BAYESIAN INFERENCE OF HAEMATOPOIETIC STEM/PROGENITOR CELL DIFFERENTIATION PHENOTYPIC MANIFOLDS AND THEIR BIFURCATION POINTS USING GAUSSIAN PROCESS AND GIBBS SAMPLING

by

Riaan Deon Dowling

Submitted in partial fulfillment of the requirements for the degree
Master of Engineering (Electronic Engineering)

in the

Department of Electrical, Electronic and Computer Engineering
Faculty of Engineering, Built Environment and Information Technology

UNIVERSITY OF PRETORIA

March 2021

SUMMARY

BAYESIAN INFERENCE OF HAEMATOPOIETIC STEM/PROGENITOR CELL DIFFERENTIATION PHENOTYPIC MANIFOLDS AND THEIR BIFURCATION POINTS USING GAUSSIAN PROCESS AND GIBBS SAMPLING

by

Riaan Deon Dowling

Supervisor(s):	Prof. J. P. de Villiers and Prof. M. S. Pepper
Department:	Electrical, Electronic and Computer Engineering
University:	University of Pretoria
Degree:	Master of Engineering (Electronic Engineering)
Keywords:	Bayesian inference, Bayes factor, bifurcation points, cell differentiation, cell lineages, continuous modelling, Gaussian process, Gibbs sampler, gene expression, haematopoietic stem/progenitor cells, model selection, phenotypic manifold

Cell differentiation is the process by which cells progress through different stages of maturation to become specialised cell types. This process is complex and involves up- and down-regulation of many genes. A common objective for the field of developmental biology is to understand protein synthesis, which is regulated by gene expression that governs the cell differentiation process. Since the molecular signatures that govern cell differentiation are unique to specific cell types, researchers are turning to gene expression data to shed light on complex biological processes. Rapid advancements in high throughput genome-scale sequencing allow researchers to exploit gene expression data, and to characterise biological processes such as cell differentiation in new ways. Therefore, various mathematical models have been developed to better understand the cell differentiation process.

In this dissertation an algorithm is developed that focuses on haematopoietic cell differentiation where haematopoietic stem/progenitor cells progress through different stages of differentiation to produce mature blood cells. Hence, BAGEL: **B**ayesian **A**nalysis of **G**ene **E**xpression **L**ineages is presented, which contributes to the body of knowledge by: (i) transforming cell differentiation pseudo-time to a representation of sequential windows that are translated and rotated within a combined pseudo-time-principal-component space using the tangent vectors of window-based Frenet frames; (ii) inferring bifurcation points via a Gibbs sampler and Bayesian model selection; (iii) constructing cell lineage representations of single-cell gene expression data in the combined pseudo-time-principal-component space, that describe the cell developmental trajectory from the start of cell differentiation to a terminal state within a given dataset; (iv) modelling cell differentiation as a continuous process using a Gaussian process; and (v) projecting a related dataset onto the cell developmental trajectory of a primary dataset's phenotypic manifold. Using this projection process, even single-cell gene expression data from a different species can also be compared.

BAGEL was applied to two primary datasets namely mouse bone marrow single-cell RNA-seq and human bone marrow single-cell RNA-seq to confirm its functional capabilities on haematopoietic differentiation. A secondary human umbilical cord blood single-cell RNA-seq dataset was projected onto the human bone marrow data set and was deemed biologically sound as it corresponds to a well-established haematopoietic cell differentiation pathway previously described. Finally, human umbilical cord blood was projected onto mouse bone marrow to determine if there are similarities in gene expression patterns between different species. The significance of BAGEL is that it will allow researchers to gain new insights into developmental processes based on (i) a sound Bayesian inference approach to model cell differentiation; and (ii) an effective projection method which opens the door to visualise and investigate the similarities and differences between intra- and inter-species single-cell gene expression datasets.

LIST OF ABBREVIATIONS

b	binormal vector
BMI	basic marginal likelihood identity
cDNA	complementary deoxyribonucleic acid
CLP	common lymphoid progenitor
CMP	common myeloid progenitor
con	connectivity
cov	covariance
Dir	Dirichlet distribution
DNA	deoxyribonucleic acid
exp	exponent
GMP	granulocyte–macrophage progenitors
$\mathcal{GP}$	Gaussian process
GRNs	gene regulatory networks
HSPCs	haematopoietic stem and progenitor cells
LT-HSPCs	long-term-haematopoietic stem and progenitor cells
MCMC	Markov chain Monte-Carlo
MEP	megakaryocyte–erythrocyte progenitors
MPP	multipotent progenitors
mRNA	messenger ribonucleic acid
Mult	multinomial distribution
n	normal vector
$\mathcal{N}$	Gaussian/normal distribution
$\mathcal{NIW}$	normal-inverse-Wishart distribution
NK	natural killer
PC	principal component
PCA	principal component analysis
PO	percentage overlap
RNA	ribonucleic acid
RNA-seq	single-cell sequencing of ribonucleic acid
ST-HSPCs	short-term-haematopoietic stem and progenitor cells
UCB	umbilical cord blood
t	tangent vector
Tr	trace function

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION

1.1 PROBLEM STATEMENT

1.1.1 Context of the problem

Cell differentiation is a fundamental process in biology which is complex and not well understood [1–8]. This process plays a key role throughout cellular development, where a single-cell progresses through a series of orchestrated and continuous differentiation stages to form a terminal cell. Differentiation usually starts with an immature cell that differentiates into a diverse set of mature cell types [6, 9] through up- and down-regulation of many genes. Haematopoiesis is one example of cell differentiation where haematopoietic stem/progenitor cells (HSPCs; immature cells) progress through different stages of differentiation to produce and maintain a continuous supply of mature blood cells [10].

Rapid advances in high throughput genome-scale sequencing enable researchers to exploit biological processes such as cell differentiation in new ways [1, 4, 11, 12]. A common objective of the field of developmental biology is to understand protein synthesis [13], which is regulated by gene expression [14] that governs the cell differentiation process. Since the molecular signatures that govern cell differentiation are unique to specific cell types, researchers are turning to single-cell gene expression data to shed light on complex biological processes. To better understand the process of cell differentiation, many mathematical models have been developed to predict gene expression lineages during differentiation based on these unique gene expression patterns [1–9, 15]. The mathematical models assist in better understanding the cell differentiation process by providing new and novel insights into the differentiation process [1, 3].

1.1.2 Research gap

The use of Bayesian inference to model cell differentiation with Gaussian processes was recently discovered. In [16] maximum likelihood, a frequentist estimation approach, is used to estimate two bifurcation points within a cell's developmental trajectory. However, Bayesian inference can provide a quantitative measure of multiple bifurcation points and select the best one by utilising Bayes factor [17].

In the field of developmental biology it may be difficult to provide sampled single-cell gene expression data that is competitive in the single-cell field due to influencing factors such as (i) the high cost associated with the consumables when sampling single-cell gene expression data; and (ii) various quality control measures along the sampling process. Therefore, the Bayesian inference model is extended to address a need to visualise small numbers of sampled cells on the cell developmental trajectory of optimally sequenced dataset consisting of many cells.

1.2 RESEARCH OBJECTIVE AND QUESTIONS

The objective of this research is to provide researchers with novel insight about cell differentiation. This objective is met by addressing the following research questions:

- Can a single-cell gene expression dataset be used to visualise cell differentiation in a combined pseudo-time-principal-component space?
- Can Bayesian inference provide accurate bifurcation points within a cell's developmental trajectory in the combined pseudo-time-principal-component space?
- Can the detected bifurcation points be used to construct cell lineages, that represent the cell developmental trajectory from the start of cell differentiation to a terminal state in the combined pseudo-time-principal-component space?
- Can the constructed cell lineages be modelled as a continuous process with a Gaussian process in the combined pseudo-time-principal-component space?
- Can a secondary sub-sampled gene expression dataset be projected onto the phenotypic manifold of a primary optimally sequenced gene expression dataset consisting of many cells?
- Can the cell developmental trajectory of cell differentiation be represented with the tangent vector of a window-based Frenet frame in the combined pseudo-time-principal-component space?

1.3 HYPOTHESIS AND APPROACH

Owing to modern sampled cell differentiation data, the cell differentiation process will be modelled as a continuous Gaussian process with distinct bifurcation points indicating a cell's fate choices. During this modelling process, a sub-sampled secondary dataset will be accurately projected onto the phenotypic manifold of an optimally sequenced dataset consisting of many cells.

The approach to address this hypothesis is to investigate the four main steps required when modelling cell differentiation, described by state-of-the-art literature [1], shown in Figure 1.1. The first step is to clean up and normalise the sampled single-cell gene expression data, referred to as data pre-processing (P.1). The next step is to use the processed data to develop the underlying phenotypic manifold of the cell differentiation process via dimensionality reduction techniques (P.2). The phenotypic manifold is then used as input for trajectory inference (P.3) to estimate gene expression lineages. Finally, the results of the simulations are visualised with graphs and figures (P.4).

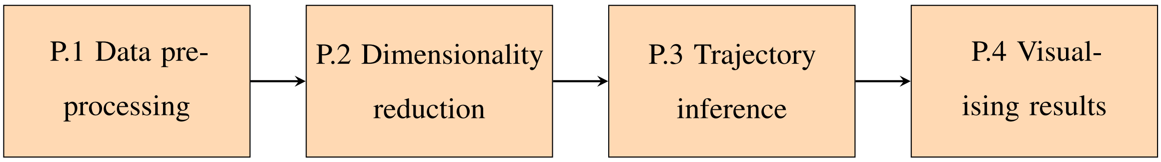


Figure 1.1. A flow diagram of the four steps to model cell differentiation. The first step is to clean up and normalise the sampled single-cell gene expression data, referred to as data pre-processing (P.1). The next step is to use the processed data to develop the underlying phenotypic manifold of the cell differentiation process via dimensionality reduction techniques (P.2). The phenotypic manifold is then used as input for trajectory inference (P.3) to estimate gene expression lineages. Finally, the results of the simulations are visualised with graphs and figures (P.4).

1.4 RESEARCH GOALS

A common goal for the field of developmental biology is to understand protein synthesis [13], which is regulated by gene expression [14] that governs the cell differentiation process. As the process of cell differentiation is a fundamental problem in biology [1], the goal is to develop a mathematical model to address this need in the biological world towards better understanding the cell differentiation process. Hence, the developed algorithm will focus on the modelling of the process whereby haematopoietic stem/progenitor cells differentiate into myeloid and erythroid lineages. This will involve the analysis and modelling of gene expressions over the course of multiple divisions, as the HSPCs differentiate into their final fate. The modelling of cell differentiation via computational analysis algorithms will provide new insights into the differentiation process as well as the biology of haematopoiesis.

1.5 RESEARCH CONTRIBUTION

In this dissertation an algorithm is developed, which contributes to the body of knowledge by (i) transforming cell differentiation pseudo-time to a representation of sequential windows that are translated and rotated within a combined pseudo-time-principal-component space using the tangent vectors of window-based Frenet frames; (ii) inferring bifurcation points via a Gibbs sampler and Bayesian model selection; (iii) constructing cell lineage representations of single-cell gene expression data in the combined pseudo-time-principal-component space, that describe the cell developmental trajectory from the start of cell differentiation to a terminal state within a given dataset; (iv) modelling cell differentiation as a continuous process using a Gaussian process; and (v) projecting a related dataset onto the cell developmental trajectory of a primary dataset's phenotypic manifold. Using this projection process, even single-cell gene expression data from a different species can also be compared.

1.6 RESEARCH OUTPUTS

The following article is being prepared for a peer-reviewed and ISI accredited journal.

- R.D. Dowling, J. Mellet, E. Wolmarans, C. Durandt, F. Joubert, J.P. de Villiers, M.S. Pepper, "Bayesian inference of haematopoietic stem/progenitor cell differentiation phenotypic manifolds and their bifurcation points using Gaussian processes and Gibbs sampling", preparing manuscript for journal.

1.7 OVERVIEW OF STUDY

The primary modelling goal of this dissertation is to develop an algorithm that is able to determine when a cell's fate choices are made. This primary modelling goal is achieved by (i) estimating bifurcation points via Bayesian inference, and (ii) modelling cell differentiation as a continuous process with a Gaussian process. A bifurcation point, in terms of biology, is defined as the exact instant a change in a cell's fate is detected along its developmental trajectory. An overly simplified visual illustration of this primary modelling goal is shown in Figure 1.2. As seen in Figure 1.2, cell differentiation can be visualised in two dimensions where the y-axis represents an arbitrary gene expression and the x-axis the pseudo-time ordering of cells. Starting at the green point, cells propagate (differentiate) towards their different terminal states (blue points) in a continuous fashion. Along this continuous cell developmental trajectory a clear bifurcation point is indicated that answers the following question: When are a cell's fate choices made? The secondary modelling goal is to develop a method that accurately projects a secondary sub-sampled single-cell gene expression dataset onto the phenotypic manifold of a primary optimally sequenced single-cell gene expression dataset consisting of many cells.

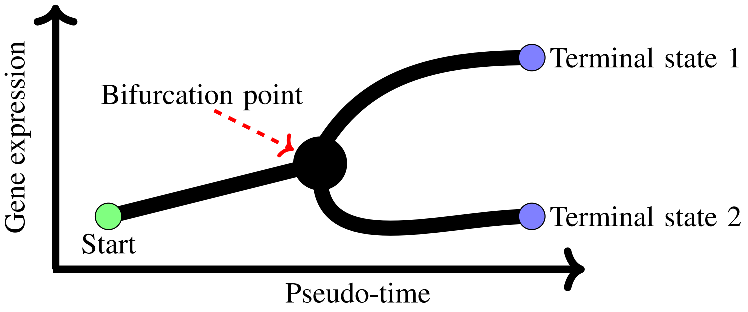


Figure 1.2. An overly simplified visual illustration of the algorithm's primary modelling goal. Cell differentiation can be visualised in two dimensions where the y-axis represents an arbitrary gene expression and the x-axis the pseudo-time ordering of cells. Starting at the green point, cells propagate (differentiate) towards their different terminal states (blue points) in a continuous fashion. Along this continuous developmental trajectory, a clear bifurcation point is indicated that answers the question: When is a cell's fate choice made?

The modelling goals of this dissertation is achieved by first presenting the relevant literature of gene expression modelling in Chapter 2. The relevant literature starts by providing a biological overview of cell differentiation and gene expression followed by pioneering modelling techniques. Next, all the statistical modelling mathematics required to understand the developed algorithm in Chapter 4 is discussed in Chapter 3. Chapter 5 presents the single-cell gene expression data used in the study as well as the experimental setup to obtain the results, as presented in Chapter 6. Finally, a sound conclusion about the conducted research is presented in Chapter 7.

CHAPTER 2 LITERATURE STUDY

2.1 CHAPTER OBJECTIVES

This chapter presents the literature study for this dissertation. Section 2.2 discusses the process of cell differentiation and how it is regulated by gene expression. This section is concluded by providing basic knowledge on how single-cell gene expression datasets are obtained. Section 2.3 discusses the processes involved in utilising the obtained single-cell gene expression datasets to develop cell differentiation models. Finally, Section 2.4 discuses the reasoning for this dissertation focusing on haematopoietic cell differentiation modelling, before concluding remarks in Section 2.5.

2.2 CELL DIFFERENTIATION

Cell differentiation is a fundamental process in biology which is complex and not well understood [1–8]. This process plays a key role throughout cellular development, where a single-cell progresses through a series of orchestrated and continuous differentiation stages to form a terminal cell. Differentiation usually starts with an immature cell that differentiates into a diverse set of mature cell types [6, 9] through up- and down-regulation of many genes. Haematopoiesis is one example of cell differentiation where haematopoietic stem/progenitor cells (HSPCs; immature cells) progress through different stages of differentiation to produce and maintain a continuous supply of mature blood cells [10]. In this dissertation, the foucs will be on haematopoietic cell differentiation in the human body, as well as haematopoietic cell differentiation in mice. However, as the cell differentiation between humans and mice are similar [18] only human cell differentiation will be discussed in this chapter. Hence, to approach the modelling of haematopoietic cell differentiation, it is necessary to understand the process of cell differentiation in the human body from fertilisation to a complete organism.

A fertilised egg cell (zygote) and consequently every cell in the human body contains approximately 6 billion DNA (deoxyribonucleic acid) base pairs (3 billion from the father and 3 billion from the mother). Approximately 1.5% of these base pairs are sequences, which are classified as genes that can result in the manufacturing of proteins. There are approximately 20 000 genes in the cell, of which only 3000 to 5000 cells are expressed (active) at any time, and the rest are suppressed (inactive). One half of the active genes perform basic metabolic functions, such as manufacturing energy. These are referred to as "housekeeping genes". The other half are expressed in a manner that is cell-type specific, and this determines the nature of the cell. The DNA sequence (and hence the sequence of the genes that an organism carries within its DNA) is known as its genotype. The genes that are expressed determine how cells will differentiate into a "final product" and how they will behave, and this is known as a phenotype.

There are approximately 350 cell types into which a zygote can differentiate. A zygote is a totipotent cell, which means that it has the potential to differentiate into all the different cell types that comprise a complete organism. Totipotent cells give rise to pluripotent cells, which can differentiate into cells of the three embryonic germ layers: endoderm, mesoderm and ectoderm. Between 10^{12} and 10^{14} cell divisions are required from a zygote to generate a complete organism. Over the course of many cell divisions, the differentiation potential of the cell decreases as the fate of the cell becomes more apparent. A cell with reduced differentiation potential compared to the totipotent cell is known as a multipotent cell. These cells do not have the ability to become a complete organism, but rather a few known cell types along their differentiation path to their final fate [14]. The cell differentiation process in the human body can, therefore, be defined as the sum of epigenetic, genetic and microenvironmental influences, referred to as healthy cell differentiation (differentiation process of interest). An example of unhealthy cell differentiation is cancer cells that have mutation variance during their cell differentiation processes [1, 19, 20].

2.2.1 Gene expression

The genetic information of every living organism on earth is stored in a molecule known as DNA [14]. This genetic information is divided into functional units called genes. Some of the genes act as instructions to produce functional products that perform a job inside a cell. These functional units are most likely protein, or more accurately, polypeptides (amino acid chains). The produced amino acid chains form a sequence of polypeptides, which determines the final three-dimensional structure and specific function of the produced polypeptides, e.g. muscle. Hence, the process by which the genetic code of a gene directs protein synthesis and produces a cell structure is known as gene expression. Gene expression consists of two main steps, namely transcription and translation. It should be noted that the following description of these two steps is on eukaryotic cells that contain a nucleus [14], as these cells are the focus of the dissertation.

Transcription is the process by which information stored in the DNA sequence of a gene is copied (transcribed) to a similar molecule called RNA (ribonucleic acid) within a cell nucleus. In essence, this is the process of RNA synthesis. To form a protein-coding gene, the RNA transcript is subjected to an additional series of processes to become a messenger RNA (mRNA). mRNA obtained its name due to its functionality, which is to transport genetic information from the DNA out of the nucleus. This step marks the end of transcription and the start of the translation process.

Translation is the process by which the gene information contained within the mRNA is used to build proteins. The process occurs within complex structures called ribosomes. Ribosomes "read" the information from the mRNA to orderly link the specific amino acids to build polypeptide chains. The nucleotides of the mRNA are read in pairs of three, known as codons, which represent a specific amino acid. When the ribosomes latch onto the mRNA, it will first find the "start" codon. Once the start codon is found, it will travel quickly down the mRNA, one codon at a time. This process will gradually build a chain of amino acids until it reaches three stop codons within the genetic information. The final constructed amino acid chain will precisely represent the codons of the mRNA. Hence, translation is the process in which mRNA directs protein synthesis. These steps of the process of gene expression can be visualised as shown in Figure 2.1. As seen, in Figure 2.1 the DNA (purple) gene information sequence is firstly transcribed (transcription in green) to an mRNA sequence (orange). Next, the mRNA sequence is used to direct protein synthesis through the process of translation (light blue) to form an amino acid chain (polypeptide in pink).

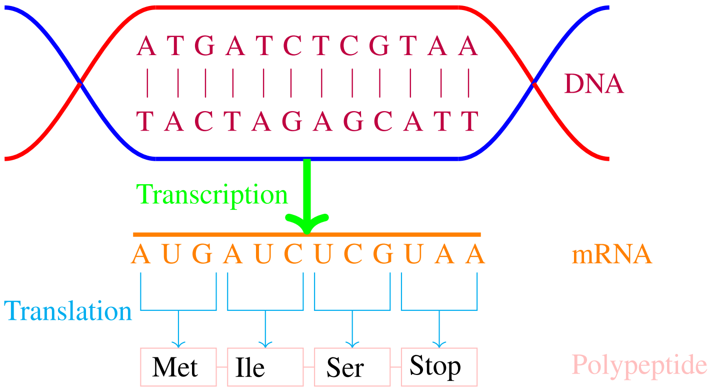


Figure 2.1. The DNA (purple) gene information sequence is firstly transcribed (transcription in green) to an mRNA sequence (orange). Next, the mRNA sequence is used to direct protein synthesis through the process of translation (light blue) to form an amino acid chain (polypeptide in pink) (Adapted from [14], with permission).

2.2.2 Gene Regulatory Networks (GRNs)

Gene regulation is a sequential process that regulates gene expression by determining which genes are expressed and which are inactive. A principal regulatory point in the cell differentiation is transcription. This is because transcription maintains gene expressions [14]. What this implies is that in order to model gene expression lineages, it is necessary to understand their regulatory landscape [7]. Developed mathematical models tend to describe regulation interactions between gene products with statistical techniques [21]. In [8] it was shown that by modelling the transcriptional dynamics of cell differentiation, unknown regulatory factors were discovered that influence gene expression. Hence, it is important to understand how a GRN regulates gene expressions to be able to develop the best possible model of cell differentiation.

There are two ways to visualise GRNs, the first being a biophysical model and the second an abstracted structure. Both these networks are depicted in Figure 2.2 and Figure 2.3 respectively. In Figure 2.2 (biophysical model), (i) different colours represent gene translation and transcription stages; (ii) the DNA of a specific gene is represented with pill-shaped blocks; (iii) the polypeptide of a specific gene due to translation and transcription is represented with ovals; and (iv) the arrows indicate the regulation of the network. Similarly, in Figure 2.3, (abstract model) (i) different colours represent the combined process of gene translation and transcription; (ii) gene expression is represented with a circle; and (iii) the arrows indicate the regulation of the network.

The biophysical model (Figure 2.2) indicates two types of regulation in terms of translation and transcription. The first regulation process is direct regulation performed on Gene 2 from Gene 1. This means that the polypeptide directly influences the gene expression of Gene 2. The second regulation is combinatorial regulation via complex formation by Gene 1 and Gene 2 on Gene 3. This indicates that the gene expression of Gene 3 is regulated by Gene 1 and Gene 2. The abstract structure is a simplification of the biophysical model, which only uses G to indicate a specific gene. Hence, the abstracted illustration removes the physical biological components and simply indicates which gene regulates which gene expression with the help of arrows.

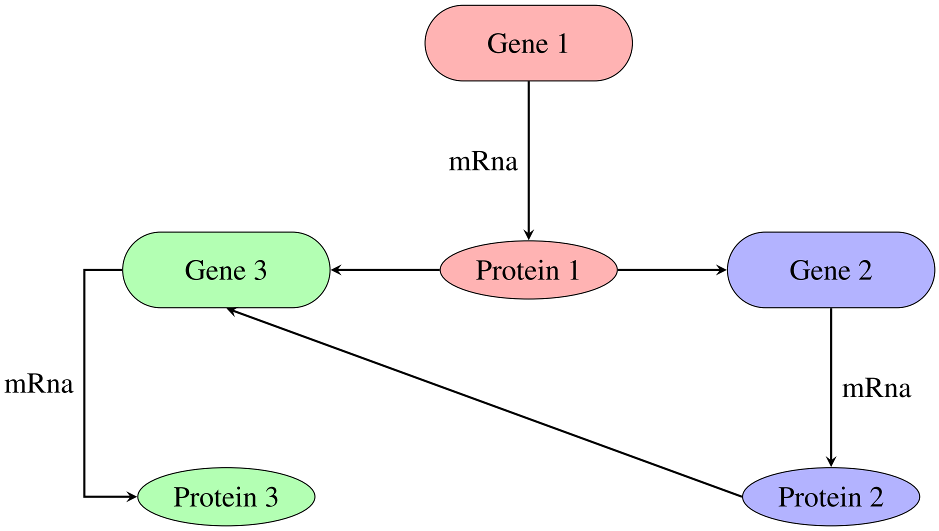


Figure 2.2. Biophysical model of GRN. In this model, (i) different colours represent gene translation and transcription stages; (ii) the DNA of a specific gene is represented with pill-shaped blocks; (iii) the polypeptide of a specific gene due to translation and transcription is represented with ovals; and (iv) the arrows indicate the regulation of the network (Adapted from [21], with permission).

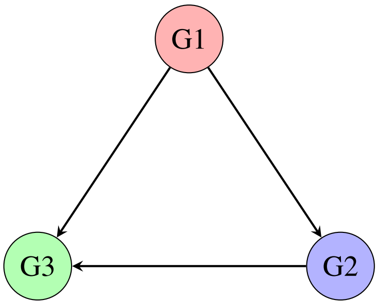


Figure 2.3. Abstracted structure of a GRN. In this model, (i) different colours represent the combined process of gene translation and transcription; (ii) gene expression is represented with a circle; and (iii) the arrows indicate the regulation of the network (Adapted from [21], with permission).

2.2.3 Sampling cell differentiation data

An important controlling factor in modelling cell fate choices, is the type of sampled data used. Until recently, gene expression studies have been limited to pooled populations of cells in order to obtain sufficient amounts of RNA, referred to as bulk RNA [22]. This provides an average of levels of gene expression across the many cell types present. Gene expression studies using RNA from pooled cell populations mask the uniqueness and heterogeneity of gene expression patterns present in individual cell types. This results mainly in abundant cell types being studied, while rare cell populations remain poorly characterised [12, 23]. The average gene expression is determined by weighting the summed cell type-specific gene expressions by cell type proportions. The first mammalian single-cell sequencing of RNA (RNA-seq) was performed in 2009 [24]. Single-cell RNA sequencing enables researchers to uncover the uniqueness of each cell to investigate cell differentiation and cell fate [1, 3–5, 7, 8, 12, 20, 23, 25]. This provides a novel view of cellular states enabling a deeper understanding of complex biological systems and processes that individual cells are involved in which is not possible with bulk RNA [23, 25, 26]. It has been shown that single-cell data has a high correlation with bulk RNA data when sampled from the same cell population [4], however, both methods suffer from technical noise when sequenced [1–8, 12, 13, 20, 23, 25, 27]. Recently researchers have been using single-cell gene expression data as opposed to bulk RNA when modelling cell differentiation, [1] since single-cell gene expression data provide an increased resolution in sampled data, [11] by determining a cell-type utilising the entire transcriptome of thousands of individual cells [28].

