorge Silva, John Paisley, Chunping Wang, David Dunson, and Lawrence Carin. “Compressive sensing on manifolds using a nonparametric mixture of factor analyzers: Algorithm and performance bounds.” In: *IEEE Transactions on Signal Processing* 58.12 (2010), pp. 6140–6155.

[26] S J Clark, R Argelaguet, C A Kapourani, T M Stubbs, and others. “Joint Profiling Of Chromatin Accessibility, DNA Methylation And Transcription In Single Cells.” In: *bioRxiv* (2017).

[27] Mihails Delmans and Martin Hemberg. “Discrete distributional differential expression (D3E)–a tool for gene expression analysis of single-cell RNA-seq data.” en. In: *BMC Bioinformatics* 17 (2016), p. 110.

[28] Xiaolan Deng, Gottfried Von Keudell, Takehiro Suzuki, Naoshi Dohmae, Makoto Nakakido, Lianhua Piao, Yuichiro Yoshioka, Yusuke Nakamura, and Ryuji Hamamoto. “PRMT1 promotes mitosis of cancer cells through arginine methylation of INCENP.” In: *Oncotarget* 6.34 (2015), p. 35173.

[29] D van Dijk, J Nainys, R Sharma, P Kathail, A J Carr, and others. “MAGIC: A diffusion-based imputation method reveals gene-gene interactions in single-cell RNA-sequencing data.” In: *bioRxiv* (2017).

[30] Chenfang Dong, Tingting Yuan, Yadi Wu, Yifan Wang, Teresa WM Fan, Sumitra Miriyala, Yiwei Lin, Jun Yao, Jian Shi, Tiebang Kang, et al. “Loss of FBP1

by Snail-mediated repression provides metabolic advantages in basal-like breast cancer." In: *Cancer Cell* 23.3 (2013), pp. 316–331.

[31] Early Breast Cancer Trialists' Collaborative Group (EBCTCG). "Relevance of breast cancer hormone receptors and other factors to the efficacy of adjuvant tamoxifen: patient-level meta-analysis of randomised trials." In: *Lancet* 378.9793 (), pp. 771–784.

[32] Michael Eisenstein. *Startups use short-read data to expand long-read sequencing market*. 2015.

[33] Andrea Facciabene, Gregory T Motz, and George Coukos. "T-regulatory cells: key players in tumor immune escape and angiogenesis." In: *Cancer Research* 72.9 (2012), pp. 2162–2171.

[34] Napoleone Ferrara. "VEGF and the quest for tumour angiogenesis factors." In: *Nature Reviews Cancer* 2.10 (2002), pp. 795–803.

[35] Greg Finak et al. "MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data." en. In: *Genome Biol.* 16 (2015), p. 278.

[36] C Fraley, A E Raftery, T B Murphy, and L Scrucca. "mclust Version 4 for R: Normal Mixture Modeling for Model-Based Clustering, Classification, and Density Estimation. 2012." In: *University of Washington: Seattle* ().

[37] Alyssa C Frazee, Andrew E Jaffe, Ben Langmead, and Jeffrey T Leek. "Polyester: simulating RNA-seq datasets with differential transcript expression." en. In: *Bioinformatics* 31.17 (2015), pp. 2778–2784.

[38] Yanyan Gao, Yaping Zhao, Juechao Zhang, Yang Lu, Xin Liu, Pengyu Geng, Baiqu Huang, Yu Zhang, and Jun Lu. "The dual function of PRMT1 in modu-

lating epithelial-mesenchymal transition and cellular senescence in breast cancer cells through regulation of ZEB1." In: *Scientific reports* 6 (2016).

[39] Andrew Gelman, Daniel Lee, and Jiqiang Guo. "Stan A Probabilistic Programming Language for Bayesian Inference and Optimization." In: *Journal of Educational and Behavioral Statistics* (2015), p. 1076998615606113.

[40] Reference Genome Group of the Gene Ontology Consortium et al. "The Gene Ontology's Reference Genome Project: a unified framework for functional annotation across species." In: *PLoS Comput Biol* 5.7 (2009), e1000431.

[41] J J P Gille et al. "Genomic deletions of MSH2 and MLH1 in colorectal cancer families detected by a novel mutation detection approach." en. In: *Br. J. Cancer* 87.8 (2002), pp. 892–897.

[42] Roger B Grosse and David K Duvenaud. "Testing mcmc code." In: *arXiv preprint arXiv:1412.5218* (2014).