State-of-the-art algorithms in the literature require that the sampled single-cell gene expression data be in the format of a cell-by-gene matrix of counts. In essence, to understand this matrix, it is necessary to understand the sequential steps involved in obtaining a single-cell RNA-seq cell-by-gene matrix of counts [29]. These sequential steps are discussed below and can be observed in the flow diagram of Figure 2.4.

- Step 1: Obtain an environment sample (ES.1) to represent a specific time step in the cell differentiation process.
  - Step 1.1: Apply a sampling method (S.1) of choice (SM.1, SM.2 or SM.3) to obtain a single-cell from the environment sample.
  - Step 1.2: Obtain the complementary deoxyribonucleic acid (cDNA) (CD.1) of the sampled

single-cell by implementing reverse transcription. The obtained cDNA is then randomly cut into large, equally-sized fragments with predefined lengths, known as read pairs. These read pairs are then aligned with a reference genome to see how many read pairs are mapped to each gene [14].

- Step 1.3: Count the total number of read pairs to form a count (C.1) of the genes expressed by the sampled single-cell.
- Step 1.4: Save the obtained counts of the sampled single-cell in a cell-by-gene matrix of counts.
- Step 1.5: Go to Step 1.1 until all of the single-cells of the environment sample have been sampled and saved.

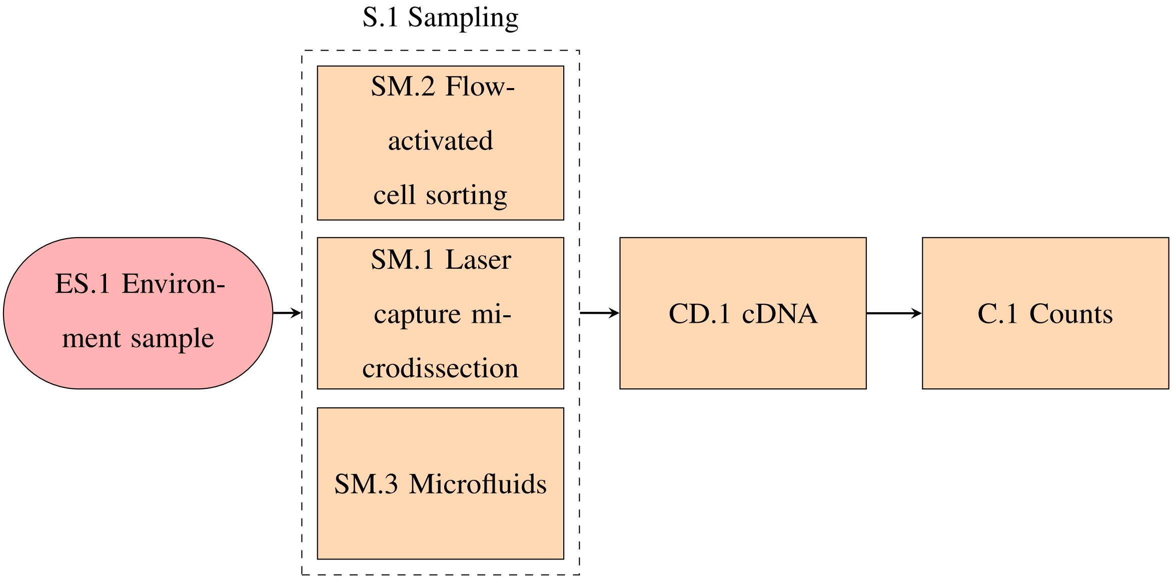


Figure 2.4. Flow diagram of the process whereby a single-cell RNA-seq cell-by-gene matrix of counts is obtained. First, an environment sample (ES.1) is taken at a given cell differentiation time step. Next, a single-cell is sampled (S.1) from the environment sample using one of the various sampling methods (SM.1, SM.2 or SM.3). After the single-cell has been sampled its cDNA (CD.1) is used to obtain a count (C.1) of the genes expressed by the single-cell. The obtained count of the sampled single-cell is then saved in a cell-by-gene matrix of counts. This process of sampling and saving individual single-cells is repeated until all of the single-cells of the environment sample have been sampled and saved.

The main concern when obtaining a cell-by-gene matrix of counts via the sc-RNA-seq method is Step 1.3. This concern is because this step involves counting the gene expressions which are used to produce the rows and columns of the cell-by-gene matrix of counts. Step 1.3 can be thought of as having a bag of marbles (single-cell sample) containing different coloured marbles (genes). A marble is then removed from the bag and will contribute one count towards a specific marble colour (gene). This process is repeated until all of the marbles are counted, which provides an estimate of the total number of same-coloured marbles (counts per gene) of the bag (single-cell sample) [14]. As seen, when there is a miscount of marbles (genes) in the bag (single-cell sample) Step 1.3 can produce an inaccurate cell-by-gene matrix of counts. When the cell-by-gene matrix of counts is completed it can be visualised in the form of a heat map.

An example of a heat map that depicts an inverse cell-by-gene matrix of count (for the ease of visualisation) is represented in Figure 2.5. These maps are most useful to display many data samples, many genes, or even both. Usually each column of the heat map represents an asynchronous sample (single-cell sample) of the data (environment sample) and each row represents the total number of counts for a given gene. The colour scheme of the heat map can be interpreted as (i) red indicating many counts, which implies high expression levels; (ii) green indicating a low level of count, which implies low expression levels; and (iii) black, which indicate the transition from green to red and vice versa. Another key aspect of a heat map is that similarities between gene expression can become evident, when the genes of the heat map are hierarchically clustered together in a tree-like structure called a dendrogram [30] (left of Figure 2.5).

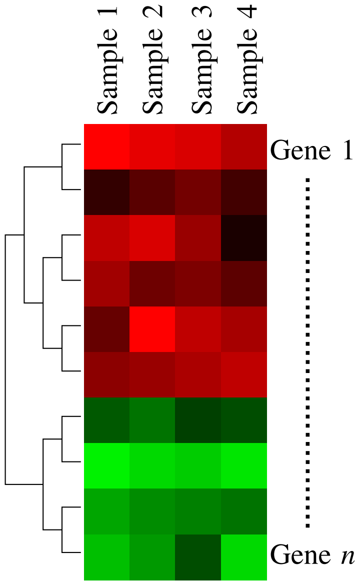


Figure 2.5. Heat map and dendrogram of an inverse cell-by-gene matrix of counts. Four different asynchronous samples (single-cells) are indicated in each column of the heat map, whereas each row represents one of a total of *n* genes. The colour scheme of the heat map can be interpreted as (i) red indicating many counts, which implies high expression levels; (ii) green indicating a low level of count, which implies low expression levels; and (iii) black, which indicate the transition from green to red and vice versa. As seen, a dendrogram is conducted on the left-hand side of the figure with the purpose of clustering genes (Adapted from [31], with permission).

2.3 CELL DIFFERENTIATION MODELLING

Rapid advances in high throughput genome-scale sequencing enable researchers to exploit biological processes such as cell differentiation in new ways [1,4,11,12]. A common objective of the field of developmental biology is to understand protein synthesis [13], which is regulated by gene expression [14] that governs the cell differentiation process. Since the molecular signatures that govern cell differentiation are unique to specific cell types, researchers are turning to single-cell gene expression data to shed light on complex biological processes. To better understand the process of cell differentiation, many mathematical models have been developed to predict gene expression lineages during differentiation based on these unique gene expression patterns [1–9,15]. The mathematical models assist in better understanding the cell differentiation process by providing new and novel insights into the differentiation process [1,3].

Early mathematical models attempted to cluster gene expressions together to determine co-regulated genes. A common approach in clustering gene expressions is with the use of a dendrogram [32–35]. A much harder problem is to determine the underlying structure of the transcriptional regulation process during cell differentiation [13]. Although these algorithms provide valuable insight into gene regulation, they do not provide any information on when and how cell fate choices are made. As stated, the key to understanding a cell's fate choices are gene regulatory networks (GRNs) [36]. These networks describe how gene expressions and protein synthesis influence a cell's differentiation path. Modern algorithms estimates well-defined bifurcation points to indicate where and how a cell's fate choices are made, by modelling cell differentiation as a discrete process with limited cell lineages [3, 6]. However, in the state-of-the-art literature, a cell's fate choices were modelled as a continuous process that included multiple bifurcation points [1]. These cell differentiation modelling techniques are referred to as trajectory inference models.

2.3.1 Trajectory inference

Trajectory inference is the process by which sampled and profiled single-cell gene expression data is automatically reconstructed as a cellular dynamic process. The datasets used in these methods can either be (i) a single snapshot that consists of a mixture of cells which are in different stages (e.g. immune cells in the bone marrow); or (ii) a set of cell samples that have been collected at different points in time. The aim of these methods is to order the sampled and profiled high-dimensional single-cell gene expression data to explain cell heterogeneity in the data sample with respect to the defined underlying dynamic process. This dynamic process is accomplished by structuring and ordering the individual cells from a given dataset along a cell differentiation trajectory according to their progression along their developmental path. This defined trajectory is then used to identify different stages as well as interrelationships of the dynamic process.

An important aspect of trajectory inference methods is their ability to assign a so-called pseudo-time to every cell. Pseudo-time is a numeric value with arbitrary units, which is a measure that defines how far a particular cell is along the cell developmental trajectory of the obtained dynamic process. When ordering cells according to their pseudo-time, all the different transition stages that a cell progresses through during the defined dynamic process can be identified. The dynamic process can either be linear or non-linear. Non-linear processes are more common as they represent a branching process, which is often encountered in cell differentiation.

Subsequently trajectory inference models can be categorised into two main parts: (i) dimensionality reduction; and (ii) trajectory inference. In Figure 2.6 the building blocks that are used by some of the possible trajectory inference models to model cell differentiation are shown. In this figure, the trajectory inference is divided into two parts: (i) dimensionality reduction; and (ii) trajectory inference as expected. The mathematical technique path used by each of the possible models (named on the right-hand side of the figure) is colour-coded with a continuous line [37] .

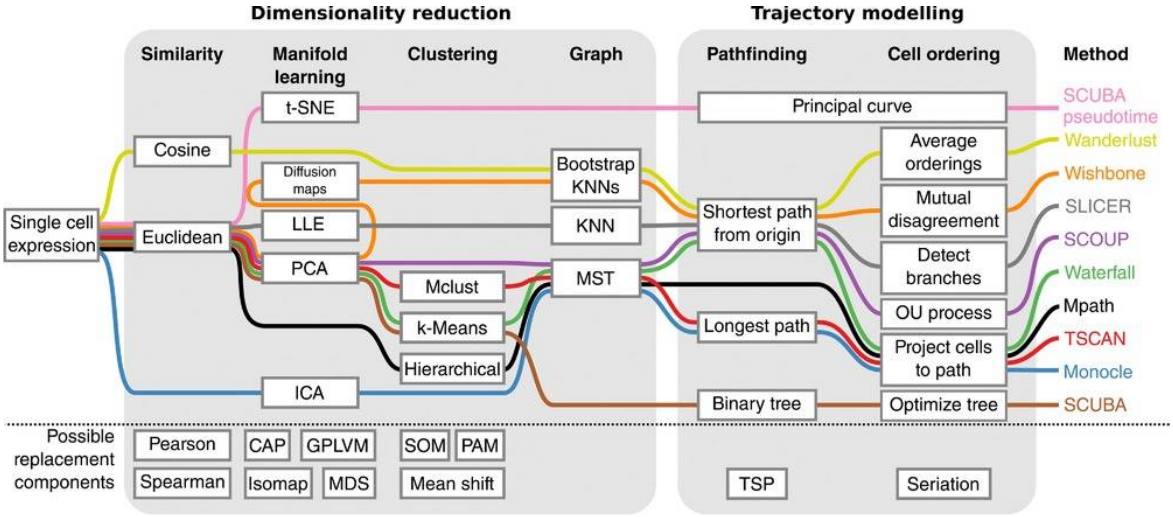


Figure 2.6. Steps for trajectory inference modelling. The trajectory inference models are categorised into two main parts: (i) dimensionality reduction (left); and (ii) trajectory inference (right). The mathematical technique path used by each of the possible models (named on the right-hand side of the figure) is colour-coded with a continuous line (Taken from [37], with permission).

2.3.1.1 Dimensionality reduction

There is a similar strategic approach, which is to pre-process sampled cell differentiation data before utilising mathematical models to estimate gene expression lineages. The sampled cell differentiation data is stored in a high-dimensional cell-by-gene matrix of counts. This means that the sampled data suffers from a phenomenon known as the “curse of dimensionality”. In order to find meaningful structures and geometries within high-dimensional data, dimensionality reduction [38] (used in state-of-the-art literature [1]) or feature selection [28] can be applied. In the case of dimensionality reduction of single-cell gene expression data, an underlying low-dimensional phenotypic manifold of the cell differentiation process is defined. A well-known method for representing single-cell gene expression data is to first represent the cells with a nearest-neighbour graph, where cells with similar states are in close proximity to each other. The axes of this high dimensional representation of the dataset are the known gene expression lineages markers. The high dimensionality is then reduced by applying diffusion maps, which effectively capture the major axes of variation in the data. This newly developed phenotypic manifold represents the gene expression of cell differentiation within two dimensions, where each cell datapoint represents the gene expression of a single-cell [1].

A review paper on gene expression modelling developed the flow diagram shown in Figure 2.7 for known phenotypic manifold development methods [28]. In the flow diagram, colours are used to indicate which part of the gene expression matrix is adjusted after each step. First, the sampled gene expression matrix undergoes feature selection, which selects specific genes to reduce the high dimensionality of the data. This is usually accomplished by eliminating genes with low expression levels. Next, the dimensionality of the expression matrix is reduced with feature and selection dimensionality reduction techniques. Lastly, the dimensionally-reduced data are clustered together. The developed phenotypic manifold is used as the input for a given mathematical model to estimate gene expression lineages.

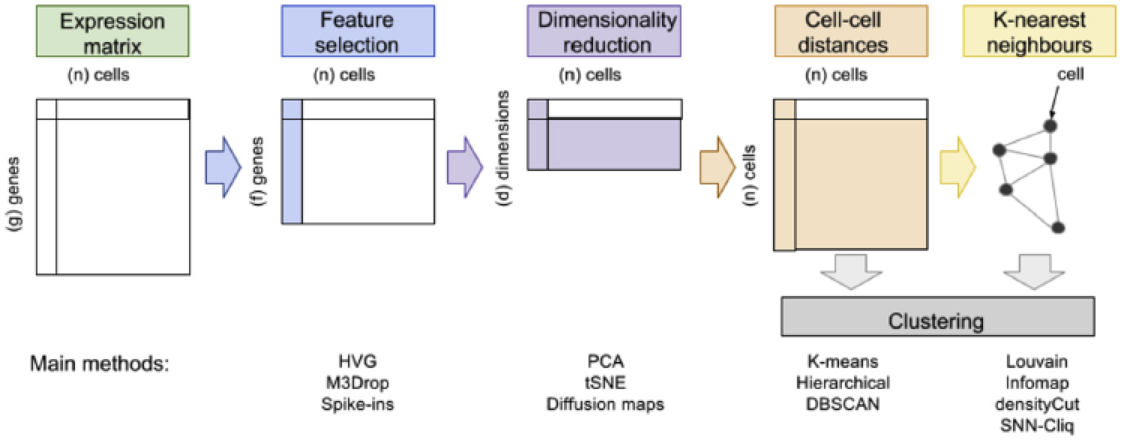


Figure 2.7. Steps for developing a phenotypic manifold displayed with a flow diagram. In the flow diagram, colours are used to indicate which part of the gene expression matrix is adjusted after each step. First, the sampled gene expression matrix undergoes feature selection which selects specific genes to reduce the high dimensionality of the data. This is usually accomplished by eliminating genes with low expression levels. Next, the dimensionality of the expression matrix is reduced and finally the data are clustered together (Taken from [28], with permission).

2.3.1.2 Trajectory modelling

A particular field of interest with regard to trajectory modelling is Gaussian processes. This interest in Gaussian processes is due to their ability to produce reliable regression models from noisy observed targets/ measurements by explicitly representing the uncertainty presented in the data [17, 39, 40].

These reliable results are because the Gaussian process provide a prior distribution over functions that has one or multiple input variables [41, 42]. Hence, Gaussian processes have been used for (i) dimensionality reduction to produce a pseudo-temporal ordering of cells [40, 43–45]; (ii) phenotypic manifold alignment, which is the process whereby multiple types of data describing different features of a phenomenon are integrated together [46–48]; and (iii) to model the cell developmental trajectory of bifurcation points within a cell's differentiation path [16, 49].

A model of interest in this dissertation is the GPfates [16] model, which utilises maximum likelihood, a frequentist estimation approach, to detect a bifurcation point within a cell's developmental trajectory. GPfates detects a bifurcation point by utilising a Gaussian process latent variable model (GPLVM) [50] and overlapping mixtures of Gaussian processes (OMGP) [51] to (i) infer a phenotypic manifold of a sc-RNA-seq dataset; (ii) infer pseudo-time; and (iii) to model the temporal dynamics of gene expression profiles with a mixture model. Essentially, GPfates detects cell lineages with distinct cell fates by associating each component of the OMGP with a different cell lineage. After this association, GPfates utilises each gene, to test whether the likelihood of the data is significantly increased by a model with a bifurcation point as opposed to a model without a bifurcation point. This likelihood association provides a measure to identify diverging global trends within the gene expressions dataset.

2.4 MOTIVATION FOR FOCUSING ON MODELLING HAEMATOPOIESIS

According to the classical model of haematopoiesis, haematopoietic stem and progenitor cells (HSPCs) can be divided into CD34- long-term (LT)-HSPCs and CD34+ short-term (ST)-HSPCs, with LT-HSPCs being at the apex of the haematopoietic hierarchy. Long-term HSPCs have self-renewal capabilities and give rise to ST-HSPCs with limited self-renewal capabilities. ST-HSPCs differentiate to form multipotent progenitors (MPP), which are precursors of common lymphoid and common myeloid progenitors (CLP/CMP). MPPs have no self-renewal abilities but are capable of differentiation into all lineages. CLPs differentiate into lymphoid and natural killer (NK) cells, while CMPs differentiate into granulocyte–macrophage progenitors (GMP) and megakaryocyte–erythrocyte progenitors (MEP), which will further differentiate into granulocytes and macrophages, and erythrocytes and megakaryocytes, respectively [52].

Even though haematopoiesis has been studied for many years, the HSPC hierarchy is more complex than previously assumed. Recent advances in single-cell technologies provide data that cannot be explained by the classical model and have resulted in a revised model of haematopoiesis. Multipotent progenitors (MPP) 1-4 with lineage bias have recently been identified. A megakaryocyte-biased MPP2 population has been shown to differentiate directly into the megakaryocyte-erythroid lineage, bypassing CMP/MEP progenitors [53]. Increasing evidence also suggests that self-renewing HSPCs in the bone marrow have bias in differentiating into the megakaryocyte-erythroid lineage based on the expression of GATA1 [53–56, 56, 57].

2.5 CONCLUDING REMARKS

Cell differentiation is a fundamental process in biology which is complex and not well understood [1–8]. This process plays a key role throughout cellular development, where a single-cell progresses through a series of orchestrated and continuous differentiation stages to form a terminal cell. Differentiation usually starts with an immature cell that differentiates into a diverse set of mature cell types [6, 9] through up- and down-regulation of many genes. Haematopoiesis is one example of cell differentiation where haematopoietic stem/progenitor cells (HSPCs; immature cells) progress through different stages of differentiation to produce and maintain a continuous supply of mature blood cells [10].

According to the classical model of haematopoiesis, HSPCs can be divided into CD34- long-term (LT)-HSPCs and CD34+ short-term (ST)-HSPCs, with LT-HSPCs being at the apex of the haematopoietic hierarchy. LT-HSPCs have self-renewal capabilities and give rise to ST-HSPCs with limited self-renewal capabilities. ST-HSPCs differentiate to form multipotent progenitors (MPP), which are precursors of common lymphoid and common myeloid progenitors (CLP/CMP). MPPs have no self-renewal abilities but are capable of differentiation into all lineages. CLPs differentiate into lymphoid and natural killer (NK) cells, while CMPs differentiate into granulocyte–macrophage progenitors (GMP) and megakaryocyte–erythrocyte progenitors (MEP), which will further differentiate into granulocytes and macrophages, and erythrocytes and megakaryocytes, respectively [52].

Even though haematopoiesis has been studied for many years, the HSPC hierarchy is more complex than the classical model previously put forward. Deeper understanding haematopoiesis resulting in the adaptation of the classical hierarchical model is directly correlated to the first mammalian single-cell sequencing of RNA that was performed in 2009 [24]. Single-cell RNA sequencing (RNA-seq) enables researchers to uncover the uniqueness of each cell to investigate cell differentiation and cell fate [1, 3–5, 7, 8, 12, 20, 23, 25]. This provides a novel view of cellular states enabling in-depth understanding of complex biological systems and processes that individual cells are involved in which is not possible with bulk RNA [23, 25, 26].

Hence, to better understand and interpret the data provided by these advancements, many mathematical models have been developed to predict gene expression lineages during differentiation based on the unique gene expression patterns uncovered during single-cell sampling [1–9, 15]. The use of Bayesian inference to model cell differentiation with Gaussian processes was recently discovered. In [16] maximum likelihood, a frequentist estimation approach, is used to estimate a single bifurcation point within a cell's developmental trajectory. However, Bayesian inference can provide a quantitative measure of multiple bifurcation points (e.g. no bifurcation point, one bifurcation point or multiple bifurcation points) within a cell's developmental trajectory and validate the evidence of these bifurcation points by utilising Bayes factor [17]. This property of Bayesian inference can be extended into regression modelling with particular focus on Gaussian processes. Gaussian processes can be utilised to produce reliable continuous cell differentiation regression models from noisy targets/measurements, by explicitly representing the uncertainty presented in the data. The validity of Gaussian processes ability to provide quantitative insight into the process of cell differentiation can be seen in its vast exploitation in the literature [16, 40, 43–49].

CHAPTER 3 BACKGROUND THEORY

3.1 CHAPTER OVERVIEW

This chapter provides a general overview of the basic statistical background theory applied in this dissertation. The background theory starts by providing basic mathematical concepts and an introductory discussion about discrete and continuous probability distributions. This discussion is followed by a defence of why the methods reported in this dissertation favour Bayesian inference instead of frequentist inference, followed by an in-depth analysis of Bayesian inference. After the discussion on Bayesian inference, some technical background of two techniques used in machine learning are discussed, namely supervised and unsupervised learning.

3.2 FRENET FRAME

Frenet frames are used to describe local kinematic properties (tangent, normal and binormal vectors) of a particle/point/window (see Chapter 4) that follows a differentiable curve in a three-dimensional Euclidean space [58]. An example of a Frenet frame (green dashed vectors) representation of a curve in space (black solid line) is shown in Figure 3.1. In Figure 3.1 the tangent vector is indicated with t, the normal vector with n and the binormal vector with b.

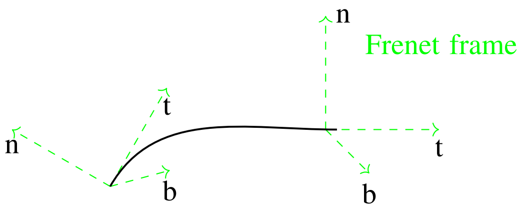


Figure 3.1. A Frenet frame (green dashed vectors) representation of a curve in space (black solid line), where the tangent vector is indicated with t, the normal vector with n and the binormal vector with b (Adapted from [59], with permission).

3.3 EUCLIDEAN DISTANCE

Euclidean distance is a measuring tool that defines the length of a straight line path between two n-dimensional points $\mathbf{p}_1$ and $\mathbf{p}_2$ [60] as

$$||\mathbf{p}_1 - \mathbf{p}_2|| = \sqrt{\sum_{j=1}^{n} (p_{1,j} - p_{2,j})^2}, \tag{3.1}$$

where $p_{1,j}$ and $p_{2,j}$ are the j^{th} components of $\mathbf{p}_1$ and $\mathbf{p}_2$ respectively.. A simple graphical illustration of the euclidean distance between two arbitrary points $\mathbf{p}_1$ and $\mathbf{p}_2$ when $n = 2$ is shown in Figure 3.2.

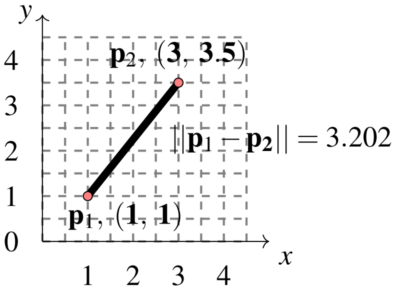


Figure 3.2. Graphical illustration of the Euclidean distance between arbitrary points $\mathbf{p}_1$ and $\mathbf{p}_2$ when $n = 2$.

3.4 PROBABILITY DISTRIBUTIONS

A probability distribution is a statistical function $p(x)$ that describes the probability (e.g. likelihood) of a set of random variables $x_1, \ldots, x_N$. These random variables can either be continuous or discrete and in the case of this dissertation, independent and identically distributed. These distributions are known as parametric distributions, as they are governed by a few a adaptive parameters [17, 61].

3.4.1 Discrete distributions

3.4.1.1 Multinomial distribution

A multinomial distribution is obtained by generalising the Bernoulli distribution to a multivariate space. Consider a K-dimensional binary variable x consisting of x_k components with $x_k \in \{0, 1\}$ such that $\sum_k x_k = 1$. The discrete multinomial distribution is defined by

$$p(x) = \prod_{k=1}^{K} \mu_k^{x_k}, \tag{3.2}$$

where μ is a parameter representing probability and is constrained such that $\mu_k \geq 0$ and $\sum_k \mu_k = 1$ [17].

3.4.2 Continuous distributions

3.4.2.1 Gaussian distribution

One of the most important distributions of continuous random variables is the Gaussian or Normal distribution [17]. The general form of a single real-valued variable x known as a univariate Gaussian distribution is defined as

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right), \tag{3.3}$$

and is governed by a mean parameter μ as well as a variance parameter σ^2. The variance provides a measure of the variability around the mean of the distribution and can give rise to two new variables. The first variable is the standard deviation σ which provides a measure of how spread out the data is, obtained by taking the square root of the variance. The second variable is precision, which is usually used instead of the variance as it is computational, more convenient and is defined as the reciprocal of the variance by $\beta = \frac{1}{\sigma^2}$. An illustration of the univariate Gaussian distribution with parameters $\mu = 0$ and $\sigma^2 = 1,\ 2$ and 3 is shown in Figure 3.3.

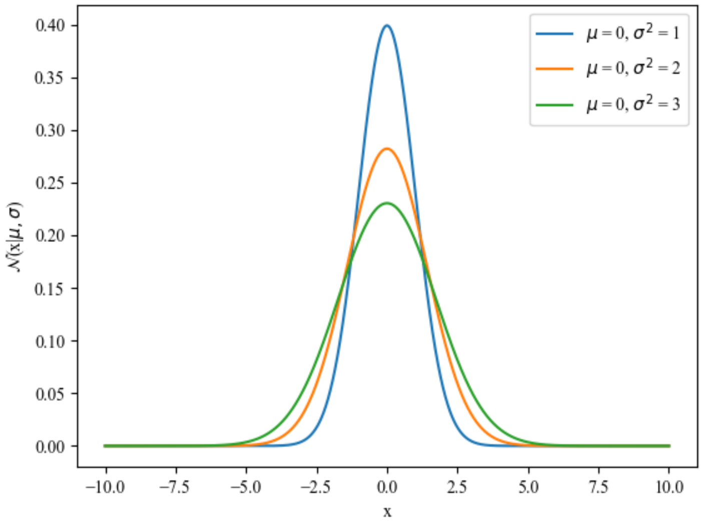
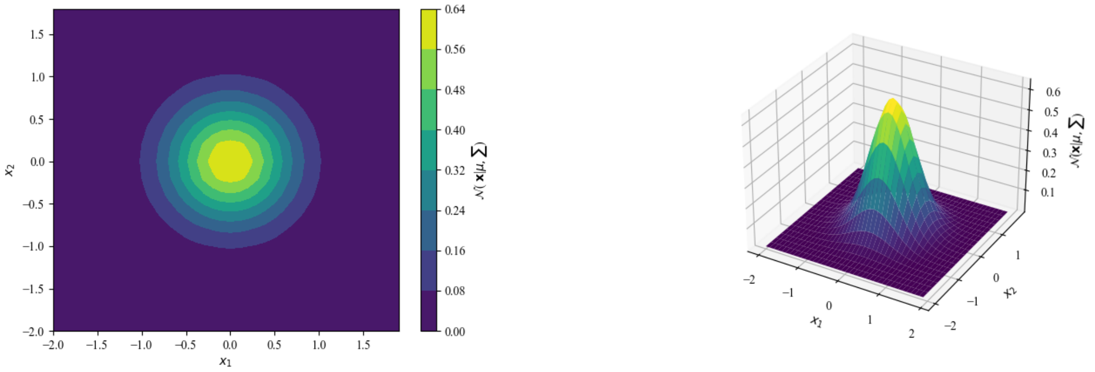


Figure 3.3. Univariate Gaussian distribution with parameters $\mu = 0$ and $\sigma^2 = 1,\ 2$ and 3.

The univariate Gaussian distribution can be extended to an D- dimensional input vector $\mathbf{x}$ known as a multivariate Gaussian distribution. The multivariate Gaussian distribution is defined as

$$\mathcal{N}(\mathbf{x}|\mu,\Sigma) = \frac{1}{(2\pi)^{\frac{D}{2}}} \frac{1}{\sqrt{|\Sigma|}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1}(\mathbf{x}-\mu)\right), \tag{3.4}$$

where Σ is a $D \times D$ covariance matrix and $|\Sigma|$ is the determinant of the covariance matrix. An illustration of the multivariate Gaussian distribution with parameters $\mu = 0$ and $\Sigma = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$ is shown in Figure 3.3.



(a) Two-dimensional representation of a multivariate Gaussian distribution.

(b) Three-dimensional representation of a multivariate Gaussian distribution.

Figure 3.4. Multivariate Gaussian distribution with two random variables x_1 and x_2 and parameters $\mu = 0$ and $\Sigma = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$.

3.4.2.2 Dirichlet distribution

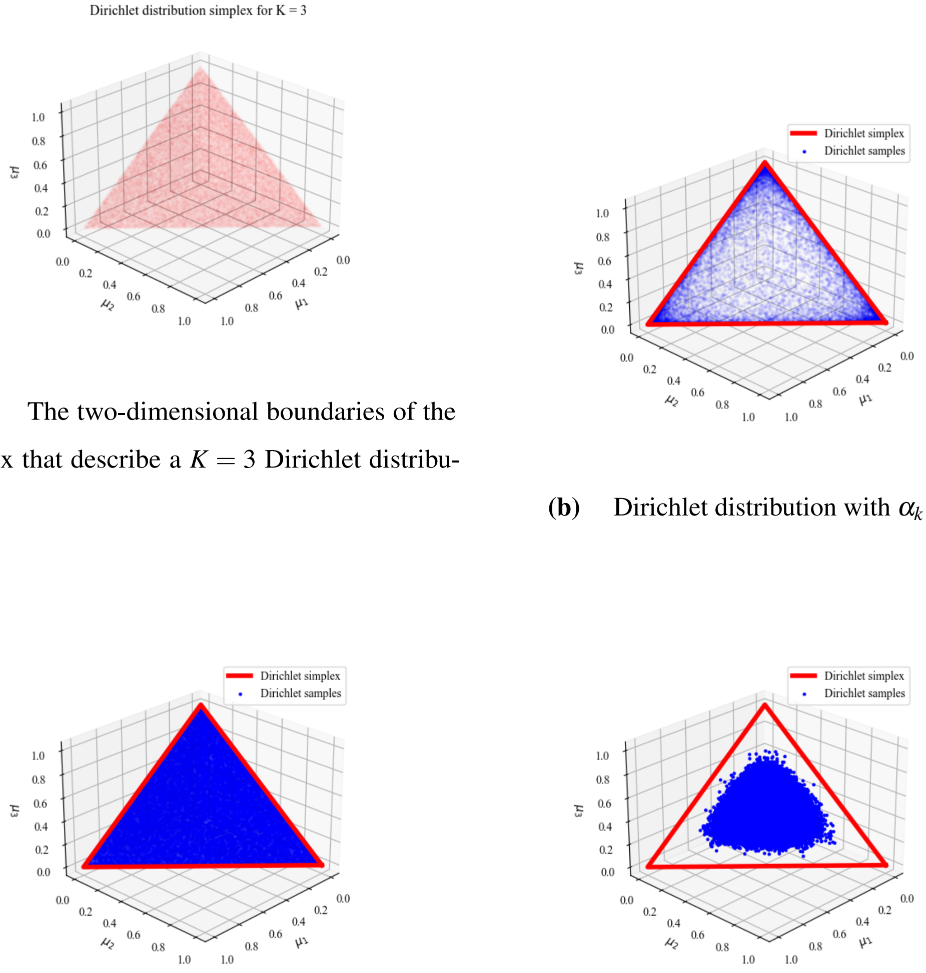
A Dirichlet distribution is obtained by generalising the Beta distribution to a multivariate space [17]. The normalised form of the Dirichlet distribution is defined with

$$Dir(\mu|\alpha) = \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_1)\ldots\Gamma(\alpha_k)} \prod_{k=1}^{K} \mu_k^{\alpha_k - 1}, \tag{3.5}$$

where α is the concentration hyperparameter which is a positive real number parameter vector defined as $\alpha = (\alpha_1, \ldots, \alpha_k)$, Γ is a gamma function with

$$\alpha_0 = \sum_{k=1}^{K} \alpha_k, \tag{3.6}$$

and μ is a parameter representing probability that is constrained such that $\mu_k \geq 0$ and $\sum_k \mu_k = 1$. Due to the summation constraint of μ_k the Dirichlet distribution of K random variables is confined to a $K-1$ dimensional probability simplex that exists on the μ_k space. An illustration of the confinement of a $K = 3$ simplex (in red) as well as the probability distribution (in blue) of a $K = 3$ simplex where $\alpha_k = 0.1,\ 1$ and 10 is shown in Figure 3.5.



(a) The two-dimensional boundaries of the simplex that describe a $K = 3$ Dirichlet distribution.

(b) Dirichlet distribution with $\alpha_k = 0.1$

(c) Dirichlet distribution with $\alpha_k = 1$

(d) Dirichlet distribution with $\alpha_k = 10$

Figure 3.5. Dirichlet distribution of $K = 3$, which is confined to a two-dimensional simplex indicated with red. As seen the densities of the Dirichlet distributions in blue differ between graphs when the concentration parameter α_k is changed.

3.4.2.3 normal-inverse-Wishart distribution

The normal-inverse-Wishart ($\mathcal{NIW}$) is a multivariate distribution that has a crucial role in Bayesian statistics, as it ensures computational tractability when sampling a multivariate Gaussian distribution when its mean (μ) and covariance (Σ) are unknown. This tractability is accomplished due to the $\mathcal{NIW}$ distribution being a conjugate prior (see Section 3.6.2) of a multivariate Gaussian distribution.

A $\mathcal{NIW}$ distribution is a four-parameter multivariate continuous distribution obtained by generalising the normal-inverse-χ^2 distribution to a d-dimensional multivariate space defined as

$$P(\mu,\Sigma) = \mathcal{NIW}(\mu_0,\kappa_0,\Lambda_0,\nu_0), \tag{3.7}$$

$$= \frac{1}{Z}|\Sigma|^{-(((\nu_0+d)/2)+1)} exp\left(\frac{-1}{2}\text{Tr}(\Lambda_0\Sigma^{-1} - \frac{\kappa_0}{2}(\mu-\mu_0)^T\Sigma^{-1}(\mu-\mu_0)\right), \tag{3.8}$$

where the normalising constant Z is equal to

$$Z = \frac{2^{\nu_0 d/2}\Gamma_d(\nu_0/2)(2\pi/\kappa_0)^{d/2}}{|\Lambda_0|^{\nu_0/2}}, \tag{3.9}$$

and the hyperparameters defined as, μ_0 a $d\times 1$ vector that defines the prior mean of the Gaussian distribution, κ_0 a positive scalar that defines the degree of belief for the prior mean, Λ_0 a positive definite matrix of dimension $d\times d$ and ν is a positive scalar that defines the degree of freedom where $\nu > d-1$. The functions used by $\mathcal{NIW}$ are the trace function defined as $Tr(\cdot)$ and a multivariate gamma function defined as $\Gamma_d(\cdot)$ [62–64]. A visual illustration of an univariate $\mathcal{NIW}$ distribution with $\mu_0 = 0$, $\kappa_0 = 1$, $\Lambda_0 = 1$ and $\nu_0 = 1$ is shown in Figure 3.6.

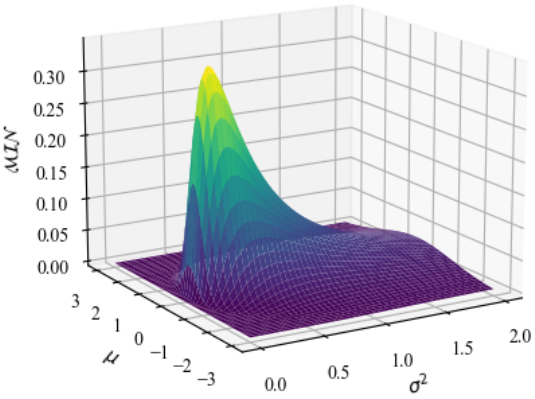


Figure 3.6. Univariate $\mathcal{NIW}$ distribution with $\mu_0 = 0$, $\kappa_0 = 1$, $\Lambda_0 = 1$ and $\nu_0 = 1$. As seen, there is a Gaussian distribution over the mean parameter (μ) and an inverse-χ^2 over the variance parameter (σ^2).

3.5 BAYESIAN VS FREQUENTIST STATISTICS

In statistics, there are mainly two philosophical interpretations of probability, namely Bayesian and frequentist, also referred to as classical statistics. Both of these philosophies aim to provide meaningful insight about a dataset $\mathcal{D}$ and a hypothesis $\mathcal{H}$, but differ in the manner that insight is provided. There has been much controversy and debate in the statistical world regarding these two approaches to define which philosophy is best. In order to defend the statistical philosophy used in this dissertation, a short comparison is provided between the different approaches [17, 65–67].

3.5.1 Bayesian statistics

Given the dataset $\mathcal{D}$ Bayesian statistics provides a quantitative measure of the uncertainty of the hypothesis defined as $p(\mathcal{H}|\mathcal{D})$. This means that in Bayesian statistics the hypothesis (model parameter) is defined as a random variable quantified by a probability distribution that describes the uncertainty of the hypothesis, and the observed data is treated as a fixed quantity. An important factor of Bayesian statistics is that it depends on subjective prior knowledge of the hypothesis. Hence, the definition of probabilities of Bayesian statistics are fundamentally related to beliefs about the hypothesis.

3.5.2 Frequentists statistics

Given the hypothesis $\mathcal{H}$ frequentist statistics provide a quantitative measure of the uncertainty of the data after a long run of independent trials defined as $p(\mathcal{D}|\mathcal{H})$. This means that in frequentist statistics, the observed data is defined as a random variable quantified by a probability distribution that describes the uncertainty of the data, and the hypothesis (model parameter) is treated as a fixed quantity. The frequentist approach, therefore, defines probability as the limit of the relative frequency of an event after a long run (possible infinite) of repeatable experiments (trials). Hence, the definition of probabilities of frequentist statistics is fundamentally related to the frequency of the observed data.

3.5.3 Bayesian defence

In the field of machine learning, Bayesian statistics are preferred to frequentist statistics when inferring model parameters (hypothesis). This is because in a frequentist domain, inferred model parameters are defined by their point estimates. The point estimates are obtained by maximising a likelihood function $\mathcal{L}(\mathcal{H}|\mathcal{D}) = p((\mathcal{D}|\mathcal{H})$ which is a function of the hypothesis. However, in a Bayesian domain, inference of model parameters is obtained by a distribution that defines the uncertainty about the parameters. This means that in the Bayesian domain, there is a distribution over the model parameters (hypothesis).

Another reason why Bayesian inference is preferred is due to its model selection capabilities. Frequentist inference compares models by their likelihood functions, and hence suffers from a phenomenon known as over-fitting. Over-fitting occurs when the model used to describe the observed dataset has a complexity level above what is required, leading to models that fit the data exactly. As observed, data usually suffers from measurement noise the increased level of complexity is unwanted because it will incorporate measurement noise in the model. Hence, over-fitting occurs when the obtained model fits the observed noisy dataset so well that when new model inputs are provided, it negatively influences their output. To mitigate over-fitting, frequentists introduced various “information criteria” that have been designed to penalise more complex models, which often lead to choosing overly simplified models. In contrast, Bayesian inference provides a quantitative measure of the goodness of fit of the observed dataset for a given model. The quantitative measure of multiple models can then be used to make an informed decision when selecting the optimal model for the data. Therefore, due to the vastly superior Bayesian statistics in data modelling, the focus of this dissertation will only be on Bayesian inference.

3.6 BAYESIAN INFERENCE

3.6.1 Bayesian probability

The philosophy of Bayesian probability utilises Bayes’s theorem when defining probability. Bayes’s theorem is a statistical method that defines conditional probabilities. Hence, Bayesian probability is defined as a quantitative measure of uncertainty. As this dissertation focuses on data modelling, Bayesian probability is defined in terms of data modelling as

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{\theta}) \cdot p(\boldsymbol{\theta})}{p(\mathcal{D})}, \tag{3.10}$$

where $p(\boldsymbol{\theta}|\mathcal{D})$ is the conditional probability of observing the model parameter vector $\boldsymbol{\theta}$ given the observed dataset $\mathcal{D}$ known as the posterior distribution, $p(\boldsymbol{\theta})$ is the marginal distribution of the model parameter vector $\boldsymbol{\theta}$ that describes assumptions about the parameter vector known as the prior, $p(\mathcal{D}|\boldsymbol{\theta})$ is the function of the parameter vector $\boldsymbol{\theta}$ describing the goodness of fit of the parameter vector to the data known as the likelihood function and $p(\mathcal{D})$ s a normalising constant, also known as the model evidence, that ensures that the posterior probability is valid and integrates to one. Hence, Bayesian probability in words is defined as

$$posterior \propto likelihood \times prior. \tag{3.11}$$

Similar to the definition in words, Bayesian probability can be visualised as shown in Figure 3.7 where the posterior distribution, indicated with a black line, is equal to the product of the likelihood, indicated with a red dotted line, times the prior, indicated with the green dash and dot line.

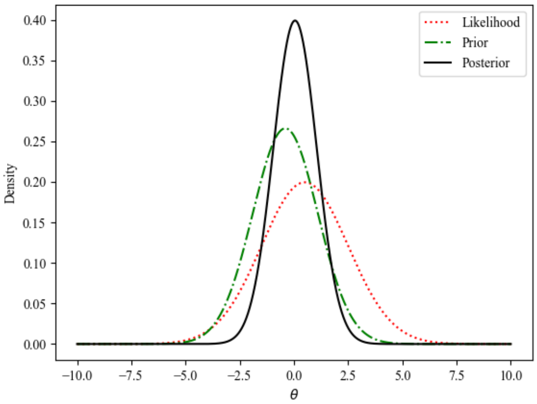


Figure 3.7. Visual illustration of Bayesian probability where the posterior distribution, indicated with a black line, is equal to the product of the likelihood, indicated with a red dotted line, and the prior, indicated with the green dash and dot line. The x-axis represents the model parameter θ, whereas the y-axis represents the distribution density.

3.6.2 Conjugate priors

In the field of Bayesian inference choosing the correct prior distribution is crucial as the incorrect prior can (i) produce poor results with a high confidence [17]; (ii) as well as make the posterior distribution computationally intractable [68, 69]. To mitigate computational intractability, pioneers define the concept of conjugate priors. Conjugate priors are prior distributions that are conjugate to the likelihood function ensuring that the posterior distribution has the same distributional form as the prior. Hence, a prior is said to be conjugate for the likelihood function $L(\theta|\mathcal{D})$ when the posterior distribution $p(\theta|\mathcal{D})$ belongs to the same family of distributions that defines the probability distributions over the parameter vector θ [17, 69]. A list of likelihood functions with their corresponding conjugate prior distributions that is of interest in this dissertation is provided in Table 3.1. As seen in Table 3.1, conjugate priors are governed by hyperparameters that control their distributions (see Section 3.4).

Table 3.1. Likelihood functions with their corresponding conjugate prior distributions.

Likelihood function	Conjugate prior
multinomial	Dirichlet
multivariate Gaussian/normal	normal-inverse-Wishart

3.6.3 Bayes's model selection

The Bayes model selection process starts by believing that the observed dataset D can be described by a set of possible candidate models $\mathcal{M}_1, \ldots, \mathcal{M}_k$ each representing a different probability distribution over the data. After the possible candidate models are defined, Bayesian model selection is achieved by selecting the model that best describes the observed data by utilising Bayes factor [70]. The Bayesian probability of (3.10) can be extended to incorporate different candidate models defined by

$$p(\boldsymbol{\theta}|\mathcal{D},\mathcal{M}_i) = \frac{p(\mathcal{D}|\boldsymbol{\theta},\mathcal{M}_i)p(\boldsymbol{\theta}|\mathcal{M}_i)}{p(\mathcal{D}|\mathcal{M}_i)}, \tag{3.12}$$

where $\boldsymbol{\theta}$ is the model parameters, $\mathcal{D}$ is the observed data and $\mathcal{M}_i$ is the i^{th} candidate model (distribution). The normalising term of the denominator is known as the marginal likelihood, also called the model evidence. The model evidence shows the preference of the i^{th} candidate model for a given dataset $\mathcal{D}$ and is used for the basis of Bayes factor analysis. Hence, Bayes factor is defined by

$$BF = \frac{p(\mathcal{D}|\mathcal{M}_i)}{p(\mathcal{D}|\mathcal{M}_j)}, \tag{3.13}$$

which is the ratio of the evidence for the i^{th} candidate model divided by the evidence for the j^{th} candidate model. If the Bayes factor is greater than one, the dataset prefers the i^{th} candidate model and if the Bayes factor is smaller than one, the dataset prefers j^{th} candidate model [17]. Table 3.2 shows a scale on how to best interpret the ranges of Bayes factor [71].

Table 3.2. Bayes factor interpretation (From [71]).

BF_{ij}	**Evidence for** $\mathcal{M}_i$
< 1	$\mathcal{M}_j$ is selected
1 - 3	Not bad
3 - 12	Substantial
12 - 150	Strong
>150	Decisive

3.6.4 Sampling methods

Calculating the evidence of Bayes's theorem has always been a daunting task, especially for a large model parameter vector $\boldsymbol{\theta}$. However, a revolutionary step in Bayesian statistics that can simplify the evidence calculation is known as Markov chain Monte Carlo (MCMC) methods. The idea of MCMC methods is to draw samples from a distribution, in essence to obtain an estimate of its expected value, in cases where the expected value is analytically to difficult to compute. To understand why MCMC methods are able to provide an estimate of the expected value via sampling, an understanding of Markov chains is first required [17, 72, 73].

3.6.4.1 Markov chains

Consider a first-order Markov chain that is defined as a sequence of random variables $\theta^{(t)} \in \theta^{(1)}, \dots, \theta^{(t)}$ that satisfy a conditional independence property. This property states that the distribution of $P(\theta^{(t)})$ should only be dependent on the previous distribution of $P(\theta^{(t-1)})$ defined as

$$p(\theta^{(m+1)}|\theta^{(1)}, \dots, \theta^{(m)}) = p(\theta^{(m+1)}|\theta^{(m)}), \tag{3.14}$$

where $m \in \{1, \dots, t-1\}$. A Markov chain can then be specified by defining an initial probability variable of $p(\theta^{(0)})$, together with transition probabilities that describe the conditional probabilities for subsequent variables as

$$T_m(\theta^{(m)}, \theta^{(m+1)}) \equiv p(\theta^{(m+1)}|\theta^{(m)}). \tag{3.15}$$

A special case of the Markov chains is known as a homogeneous Markov chain, which occurs when all the transition probabilities are the same for all the possible states.

The specific Markov chains of interest concerning MCMC methods are ergodic homogeneous Markov chains. An ergodic Markov chain implies that the chain (i) converges to a unique target distribution irrespective of the initial variable of $p(\theta^{(0)})$; (ii) does not get trapped between state cycles; and (iii) there is a positive probability that each of the states can reach every other state with one or multiple steps. In essence, for a Markov chain to be viable for MCMC it should also converge to a desired distribution. A Markov chain converges when it is defined as a stationary/invariant chain. A Markov chain is said to be stationary/invariant when the vector probability of being in any given state is independent of the initial variable of $p(\theta^{(0)})$.

3.6.4.2 Markov chain Monte-Carlo (MCMC)

MCMC methods construct a stationary ergodic Markov chain such that its equilibrium probability distribution corresponds to a desired distribution. In most cases the desired distribution to be estimated is an analytical complex posterior distribution of (3.10). To estimate the posterior distribution, the constructed Markov chain draws sequential samples (t) of the model parameter vector θ from an approximation of the posterior distribution $p(\theta^{(t)}|\mathcal{D})$ to obtain an estimate of the expected posterior distribution $\mathbb{E}[p(\theta|\mathcal{D})]$. Hence, the Monte-Carlo (MC) expected value of the posterior distribution is defined as

$$\mathbb{E}[p(\boldsymbol{\theta}|\mathcal{D})] = \int_{\theta} p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}, \\ \approx \frac{1}{n_s}\sum_{t=1}^{n_s} p(\boldsymbol{\theta}^{(t)}|\mathcal{D}), \tag{3.16}$$

where n_s is the number of samples. As seen from (3.16), the accuracy of MCMC is directly proportional to the total number of samples. This means that as n_s tends to ∞, the MCMC approximation will become more accurate.

MCMC methods do not begin their sampling at a stationary distribution due to the Markov chain that still needs to converge. Hence, the first couple of samples of the MCMC simulation is disregarded. This period of samples that are disregarded due to the MCMC method converging to a target distribution is known as the burn-in period.

3.6.4.3 Gibbs sampler

Gibbs sampling is a particular method of MCMC that provides a useful approach to iteratively draw samples from full conditional distribution when their joint distribution is complex. The aim of the Gibbs sampler is to use the draws from the full conditional distribution to obtain an expected value of their joint distribution. Gibbs sampling can be applied for any joint distribution that has full conditional probabilities, which in the case of this dissertation is the posterior distribution of (3.12). Consider the following joint distribution:

$$p(\boldsymbol{\theta}|\mathcal{D}) = p(\theta_1, \ldots, \theta_z|\mathcal{D}), \tag{3.17}$$

where $\boldsymbol{\theta}$ is the model parameter vector of length z with unknown quantities and $\mathcal{D}$ is the observed data. During every Gibbs sampler iteration, a cycle is completed in which each of the parameters in the parameter vector is sampled condition on all of the other parameters and observed data. The full conditional distribution of the posterior distribution in (3.17) is defined as

$$p(\theta_i|\mathcal{D}, \theta_j^{(t-1)}, j \neq i), \quad i = 1, \ldots, z, \tag{3.18}$$

where t refers to an iteration of the Gibbs sampler, and θ_j^{t-1} refers to all the components of the parameter vector as

$$\theta_j^{(t-1)} = (\theta_1^{(t+1)}, \ldots, \theta_{i-1}^{(t+1)}, \theta_{i+1}^{(t)}, \ldots, \theta_z^{(t)}). \tag{3.19}$$

This means that each sample of the Gibbs sampler has full knowledge of all the previous samples in a cycle as well as samples from the previous cycle. The pseudo code of a Gibbs sampler sampling from the full conditionals of the joint posterior in (3.17) is summarised in Algorithm 1.