[43] Dominic Grün, Anna Lyubimova, Lennart Kester, Kay Wiebrands, Onur Basak, Nobuo Sasaki, Hans Clevers, and Alexander van Oudenaarden. "Single-cell messenger RNA sequencing reveals rare intestinal cell types." en. In: *Nature* 525.7568 (2015), pp. 251–255.

[44] Anupam Gupta and Ziv Bar-Joseph. "Extracting dynamics from static cancer expression data." en. In: *IEEE/ACM Trans. Comput. Biol. Bioinform.* 5.2 (Apr. 2008), pp. 172–182.

[45] L. Haghverdi, F. Buettner, and F. J. Theis. "Diffusion maps for high-dimensional single-cell analysis of differentiation data." In: *Bioinformatics* May (2015), pp. 1–10. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btv325. URL: http://www.ncbi.nlm.nih.gov/pubmed/26002886.

[46] Laleh Haghverdi, Maren Buettner, F Alexander Wolf, Florian Buettner, and Fabian J Theis. "Diffusion pseudotime robustly reconstructs lineage branching." In: *Nature Methods* (2016).

[47] Tamar Hashimshony, Florian Wagner, Noa Sher, and Itai Yanai. "CEL-Seq: single-cell RNA-Seq by multiplexed linear amplification." In: *Cell reports* 2.3 (2012), pp. 666–673.

[48] Trevor Hastie and Werner Stuetzle. "Principal Curves." en. In: (Mar. 2012). URL: http://www.tandfonline.com/doi/abs/10.1080/01621459.1989.10478797.

[49] James Hensman, Nicolo Fusi, and Neil D Lawrence. "Gaussian processes for big data." In: *arXiv preprint arXiv:1309.6835* (2013).

[50] S C Hicks, M Teng, and R A Irizarry. "On the widespread and critical impact of systematic bias and batch effects in single-cell RNA-Seq data." In: *bioRxiv* (2015).

[51] Stephanie C Hicks, F William Townes, Mingxiang Teng, and Rafael A Irizarry. "Missing Data and Technical Variability in Single-Cell RNA-Sequencing Experiments." In: *bioRxiv* (2017), p. 025528.

[52] Geoffrey E Hinton and Sam T Roweis. "Stochastic neighbor embedding." In: *Advances in neural information processing systems*. 2002, pp. 833–840.

[53] Matthew D Hoffman and Andrew Gelman. "The No-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo." In: *J. Mach. Learn. Res.* 15.1 (2014), pp. 1593–1623.

[54] Matthew D Hoffman, David M Blei, Chong Wang, and John William Paisley. "Stochastic variational inference." In: *Journal of Machine Learning Research* 14.1 (2013), pp. 1303–1347.

[55] Youjin Hu, Kevin Huang, Qin An, Guizhen Du, Ganlu Hu, Jinfeng Xue, Xianmin Zhu, Cun-Yu Wang, Zhigang Xue, and Guoping Fan. "Simultaneous profiling of transcriptome and DNA methylome from a single cell." en. In: *Genome Biol.* 17 (2016), p. 88.

[56] Saiful Islam, Una Kjällquist, Annalena Moliner, Pawel Zajac, Jian-Bing Fan, Peter Lönnerberg, and Sten Linnarsson. "Characterization of the single-cell transcriptional landscape by highly multiplex RNA-seq." In: *Genome research* 21.7 (2011), pp. 1160–1167.

[57] Zhicheng Ji and Hongkai Ji. "TSCAN: Pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis." en. In: *Nucleic Acids Res.* 44.13 (2016), e117.

[58] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

[59] Peter V Kharchenko, Lev Silberstein, and David T Scadden. "Bayesian approach to single-cell differential expression analysis." In: *Nature methods* 11.7 (July 2014), pp. 740–2. ISSN: 1548-7105. DOI: 10.1038/nmeth.2967. URL: http://www.ncbi.nlm.nih.gov/pubmed/24836921.

[60] Diederik P Kingma and Max Welling. "Auto-Encoding Variational Bayes." In: (2013). arXiv: 1312.6114v10 [stat.ML].

[61] Allon M Klein, Linas Mazutis, Ilke Akartuna, Naren Tallapragada, Adrian Veres, Victor Li, Leonid Peshkin, David A Weitz, and Marc W Kirschner. "Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells." In: *Cell* 161.5 (2015), pp. 1187–1201.