Algorithm 1 Gibbs sampler

```
Initialise starting values: $\theta_1^{(0)}, \ldots, \theta_z^{(0)}$
Define the total number of Gibbs iterations: $T$
for $t = 1, \ldots, T$ : do
    Sample $\theta_1^{(t+1)} \sim p(\theta_1 | \theta_2^{(t)}, \theta_3^{(t)}, \ldots, \theta_z^{(t)})$
    Sample $\theta_2^{(t+1)} \sim p(\theta_2 | \theta_1^{(t+1)}, \theta_3^{(t)}, \ldots, \theta_z^{(t)})$
    Sample $\theta_3^{(t+1)} \sim p(\theta_3 | \theta_1^{(t+1)}, \theta_2^{(t+1)}, \ldots, \theta_z^{(t)})$
    $\vdots$
    Sample $\theta_j^{(t+1)} \sim p(\theta_j | \theta_1^{(t+1)}, \ldots, \theta_{i-1}^{(t+1)}, \theta_{i+1}^{(t)}, \ldots, \theta_z^{(t)})$
    $\vdots$
    Sample $\theta_z^{(t+1)} \sim p(\theta_z | \theta_1^{(t+1)}, \theta_2^{(t+1)}, \ldots, \theta_{z-1}^{(t+1)})$
end for
```

3.6.5 Estimating marginal likelihood/evidence via a Gibbs sampler

As the model evidence of (3.12) for a single candidate model is defined as

$$p(\mathcal{D}|\mathcal{M}) = \int \frac{p(\mathcal{D}|\theta, \mathcal{M}) p(\theta|\mathcal{M})}{p(\theta, \mathcal{D}|\mathcal{M})} d\theta. \tag{3.20}$$

it can be clearly seen that the complexity of the marginalisation integral of the evidence is directly proportional to the number of model parameters. This relationship between model parameters and the complexity of the marginalisation integral is why the evidence $P(\mathcal{D})$ is so daunting to calculate. However, a "gold standard" to estimate the marginal likelihood is known as the basic marginal likelihood identity (BMI) proposed in [74]. The BMI is a rearranged Bayes theorem that holds for any θ defined by

$$p(\mathcal{D}) = \frac{p(\mathcal{D}|\theta) p(\theta)}{p(\theta|\mathcal{D})}. \tag{3.21}$$

A more computational convenient manner in which to represent (3.21) is to write it in a logarithmic scale defined by

$$log(p(\mathcal{D})) = log(p(\mathcal{D}|\theta^*)) + log(p(\theta^*)) - log(p(\theta^*|\mathcal{D})), \tag{3.22}$$

where the posterior density $p(\theta^*|\mathcal{D})$ is an estimate of the density at a selected point θ^*. It is preferred that the selected point θ^* be at a high density point, although the equality of BMI holds for any selected θ^* [74]. As seen in (3.22), all that is required to estimate the marginal likelihood is the log likelihood, log prior and an estimate of the posterior density. The logs of the likelihood and the prior are usually easily computed in closed form, whereas the posterior density can be estimated by a Monte Carlo average that is based on draws from a Gibbs sampler. BMI starts by utilising the law of total probability that allows the posterior distribution at the selected point $\theta^* = [\theta_1^*, \ldots, \theta_B^*]$ to be written as

$$p(\theta^*|\mathcal{D}) = p(\theta_1^*|\mathcal{D}) \times p(\theta_2^*|\mathcal{D}, \theta_1^*) \times \cdots \times p(\theta_B^*|\mathcal{D}, \theta_1^* \ldots \theta_{B-1}^*). \tag{3.23}$$

The first term is known as the marginal density and is obtained from draws of the initial Gibbs run, whereas the rest of the terms, which are known as the reduced conditional densities, are estimated with

$$p(\theta_r^*|\mathcal{D}, \theta_1^*, \theta_2^*, \ldots, \theta_{r-1}^*) = \int p(\theta_r^*, \theta_{r+1}, \ldots, \theta_B, \mathbf{z}|, \theta_1^*, \theta_2^*, \ldots, \theta_{r-1}^*, \theta_l(l > r), \mathcal{D}) \\ d(\theta_{r+1}, \ldots, \theta_B, \mathbf{z}), \tag{3.24}$$

where $\mathbf{z}$ is a latent variable. BMI then estimates the integral of the reduced conditional densities by continuing to sample from the full conditional densities of $p(\theta_r^*, \theta_{r+1}^*, \ldots, \theta_B^*, \mathbf{z})$ and by setting θ_s to its sample values $\theta_s^*, (s \leq r-1)$ for each of these the full conditional densities. This means that the samples from a reduced complete conditional Gibbs run can be denoted with $p(\theta_r^{(j)}, \theta_{r+1}^{(j)}, \ldots, \theta_B^{(j)}, \mathbf{z}^{(j)})$ leading to the estimate of reduced conditional densities as

$$\widehat{p}(\theta_r^*|\mathcal{D}, \theta_s^*(s < r)) = G^{-1} \sum_{j=1}^{G} p(\theta_r^*|\mathcal{D}, \theta_1^*, \theta_2^* \ldots, \theta_{r-1}^* \theta_l^{(j)}(l > r), \mathbf{z}^{(j)}), \tag{3.25}$$

where G is the total number samples. Finally, the log marginal likelihood can be denoted by

$$log(p(\mathcal{D})) = log(p(\mathcal{D}|\theta^*)) + log(p(\theta^*)) - \sum_{r=1}^{B} log(\widehat{p}(\theta_r^*|\mathcal{D}, \theta_s^*(s < r))). \tag{3.26}$$

3.7 SUPERVISED VS UNSUPERVISED LEARNING

To solve the problem of modelling cell differentiation to estimate gene expression lineages in a novel manner, artificial intelligence is utilised. Artificial intelligence is a crucial contributor to the field of science of learning, e.g. the process of learning from data [75]. The specific branch of artificial intelligence of interest in this dissertation is machine learning which is contained within the framework of statistical learning theory. Statistical learning theory refers to a vast set of tools that is used to model and understand complex datasets to define a predictive function [76]. Most of these tools fall into two categories known as supervised and unsupervised learning [17, 75, 76]. A simple graphical illustration of the tools used in the dissertation under the dominion of statistical learning is shown in Figure 3.8. As seen, artificial intelligence (grey pill) is govern by statistical learning (dashed line). The branch of interest of artificial intelligence is machine learning (grey pill). Machine learning can be categorised as supervised (green pill) or unsupervised learning (blue pill). Under each of these two categories subcategories are defined with a red pill.

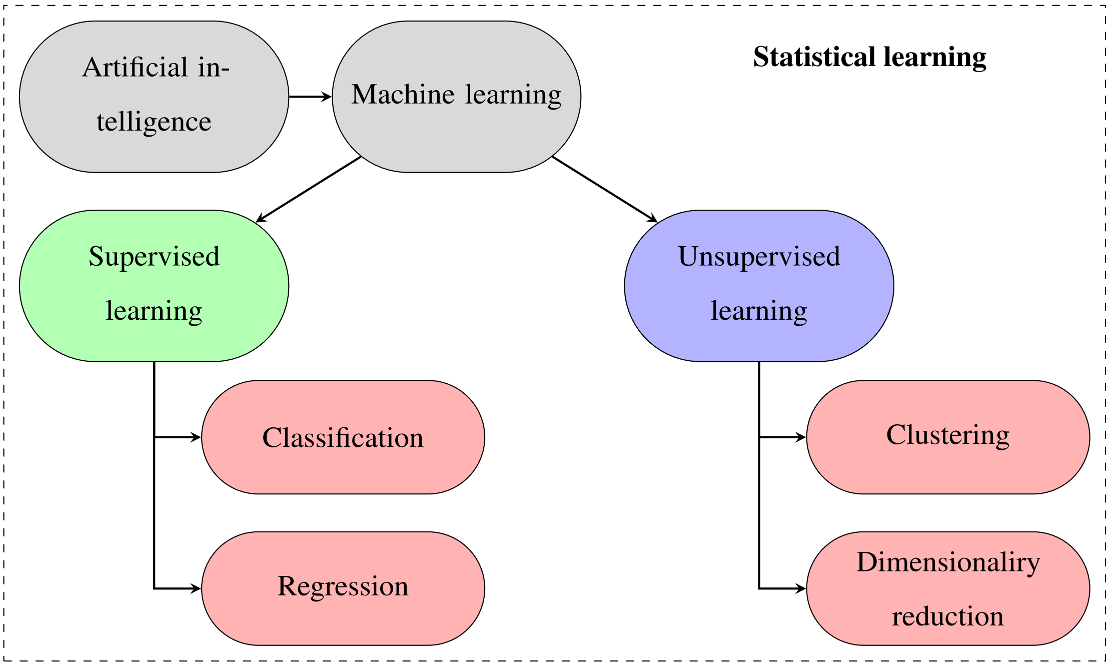


Figure 3.8. A simple graphical illustration of the tools used in the dissertation under the dominion of statistical learning. As seen, artificial intelligence (grey pill) is governed by statistical learning (dashed line). The branch of interest of artificial intelligence is machine learning (grey pill). Machine learning can be categorised as supervised (green pill) or unsupervised learning (blue pill). Under each of these two categories, subcategories are defined with a red pill.

3.7.1 Supervised learning

Supervised learning is used the by the majority of biologists instead of unsupervised learning [77]. All supervised learning models consist of a set of input variables (observations/measurements) that have an influence on a set of one or multiple corresponding output target variables (observations/measurements). The goal of these models is to use an algorithm to learn a mapping function that best describes the relationship between the input and output variables. The idea of this mapping function is that (i) it can be used to accurately predict target variables of future input variable; or (ii) to provide a better understanding of the relationship between input and output variables. Hence, supervised learning can be divided into two categories, namely classification and regression [17, 75, 76]. Classification supervised learning is used when the output variable consists of a finite number of discrete categories and the aim is to assign each of the input variables to it; whereas regression supervised learning is used when the output variable contains one or multiple continuous variables [17]. The supervised learning technique of interest in this dissertation is Gaussian process regression.

3.7.1.1 Gaussian process

A Gaussian process is defined as a stochastic process consisting of a collection of random variables that are usually indexed by time and/or space, such that every finite subset of random variables have a joint Gaussian distribution. In essence, a Gaussian process is used to describe distributions over functions [17, 39]. The goal of a Gaussian process is to predict a continuous target vector $\mathbf{y}$, which consists of one ore more target variables, given a training dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)|i = 1, \ldots, n\}$ of n samples, where $\mathbf{x}$ is a D-dimensional input vector and y is a scalar value of the target variables. The training dataset can is defined as $\mathcal{D} = (X, \mathbf{y})$ when the input vector is represented with a design matrix X of dimension $\mathcal{D} \times n$. The design matrix is obtained by aggregating all the n cases of the input column vector.

3.7.1.1.1 GAUSSIAN PROCESS DEFINITION Consider the following Bayesian linear regression model:

$$f(\mathbf{x}) = \phi(\mathbf{x})^T \theta, \tag{3.27}$$

where $\mathbf{x}$ is an input vector. As seen, this Bayesian model is described by a linear combination of m-dimensional basis functions denoted by $\phi(\mathbf{x}) = (\phi_0(\mathbf{x}), \ldots, \phi_{m-1}(\mathbf{x}))$. A basis function is used to project the input vector $\mathbf{x}$ into a higher N-dimensional space, known as a feature space, before applying linear regression. The reason for this projection is to allow the function $f(\mathbf{x})$ to be non-linear with respect to the input vector. Finally, the Bayesian model is governed by θ, which is a m-dimensional parameter vector of weights. Owing to the model being Bayesian, a prior probability distribution is defined over the parameter vector θ as

$$p(\theta) = \mathcal{N}(\theta|0, \Sigma_p), \tag{3.28}$$

which is a zero mean Gaussian distribution with a covariance matrix $\Sigma_p = \alpha^{-1}\mathbf{I}$ that is governed by a hyperparameter α and identity matrix $\mathbf{I}$. Owing to this prior distribution, a posterior distribution is defined of the parameter vector that is analogous to (3.10). Hence, when observing (3.27) it is clear that a different function can be specified for each possible set of values from the parameter vector weights. As the parameter vector weights are defined with a distribution, it induces a distribution over the possible functions of the linear model, in essence resulting in a distribution over functions.

To obtain a Gaussian process in terms of a distribution over functions from the Bayesian linear regression model is quite simple. This is because the function $f(\mathbf{x})$ is Gaussian distribution due to the linear regression model being defined by a linear combination of the parameter vector θ, which consists of Gaussian distributed variables. Therefore, a Gaussian process can be completely defined by its mean and covariance functions as

$$\mathbb{E}[f(\mathbf{x})] = \phi(\mathbf{x})^T\mathbb{E}(\theta) = 0, \tag{3.29}$$

$$\mathbb{E}[f(\mathbf{x}), f(\mathbf{x}')] = \phi(\mathbf{x})^T\mathbb{E}(\theta\theta^T)\phi(\mathbf{x}') = \phi(\mathbf{x})^T\Sigma_p\phi(\mathbf{x}'), \tag{3.30}$$

where $\mathbf{x}'$ is another input vector and $\mathbb{E}[f(\mathbf{x})]$ is a zero mean that ensures symmetry, to simplify the mathematical analysis of a Gaussian process because usually there is no prior knowledge about the mean and $\mathbb{E}[f(\mathbf{x}), f(\mathbf{x}')]$ is the covariance of function values $f(\mathbf{x})$ and $f(\mathbf{x}')$. From this analysis it is clear that the function values of $f(\mathbf{x})$ and $f(\mathbf{x}')$ have a joint Gaussian distribution. In a more general sense, when there are n input vectors, all of the corresponding function values $f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)$ will have a joint Gaussian distribution. The total number of input variables should be greater then the dimension of the basis function, $n < N$ otherwise the covariance matrix will be of rank N and the joint Gaussian distribution will be singular. The standard form of the Gaussian process can therefore be defined as

$$f(\mathbf{x}) \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}')), \tag{3.31}$$

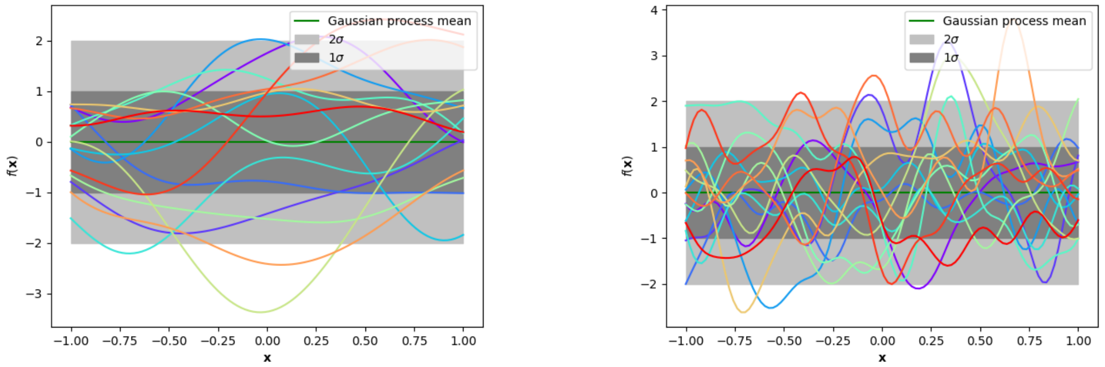
where $k(\mathbf{x}, \mathbf{x'})$ is a kernel function used to populate the covariance matrix. To ensure optimal performance from a Gaussian process, the correct kernel function should be chosen. A popular kernel used in Gaussian process analysis is the exponential of a quadratic kernel. The popularity of this kernel is due to it corresponding to a Bayesian linear regression model consisting of an infinite number of basis functions, which implies a distribution over functions. The exponential of a quadratic kernel with an additional constant and linear term used in this dissertation is denoted as

$$k(x_n, x_m) = \mathbf{w}_0 exp\big(-\frac{\mathbf{w}_1}{2}||x_n - x_m||^2\big) + \mathbf{w}_2 + \mathbf{w}_3 x_n^T x_m, \tag{3.32}$$

where x_n and x_m are two input values, governed by the weight hyperparameter vector $\mathbf{w}$. The weight hyperparameter vector consists of $\mathbf{w}_0$ which influences the variance, $\mathbf{w}_1$ known as the length scale, which influences the smoothness of the function, $\mathbf{w}_2$ which is a constant that offsets the kernel and $\mathbf{w}_3$ which is a term that can be incorporated to address noise variance in the data. A Gaussian process prior is, therefore, defined as a distribution over functions by

$$f_* \sim \mathcal{N}(0, K(X_*, X_*), \tag{3.33}$$

where the dataset X_* corresponds with any chosen number of points and K is a Gram matrix of the dataset. In Figure 3.9, two different arbitrary graphical examples of samples taken form the Gaussian process prior of (3.33) is shown. In this figure the samples are plotted with their corresponding inputs, where X_* is chosen as 1000 samples and the covariance function of (3.32) is populated with different hyperparameter ($\mathbf{w}$) values. The mean of the Gaussian process is indicated with a green line, the first standard deviation 1σ with dark grey, the second standard deviation 2σ with light grey, whereas the samples are indicated with different colours.



(a) Gaussian process samples with the weight hyperparameter vector set equal to $\mathbf{w} = (1,\ 4,\ 0,\ 0)$.

(b) Gaussian process samples with the weight hyperparameter vector set equal to $\mathbf{w} = (1,\ 64,\ 0,\ 0)$.

Figure 3.9. Two different arbitrary graphical examples of samples taken form the Gaussian process prior of (3.33). In this figure, the samples are plotted with their corresponding inputs, where X_* is chosen as 1000 samples and the covariance function of (3.32) is populated with different hyperparameter ($\mathbf{w}$) values. The mean of the Gaussian process is indicated with a green line, the first standard deviation 1σ with dark grey, the second standard deviation 2σ with light grey, whereas the samples are indicated with different colours.

3.7.1.1.2 INFERENCE WITH GAUSSIAN PROCESS Consider the following noisy function value:

$$y = f(\mathbf{x}) + \varepsilon, \tag{3.34}$$

where ε is additive Gaussian noise obtained from an independent, identically distributed Gaussian distribution that has a mean of zero and a variance of σ_n^2 defined as

$$\varepsilon \sim \mathcal{N}(0, \sigma_n^2). \tag{3.35}$$

In practice, we usually do not want to obtain just one function value $f(\mathbf{x})$ but rather the joint distribution of function values $f(\mathbf{x_1}), \ldots, f(\mathbf{x}_n)$. The joint distribution with additive noise is defined as

$$\mathbf{y} = \Phi\theta + \sigma_n^2 \mathbf{I}, \tag{3.36}$$

where Φ is a design matrix of the basis functions. Therefore, the covariance matrix of the joint distribution of function variables with additive noise is defined as

$$\mathrm{cov}(\mathbf{y}) = K(X,X) + \sigma_n^2 \mathbf{I}, \tag{3.37}$$

where K is a Gram matrix. In essence to make predictions a joint distribution needs to be defined that incorporates new training function values f as well as function values from the prior f_*. If there are n training datapoints and n_* prior datapoints the joint distribution of f and f_* is defined as

$$\begin{bmatrix} f \\ f_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X,X)+\sigma_n^2\mathbf{I} & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right), \tag{3.38}$$

where $K(X,X)$ is a $n \times n$ Gram matrix of the training data, $K(X,X_*) = K(X_*,X)$ is a $n \times n_*$ Gram matrix of the training data and prior and $K(X_*,X_*)$ is a $n_* \times n_*$ Gram matrix of the prior. Since the prior and the training data is jointly Gaussian distributed the posterior distribution will also be a Gaussian defined as

$$f_*|X_*,X,f \sim \mathcal{N}(\overline{f}_*, \text{cov}(f_*)), \tag{3.39}$$

where the mean and covariance of the posterior are defined with

$$\overline{f}_* \triangleq \mathbb{E}[f_*|X_*,X,f] = K(X_*,X)[K(X,X)+\sigma_n^2\mathbf{I}]^{-1}\mathbf{y}, \tag{3.40}$$

$$\text{cov}(f_*) = K(X_*,X_*) - K(X_*,X)[K(X,X)+\sigma_n^2\mathbf{I}]^{-1}K(X,X_*). \tag{3.41}$$

A numerical, more stable and computational faster approach to solve the inverse of $[K(X,X)+\sigma_n^2\mathbf{I}]^{-1}$ is to use the Cholesky decomposition. The Cholesky decomposition is used to decompose a positive definite, symmetric matrix A into a product of matrices. The Cholesky decomposition of matrix A is defined as

$$A = LL^T, \tag{3.42}$$

where L is the Cholesky factor defined as a lower triangular matrix, and L^T is the conjugate of the Cholesky factor. A simple example of Cholesky decomposition of a 3×3 matrix is defined as

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \begin{bmatrix} L_{11} & L_{12} & L_{13} \\ 0 & L_{22} & L_{23} \\ 0 & 0 & L_{33} \end{bmatrix}. \tag{3.43}$$

Consider the definition of the inverse of matrix A as [78]

$$AA^{-1} = \mathbf{I}, \tag{3.44}$$

where $\mathbf{I}$ is an identity matrix. The inverse of matrix A can therefore be obtained by decomposing A with a Cholesky decomposition [17,39] and then solving the obtained linear system. The decomposition of A in (3.44) gives rise to the following equation:

$$LL^TA^{-1} = \mathbf{I}. \tag{3.45}$$

By definition, the Cholesky decomposition can solve (3.45) by first using forward substitution to solve the triangular system defined as

$$L\mathbf{z} = \mathbf{I}, \tag{3.46}$$

and then uses back substitution to solve the triangular system defined as

$$L^T A^{-1} = \mathbf{z}. \tag{3.47}$$

Hence, the inverse of matrix A due to the Cholesky decomposition is defined as

$$A^{-1} = \frac{(\frac{\mathbf{I}}{L})}{L^T}. \tag{3.48}$$

The Cholesky decomposition can also be used to simplify the notation of (3.41). Consider the second element of (3.41) where $A = [K(X,X) + \sigma_n^2 \mathbf{I}]$

$$\begin{aligned} K(X_*,X)A^{-1}K(X,X_*) &= K(X,X_*)^T (LL^T)^{-1} K(X,X_*) \\ &= K(X,X_*)^T (L^T)^{-1} L^{-1} K(X,X_*) \\ &= \mathbf{v}^T \mathbf{v}, \end{aligned} \tag{3.49}$$

where $\mathbf{v} = \frac{K(X,X_*)}{L}$. These simplifications due to the Cholesky decomposition give rise to the pseudo code for a Gaussian process regression, as denoted in Algorithm 2.

Algorithm 2 Gaussian process

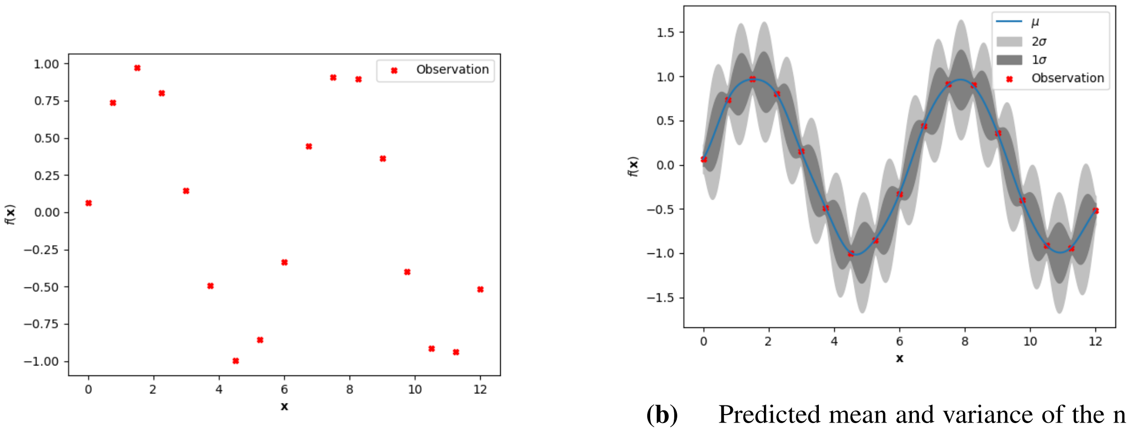
Input: X (inputs), $\mathbf{y}$ (targets), k (covariance function of Gram matrix K), σ_n^2 (noise level), X_* (prior datapoints)

```
L := Cholesky($[K(X,X) + \sigma_n^2 \mathbf{I}]$)
$A^{-1}\mathbf{y} := \frac{(\frac{\mathbf{I}}{L})}{L^T}$                }
$\overline{f}_* := K(X,X_*)^T A{-}1\mathbf{y}$                  }    predictive mean (3.40)
$\mathbf{v} := \frac{K(X,X_*)}{L}$                        }
$cov\overline{f}_* := K(X_*,X_*) - \mathbf{v}^T\mathbf{v}$             }    predictive covariance (3.41)
```

A graphical illustration of the functional capabilities of the Gaussian process regression in Algorithm 2 is shown in Figure 3.10. In Figure 3.10, noisy observations (indicated with red crosses) are modelled to obtain a predicted mean (indicated with a blue line), the first standard deviation 1σ (indicated with dark grey shading) and the second standard deviation 2σ (indicated with light grey shading).

3.7.2 Unsupervised learning

Unsupervised learning is rarely used by biologists, probably owing to it being a newer approach, as well as that it requires more advanced computer programming skills than supervised learning [77]. The reason why more advanced computer programming skills are required is due to unsupervised learning describing a more challenging problem than supervised learning [76]. The more challenging problem is that all unsupervised learning models consist of only a set of input variables (observations/measurements) without any corresponding output target variables (observations/measurements).



(a) Noisy data observations.

(b) Predicted mean and variance of the noisy data.

Figure 3.10. A graphical illustration of the functional capabilities of the Gaussian process regression in Algorithm 2. In this figure, noisy observations (indicated with red crosses) are modelled to obtain a predicted mean (indicated with a blue line), the first standard deviation 1σ (indicated with dark grey shading) and the second standard deviation 2σ (indicated with light grey shading).

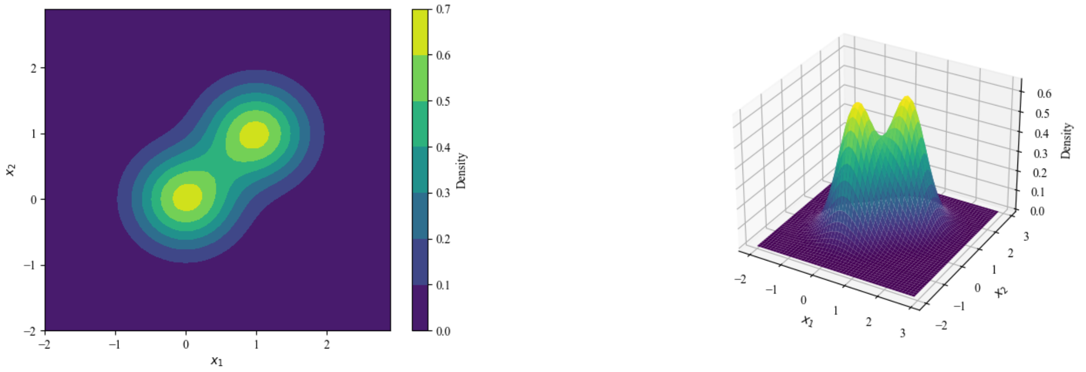
Hence, the unsupervised learning goal is (i) to cluster datasets; (ii) determine their density estimation; and (iii) to reduce the dimensionality of high-dimensional datasets for visualisation purposes without explicit defined labels (output/target variables) [17]. The unsupervised learning techniques of interest in this dissertation are clustering and dimensionality reduction.

3.7.2.1 Finite Gaussian mixture models

3.7.2.1.1 MIXTURE MODELS A mixture model is a probabilistic model that is defined as a linear combination of basic probability distributions that are used to describe more complex probability distributions [17, 79, 80]. A simple example of a two-component Gaussian mixture model used to visualise the mixture model definition is shown in Figure 3.11.