[62] Aleksandra A Kolodziejczyk, Jong Kyoung Kim, Jason CH Tsang, Tomislav Ilicic, Johan Henriksson, Kedar N Natarajan, Alex C Tuck, Xuefei Gao, Marc Bühler,

Pentao Liu, et al. “Single cell RNA-sequencing of pluripotent states unlocks modular transcriptional variation.” In: *Cell Stem Cell* 17.4 (2015), pp. 471–485.

[63] Aleksandra A Kolodziejczyk, Jong Kyoung Kim, Valentine Svensson, John C Marioni, and Sarah A Teichmann. “The technology and biology of single-cell RNA sequencing.” en. In: *Mol. Cell* 58.4 (2015), pp. 610–620.

[64] Keegan D Korthauer, Li-Fang Chu, Michael A Newton, Yuan Li, James Thomson, Ron Stewart, and Christina Kendziorski. “A statistical approach for identifying differential distributions in single-cell RNA-seq experiments.” en. In: *Genome Biol.* 17.1 (2016), p. 222.

[65] Marcin Kosinski and Przemyslaw Biecek. *RTCGA: The Cancer Genome Atlas Data Integration*. R package version 1.4.0. 2016. URL: https://rtcga.github.io/RTCGA.

[66] Monika S Kowalczyk, Itay Tirosh, Dirk Heckl, Tata Nageswara Rao, Atray Dixit, Brian J Haas, Rebekka K Schneider, Amy J Wagers, Benjamin L Ebert, and Aviv Regev. “Single-cell RNA-seq reveals changes in cell cycle and differentiation programs upon aging of hematopoietic stem cells.” en. In: *Genome Res.* 25.12 (Dec. 2015), pp. 1860–1872.

[67] Eric W-F Lam, Jan J Brosens, Ana R Gomes, and Chuay-Yeng Koo. “Forkhead box proteins: tuning forks for transcriptional harmony.” In: *Nature Reviews Cancer* 13.7 (2013), pp. 482–495.

[68] Charity W Law, Yunshun Chen, Wei Shi, and Gordon K Smyth. “voom: Precision weights unlock linear model analysis tools for RNA-seq read counts.” en. In: *Genome Biol.* 15.2 (2014), R29.

[69] Neil D Lawrence. "Gaussian process latent variable models for visualisation of high dimensional data." In: *Advances in neural information processing systems*. 2004, pp. 329–336.

[70] Neil Lawrence. "Probabilistic Non-linear Principal Component Analysis with Gaussian Process Latent Variable Models." In: *J. Mach. Learn. Res.* 6.Nov (2005), pp. 1783–1816.

[71] Quoc V Le, Alex J Smola, and Stéphane Canu. "Heteroscedastic Gaussian process regression." In: *Proceedings of the 22nd international conference on Machine learning*. ACM. 2005, pp. 489–496.

[72] Ryan Lister, Ronan C O'Malley, Julian Tonti-Filippini, Brian D Gregory, Charles C Berry, A Harvey Millar, and Joseph R Ecker. "Highly integrated single-base resolution maps of the epigenome in Arabidopsis." In: *Cell* 133.3 (2008), pp. 523–536.

[73] Tapio Lönnberg, Valentine Svensson, Kylie R James, Daniel Fernandez-Ruiz, Ismail Sebina, Ruddy Montandon, Megan SF Soon, Lily G Fogg, Michael JT Stubbington, Frederik Otzen Bagger, et al. "Temporal mixture modelling of single-cell RNA-seq data resolves a CD4+ T cell fate bifurcation." In: *bioRxiv* (2016), p. 074971.

[74] Michael I Love, Wolfgang Huber, and Simon Anders. "Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2." en. In: *Genome Biol.* 15.12 (2014), p. 550.

[75] Aaron T L Lun, Karsten Bach, and John C Marioni. "Pooling across cells to normalize single-cell RNA sequencing data with many zero counts." en. In: *Genome Biol.* 17 (2016), p. 75.

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

[77] Laurens van der Maaten. "Preserving local structure in Gaussian process latent variable models." In: *Proceedings of the 18th Annual Belgian-Dutch Conference on Machine Learning*. 2009, pp. 81–88.

[78] Iain C Macaulay, Chris P Ponting, and Thierry Voet. "Single-Cell Multiomics: Multiple Measurements from Single Cells." en. In: *Trends Genet.* 33.2 (Feb. 2017), pp. 155–168.

[79] Iain C Macaulay et al. "G&T-seq: parallel sequencing of single-cell genomes and transcriptomes." en. In: *Nat. Methods* 12.6 (2015), pp. 519–522.

[80] Iain C Macaulay, Valentine Svensson, Charlotte Labalette, Lauren Ferreira, Fiona Hamey, Thierry Voet, Sarah A Teichmann, and Ana Cvejic. "Single-Cell RNA-Sequencing Reveals a Continuous Spectrum of Differentiation in Hematopoietic Cells." en. In: *Cell Rep.* 14.4 (2016), pp. 966–977.

[81] Iain C Macaulay, Valentine Svensson, Charlotte Labalette, Lauren Ferreira, Fiona Hamey, Thierry Voet, Sarah A Teichmann, and Ana Cvejic. "Single-cell RNA-sequencing reveals a continuous spectrum of differentiation in hematopoietic cells." In: *Cell reports* 14.4 (2016), pp. 966–977.

[82] Evan Z Macosko et al. "Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets." en. In: *Cell* 161.5 (2015), pp. 1202–1214.

[83] Paul M Magwene, Paul Lizardi, and Junhyong Kim. "Reconstructing the temporal ordering of biological samples using microarray data." en. In: *Bioinformatics* 19.7 (2003), pp. 842–850.

[84] Qi Mao, Li Wang, Ivor W Tsang, and Yijun Sun. "A novel regularized principal graph learning framework on explicit graph representation." In: *arXiv preprint arXiv:1512.02752* (2015).

[85] Eugenio Marco, Robert L Karp, Guoji Guo, Paul Robson, Adam H Hart, Lorenzo Trippa, and Guo-Cheng Yuan. "Bifurcation analysis of single-cell gene expression data reveals epigenetic landscape." en. In: *Proc. Natl. Acad. Sci. U. S. A.* 111.52 (2014), E5643–50.

[86] Eugenio Marco, Robert L Karp, Guoji Guo, Paul Robson, Adam H Hart, Lorenzo Trippa, and Guo-Cheng Yuan. "Bifurcation analysis of single-cell gene expression data reveals epigenetic landscape." In: *Proceedings of the National Academy of Sciences* 111.52 (2014), E5643–E5650.

[87] Davis J McCarthy, Kieran R Campbell, Aaron T L Lun, and Quin F Wills. "Scater: pre-processing, quality control, normalization and visualization of single-cell RNA-seq data in R." en. In: *Bioinformatics* (2017).

[88] Gil McVean. "A genealogical interpretation of principal components analysis." In: *PLoS Genet* 5.10 (2009), e1000686.

[89] S Michel, A Benner, M Tariverdian, N Wentzensen, P Hoefler, T Pommerencke, N Grabe, M von Knebel Doeberitz, and M Kloor. "High density of FOXP3-positive T cells infiltrating colorectal cancers with microsatellite instability." In: *British journal of cancer* 99.11 (2008), pp. 1867–1873.

[90] Jeffrey W Miller and Matthew T Harrison. "A simple example of Dirichlet process mixture inconsistency for the number of components." In: *Advances in neural information processing systems*. 2013, pp. 199–206.

[91] Alyssa M Molinaro and Bret J Pearson. “In silico lineage tracing through single cell transcriptomics identifies a neural stem cell population in planarians.” In: *Genome biology* 17.1 (2016), p. 1.

[92] Ali Mortazavi, Brian A Williams, Kenneth McCue, Lorian Schaeffer, and Barbara Wold. “Mapping and quantifying mammalian transcriptomes by RNA-Seq.” In: *Nature methods* 5.7 (2008), pp. 621–628.

[93] Keefe Murphy, Isobel Claire Gormley, and Cinzia Viroli. “Infinite Mixtures of Infinite Factor Analysers: Nonparametric Model-Based Clustering via Latent Gaussian Models.” In: *arXiv preprint arXiv:1701.07010* (2017).

[94] Ugrappa Nagalakshmi, Zhong Wang, Karl Waern, Chong Shou, Debasish Raha, Mark Gerstein, and Michael Snyder. “The transcriptional landscape of the yeast genome defined by RNA sequencing.” In: *Science* 320.5881 (2008), pp. 1344–1349.