A key feature of mixture models is that they can be used to identify clusters of sub-populations in an observed dataset. In cases where the total number of distributions of the mixture model is known, it is referred to as a finite mixture model.

Consider an observed dataset $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_n)$ that has a total of n samples coinciding in a $m-$dimensional space. The mixture distribution of the datapoint $\mathbf{y}_i$ with a known finite number



(a) Two-dimensional representation of a multivariate Gaussian mixture model.

(b) Three-dimensional representation of multivariate Gaussian mixture model.

Figure 3.11. Example of a two-component multivariate Gaussian mixture model where brighter colours are used to indicate higher density values.

of components K can be represented with a sum of weighted densities defined as

$$p(\mathbf{y}_i|\theta) = \sum_{k=1}^{K} \pi_k p_k(\mathbf{y}_i|\theta_k), \tag{3.50}$$

where $p_k(\mathbf{y}_i|\theta_k)$ is the distribution of the kth component, with a parameter vector θ_k, the model parameter vector θ for all components is defined as $\theta = (\pi_1, \ldots, \pi_K; \theta_1, \ldots, \theta_K)$ and π_k is known as the mixing coefficient, which is a vector containing the probabilities that the observed datapoint $\mathbf{y}_i$ belong to component k. The properties of the mixing coefficient is that it is always positive $\pi > 0$ and that the vector adds up to one $\sum_{k=1}^{K} \pi_k = 1$.

A mixture model can be simplified by introducing a binary random variable z known as a latent variable that has K-dimensions. The latent variable is defined such that

$$z_{i,k} = \begin{cases} 1 & \text{if } \mathbf{y}_i \sim p_k(\mathbf{y}_i|\theta_k) \\ 0 & \text{otherwise} \end{cases}, \tag{3.51}$$

and

$$\sum_{K} z_k = 1. \tag{3.52}$$

This means that at any given state, if z_k is equal to one and all the rest of the latent variables are zero. The probability distribution of the latent variable is defined by

$$p(\mathbf{z}, \theta) = \prod_{k=1}^{K} \pi_k^{z_k}. \tag{3.53}$$

Consider the case of a K component Gaussian mixture distribution defined by

$$p(\mathbf{y}_i|\theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{y}|\mu_k, \Sigma_k). \tag{3.54}$$

The conditional distribution of the data, given its corresponding latent variable, is also a Gaussian distribution defined as

$$p(\mathbf{y}_i = |z_k = 1, \theta_k) = p_k(\mathbf{y}_i|\theta_k), \tag{3.55}$$

$$= \mathcal{N}(\mathbf{y}|\mu_k, \Sigma_k), \tag{3.56}$$

and can be standardised to

$$p(\mathbf{y}_i|\mathbf{z}, \theta) = \prod_{k=1}^{K} p_k(\mathbf{y}_i|\theta_k)^{z_k}, \tag{3.57}$$

$$= \prod_{k=1}^{K} \mathcal{N}(\mathbf{y}|\mu_k, \Sigma_k)^{z_k}. \tag{3.58}$$

This means that the marginal distribution of a finite-mixture distribution is simplified by summing over the joint distribution of (3.53) and (3.57) such that

$$p(\mathbf{y}_i|\theta) = \sum_{z} p(\mathbf{z}, \theta) p(\mathbf{y}_i|\mathbf{z}, \theta) = \sum_{k=1}^{K} \pi_k p_k(\mathbf{y}_i|\theta_k). \tag{3.59}$$

An important quantity when using the latent variable approach is the posterior probability of $z_k = 1$ when a datapoint y_i is observed. The expected value of the posterior probability is defined according to Bayes's theorem as

$$\tau_{i,k} = p(z_{i,k}|y_i) = \frac{\pi_k \mathcal{N}(y_i|\mu_k, \Sigma_k)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(y_i|\mu_k, \Sigma_k}. \tag{3.60}$$

3.7.2.1.2 BAYESIAN MIXTURE MODELS WITH UNKNOWN PARAMETERS When estimating finite-mixture models in a Bayesian setting, a conjugate prior distribution (see Section 3.6.2) is defined for all the unknown model parameters (θ) in (3.59). This means that the first step to define a Bayesian mixture model is to obtain model parameter values by sampling them from their prior distributions. Hence, the model parameter samples are acquired by sequentially sampling from the following prior distributions

$$\begin{aligned} \pi|\alpha &\sim Dir(\frac{\alpha_1}{K}, \ldots, \frac{\alpha_K}{K}), \\ z_i &\sim Mult(1 : \pi_1, \ldots, \pi_K), \\ \theta_{z_i}|G_0 &\sim G_0, \\ \mathbf{y}_i|\theta_{z_i} &\sim p_k(\mathbf{y}_i|\theta_{z_i}), \end{aligned} \tag{3.61}$$

where Dir is a Dirichlet distribution (see Section 3.4.2.2),with a concentration hyperparameter α, which is the conjugate prior to the latent variable z, $Mult$ is a multinomial distribution (see Section 3.4.1.1) with mixing proportions π, G_0 is a base prior distribution and $p_k(\mathbf{y}_i|\theta_{z_i})$ is the conditional distribution with the parameter θ_{z_i}. A graphical model that illustrates the conditional dependencies in a Bayesian mixture model is shown in Figure 3.12. In Figure 3.12, a grey circle represents an observed value, a white circle represents a random variable, a dashed-line block indicates a repetition of K or N times, and variables that are not placed within any constraint are hyperparameters, and arrows indicate the conditional dependencies.

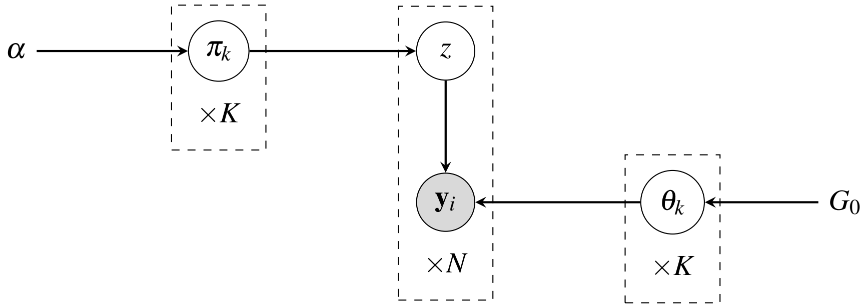


Figure 3.12. A simple graphical illustration of the conditional dependencies in a Bayesian mixture model. In this graph, a grey circle represents an observed value, a white circle represents a random variable, a dashed-line block indicates a repetition of K or N times, variables that are not placed within any constraint are hyperparameters, and arrows indicate the conditional dependencies (Adapted from [79], with permission).

3.7.2.1.3 BAYESIAN APPROACH TO MULTIVARIATE GAUSSIAN MIXTURE MODELS WITH UNKNOWN PARAMETERS The Bayesian mixture model of unknown parameters can be extended to a multivariate Gaussian approach. A multivariate Gaussian is governed by its μ and Σ parameters, as seen in (3.4). Hence, the base prior distribution G_0 is a NIW distribution ((3.7), see Section 3.6.2), which is the conjugate prior of μ and Σ. This is because the conjugate prior of μ is a multivariate Gaussian distribution, whereas the conjugate prior of Σ is an inverse-Wishart distribution[1]. Thus, the sequential sampling of a Bayesian approach to define a multivariate Gaussian mixture model of (3.59) is to sample from the following distributions

$$\pi|\alpha \sim Dir(\alpha_1,\ldots,\alpha_K), \tag{3.62}$$

$$z_i \sim Mult(1:\pi_1,\ldots,\pi_K), \tag{3.63}$$

$$\Sigma_{z_i} \sim inverse-Wishart(\nu_0,\Gamma_0), \tag{3.64}$$

$$\mu_{z_i} \sim \mathcal{N}(\mu_0,\frac{\Sigma_{z_i}}{\kappa_0}), \tag{3.65}$$

$$\mathbf{y}_i|\mu_{z_i},\Sigma_{z_i} \sim \mathcal{N}(\mathbf{y}_i|\mu_{z_i},\Sigma_{z_i}), \tag{3.66}$$

[1] To explain the sampling process of the covariance and mean the $\mathcal{NIW}$ distribution is split into an inverse-Wishart distribution (3.64) and a normal distribution (3.65). However, in practice only a $\mathcal{NIW}$ distribution is implemented.

where *Dir* is a Dirichlet distribution, *Mult* is a multinomial distribution with mixing proportions π, Σ and μ is sampled from a $\mathcal{NIW}$ distribution, ν_0, Γ_0, μ_0 and κ_0 are hyperparameters, $\mathbf{y}_i|\mu_{z_i},\Sigma_{z_i}$ is the conditional distribution with the parameters μ_{z_i} and Σ_{z_i}. A graphical model that illustrates the conditional dependencies in a Bayesian approach to a multivariate Gaussian mixture model is shown in Figure 3.12. In Figure 3.12, a grey circle represents an observed value, a white circle represents a random variable, a dashed-line block indicates a repetition of K or N, variables that are not placed within any constraint are hyperparameters, and arrows indicate the conditional dependencies.

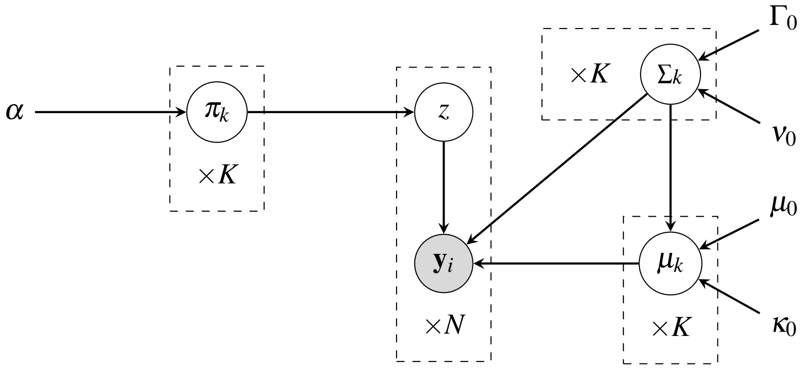


Figure 3.13. A simple graphical illustration of the conditional dependencies in a Bayesian approach to a multivariate Gaussian mixture model. In this graph, a grey circle represents an observed value, a white circle represents a random variable, a dashed-line block indicates a repetition of K or N times, variables that are not placed within any constraint are hyperparameters, and arrows indicate the conditional dependencies(Adapted from [79], with permission).

3.7.2.2 Principal Component Analysis (PCA)

Principal Component Analysis is a linear machine learning technique that forms a basis on which to evaluate multivariate data [17, 81]. This is because PCA can be used to reduce dimension of datasets in an attempt to increase computational performance and/or visualise the data. The applications of PCA were also extended to compare lossy data or to extract features from the multivariate dataset. There are two different mathematical definitions of PCA, but only one will be considered here as these definitions both result in the same algorithm. PCA is, therefore, defined as the process that maximises the orthogonal projection of a multivariate dataset onto a principal subspace where the principal subspace is a lower linear dimension then the original multivariate dataset.

Consider a dataset of dimensionality $\mathcal{K}$ consisting of N Euclidean variables such that the dataset is defined as $\{\mathbf{y}_n\}$ where $n = 1\ldots,N$. The goal of PCA is to project this dataset onto a dimension m where $m < \mathcal{K}$ such that the variance of the projected data is maximised. To explain how PCA obtains these projections, consider the projection onto a one-dimensional space $m = 1$. The vector that describes this projection is a unit vector u_1 with $\mathcal{K}$ dimensions such that $u_1^T u_1 = 1$. The variance of the projected data onto $m = 1$ is defined as

$$\sigma^2 = \frac{1}{N}\sum_{n=1}^{N}\{\mu_1^T \mathbf{y}_n - \mu_1^T \bar{\mathbf{y}}\}^2 \tag{3.67}$$

$$= \mu_1^T \mathbf{S} \mu_1, \tag{3.68}$$

where $\mu_1^T \mathbf{y}_n$ is the scalar projection of $\mathbf{y}_n$ onto the μ_1, $\mu_1^T \bar{\mathbf{y}}$ is the mean of the projected data, $\bar{\mathbf{y}}$ is the mean of the dataset defined by

$$\bar{\mathbf{y}} = \sum \frac{1}{N}\sum_{n=1}^{N} \mathbf{y}_n, \tag{3.69}$$

and $\mu_1^T \mathbf{S} \mu_1$ is the covariance of the dataset defined by

$$\mu_1^T \mathbf{S} \mu_1 = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{y}_n - \bar{\mathbf{y}})(\mathbf{y}_n - \bar{\mathbf{y}})^T. \tag{3.70}$$

To obtain the PCA of the dataset, we want to maximise the variance $\mu_1^T \mathbf{S} \mu_1$ with respect to the unit vector μ_1. However, maximisation is not as simple as it seems. This is because when the variance is maximised with respect to the unit vector, the maximisation tends to infinity. To mitigate this tendency to infinity, the maximising has to be constrained with a Lagrange multiplier λ_1 and the normalising condition of $\mu_1^T \mu_1 = 1$. This means that the variance is defined as

$$\mathcal{L}(\mu_1, \lambda_1) = \mu_1^T \mathbf{S} \mu_1 + \lambda_1(1 - \mu_1^T \mu_1), \tag{3.71}$$

and the maximisation thereof can be treated as an unconstrained maximisation. The derivative of the new variance with respect to the unit vector is defined as

$$\frac{\delta L}{\delta \lambda_1} = 2\mathbf{S}\mu_1 - \lambda_1 \mu_1, \tag{3.72}$$

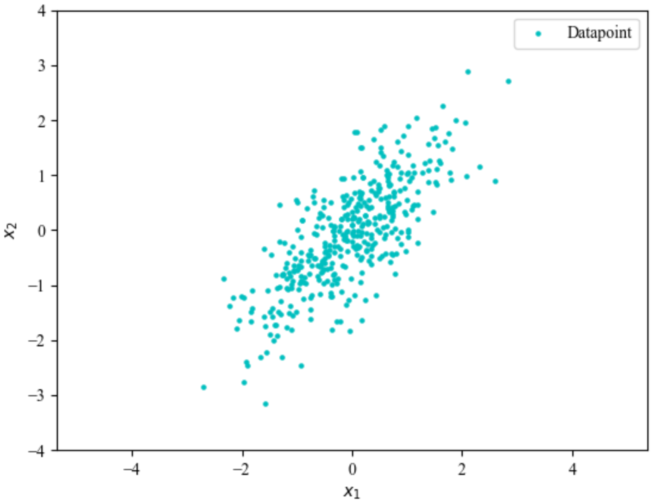
and when set equal to zero to obtain the stationary point that indicates maximisation, we obtain

$$\mathbf{S}\mu_1 = \lambda_1 \mu_1. \tag{3.73}$$

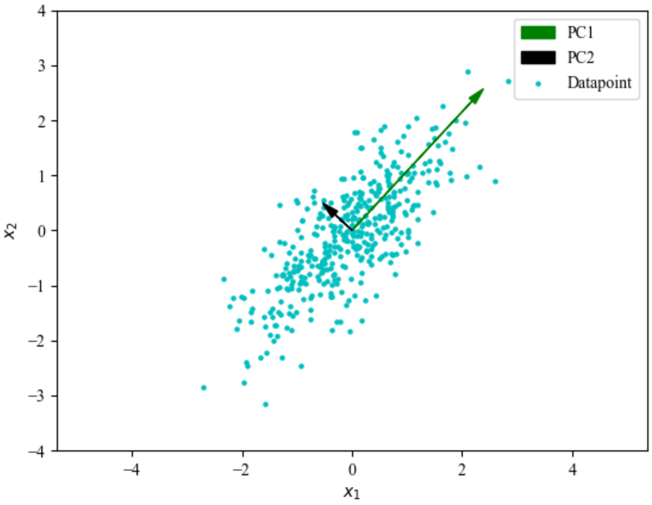
What this all boils down to is that the desired unit vector μ_1 is a eigenvector of the covariance matrix $\mathbf{S}$. Consider a rearranged version of the 3.73 defined as

$$\mu_1^T \mathbf{S} \mu_1 = \lambda_1. \tag{3.74}$$

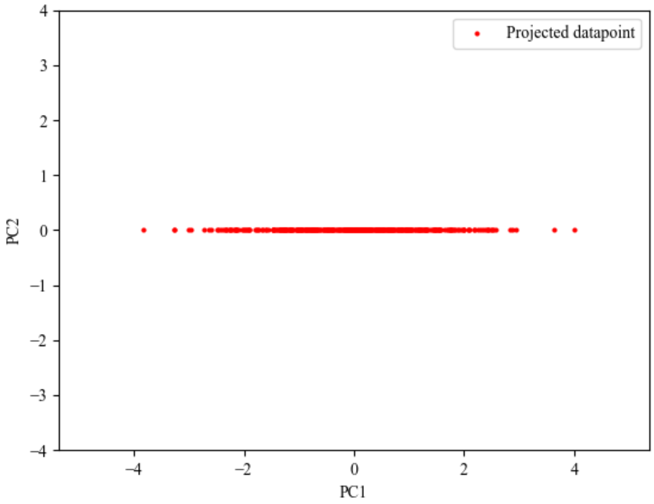
From (3.74) it is clear that the eigenvector that maximises the variance is the eigenvector with the largest eigenvalue (λ_1), due to the normalising property of $\mu_1^T \mu_1 = 1$. This eigenvector μ_1 is referred to as the first principal component. Hence, the principal components of a dataset are the eigenvectors of the covariance matrix, arranged decreasingly according to their eigenvalues to ensure maximum variance. A key property of the covariance matrix is that is it symmetric. This symmetric property is important in PCA as the eigenvectors of a symmetric matrix are always orthogonal to each other. This means that all the obtained principal components are always orthogonal to each other. A simple PCA analysis of a two-dimensional dataset with random variables x_1 and x_2 is shown in Figure 3.14. In Figure 3.14, (a) the datapoints of the random variable dataset is represented with blue dots, (b) the first principal component that defines the axis with the maximum variance of the dataset is a green vector, whereas the black orthogonal vector is the second principal component and describes the axis with the second most variance of the dataset. Finally, (c) the projected datapoints onto the first principal component is indicated with red dots..



(a) The datapoints of the two-dimensional dataset.



(b) The datapoints of the two-dimensional dataset with the first and second principal components.



(c) The projected datapoints onto the first principal component.

Figure 3.14. A simple PCA analysis of a two-dimensional dataset with random variables x_1 and x_2 is shown. In these figures, (a) the datapoints of the random variable dataset is represented with blue dots, (b) the first principal component that defines the axis with the maximum variance of the dataset is a green vector, whereas the black orthogonal vector is the second principal component and describes the axis with the second most variance of the dataset. Finally, (c) the projected datapoints onto the first principal component is indicated with red dots.

3.7.2.3 Diffusion maps

Diffusion maps is a non-linear dimensionality reduction technique that is used to find the underlining manifold (geometry) of a high-dimensional dataset [82, 83]. The framework on which diffusion maps are built is connectivity and diffusion distance.

3.7.2.3.1 CONNECTIVITY Connectivity is the probability of jumping from one datapoint to another when taking a random walk through the dataset. Hence, connectivity provides an important relationship between probability and distances in the feature space defined as

$$\text{con}(x,y) = p(x,y), \tag{3.75}$$

where $p(x,y)$ is the probability of reaching point y when starting at point x in one step of the random walk. A popular manner in which to describe the probability $p(x,y)$ is to use a Gaussian kernel defined by

$$k(x,y) = \exp\left(\frac{|x-y|^2}{\alpha}\right), \tag{3.76}$$

where α is the width of the kernel which is used to define a local neighbourhood around the datapoint x where similarity measurements are trusted. The constructed neighbourhood consists of all elements of y where $k(x,y) \geq \varepsilon$ and $0 < \varepsilon << 1$. The connectivity between x and y can, therefore, be defined as

$$\text{con}(x,y) = p(x,y) = \beta_x k(x,y), \tag{3.77}$$

where $\beta_x = \frac{1}{\sum_y k(x,y)}$ is a row-normalising constant that ensures that the kernel describes the probability of a single step. The obtained connectivity is used to populate a diffusion matrix P that describes the one step probabilities between datapoints. Hence, a diffusion matrix of a high dimensional dataset $\{X_i\}_{i=1}^n$ that consists of $n = 2$ datapoints is defined as

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}, \tag{3.78}$$

where P_{ij} corresponds to the connectivity (probability of jumping in one step) between X_i and X_j. This diffusion matrix is then used to define the probability of reaching X_j from X_i in t steps as

$$P^t = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}^t, \tag{3.79}$$

where P_{ij}^t is a summation of probabilities of all the possible paths between X_i and X_j. If $t = 2$, the diffusion matrix after two steps will be

$$P^2 = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}^2, \tag{3.80}$$

$$= \begin{bmatrix} p_{11}p_{11} + p_{12}p_{21} & p_{12}p_{22} + p_{11}p_{12} \\ p_{21}p_{12} + p_{22}p_{21} & p_{22}p_{22} + p_{21}p_{12} \end{bmatrix}, \tag{3.81}$$

where $P_{11} = p_{11}p_{11} + p_{12}p_{21}$, which describes the transition probability from point X_1 to X_1 after two steps. Hence, the diffusion matrix over multiple steps defines the true underlying structure of the dataset, as paths along this structure will have a high probability. This high probability is because datapoints along the true geometric structure of the data are densely populated. In layman terms this implies that locally defined geometric structures of the dataset can be used to define its global geometric structure.

3.7.2.3.2 DIFFUSION DISTANCE Diffusion distance is a tool used to estimate distance between datapoints, similar to Euclidean distance (Section 3.3). However, diffusion distance expands the constructed diffusion matrix of (3.79), which is based on connectivity (3.77), to define distance. What this means is that diffusion distance defines distance based on how well datapoints are connected. A simple illustration of the difference between Euclidean distance and diffusion distance is shown in Figure 3.15. In Figure 3.15, a dataset (indicated with circles) is considered where the distances between points A and B (AB) as well as points A and C (AC) are unknown. When examining these distances, the Euclidean distance (light blue dashed line), straight line distance, of AB and AC almost seem to be exactly the same. However, diffusion distance has a different conclusion. The diffusion distance of AB is small as there are many high probability paths connecting them, indicated with dark blue arrows. In contrast, the diffusion distance between AC is large due to the bottleneck between the points which decreases the number of high probability paths between them, indicated with red. Hence, the diffusion distance concludes that AB $<<$ AC whereas Euclidean distance concludes that AB $\approx$ AC.

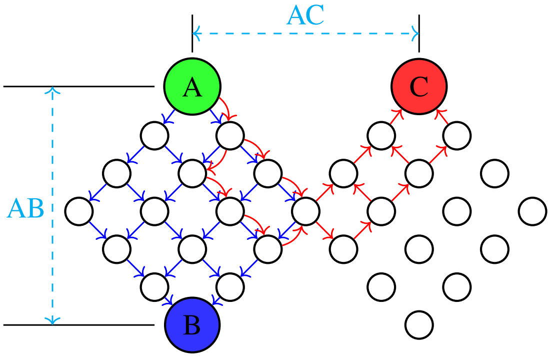


Figure 3.15. The difference between Euclidean distance and diffusion distance. Consider a dataset (indicated with circles) where the distances between points A and B (AB) as well as points A and C (AC) are unknown. When examining these distances, the Euclidean distance (light blue dashed line), straight line distance of AB and AC almost seem to be exactly the same. However, diffusion distance has a different conclusion. The diffusion distance of AB is small as there are many high probability paths connecting them, indicated with dark blue arrows. In contrast, the diffusion distance between AC is large due to the bottleneck between points which decreases the number of high probability paths between them, indicated with red. Hence, the diffusion distance concludes that AB $<<$ AC, whereas Euclidean distance concludes that AB $\approx$ AC (Adapted from [83], ©IEEE 2013).

Mathematically, the diffusion distance of Figure 3.15 is defined as

$$D_t(X_i,X_j)^2 = \sum_{u\in X} |p_t(X_i,u) - p_t(X_j,u)|^2 \tag{3.82}$$

$$= \sum_{k} |P^t_{ik} - P^t_{kj}|^2, \tag{3.83}$$

where $p_t(X_i,u)$ is the summation of all of the possible paths that define the probability of reaching any arbitrary point u in the dataset when starting at point X_i within t steps. In essence, for the diffusion distance to obtain a small value, the path probabilities between X_i and u should be similar to X_j and u. The only scenario in which they are similar is when both X_i and X_j are well connected by the arbitrary point u. This proves the statement that diffusion distance is small along the underlying geometric structure of the dataset because the true geometric structure is densely populated.

3.7.2.3.3 DIFFUSION MAP The first step in obtaining a diffusion map is to map the dataset onto a Euclidean space based on their diffusion distances. In this new diffusion space of the data, the diffusion distances become Euclidean distance. The mapping onto the new diffusion space is defined as

$$Y_i = \begin{bmatrix} p_t(X_i,X_1) \\ p_t(X_i,X_1) \\ \vdots \\ p_t(X_i,X_n) \end{bmatrix}, \tag{3.84}$$

where Y_i is a collection of vectors that describe the connectivity between point X_i and the dataset. The Euclidean distance between these vectors corresponds to the diffusion distance as

$$\begin{aligned} ||Y_i - Y_j||^2 &= \sum_{u\in X} |p_t(X_i,u) - p_t(X_j,u)|^2, \\ &= \sum_k = |P^t_{ik} - P^t_{kj}|^2, \\ &= D_t(X_i,X_j)^2. \end{aligned}$$

Similar to PCA (Section 3.7.2.2), diffusion maps also rely on eigenvalues and eigenvectors to reduce the dimensionality of the diffusion space. Consider the normalised diffusion matrix

$$P = D^{-1}K, \tag{3.85}$$

where K is kernel matrix such that $K_{ij} = k(X_i,X_j)$ and D^{-1} is a normalising diagonal matrix that consists of the row sums of K. The left eigenvectors and eigenvalues of (3.85) can be used to represent the diffusion distances of (3.84) such that the diffusion space is

$$Y_i = \begin{bmatrix} \lambda_1^t \psi_1(i) \\ \lambda_2^t \psi_1(i) \\ \vdots \\ \lambda_n^t \psi_1(i) \end{bmatrix}, \tag{3.86}$$

where λ_1^t is the eigenvalue of the first eigenvector of P, indicating its importance and $\psi_1(i)$ is the i^{th} element of that vector. The dimensionality of the diffusion space is then reduced by (i) ordering the eigenvectors decreasingly according to their eigenvalues; and (ii) selecting the first m vectors that approximate the diffusion distance best. As the diffusion matrix P is a symmetric matrix, all the obtained eigenvectors are orthogonal to each other.

3.8 CONCLUDING REMARKS

This section provided an overview of the basic statistical background required to understand the functionality of the novel approach to model cell differentiation in Chapter 4. The two main categories for developing a cell differentiation algorithm as stated in Section 2.3.1 are (i) dimensionality reduction; and (ii) trajectory inference. Hence, the algorithm of Chapter 4 utilises PCA (see Section 3.7.2.2) and diffusion maps (see Section 3.7.2.3) to perform dimensionality reduction to obtain a phenotypic manifold, which is used as input for trajectory inference. During trajectory inference, bifurcation points are estimated with Bayesian inference, (see Section 3.6) specifically Bayesian model selection of (i) a multivariate Gaussian distributions with unknown model parameters; as well as (ii) a multivariate Gaussian mixture model with unknown parameters (see Section 3.7.2.1.3). Both of these Bayesian models are defined by a combination of various distributions (see Section 3.4). During model selection of the Bayesian models, PCA and Frenet frames (see Section 3.2) are cleverly utilised to enhance the novel algorithm's performance. Finally, the trajectory inference step is concluded by utilising Euclidean distance (see Section 3.3) and Gaussian processes (see Section 3.7.1.1) to model cell differentiation as a continuous process.

CHAPTER 4 BAGEL: BAYESIAN ANALYSIS OF GENE EXPRESSION LINEAGES

4.1 CHAPTER OVERVIEW

In this chapter, an algorithm is developed known as BAGEL: **B**ayesian **A**nalysis of **G**ene **E**xpression **L**ineages, which estimates bifurcation points within a combined pseudo-time-principal-component space of (i) a primary single-cell gene expression dataset; or (ii) the projection of a sub-sampled secondary single-cell gene expression dataset onto the phenotypic manifold of a primary single-cell gene expression dataset, by utilising a Gibbs sampler and Bayesian model selection. These detected bifurcation points are then used to construct a continuous representation of cell lineages with a Gaussian process known as PC-lineages[1]. The process of developing BAGEL starts by first defining primary and secondary modelling goals, followed by its flow diagram. Next, an in-depth description of how BAGEL applies the three main parts of gene expression modelling: (i) data import; (ii) dimensionality reduction; and (iii) trajectory inference is discussed. Finally, as gene expression modelling is complex, the chapter is concluded by a summary of the assumptions and limitations of BAGEL.

4.2 BAGEL MODELLING GOALS

The primary modelling goal of BAGEL is to determine when a cell's fate choices are made. This primary modelling goal is achieved by (i) estimating bifurcation points with Bayesian inference; and (ii) to model cell differentiation as a continuous process with a Gaussian process. A bifurcation point, in terms of biology, is defined as the exact instant that a change in a cell's fate is detected along its cell developmental trajectory. An overly simplified visual illustration of BAGEL's primary goal is shown in Figure 4.1. As seen in Figure 4.1, cell differentiation can be visualised in two dimensions where the y-axis represents an arbitrary gene expression and the x-axis the pseudo-time ordering of cells. Starting

[1] PC-lineages is the reduced dimensional PCA representation of lineages of single-cell gene expression dataset in the combined pseudo-time-principal-component space.

at the green point, cells propagate (differentiate) towards their different terminal states (blue points) in a continuous fashion. Along this continuous cell developmental trajectory, a clear bifurcation point is indicated that answers the question: When is a cell's fate choice made? The secondary modelling goal of BAGEL is to develop a method that accurately projects a secondary sub-sampled single-cell gene expression dataset onto the phenotypic manifold of a primary optimally sequenced single-cell gene expression dataset consisting of many cells.

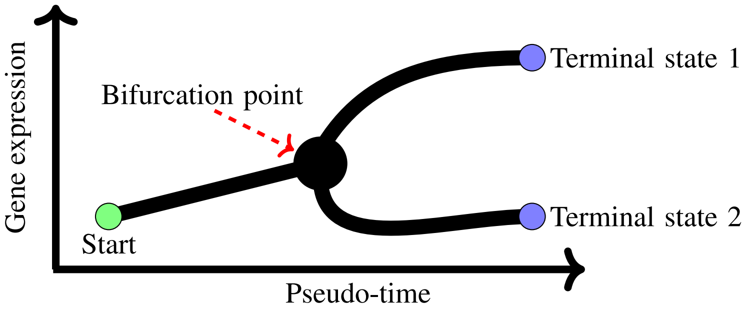


Figure 4.1. An overly simplified visual illustration of BAGEL's primary modelling goal. As seen, cell differentiation can be visualised in two dimensions where the y-axis represents an arbitrary gene expression and the x-axis the pseudo-time ordering of cells. Starting at the green point, cells propagate (differentiate) towards their different terminal states (blue points) in a continuous fashion. Along this continuous cell developmental trajectory, a clear bifurcation point is indicated that answers the question: When is a cell's fate choice made?

4.3 PROCESS TO ACHIEVE THESE MODELLING GOALS

The four main steps required when developing an cell differentiation modelling algorithm, as described by state-of-the-art literature [1], are shown in Figure 4.2. The first step is to clean up and normalise the sampled single-cell gene expression data, referred to as data pre-processing (P.1). The next step is to use the processed data to develop the underlying phenotypic manifold of the cell differentiation process via dimensionality reduction techniques (P.2). The manifold is then used as input for trajectory inference (P.3) to estimate gene expression lineages. Finally, the results of the simulations are visualised with graphs and figures (P.4).

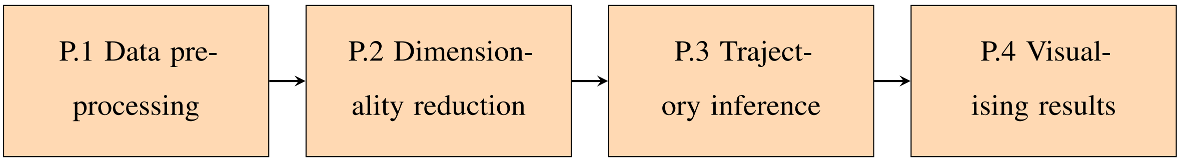


Figure 4.2. A flow diagram of the four main steps involved when developing an cell differentiation modelling algorithm. The first step is to clean up and normalise the sampled single-cell gene expression data, referred to as data pre-processing (P.1). The next step is to use the processed data to develop the underlying phenotypic manifold of the cell differentiation process via dimensionality reduction techniques (P.2). The manifold is then used as input for trajectory inference (P.3) to estimate gene expression lineages. Finally, the results of the simulations are visualised with graphs and figures (P.4).

4.4 BAGEL: BAYESIAN ANALYSIS OF GENE EXPRESSION LINEAGES

BAGEL models the complex biological process of cell differentiation by sequentially executing the steps below.

1. Define an underlying low dimensional phenotypic manifold of the cell differentiation process based on the Palantir algorithm [1] by applying: (i) PCA_d[2] (top 300 components); (ii) diffusion maps (top five diffusion components, although the Palantir algorithm results are shown to be robust to the number of diffusion components); and (iii) PCA_v[3] for the purpose of visualising the phenotypic manifold. Hence, the single-cell gene expression dataset is reduced to two visual dimensions, namely $\text{PC}_\text{v}1$ and $\text{PC}_\text{v}2$.
2. Project a secondary sub-sampled single-cell gene expression dataset (see Section 5.3) onto the phenotypic manifold of a primary optimally sequenced single-cell gene expression dataset consisting of many cells by manipulating the PCA_d step of dimensionality reduction (see Section 4.6.1 for an in-depth description).
3. Incorporate the pseudo-time of the Palantir algorithm, which is a numeric value with arbitrary units, that describes a measure of how far a particular cell datapoint is along the cell developmental trajectory of cell differentiation.
4. Transform a sequence of pseudo-time-intervals to a sequence of sequential windows that are translated and rotated within a combined pseudo-time-principal-component space, $\mathcal{P} =$ [pseudo-time,

[2] PCA_d is the PCA used for dimensionality reduction purposes characterised with a subscript-d.

[3] PCA_v is the PCA used for visualisation purposes characterised with a subscript-v.

$\mathrm{PC_v1},\ \mathrm{PC_v2}] = [\mathcal{T}_p,\ \mathrm{PC_v1},\ \mathrm{PC_v2}]$, using the tangent vectors of window-based Frenet frames[4]. The tangent vectors of these window-based Frenet frames are parallel to the cell developmental trajectory of the cell datapoints of each window. The tangent vector is obtained by determining the first principal component $\mathrm{PCA_w}$ [5] of the cell datapoints of a pseudo-time-interval. When concatenating the obtained tangent vectors of the window-based Frenet frames of all the sequential windows, a tangent vector window-based Frenet frame of the entire cell developmental trajectory can be defined.

5. Infer bifurcation points via Bayesian model selection and a Gibbs sampler along the transformed pseudo-timeline.
6. Construct a continuous representation of cell lineages with a Gaussian process in the space $\mathcal{P}$ known as PC-lineages.

A flow diagram on how BAGEL achieves all of these modelling goals is shown in Fig 4.3. The process of cell differentiation modelling starts (S.1) by defining initial model parameters (P.1.1) as well as input single-cell gene expression data (P.1.2). A key property of BAGEL is its capability of using two different datasets as input defined as the primary and secondary dataset respectively. The capability of utilising two datasets allows for the opportunity to combine the datasets by projecting the secondary dataset onto the phenotypic manifold of the primary dataset. This functionality allows users to visualise small numbers of sub-sampled cells[6] from the secondary dataset on the cell developmental trajectory of the optimally sequenced primary dataset, which consists of many cells. The first restriction of this functionality is that the primary dataset should contain an increased number of cell datapoints compared to the secondary dataset. This increased number of primary cell datapoints are to ensure that the secondary dataset does not distort the visualisation step of the phenotypic manifold of the primary dataset. The rule of thumb based on the datasets used during BAGEL implementation, is that the total number of cell datapoints of the secondary dataset, should be at most be equal to ten percent (%10) of the total number of primary cell datapoints. The second restriction is that the datasets should express the similar differentiation precursors. It was found that BAGEL produced credible results even when the main and secondary datasets were sequenced differently.

[4] A Frenet frame is typically applied to each point on a curve however, in the case of the developed algorithm a Frenet frame is applied to a pseudo-time-interval of cell datapoints along the phenotypic manifold. This is referred to as a window-based Frenet frame.

[5] $\mathrm{PCA_w}$ is the PCA used to obtain the tangent vector of the window-based Frenet frame characterised with a subscript-w.

[6] Cells are usually sub-sampled by influencing factors as discussed in Section 5.3.

The first decision block (D.1) allows input of one (P.2.1) or two (P.2.2) datasets to BAGEL. If one dataset is used (P.2.1), only the visualisation step of the Palantir algorithm's dimensionality reduction technique is changed compared to the original Palantir algorithm (see Section 4.6.1). However, when two datasets are used (P.2.2), PCA_d is used to minimise the influence of the projection of the secondary data on the Palantir algorithm's low dimensional phenotypic manifold development of the primary dataset (see Section 4.6.1 for an in-depth description) before changing the visualisation step. Both P.2.1 and P.2.2 produce the Palantir algorithm's (i) well defined phenotypic manifold of the cell developmental trajectory; (ii) pseudo temporal ordering of the cells; and (iii) terminal states[7] (cell fates).

The output of the Palantir algorithm is used as input to a continuous BAGEL-loop which is the core of BAGEL. The input of BAGEL-loop is therefore the single-cell gene expression data arranged according to its two visual axes $\text{PC}_\text{v}1$ and $\text{PC}_\text{v}2$ and its pseudo-time within space $\mathcal{P}$. The purpose of BAGEL-loop is to determine if there is a bifurcation point in the input dataset. When BAGEL-loop does not detect a bifurcation point in the input dataset, the input dataset is defined as a distinct PC-lineage. However, when BAGEL-loop detects a bifurcation point in the input dataset, the bifurcation point is used to define two distinct PC-lineages from the input dataset.

BAGEL-loop consists of four main parts namely *all lineages detected* (D.2), *window method* (P.3.1), *estimate bifurcations* (P.3.2) and *associate data* (P.3.3). After receiving the input dataset in space $\mathcal{P}$ at D.2, BAGEL-loop starts at the *window method* (P.3.1) which is used to define a "window" that can be thought of as a small glimpse in pseudo-time of the cell differentiation process. The first step of the *window method* (P.3.1) is to define a pseudo-time-interval in which a user defined number of cell datapoints are selected from the input data. A characteristic of the cell datapoints inside a pseudo-time-interval is that they follow the cell developmental trajectory sequentially based on pseudo-time. The cell datapoints of the pseudo-time-interval are used to define a tangent vector PC_w parallel to the cell developmental trajectory in the space $\mathcal{P}$ (this tangent vector is the tangent vector of the window-based Frenet frame). Finally, a window of unique cell datapoints[8] are "sliced" perpendicular to the cell developmental trajectory in the space $\mathcal{P}$, by selecting a user defined number of cell datapoints from the pseudo-time-interval in the direction of the tangent vector. The output of the *window method* (P.3.1) is

[7] The obtained terminal states of the Palantir algorithm only represent the terminal states of the dataset and not the terminal states of cell differentiation, as the input single-cell gene expression dataset only contain a snapshot of cell differentiation.

[8] Cell datapoints that are represented in previous windows can by definition not be included in subsequent windows see Section 4.7.2.

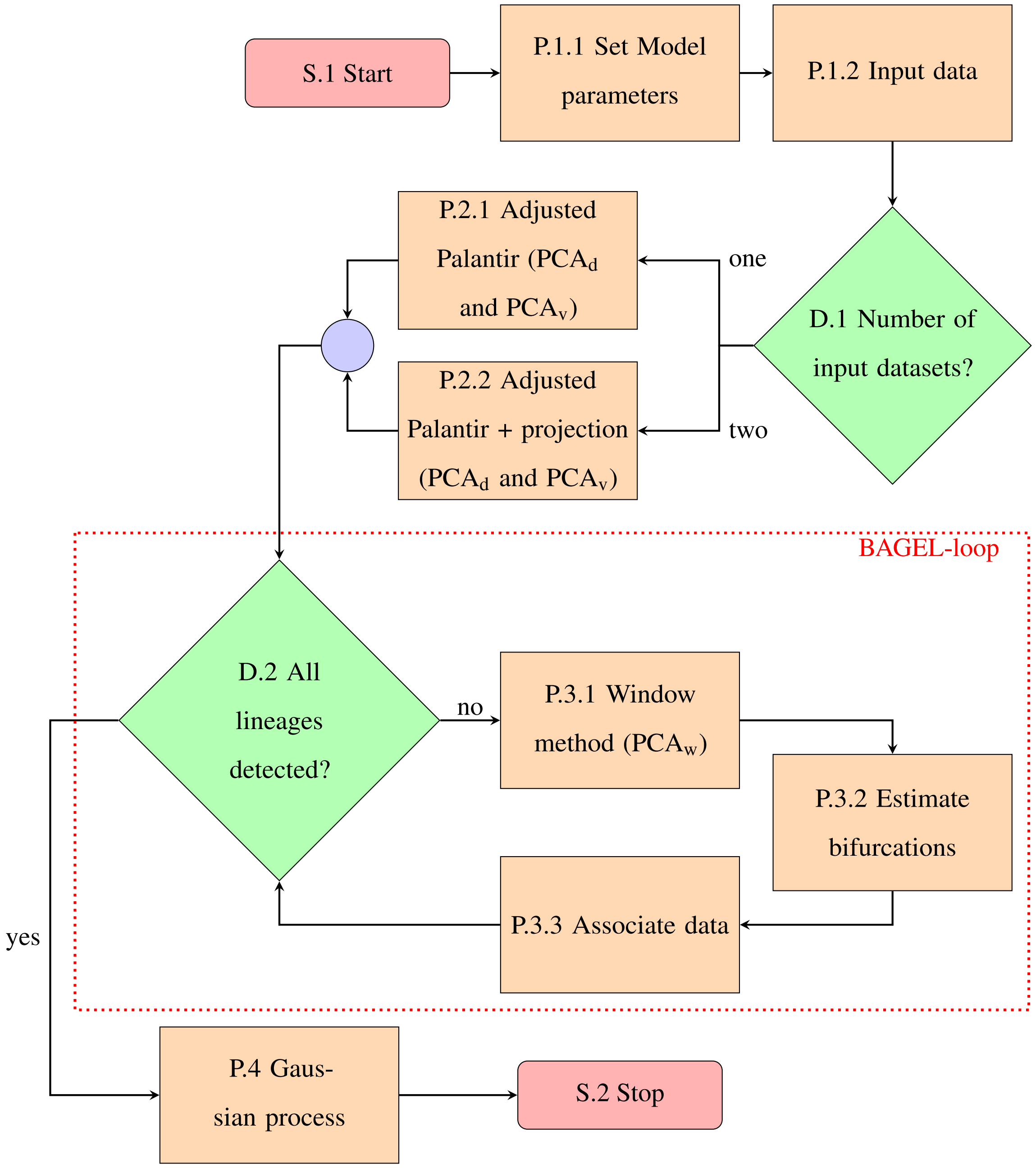

S.1 Start
P.1.1 Set Model parameters
P.1.2 Input data
D.1 Number of input datasets?
one
two
P.2.1 Adjusted Palantir (PCA_d and PCA_v)
P.2.2 Adjusted Palantir + projection (PCA_d and PCA_v)
BAGEL-loop
D.2 All lineages detected?
no
P.3.1 Window method (PCA_w)
P.3.2 Estimate bifurcations
P.3.3 Associate data
yes
P.4 Gaussian process
S.2 Stop

Figure 4.3. Flow diagram of the process by which BAGEL models cell differentiation to estimate PC-lineages using gene expressions.

(i) the tangent vector of the window-based Frenet frame; and (ii) a cluster of cell datapoints of the cell developmental trajectory partitioned according to the window. These unique cell datapoints are used to determine if there is a bifurcation point (P.3.2) within the window via Bayesian model selection.

Bayesian model selection determines whether the cell datapoints of a window is optimally represented with one (Model 1) or a mixture of two (Model 2) multivariate Gaussian distributions. If Bayesian model selection selects Model 1 as the best fit for the cell datapoints of the current window, all of the cell datapoints from the window are associated with one distinct PC-lineage (P.3.4). After associating the cell datapoints to the PC-lineage, BAGEL-loop is repeated by creating a new window (P3.1) based on the next pseudo-time-interval from the input dataset of D.2 and the processes of model selection (P.3.2) and data association (P3.3) are repeated. When no bifurcation point is detected in all of the input data (the series of sequential windows of the input dataset is best represented with Model 1), BAGEL will define the input data of D.2 as a distinct detected PC-lineage.

However, if Bayesian model selection selects Model 2 as the best fit for the datapoints of the current window, two bifurcation validation methods known as local and global validation is applied. Local validation validates the fit of Model 2 to the current window datapoints, whereas global validation compares subsequent windows by a majority voting rule to detect a bifurcation point. As seen these validation methods are used to ensure the credibility of a detected bifurcation point. When a bifurcation point is detected BAGEL defines the input dataset of D.2 as containing two distinct PC-lineages by (i) duplicating the distinct PC-lineage before the detected bifurcation point which consist of a series of sequential windows of the input dataset where Model 1 is their best fit; and (ii) associating all of the cell datapoints after the bifurcation point to the PC-lineage they most likely represent (P.3.4). After defining the two PC-lineages, each of them is used as an input to BAGEL-loop. This step refines the defined PC-lineages to ensure that they do not contain any additional bifurcation points after bifurcating. This process continues until no bifurcation points are detected in any defined PC-lineages (all of the individual PC-lineages are best represented with a series of sequential windows where Model 1 is their best fit) implying that all of the possible PC-lineages have been detected (D.2). Finally, BAGEL stops (S.2) after modelling each detected PC-lineage as a continuous process using a Gaussian process (P.4).

4.5 DATA IMPORT

In order to ensure credible results, the sampled sc-RNAseq data needs to be pre-processed. The first step is to eliminate low molecule count cells and genes with low detection rates. Cells with fewer than 1000 molecules are eliminated (fewer then 1000 genes expressed) and genes that are expressed in fewer than 10 cells are eliminated. The next step is to normalise the data to ensure that the data is not biased. This is achieved through row-normalising the dataset by dividing the different gene expression counts of each individual cell by its own total number of gene expression counts (the sum of column values for that cell/row) [1]. Finally, the data is log-transformed, which is a common practice with normalised single-cell gene expression data. The log transform of the normalised data has three important effects, which are [84]:

1. The log-transformed data provides a canonical way to measure changes in gene expressions by utilising the distances between log-transformed expression values.
2. The mean–variance relationship within a single-cell is mitigated.
3. The skewness of the normalised data is reduced.

4.5.1 Data import algorithm

The procedure for importing and pre-processing data is summarised in Algorithm 3.

Algorithm 3 Data import and pre-processing

```
Input: Single-cell gene expression data in the form of cell-by-gene matrix of counts.
Clean up data by eliminating: (i) cells with no gene expressions; and (ii) genes that were not present in any cell.
Filter the data by eliminating low molecule count cells ($< 1000$) and genes with low detection rate ($< 10$).
Normalise data.
Log transform data.
Output Pre-processed single-cell gene expression data.
```

4.6 DIMENSIONALITY REDUCTION

Dimensionality reduction refers to the process used to develop an underlying phenotypic manifold of the single-cell gene expression data. BAGEL's approach to obtain a phenotypic manifold is based on the Palantir algorithm (Appendix A). The Palantir algorithm develops the underlying phenotypic manifold by applying all three known techniques for dimensionality reduction of single-cell gene expression data [28], including: (i) $\mathrm{PCA_d}$[9] (top 300 components); (ii) diffusion maps (top five diffusion components, although the Palantir algorithm results are shown to be robust to the number of diffusion components); and (iii) t-distributed stochastic neighbour embedding (t-SNE) for visualisation (top two components). After the phenotypic manifold is obtained, the Palantir algorithm uses the Markov affinity-based graph imputation of cells (MAGIC) algorithm to de-noise phenotypic manifold data [1]. In BAGEL the visualisation step of the Palantir algorithm was changed from t-SNE to $\mathrm{PCA_v}$. This is

[9] $\mathrm{PCA_d}$ is the PCA used for dimensionality reduction purposes characterised with a subscript-d

because t-SNE is a probabilistic (non-static) transformation which means t-SNE can be viewed as a black box model that produces different outputs each time, whereas PCA$_\text{v}$ performs mathematically based (static) transformations which allows for the same result with each simulation [85]. Hence, the single-cell gene expression dataset is reduced to two visual dimensions, namely PC$_\text{v}$1 and PC$_\text{v}$2. BAGEL also does not use MAGIC imputation to de-noise the phenotypic manifold data, since the imputed data (i) does not reflect the actual sampled cells; and (ii) distorts the phenotypic manifold when incorporating pseudo-time as the third axis of visualisation.

4.6.1 Dataset projection

BAGEL allows projection of a secondary dataset onto a primary dataset's phenotypic manifold which is accomplished by manipulating the first PCA$_\text{d}$ step of the Palantir algorithm. This PCA$_\text{d}$ step is manipulated by iterating through all the secondary cell datapoints one cell datapoint at a time, as seen in the following 13 steps:

- Step 1: Filter the secondary dataset by only selecting cell datapoints with one or more genes that correspond with the primary dataset.
- Step 2: Append genes that are missing in the secondary dataset when compared to the primary dataset, and zero pad the expression rates of the genes that are not expressed in the secondary dataset.
- Step 3: Define an empty array that will be used to store the results of the projected PCA$_\text{d}$ results of the secondary dataset.
    - Step 4: Select a unique cell datapoint from the secondary dataset.
    - Step 5: Append the selected cell datapoint to the primary dataset.
    - Step 6: Pre-process the dataset.
    - Step 7: Perform PCA$_\text{d}$ of the combined primary dataset including the cell datapoint of the secondary dataset.
    - Step 8: Select the secondary PCA$_\text{d}$ cell datapoint result.
    - Step 9: Append the selected secondary PCA$_\text{d}$ cell datapoint result to the secondary PCA$_\text{d}$ result dataset.
    - Step 10: Go to step 4 until all cell datapoints within the secondary dataset have been converted to the primary dataset PCA$_\text{d}$ space.

- Step 11: Calculate PCA$_{\mathrm{d}}$ of the primary dataset.
- Step 12: Append the newly created PCA$_{\mathrm{d}}$ data frame of the secondary dataset to the PCA$_{\mathrm{d}}$ data of the primary dataset.
- Step 13: Apply diffusion maps and PCA$_{\mathrm{v}}$.

The technique of using PCA$_{\mathrm{d}}$ to project the secondary dataset onto the primary dataset is effective because the influence of the secondary dataset is directly proportional to the ratio of the number of cell datapoints in the secondary dataset to the number of cell datapoints in the primary dataset. In most cases the primary dataset will include more than 4000 cell datapoints [1,6] which implies that the influence of a single secondary cell datapoint is negligible during the PCA$_{\mathrm{d}}$ calculations. The restrictions of this functionality based on the practical implementations of Table 5.2 are (i) that the primary dataset should contain an increased number of differentiated cells compared to the secondary dataset; and (ii) the datasets should express the similar differentiation precursors.

4.6.2 Dimensionality reduction algorithm

The process of obtaining the phenotypic manifold of the primary dataset and the projection of the secondary dataset onto the phenotypic manifold of the primary dataset is summarised in Algorithm 4.

Algorithm 4 Phenotypic manifold and single-cell projection

```
Input: Pre-processed single-cell gene expression data (output of Algorithm 3).
if Two datasets == True then
    Projected-secondary-dataset→ Project the secondary dataset onto the primary dataset (see Section 4.6.1).
    PCA_d_300→ Compute top 300 PCA_d components of the primary dataset.
    Combined-datasets → Combine the projected-secondary-dataset and PCA_d_300.
    Diffusion_map→ Compute Diffusion map of combined-datasets.
    Visualise→ Compute top 2 PCA_v components of Diffusion_map.
else
    PCA_d_300→ Compute PCA_d top 300 components of primary dataset.
    Diffusion_map→ Compute Diffusion map of PCA_d_300.
    Visualise→ Compute top 2 PCA_v components of Diffusion_map.
end if
Output: Cell datapoints of phenotypic manifold.
```

4.7 TRAJECTORY INFERENCE

The core of modelling cell differentiation is the trajectory inference step, as this step is usually used to provide novel insights about cell differentiation.