[95] Radford M Neal et al. “MCMC using Hamiltonian dynamics.” In: *Handbook of Markov Chain Monte Carlo* 2 (2011), pp. 113–162.

[96] Cancer Genome Atlas Network et al. “Comprehensive molecular characterization of human colon and rectal cancer.” In: *Nature* 487.7407 (2012), pp. 330–337.

[97] Sait Ozturk, Panagiotis Papageorgis, Chen Khuan Wong, Arthur W Lambert, Hamid M Abdolmaleky, Arunthathi Thiagalingam, Herbert T Cohen, and Sam Thiagalingam. “SDPR functions as a metastasis suppressor in breast cancer by promoting apoptosis.” In: *Proceedings of the National Academy of Sciences* 113.3 (2016), pp. 638–643.

[98] F F Parl, B P Schmidt, W D Dupont, and R K Wagner. “Prognostic significance of estrogen receptor status in breast cancer in relation to tumor stage, axillary node metastasis, and histopathologic grading.” en. In: *Cancer* 54.10 (1984), pp. 2237–2242.

[99] Franziska Paul, Ya'ara Arkin, Amir Giladi, Diego Adhemar Jaitin, Ephraim Kenigsberg, Hadas Keren-Shaul, Deborah Winter, David Lara-Astiaso, Meital Gury, Assaf Weiner, et al. "Transcriptional heterogeneity and lineage commitment in myeloid progenitors." In: *Cell* 163.7 (2015), pp. 1663–1677.

[100] Tobias Petri, Evi Berchtold, Ralf Zimmer, and Caroline C Friedel. "Detection and correction of probe-level artefacts on microarrays." In: *BMC bioinformatics* 13.1 (2012), p. 114.

[101] Simone Picelli, Åsa K Björklund, Omid R Faridani, Sven Sagasser, Gösta Winberg, and Rickard Sandberg. "Smart-seq2 for sensitive full-length transcriptome profiling in single cells." en. In: *Nat. Methods* 10.11 (Nov. 2013), pp. 1096–1098.

[102] Emma Pierson and Christopher Yau. "ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis." In: *Genome Biology* 16.1 (2015), p. 1.

[103] H J Pimentel, N Bray, S Puente, P Melsted, and L Pachter. "Differential analysis of RNA-Seq incorporating quantification uncertainty." In: *bioRxiv* (2016).

[104] Alex A Pollen, Tomasz J Nowakowski, Joe Shuga, Xiaohui Wang, Anne A Leyrat, Jan H Lui, Nianzhen Li, Lukasz Szpankowski, Brian Fowler, Peilin Chen, et al. "Low-coverage single-cell mRNA sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex." In: *Nature biotechnology* 32.10 (2014), pp. 1053–1058.

[105] Jean-Francois Poulin, Bosiljka Tasic, Jens Hjerling-Leffler, Jeffrey M Trimarchi, and Rajeshwar Awatramani. "Disentangling neural cell diversity using single-cell transcriptomics." en. In: *Nat. Neurosci.* 19.9 (2016), pp. 1131–1141.

[106] Peng Qiu, Andrew J Gentles, and Sylvia K Plevritis. "Discovering biological progression underlying microarray samples." en. In: *PLoS Comput. Biol.* 7.4 (Apr. 2011), e1001123.

[107] Xiaojie Qiu, Qi Mao, Ying Tang, Li Wang, Raghav Chawla, Hannah Pliner, and Cole Trapnell. "Reversed graph embedding resolves complex single-cell developmental trajectories." In: *bioRxiv* (2017), p. 110668.

[108] Xiaojie Qiu, Andrew Hill, Jonathan Packer, Dejun Lin, Yi-An Ma, and Cole Trapnell. "Single-cell mRNA quantification and differential analysis with Census." en. In: *Nat. Methods* 14.3 (Mar. 2017), pp. 309–315.

[109] Daniel Ramsköld et al. "Full-length mRNA-Seq from single-cell levels of RNA and individual circulating tumor cells." en. In: *Nat. Biotechnol.* 30.8 (Aug. 2012), pp. 777–782.

[110] Carl Edward Rasmussen and Christopher KI Williams. *Gaussian processes for machine learning*. Vol. 1. MIT press Cambridge, 2006.

[111] Aviv Regev et al. "The Human Cell Atlas." In: *bioRxiv* (2017).