4.7.1 Pseudo-time

Pseudo-time is a numeric value with arbitrary units, which is a measure that defines how far a particular cell is along the cell developmental trajectory of cell differentiation. Pseudo-time is utilised in trajectory inference algorithms that estimate cell lineages to help understand a cell's fate choices [3]. These algorithms base the pseudo-time ordering of cells on their transcription profiles [4] and it was proven to effectively model gene expression lineages by [1,3,4,26]. BAGEL utilises the pseudo-time ordering of cells from the Palantir algorithm [1] (Appendix A.3). To visually understand pseudo-time ordering of cell datapoints, an example of arbitrary cell datapoints is shown in Figure 4.4. In Figure 4.4, the arbitrary primary cell datapoints (black dots) are reduced to one visual dimension (PC_v1), and arranged according to their pseudo-time ordering. The projection capabilities of Algorithm 4 is also shown where a secondary cell datapoint (black circle filled with red) is accurately projected onto the phenotypic manifold of the primary cell datapoints.

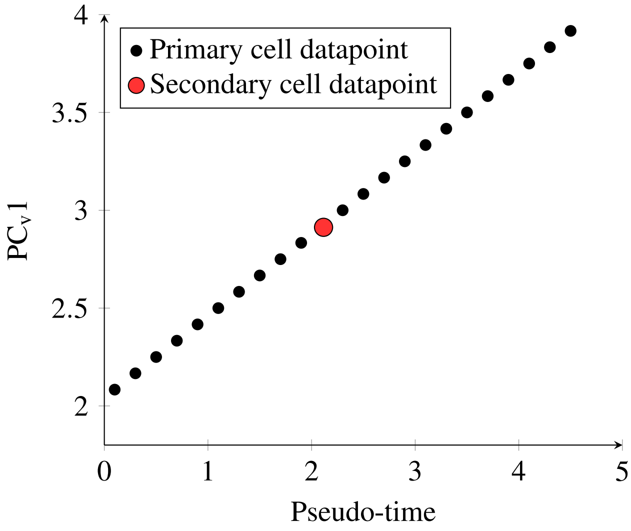


Figure 4.4. The visualisation of pseudo-time ordering of arbitrary cell datapoints. In this figure, the arbitrary primary cell datapoints (black dots) are reduced to one visual dimension (PC_v1), and arranged according to their pseudo-time ordering. The projection capabilities of Algorithm 4 is also shown where a secondary cell datapoint (black circle filled with red) is accurately projected onto the phenotypic manifold of the primary cell datapoints.

4.7.2 Window method

BAGEL transforms the obtained phenotypic manifold of the single-cell gene expression data from a sequence of pseudo-time-intervals, to a sequence of sequential windows that are translated and rotated within the space $\mathcal{P}$, using the tangent vectors of window-based Frenet frames. These sequential windows can be thought of as a small glimpse in pseudo-time of the cell differentiation process. A characteristic of a window is that each of the individual windows in a series of windows will contain an unique cluster of cell datapoints from the single-cell gene expression dataset. Hence, the process of cell differentiation can be defined as a series of sequential pseudo-time steps. The main advantages of dividing the phenotypic manifold into windows is that (i) bifurcation point detection can be simplified; and (ii) the cell developmental trajectory of cell differentiation can be represented with the tangent vector of the window-based Frenet frame.

The window method of transforming the pseudo-time is dependent on two user-defined intervals, called *pseudo-time-interval* (Δ_t) and *window-interval* (Δ_w) also known as a window, which have a relationship of $\Delta_t \geq \Delta_\mathrm{w}$. These two intervals are required for the two different types of partitioning steps implemented when defining the cluster of cell datapoints of a window namely Type 1: pseudo-time partitioning and Type 2: window partitioning. For the purpose of explaining the functionality of these intervals and the partitioning steps, the user defined intervals Δ_t and Δ_w are set to contain a total of 19 cell datapoints and 11 cell datapoints respectively.

4.7.2.1 Type 1: pseudo-time partitioning

The purpose of type 1 partitioning is (i) to define a cluster of cell datapoints within the Δ_t interval, which sequentially follow the pseudo-time ordering of the space $\mathcal{P}$, known as Δ_t-cluster; and (ii) to define a tangent vector of the window-based Frenet frame which is parallel to the cell developmental trajectory of Δ_t-cluster by utilising PCA$_\mathrm{w}$. Hence, consider Figure 4.5 where arbitrary cell datapoints represent the process of cell differentiation in the space $\mathcal{P}$ with dots. As seen, in Figure 4.5 a Δ_t-cluster is obtained by selecting a total of 19 cell datapoints (yellow dots) from the arbitrary cell datapoints, sequentially according to their pseudo-time ordering in space $\mathcal{P}$ within the interval of Δ_t (orange dotted line). Also in Figure 4.5 the characteristic of a window can be observed, where the previous defined window cell datapoints (red dots) does not influence the cell datapoints selection process of Δ_t-cluster. After Δ_t-cluster is obtained it is used to define the tangent vector of the window-based Frenet frame.

As the first principal component ($\mathrm{PC_w}1$) of a dataset describes the direction of the maximum amount of variance of that dataset with a vector [17], it is used to define a tangent vector parallel to the cell developmental trajectory. Hence, the first principal component of Δ_t-cluster is used to define the tangent vector of the window-based Frenet frame positioned at the mean of the $\mathrm{PCA_w}$. It should be noted that when computing the tangent vector $\mathrm{PC_w}$, the maximum amount of variance of Δ_t-cluster may be in the $\mathrm{PC_v}1$-$\mathrm{PC_v}2$ plane instead of the pseudo-time dimension. To mitigate this undesirable outcome, a process known as pseudo-time-scaling is introduced. Pseudo-time-scaling enforces a widening in Δ_t-cluster with respect to pseudo-time. This widening ensures that the variance of pseudo-time in Δ_t-cluster is significantly larger than the variance of the $\mathrm{PC_v}1$-$\mathrm{PC_v}2$ plane. A simple manner in which to implement this widening is to multiply each of the pseudo-time values of Δ_t-cluster with a large unit less constant value eg. 10000. This widening, therefore, ensures that the obtained principal component $\overline{\mathrm{PC_w}1}$ will always point in the direction of pseudo-time and will only change direction in the space $\mathcal{P}$ due to the influence of the $\mathrm{PC_v}1$-$\mathrm{PC_v}2$ plane.

The obtained $\overline{\mathrm{PC_w}1}$ pseudo-time $\mathrm{PCA_w}$ mean value and its pseudo-time tangent vector component does not represent Δ_t-cluster, but rather its widened pseudo-time equivalent. Hence, $\overline{\mathrm{PC_w}1}$ is un-widened by dividing its pseudo-time $\mathrm{PCA_w}$ mean value and its pseudo-time tangent vector component by the widened value which in this case is 10000. Finally, as seen in Figure 4.5 this division step provides a vector parallel to the cell developmental trajectory (blue arrow) in space $\mathcal{P}$ positioned at the $\mathrm{PCA_w}$ mean (black circle filled with green) known as $\mathrm{PC_w}1$ (blue arrow). For conceptual purposes the normal vector of the window-based Frenet frame $\mathrm{PC_w}2$ is also shown with a gray arrow.

4.7.2.2 Type 2: window partitioning

The second type of partitioning defines a cluster of cell datapoints within the Δ_w interval (also known as a window), which sequentially follow the cell developmental trajectory of cell datapoints along the tangent vector $\mathrm{PC_w}1$. Type 2 partitioning starts by projecting the obtained Δ_t-cluster onto the vector subspace of $\mathrm{PC_w}1$ [86] by utilising

$$proj_{\mathrm{PC_w}1}\bar{\mathbf{a}} = \frac{\bar{\mathbf{a}} \cdot \mathrm{PC_w}1}{|\mathrm{PC_w}1|^2} \times \mathrm{PC_w}1, \tag{4.1}$$

where $proj_{\mathrm{PC_w}1}\bar{\mathbf{a}}$ is the projection of vector $\bar{\mathbf{a}}$ which is a cell datapoint of Δ_t-cluster onto an $\mathrm{PC_w}1$ vector axis. A visual illustration of how (4.1) projects one of the datapoints of an arbitrary Δ_t-cluster (black dots) onto an obtained $\mathrm{PC_w}1$ vector axis (dark blue arrow) is shown in Figure 4.6. As seen in Figure 4.6, the projected cell datapoint (red dot) is obtained by projecting its vector (red arrow) onto the $\mathrm{PC_w}1$ vector (light blue dotted line). These projections are repeated for all the cell datapoints in the

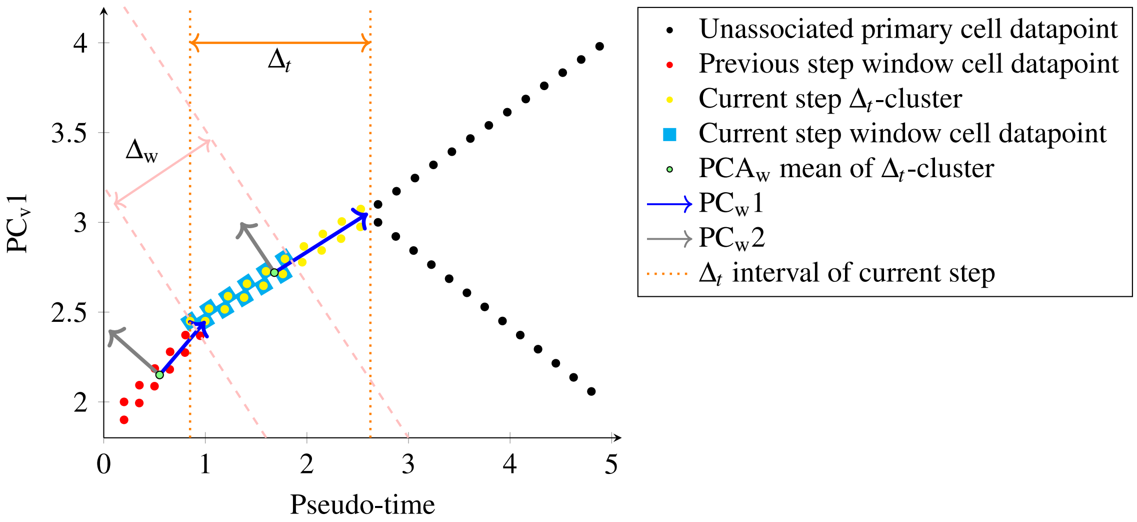


Figure 4.5. A visual illustration of how the window method sequentially divides the phenotypic manifold into windows. As seen, a Δ_t-cluster (yellow dots) is obtained by selecting a total of Δ_t cell datapoints sequentially according to their pseudo-time ordering in space $\mathcal{P}$, while excluding the cell datapoints of the previous window (red dots). Next, the tangent vector of the window-based Frenet frame PC_w1 is defined by taking the first principal component of the obtained Δ_t-cluster. This tangent vector PC_w1 is centred at the mean of the executed PCA_w of the Δ_t-cluster. Finally, a window (light blue squares) of cell datapoints is obtained by selecting a total of Δ_w cell datapoints along the obtained tangent vector PC_w1.

Δ_t-cluster until the desired cluster of cell datapoints within the Δ_w interval is reached, starting from the first datapoint (most left in the diagram) of the current time step window and progressing along PC_w1.

As seen in Figure 4.5, after the Δ_t-cluster and its tangent vector PC_w1 is defined 11 cell datapoints (light blue squares) are selected in the direction of the PC_w1 vector to define a window. Finally, after a window is defined the next subsequent window is obtained by starting at type 1 partitioning again and repeating the process.

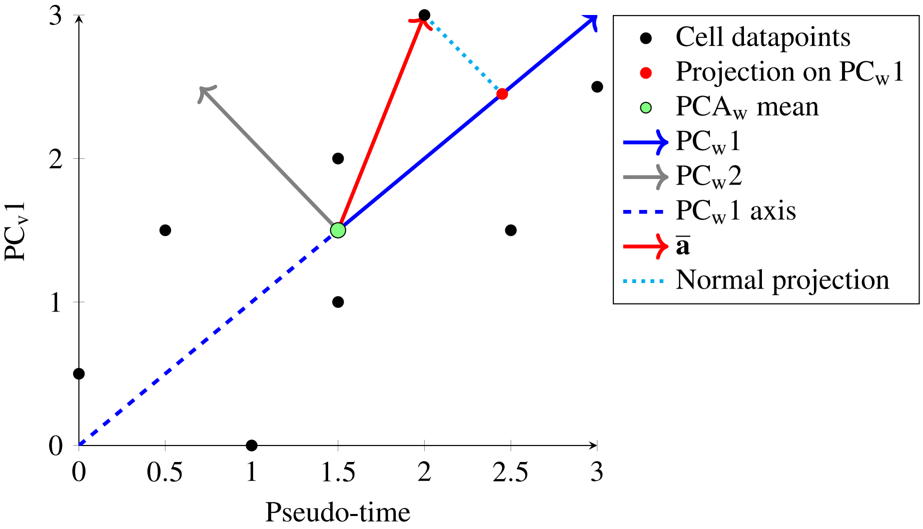


Figure 4.6. A visual illustration of how (4.1) projects one of the datapoints of an arbitrary Δ_t-cluster (black dots) onto an obtained PC_w1 vector axis (dark blue arrow). As seen, the projected cell datapoint (red dot) is obtained by projecting its vector (red arrow) onto the PC_w1 vector (light blue dotted line). These projections are repeated for all the cell datapoints in the Δ_t-cluster the desired cluster of cell datapoints within the Δ_w interval is reached, starting from the first datapoint (most left in the diagram) of the current time step window and progressing along PC_w1.

4.7.2.3 Terminal state operation

During type 1 partitioning, BAGEL automatically selects at least two consecutive Δ_t-clusters before continuing to type 2 partitioning. This is because BAGEL is constantly observing subsequent Δ_t-clusters to determine if their cell datapoints contain any terminal states. If a subsequent Δ_t-cluster contain one or more terminal states, BAGEL operates in terminal state conditions. In these conditions the interval size of Δ_t for the current Δ_t-cluster is increased to the position of the terminal state cell datapoint with the largest pseudo-time value of the subsequent Δ_t-cluster. This increased interval is known as Δ_t-extended and is used to define an extended Δ_t-cluster. Next, an original Δ_t-cluster within the user defined interval of Δ_t is selected, after the extended Δ_t-cluster, to see if they contain any terminal states. This process of extending the Δ_t-cluster continues until all the terminal states, after a positive detection (subsequent Δ_t-cluster contains one or more terminal states), are detected or if a subsequent Δ_t-cluster does not contain a terminal state. It should be noted that in terminal state conditions the cell datapoints of a window is defined to be equal to the extended Δ_t-cluster and the process of type 2 partitioning is disregarded. When a terminal state is detected in the cell datapoints of subsequent Δ_t-clusters, the algorithm defines it as a final window, indicating a PC-lineage endpoint.

4.7.2.4 Tangent vectors of window-based Frenet frames

When all of the individual detected PC-lineages within space $\mathcal{P}$ are best represented with a series of sequential windows where Model 1 is their best fit, all of their obtained tangent vectors ($\mathrm{PC_w}1$) are concatenated. These concatenated vectors are used to define a tangent vector window-based Frenet frame of the entire cell developmental trajectory.

4.7.2.5 Window method operation recommendation

To ensure that the windows accurately represent the cell developmental trajectory of the obtained phenotypic manifold, the following intervals are recommended when defining *pseudo-time-interval* (Δ_t) and window (Δ_w):

$$\frac{\Delta_\mathrm{w}}{TC} \times 100 \approx 3\% - 6\%, \tag{4.2}$$

where TC is the total number of cell datapoints in the primary single-cell gene expression dataset and

$$\Delta_t \approx 1.5 \times \Delta_\mathrm{w}. \tag{4.3}$$

These intervals are based on the accurate results obtained when modelling the single-cell gene expression datasets of Table 5.2.

4.7.2.6 Window method algorithm

The process of selecting cell datapoints inside a window and computing a window-based Frenet frame is summarised in Algorithm 5.

Algorithm 5 Window method

```
Input: Dimensionality-reduced single-cell gene expression data (output of Algorithm 4).
Input: The Palantir algorithm pseudo-time ordering of cell datapoints (Appendix A.3).
Initialise: Number of cell datapoints inside $\Delta_t$ and the Number of cell datapoints inside $\Delta_\mathrm{w}$.
$\Delta_t$-cluster $\rightarrow$ Select the user defined number of cell datapoints inside the $\Delta_t$ boundary.
Next $\Delta_t$-cluster $\rightarrow$ Select the user defined number of cell datapoints inside the next $\Delta_t$ boundary.
while Next $\Delta_t$-cluster contains a terminal state == True do        ▷ Terminal state operation
    $\Delta_t$-extended boundary = Terminal state position.
    Extended $\Delta_t$-cluster $\rightarrow$ Select the cell datapoints inside the $\Delta_t$-extended boundary.
    Next $\Delta_t$-cluster $\rightarrow$ Obtain the next sequential $\Delta_t$-cluster after the extended $\Delta_t$-cluster.
end while
if Extended $\Delta_t$-cluster then        ▷ Terminal state operation
    Pseudo-time-scaling $\rightarrow$ Widen extend $\Delta_t$-cluster pseudo-time.
    $\overline{\mathrm{PC}_\mathrm{w}1}$ $\rightarrow$ Compute first principal component of widen extend $\Delta_t$-cluster.
    Tangent vector of window-based Frenet frame $\mathrm{PC}_\mathrm{w}1$ $\rightarrow$ un-widened $\overline{\mathrm{PC}_\mathrm{w}1}$.
    Window cell datapoints $\rightarrow$ Cell datapoints inside extended $\Delta_t$-cluster.
else
    Pseudo-time-scaling $\rightarrow$ Widen $\Delta_t$-cluster pseudo-time.
    $\overline{\mathrm{PC}_\mathrm{w}1}$ $\rightarrow$ Compute first principal component of widen $\Delta_t$-cluster.
    Tangent vector of window-based Frenet frame $\mathrm{PC}_\mathrm{w}1$ $\rightarrow$ un-widened $\overline{\mathrm{PC}_\mathrm{w}1}$.
    Window cell datapoints $\rightarrow$ Utilise $\mathrm{PC}_\mathrm{w}1$ to obtain cell datapoints inside the $\Delta_\mathrm{w}$ boundary.
end if
Delete obtained window cell datapoints from dataset to ease future computation.
Output: Window, tangent vector of window-based Frenet frame.
```

4.7.3 Detecting bifurcation points with Bayesian model selection

In essence, to estimate a bifurcation point, it is assumed that the cell datapoints of each window is Gaussian distributed. Therefore, a bifurcation point along the cell's developmental trajectory is detected by determining if the cell datapoints inside an obtained window is best represented with a single multivariate Gaussian distribution (Model 1) or with a mixture of two multivariate Gaussian distributions (Model 2). This is known as a model selection problem and it is solved with Bayesian model selection and implemented with a Gibbs sampler.

4.7.3.1 Model selection concept

To explain the process of how Bayesian model selection is used to estimate bifurcation points, two different visual illustrations are presented. As seen in Figure 4.7 and Figure 4.8, the first step in detecting a bifurcation point is to obtain a window (pink dashed line) of cell datapoints (red dots). The assumption of the window (in the case of this example) is that all of its cell datapoints are from one of two different Gaussian univariate distributions. Therefore, two different models (green arrows) are then applied to the cell datapoints of the window: (i) the first model is a single univariate Gaussian

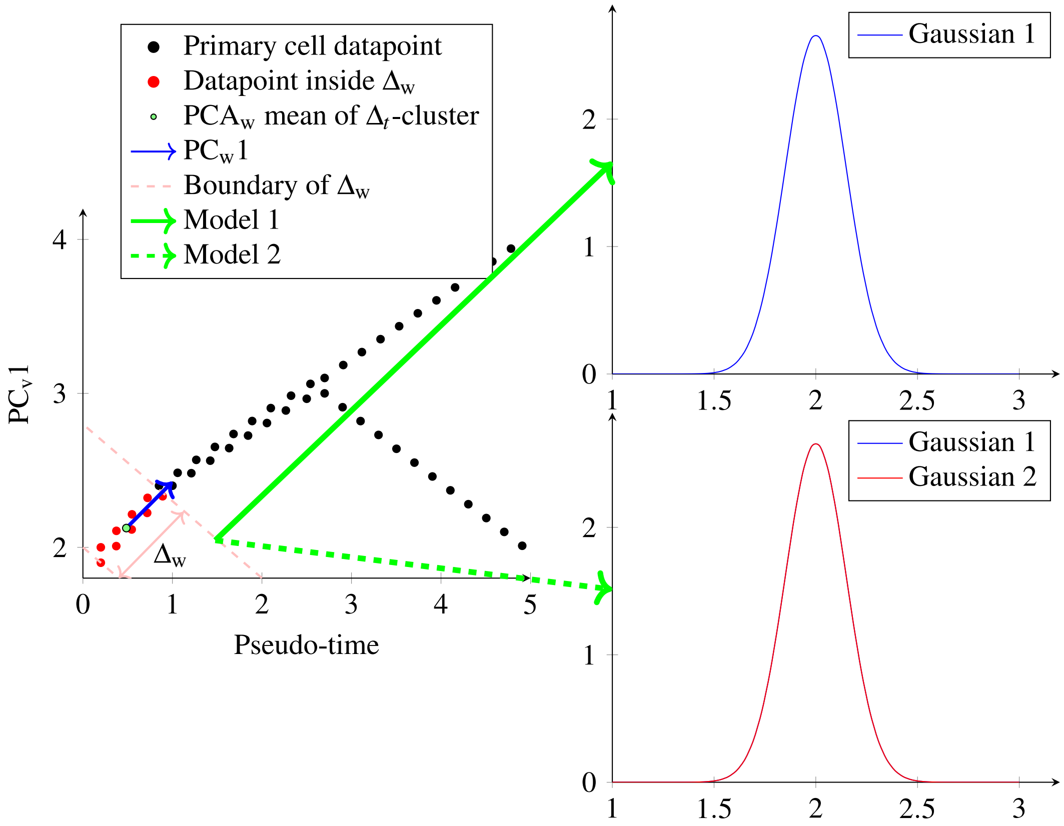


Figure 4.7. The first visual example to explain the process of how Bayesian model selection is used to estimate bifurcation points. As seen in these figures, the first step in detecting a bifurcation point is to obtain a window (pink dashed line) of cell datapoints (red dots). The assumption of the window (in the case of this example) is that all of its cell datapoints are from one of two different univariate Gaussian distributions. Therefore, two different models (green arrows) are then applied to the cell datapoints of the window: (i) the first model is a single univariate Gaussian distribution (top right); whereas (ii) the second model is a mixture of two univariate Gaussian distributions (bottom right). The model evidence for both these models is then estimated and compared using Bayes factor. In the case of this window, a single Gaussian distribution (top right) fit the data best indicating that there is no bifurcation.

distribution (top right); whereas (ii) the second model is a mixture of two univariate Gaussian distributions (bottom right). The model evidence for both these models is then estimated and compared using Bayes factor. In the case of Figure 4.7, a single Gaussian distribution (top right) fits the data best, indicating that there is no bifurcation. However, in the case of Figure 4.8, the mixture of two Gaussian distributions (bottom right) fits the data best indicating that there is a bifurcation point. Although this example uses a univariate assumption, BAGEL's implementation thereof is multivariate.

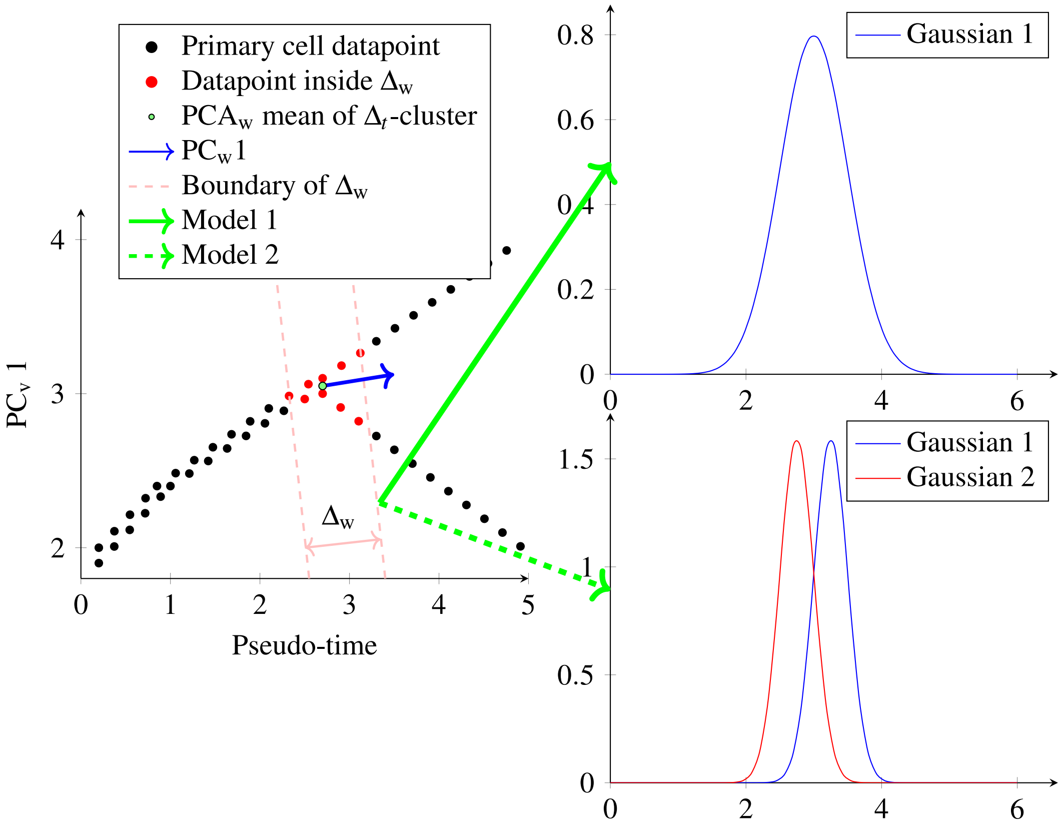


Figure 4.8. The second visual example to explain the process of how Bayesian model selection is used to estimate bifurcation points. As seen in these figures, the first step in detecting a bifurcation point is to obtain a window of cell datapoints (red dots). The assumption of the window (in the case of this example) is that all of its cell datapoints are from one of two different univariate Gaussian distributions. Therefore, two different models (green arrows) are then applied to the cell datapoints of the window: (i) the first model is a single univariate Gaussian distribution (top right); whereas (ii) the second model is a mixture of two univariate Gaussian distributions (bottom right). The model evidence for both these models is then estimated and compared using Bayes factor. In the case of this window, a mixture of two univariate Gaussian distributions (bottom right) fit the data best indicating that there is a bifurcation point.