[112] John E Reid and Lorenz Wernisch. "Pseudotime estimation: deconfounding single cell time series." en. In: *Bioinformatics* 32.19 (2016), pp. 2973–2980.

[113] Markus Ringnér. "What is principal component analysis?" In: *Nature biotechnology* 26.3 (2008), p. 303.

[114] D Risso, F Perraudeau, S Gribkova, S Dudoit, and J P Vert. "ZINB-WaVE: A general and flexible method for signal extraction from single-cell RNA-seq data." In: *bioRxiv* (2017).

[115] Mark D Robinson, Davis J McCarthy, and Gordon K Smyth. "edgeR: a Bioconductor package for differential expression analysis of digital gene expression data." en. In: *Bioinformatics* 26.1 (2010), pp. 139–140.

[116] John Salvatier, Thomas V Wiecki, and Christopher Fonnesbeck. "Probabilistic programming in Python using PyMC3." In: *PeerJ Computer Science* 2 (2016), e55.

[117] Manu Setty, Michelle D Tadmor, Shlomit Reich-Zeliger, Omer Angel, Tomer Meir Salame, Pooja Kathail, Kristy Choi, Sean Bendall, Nir Friedman, and Dana Pe'er. "Wishbone identifies bifurcating developmental trajectories from single-cell data." In: *Nature biotechnology* 34.6 (2016), pp. 637–645.

[118] Richard Sever and Joan S Brugge. "Signal transduction in cancer." en. In: *Cold Spring Harb. Perspect. Med.* 5.4 (2015).

[119] Alex K Shalek et al. "Single-cell RNA-seq reveals dynamic paracrine control of cellular variation." en. In: *Nature* 510.7505 (2014), pp. 363–369.

[120] Jaehoon Shin et al. "Single-Cell RNA-Seq with Waterfall Reveals Molecular Cascades underlying Adult Neurogenesis." English. In: *Cell Stem Cell* 17.3 (Aug. 2015), pp. 360–372. ISSN: 19345909. DOI: 10.1016/j.stem.2015.07.013. URL: http://www.cell.com/article/S1934590915003124/fulltext.

[121] Charles Spearman. """ General Intelligence," Objectively Determined and Measured." In: *The American Journal of Psychology* 15.2 (1904), pp. 201–292.

[122] Fuchou Tang et al. "mRNA-Seq whole-transcriptome analysis of a single cell." en. In: *Nat. Methods* 6.5 (May 2009), pp. 377–382.

[123] P J Tatlow and Stephen R Piccolo. "A cloud-based workflow to quantify transcript-expression levels in public cancer compendia." en. In: *Sci. Rep.* 6 (2016), p. 39259.

[124] Michael E Tipping and Christopher M Bishop. "Probabilistic principal component analysis." In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61.3 (1999), pp. 611–622.

[125] Michalis K Titsias. "Variational Learning of Inducing Variables in Sparse Gaussian Processes." In: *AISTATS*. Vol. 12. 2009, pp. 567–574.

[126] Michalis Titsias and Neil Lawrence. "Bayesian Gaussian Process Latent Variable Model." In: *Artificial Intelligence* 9 (2010), pp. 844–851. URL: http://eprints.pascal-network.org/archive/00006343/.

[127] Dustin Tran, Alp Kucukelbir, Adji B Dieng, Maja Rudolph, Dawen Liang, and David M Blei. "Edward: A library for probabilistic modeling, inference, and criticism." In: (2016). arXiv: 1610.09787 [stat.CO].

[128] Cole Trapnell. "Defining cell types and states with single-cell genomics." en. In: *Genome Res.* 25.10 (Oct. 2015), pp. 1491–1498.

[129] 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." en. In: *Nat. Biotechnol.* 32.4 (Apr. 2014), pp. 381–386.

[130] Barbara Treutlein, Doug G Brownfield, Angela R Wu, Norma F Neff, Gary L Mantalas, F Hernan Espinoza, Tushar J Desai, Mark A Krasnow, and Stephen R Quake. "Reconstructing lineage hierarchies of the distal lung epithelium using single-cell RNA-seq." en. In: *Nature* 509.7500 (2014), pp. 371–375.

[131] Po-Yuan Tung, John D Blischak, Chiaowen Joyce Hsiao, David A Knowles, Jonathan E Burnett, Jonathan K Pritchard, and Yoav Gilad. "Batch effects and the effective design of single-cell gene expression studies." In: *Scientific Reports* 7 (2017).

[132] Jengnan Tzeng, Henry Horng-Shing Lu, and Wen-Hsiung Li. "Multidimensional scaling for large genomic data sets." In: *BMC bioinformatics* 9.1 (2008), p. 179.

[133] Dmitry Usoskin et al. "Unbiased classification of sensory neuron types by large-scale single-cell RNA sequencing." en. In: *Nat. Neurosci.* 18.1 (Jan. 2015), pp. 145–153.

[134] Bo Wang, Junjie Zhu, Emma Pierson, Daniele Ramazzotti, and Serafim Batzoglou. "Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning." en. In: *Nat. Methods* 14.4 (Apr. 2017), pp. 414–416.

[135] Jiayu Wang, Binghe Xu, Peng Yuan, Pin Zhang, Qing Li, Fei Ma, and Ying Fan. "TOP2A amplification in breast cancer is a predictive marker of anthracycline-based neoadjuvant chemotherapy efficacy." In: *Breast cancer research and treatment* 135.2 (2012), pp. 531–537.

[136] Shen SJ Wang and Matt P Wand. "Using infer. net for statistical analyses." In: *The American Statistician* 65.2 (2011), pp. 115–126.

[137] John N Weinstein et al. "The cancer genome atlas pan-cancer analysis project." In: *Nat. Genet.* 45.10 (2013), pp. 1113–1120.

[138] Joshua D Welch, Alexander J Hartemink, and Jan F Prins. "SLICER: inferring branched, nonlinear cellular trajectories from single cell RNA-seq data." In: *Genome biology* 17.1 (2016), p. 106.

[139] Jonathan Welti, Sonja Loges, Stefanie Dimmeler, and Peter Carmeliet. "Recent molecular discoveries in angiogenesis and antiangiogenic therapies in cancer." In: *The Journal of Clinical Investigation* 123.8 (2013), pp. 3190–3200.

[140] Michael L Whitfield, Gavin Sherlock, Alok J Saldanha, John I Murray, Catherine A Ball, Karen E Alexander, John C Matese, Charles M Perou, Myra M Hurt, Patrick O Brown, et al. "Identification of genes periodically expressed in the human cell cycle and their expression in tumors." In: *Molecular biology of the cell* 13.6 (2002), pp. 1977–2000.

[141] Christopher Yau et al. “pcaReduce: hierarchical clustering of single cell transcriptional profiles.” In: *BMC bioinformatics* 17.1 (2016), p. 140.

[142] Ka Yee Yeung and Walter L. Ruzzo. “Principal component analysis for clustering gene expression data.” In: *Bioinformatics* 17.9 (2001), pp. 763–774.

[143] Matthew D Young, Matthew J Wakefield, Gordon K Smyth, and Alicia Oshlack. “Gene ontology analysis for RNA-seq: accounting for selection bias.” In: *Genome biology* 11.2 (2010), p. 1.

[144] Yi Yu-Rice, Yanli Jin, Bingchen Han, Ying Qu, Jeffrey Johnson, Takaaki Watanabe, Long Cheng, Nan Deng, Hisashi Tanaka, Bowen Gao, et al. “FOXC1 is involved in ERα silencing by counteracting GATA3 binding and is implicated in endocrine resistance.” In: *Oncogene* (2016).

[145] Haiyuan Yu, Katherine Nguyen, Tom Royce, Jiang Qian, Kenneth Nelson, Michael Snyder, and Mark Gerstein. “Positional artifacts in microarrays: experimental verification and construction of COP, an automated detection tool.” In: *Nucleic acids research* 35.2 (2006), e8–e8.

[146] Amit Zeisel et al. “Brain structure. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq.” en. In: *Science* 347.6226 (2015), pp. 1138–1142.

[147] Grace X Y Zheng et al. “Massively parallel digital transcriptional profiling of single cells.” en. In: *Nat. Commun.* 8 (2017), p. 14049.

[148] Fan Zhou, Xianlong Li, Weili Wang, Ping Zhu, Jie Zhou, Wenyan He, Meng Ding, Fuyin Xiong, Xiaona Zheng, Zhuan Li, et al. “Tracing haematopoietic stem cell formation at single-cell resolution.” In: *Nature* 533.7604 (2016), pp. 487–492.

[149] Mo Huang et al. “Gene Expression Recovery For Single Cell RNA Sequencing.” In: *bioRxiv* (2017).