4.7.3.2 Bayesian model selection of finite Gaussian mixture models

The purpose of model selection in BAGEL is to select a model that best fits the cell datapoints in a given window. This selection is accomplished by obtaining the evidence of each model defined in (3.20) with its estimate defined in (3.21), followed by calculating Bayes factor defined in (3.13). In order to estimate the BMI of (3.21), a posterior distribution of both a single multivariate Gaussian distribution (Model 1) and the mixture of two multivariate Gaussian distributions (Model 2) are required. Both posterior distributions for Model 1 and Model 2 are defined below with respect to a known dataset **x** with N datapoints confined to a combined pseudo-time-principal-component space (space $\mathcal{P}$) denoted by $\mathbf{x} = [\mathrm{PC_v1},\ \mathrm{PC_v2},\ \text{pseudo-time}]$.

4.7.3.2.1 MODEL 1 The posterior distribution of a single multivariate Gaussian distribution where the mean and covariance are unknown and assumed to be independent of each other can be defined as

$$p(\mu,\Sigma|\mathbf{x}) = \frac{p(\mathbf{x}|\mu,\Sigma)p(\mu)p(\Sigma)}{p(\mathbf{x})}, \tag{4.4}$$

where $p(\mathbf{x}|\mu,\Sigma)$ is the likelihood function defined by

$$p(\mathbf{x}|\mu,\Sigma) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{x}|\mu,\Sigma). \tag{4.5}$$

The conjugate prior to the likelihood function in (4.5) is a $\mathcal{NIW}$ distribution (see Section 3.6.2) defined in (3.7). As the prior is conjugate to the likelihood function, the posterior distribution will also be a $\mathcal{NIW}$ distribution defined as

$$p(\mu,\Sigma|\mathbf{x}) = \mathcal{NIW}(\mu,\Sigma|\mu_n,\kappa_n,\Lambda_n,\nu_n), \tag{4.6}$$

where the posterior hyperparameters are defined by

$$\mu_n = \frac{n\bar{\mathbf{x}} + \kappa_0\mu_0}{n+\kappa_0}, \tag{4.7}$$

$$\kappa_n = \kappa_0 + n, \tag{4.8}$$

$$\nu_n = \nu_0 + n, \tag{4.9}$$

$$\Lambda_n = \Lambda_0 + \mathbf{W} + \frac{n\kappa_n}{n+\kappa_n}(\bar{\mathbf{x}}-\mu_0)(\bar{\mathbf{x}}-\mu_0)^T, \tag{4.10}$$

with n being the total number of datapoints, $\bar{\mathbf{x}}$ being the datset mean and **W** being the sum of squares defined as

$$\mathbf{W} = \sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T. \tag{4.11}$$

Therefore, as the posterior distribution of (4.4) can be re-written as

$$\mathcal{NIW}(\mu,\Sigma|\mu_n,\kappa_n,\Lambda_n,\nu_n|\mathbf{x}) = \frac{[\prod_{n=1}^{N}\mathcal{N}(\mathbf{x}|\mu,\Sigma)]\mathcal{NIW}(\mu_0,\kappa_0,\Lambda_0,\nu_0)}{p(\mathbf{x})}, \tag{4.12}$$

the BMI of model 1 is defined as

$$p(\mathbf{x}) = \frac{[\prod_{n=1}^{N} \mathcal{N}(\mathbf{x}|\mu,\Sigma)]\mathcal{NIW}(\mu_0,\kappa_0,\Lambda_0,\nu_0)}{\mathcal{NIW}(\mu,\Sigma|\mu_n,\kappa_n,\Lambda_n,\nu_n)}. \tag{4.13}$$

4.7.3.2.2 MODEL 2 A mixture model of a two-component $K = 2$ multivariate Gaussian distribution can be described by the joint distribution of the latent variable and dataset such that

$$p(\mathbf{x},\mathbf{z}|\pi,\mu_1,\Sigma_1,\mu_2,\mu_2) = p(\mathbf{x}|\pi,\mu_1,\Sigma_1,\mu_2,\mu_2,z)p(\mathbf{z}|\pi). \tag{4.14}$$

As the number of components of the mixture model is known, the mixing proportion π is defined as

$$\sum_{K} \pi_k = 1 = \pi_1 + \pi_2,$$

which implies that

$$\pi_1 = 1 - \pi_2. \tag{4.15}$$

The relationship between the mixing portions in (4.15) can be used to define the distribution of the latent variable in (3.53) as

$$p(\mathbf{z},\theta) = (\pi_1)^{z_1}(1-\pi_1)^{z_2}. \tag{4.16}$$

The mixing portion relationship can also be used to define a likelihood function as

$$p(\mathbf{x},\mathbf{z}|\pi,\mu_1,\Sigma_1,\mu_2,\mu_2) = \prod_{i=1}^{N}[(\pi_1)\mathcal{N}(x_i|\mu_1,\Sigma_1)]^{z_{1,i}}[(1-\pi_1)\mathcal{N}(x_i|\mu_2,\Sigma_2)]^{z_{2,i}}. \tag{4.17}$$

The conjugate priors of the likelihood function are a Dirichlet distribution for the latent variable and a $\mathcal{NIW}$ distribution for the mean and covariance. As the prior is conjugate to the likelihood function, the posterior distribution will also be the product of a Dirichlet distribution and two $\mathcal{NIW}$ distributions. Therefore, the posterior distribution that consists of a mixture of two multivariate Gaussian distributions where the mean, covariance and mixing coefficient are unknown and assumed to be independent can be defined as

$$p(\pi,\mu_1,\Sigma_1,\mu_2,\Sigma_2|\mathbf{x},\mathbf{z}) = \frac{\prod[p(\mathbf{x}|\pi,\mu_1,\Sigma_1,\mu_2,\mu_2,\mathbf{z})p(z|\pi)]}{p(x)} \times \frac{p(\pi|\alpha)p(\mu_1,\Sigma_1|\mu_0,\kappa_o,\Lambda_0,\nu_0)p(\mu_2,\Sigma_2|\mu_0,\kappa_o,\Lambda_0,\nu_0)}{p(x)} \tag{4.18}$$

$$= p(\pi|\mathbf{x},\mathbf{z},\mu_1,\Sigma_1,\mu_2,\Sigma_2)p(\mu_1,\Sigma_1|\mathbf{x},\mathbf{z},\pi,\mu_2,\Sigma_2)\times p(\mu_2,\Sigma_2|\mathbf{x},\mathbf{z},\pi,\mu_1,\Sigma_1), \tag{4.19}$$

where $p(\pi|\mathbf{x},\mathbf{z},\mu_1,\Sigma_1,\mu_2,\Sigma_2)$ is a Dirichlet distribution defined as

$$p(\pi|\mathbf{x},\mathbf{z},\mu_1,\Sigma_1,\mu_2,\Sigma_2) = Dir(\alpha_1+n_1,\alpha_2+n_2), \tag{4.20}$$

with a hyperparameter α and a parameter n_k, which defines the total number of datapoints belonging to the kth component and $p(\mu_k,\Sigma_k|\mathbf{x},\mathbf{z},\pi,\mu_{\neq k},\Sigma_{\neq k})$ is a $\mathcal{NIW}$ distribution of the kth component defined as

$$p(\mu_k,\Sigma_k|\mathbf{x},\mathbf{z}\pi,\mu_{\neq k},\Sigma_{\neq k}) = \mathcal{NIW}((\mu_k,\Sigma_k|\mu_n,\kappa_n,\Lambda_n,\nu_n), \tag{4.21}$$

with hyperparameters

$$\mu_n = \frac{n_k\overline{\mathbf{x}_k}+\kappa_0\mu_0}{n+\kappa_0}, \tag{4.22}$$

$$\kappa_n = \kappa_0 + n_k, \tag{4.23}$$

$$\nu_n = \nu_0 + n_k, \tag{4.24}$$

$$\Lambda_n = \Lambda_0 + \mathbf{W}_k + \frac{{}_k n\kappa_n}{n_k+\kappa_n}(\overline{\mathbf{x}}_k-\mu_0)(\overline{\mathbf{x}}_k-\mu_0)^T, \tag{4.25}$$

and where $\mathbf{W}_k$ is the sum of squares of the kth component defined as

$$\mathbf{W}_k = \sum_{i\in k}^{n}(\mathbf{x}_i-\overline{\mathbf{x}}_k)(\mathbf{x}_i-\overline{\mathbf{x}}_k)^T. \tag{4.26}$$

Due to the posterior definition of (4.18), the BMI of model 2 can be defined as

$$p(\mathbf{x}) = \frac{\prod[p(\mathbf{x}|\pi,\mu_1,\Sigma_1,\mu_2,\mu_2,z)p(z|\pi)]p(\pi)p(\mu_1,\Sigma_1)p(\mu_2,\Sigma_2)}{p(\pi|\mathbf{x},\mathbf{z},\mu_1,\Sigma_1,\mu_2,\Sigma_2)\mathcal{NIW}(\mu_k,\Sigma_k|\mu_n,\kappa_n,\Lambda_n,\nu_n)}. \tag{4.27}$$

4.7.3.3 Summary of model selection parameters

In BAGEL the input dataset vector $\mathcal{D}$ is

$$\mathcal{D} = [\text{pseudo-time, PC}_\text{v}1,\ \text{PC}_\text{v}2]^T \tag{4.28}$$

The parameter vector $\theta_{\text{Model 1}}$ for Model 1, a single multivariate Gaussian distribution, is defined with

$$\theta_{\text{Model 1}} = [\mu,\ \Sigma]^T, \tag{4.29}$$

where μ is the mean and Σ is the covariance. The parameter vector $\theta_{\text{Model 2}}$ for Model 2, which is a mixture of two multivariate Gaussian distributions, is defined with

$$\theta_{\text{Model 2}} = [\pi,\ \mu_1,\ \Sigma_1,\ \mu_2,\ \Sigma_2]^T, \tag{4.30}$$

where π is the mixing coefficient, μ is the mean, Σ is the covariance and the subscript indicates to which one of the two mixture component the parameters belong.

4.7.3.4 Bifurcation point detection algorithm

A bifurcation point within space $\mathcal{P}$ is detected by utilising model selection to compare the model evidence of two distinct models on cell datapoints inside a window. As these evidences are difficult to calculate, the BMI of (3.21) with a Gibbs sampler (Algorithm 1) is used to estimate them. This estimation process is summarised in Algorithm 6. In Algorithm 6, the superscript t indicates the iteration of the Gibbs sampler, the subscript k indicates the mixture model component, $\mathbf{x}$ is the cell datapoints inside the window and the subscript i is a pointer to individual cell datapoints.

Algorithm 6 Bifurcation point detection via a Gibbs sampler and Bayesian model selection

```
Input: Cell datapoints inside a window
Initialise: The hyperparameters $\mathcal{H}^{(0)} = (\alpha^{(0)}, \mu_0^{(0)}, \kappa_0^{(0)}, \Gamma_0^{(0)}, \nu_0^{(0)})$, the mixing proportions $\pi^{(0)} = (\pi_1^{(0)}, 1 - \pi_1^{(0)})$, and the component parameters $\theta_k^{(0)} = (\mu^{(0)}, \Sigma^{(0)})$.
Initialise: Model 1 evidence (M1E) = 0 and Model 2 evidence (M2E) = 0.
for $t = 1$ until total number of Gibbs samples do        ▷ See Section 5.4 about recommendations
    $\Sigma^{(t)} | \mu^{(t-1)}, \mathbf{x} \sim inverse - Wishart(\nu_n, \Lambda_n)$        ▷ Model 1 covariance
    $\mu^{(t)} | \Sigma^{(t)}, \mathbf{x} \sim \mathcal{N}(\mu_n, \frac{\Sigma}{\kappa_n})$        ▷ Model 1 mean
    for $k = 1$ to $K$ do
        $z_i^{(t)} | \tau_{i,1}^{(t)}, \pi_k^{(t-1)} \theta_k^{(t-1)} \sim \mathcal{M}(1 : \tau_{i,1}^{(t)}, \dots, \tau_{i,K}^{(t)})$, where $\tau_{i,k} =$ (3.60)        ▷ Model 2 label
    end for
    $\pi^{(t)} | \tau_{i,k}^{(t)}, \mu_k^{(t-1)}, \Sigma_k^{(t-1)}, \mathbf{x} \sim Dir(\alpha_1 + n_1, \dots, \alpha_K + n_K)$        ▷ Model 2 mixing proportions
    for $k = 1$ to $K$ do
        $\Sigma_k^{(t)} | \tau_{i,k}^{(t)}, \pi_k^{(t)}, \mu_k^{(t-1)}, \mathbf{x} \sim inverse - Wishart(\nu_n, \Lambda_n)$        ▷ Model 2 covariance
        $\mu_k^{(t)} | \tau_{i,k}^{(t)}, \pi_k^{(t)}, \Sigma_k^{(t)}, \mathbf{x} \sim \mathcal{N}(\mu_n, \frac{\Sigma}{\kappa_n})$        ▷ Model 2 mean
    end for
    if $t >$ Burn-in period then        ▷ After Markov chain convergence
        BMI_1 → Compute BMI of Model 1 with (4.13)        ▷ BMI of Model 1 samples
        M1E = M1E + BMI_1        ▷ Sum BMI of Model 1
        BMI_2 → Compute BMI of model 2 with (4.27)        ▷ BMI of Model 2 samples
        M2E = M2E + BMI_2        ▷ Sum BMI of Model 2
    end if
end for
M1E = M1E / ($t$ - burn-in period)        ▷ Estimate of Model 1 evidence
M2E = M2E / ($t$ - burn-in period)        ▷ Estimate of Model 2 evidence
Bayes factor = M2E / M1E        ▷ Compute Bayes factor
if Bayes factor $> 1$ then        ▷ Bayes factor evaluation based on Table 3.2
    Model 2 → TRUE        ▷ Bifurcation point detected
else
    Model 1 → TRUE        ▷ No bifurcation point detected
end if
Output: Model 1, Model 2, Model 2 mixing proportions (if applicable)
```

4.7.4 Bifurcation validation

The accuracy in bifurcation point detection is improved by implementing local and global validation methods. These validation methods are used to verify detected bifurcation points before defining it as a bifurcation point within the cell developmental trajectory.

4.7.4.1 Local validation

The first validation method validates the cell datapoints of a window when the window is positively identified as representing a bifurcation point (Model 2). A false detection of a bifurcation point may occur when there is a sudden change in a cell's developmental trajectory captured by the cell datapoints of a window. Consider the case in Figure 4.9 where there is a sudden change in the cell's developmental trajectory captured by the window cell datapoints (red dots). When applying model selection of multivariate Gaussian distributions to the cell datapoints of this window, the sudden changes in the cell's developmental trajectory may lead to the cell datapoints being best represented with two distinct Gaussian distributions (green en blue ellipses). This is a false bifurcation point detection of the cell datapoints of a window and is known as a local false detection.

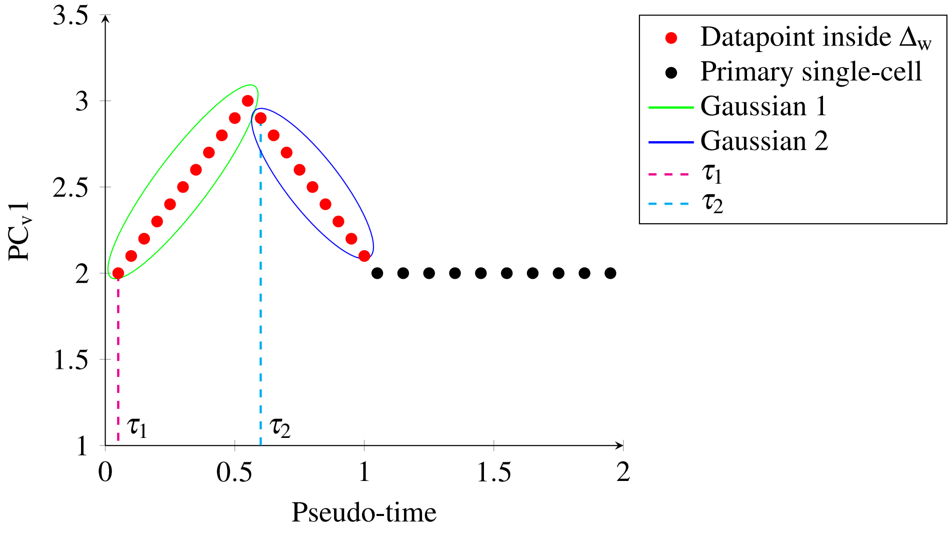


Figure 4.9. Local false detection of a bifurcation point due to when a quick and sudden change in a cell's developmental trajectory is captured by the cell datapoints (red dots) of a window. As seen when applying a multivariate model selection to the cell datapoints of this window, the sudden changes may lead to the window being best represented with two distinct multivariate Gaussian distributions (green and blue ellipses). As seen, these incorrect distributions (in terms of bifurcation) can be identified as the starting point of PC-lineage-1 (τ_1) and PC-lineage-2 (τ_2) are not in close proximity to each other.

To mitigate a local false detection, an assumption is made about bifurcation points. The assumption is that when there is a true bifurcation point captured by the cell datapoints of a window, both of the detected PC-lineages will bifurcate at the same pseudo-time. This means that the starting point of PC-lineage-1 within the window should approximately be the starting point of PC-lineage-2 within the window (*PC-lineage-1 starting point* $\approx$ *PC-lineage-2 starting point*). When both PC-lineages start at exactly the same instance the ratio of their pseudo-time starting points will be approximately one. Therefore, for a bifurcation point to pass its local validation, it should satisfy the following:

$$\frac{\tau_1}{\tau_2} > \frac{2}{3}, \tag{4.31}$$

where τ_1 is the starting point pseudo-time of PC-lineage-1 within the window and τ_2 is the starting point pseudo-time of PC-lineage-2 within the window. In (4.31) it is always ensured that $\tau_2 \geq \tau_1$ to satisfy the ratio of $\frac{2}{3}$. The ratio of $\frac{2}{3}$ is chosen based on the following two intuitions about bifurcation (i) if the ratio of (4.31) is equal to one, it means that the pseudo-time starting point for each PC-lineage is exactly the same instance; and (ii) if the ratio of (4.31) is very small the pseudo-time starting point for each PC-lineage is at opposite extremes of the window[10]. Hence, if the ratio of (4.31) is greater and equal to $\frac{2}{3}$ the starting point pseudo-time of each PC-lineage is assumed to be in close proximity to each other.

4.7.4.2 Global validation

The second method globally validates bifurcation points by observing subsequent windows that have passed their local validation. BAGEL implements the majority voting rule and defines a bifurcation point only when three consecutive windows select Model 2 as their best fit. However, there is an exception to this rule when BAGEL operates in terminal state conditions (see Section 4.7.2.3 regarding terminal state conditions). The two possible scenarios for this exception is that (i) if only one window selects Model 2 as their best fit and it is the final window, a bifurcation point is detected; or (ii) if there are two consecutive windows, where the latter is the final window, and both of them selects Model 2 as their best fit, a bifurcation point is detected. This rule is due to the terminal states of each PC-lineages in space $\mathcal{P}$ not ending at the exact same instance. After a bifurcation is detected and there is still unassociated data, the association algorithm in Section 4.7.5 mitigates it.

4.7.5 Data association

As Bayesian model selection only associates the cell datapoints of a window with one or a mixture of two multivariate Gaussian distributions and not which PC-lineage they belong to, a simple association algorithm is required. The association algorithm is divided into three parts namely (i) data association

[10] Assuming a non-zero τ_2. which would cause (4.31) to be undefined.

before a bifurcation point; (ii) data association with a bifurcation point; and (iii) data association after a bifurcation point. The last two of these association algorithms use Euclidean distance (see Section 3.3) for association.

4.7.5.1 Bifurcation validation and data association flow diagram

A flow diagram of how bifurcation validation is used to define (i) a bifurcation point within the cell developmental trajectory; and (ii) associate cell datapoints to distinct PC-lineages is shown in Figure 4.10. As seen, the process of bifurcation validation starts (S.1) at the model selection (D.1) of the nth window (where n is a global window counter) by estimating the best fit for its cell datapoints: Model 1 (orange M1) and Model 2 (light blue M2). When Model 2 is the best fit for the cell datapoints of the nth window, local validation is applied (LV.1). If local validation (LV.1) is passed a counter m, that keeps track of how many sequential local validations have been passed for the purpose of global validation, is incremented by one. After incriminating m the mixture components of the nth window is saved (SV.1) for the purpose of bifurcation detection, before continuing to the fundamental question (D.3) of global validation[11] (grey dotted box) which is: is $m = 3$? When m is fewer than three, global validation is not achieved and the next sequential nth window is evaluated by incrementing n and restarting the process from S.1.

However, when Model 1 (orange M1) is the best fit for the cell datapoints of the nth window (D.1), or when local validation (LV.1) is failed (orange F), the flow diagram enters a loop called PC-lineage-1 loop association (purple dashed loop). When this loop is entered it means that global validation has failed, and the process of searching for three sequential windows where model 2 is their best fit need to start over. Hence, in this loop the current nth window, as well as all the saved mixture components of up to two sequential windows, are associated to a distinct PC-lineage called PC-lineage-1 by (i) associating the $(n-m)$th window to PC-lineage-1 (A.1); (ii) subtracting one iteration from m to iterate through all of the saved mixture components of the sequential windows, one saved window at a time; and (iii) repeating the process until $m = -1$, which indicates that the current nth window as well as all the saved mixture components of up to two sequential windows have been associated to PC-lineage-1. After the loop is completed, the counter m is reinitialised to zero and the next sequential nth window is evaluated by incrementing n and restarting the process from S.1.

[11] When BAGEL operates in terminal state conditions (see Section 4.7.2.3) the total number of windows in global validation may be fewer then three. However, the process of data association (A.2) remains the same and is only limited to the step matching the total number of windows available for association (see Section 4.7.5.3).

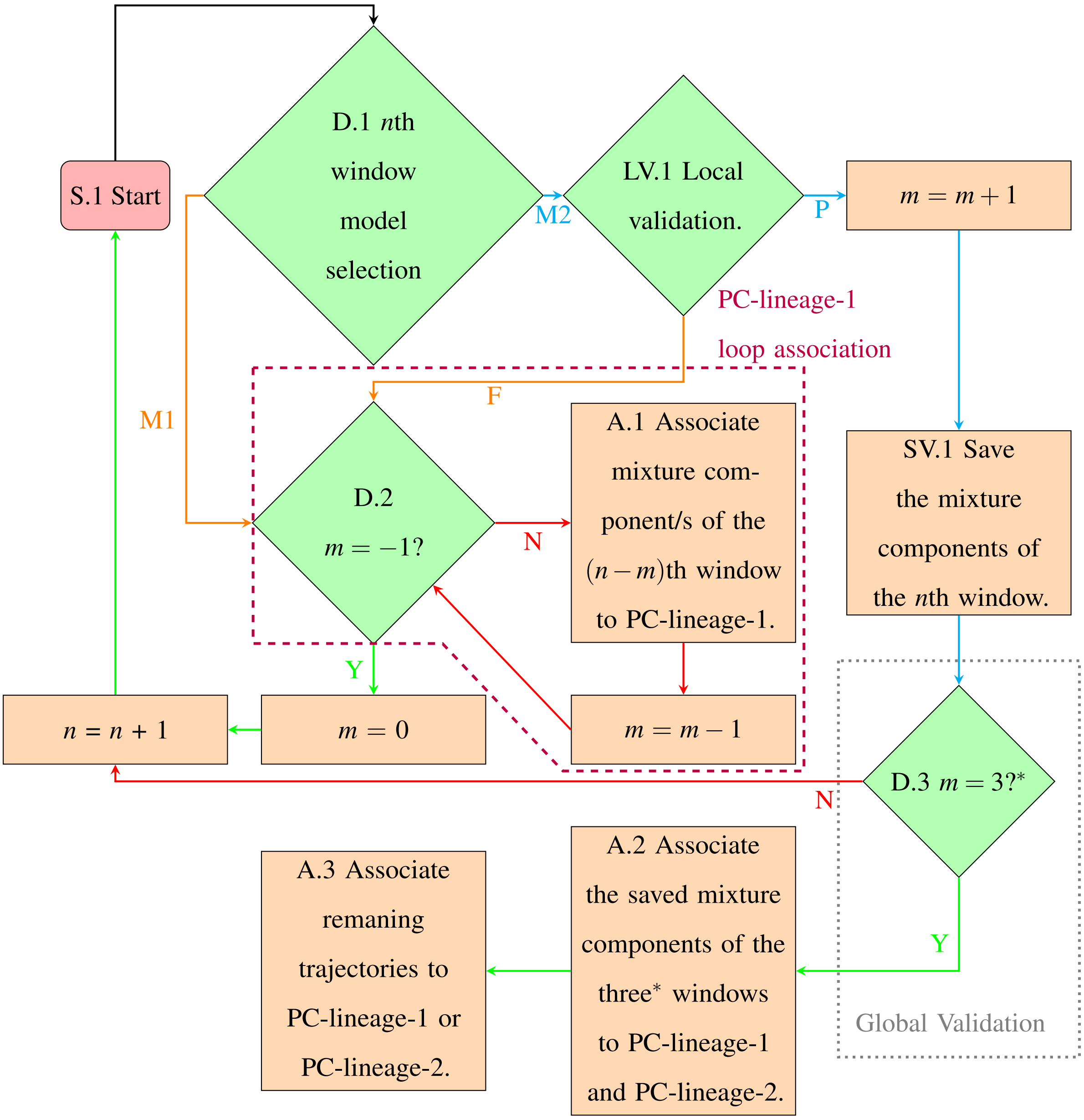

S.1 Start
D.1 nth window model selection
M2
LV.1 Local validation.
P
$m = m+1$
PC-lineage-1 loop association
M1
F
D.2 $m = -1$?
N
A.1 Associate mixture component/s of the $(n-m)$th window to PC-lineage-1.
SV.1 Save the mixture components of the nth window.
Y
$n = n + 1$
$m = 0$
$m = m-1$
D.3 $m = 3$?*
N
A.3 Associate remaning trajectories to PC-lineage-1 or PC-lineage-2.
A.2 Associate the saved mixture components of the three* windows to PC-lineage-1 and PC-lineage-2.
Y
Global Validation

Figure 4.10. Flow diagram of the process by which bifurcation validation, bifurcation detection and data association are achieved. In this flow diagram both loop-variables are initialised as $n = 0$ and $m = 0$. The variable n is the global window counter, and m keeps track of how many sequential local validations have been passed. *When BAGEL operates in terminal state conditions (see Section 4.7.2.3) the total number of windows in global validation may be fewer then three. However, the process of data association (A.2) remains the same and is only limited to the step matching the total number of windows available for association (see Section 4.7.5.3).

When global validation is passed (D.3, $m = 3$), the saved mixture components of the three sequential windows are associated to one of two distinct PC-lineages namely PC-lineage-1 and PC-lineage-2 by utilising A.2 (see Section 4.7.5.3). During the association of A.2, the position of the detected bifurcation point in space $\mathcal{P}$ is defined at the maximum pseudo-time value of the window preceding the saved windows as well as, its $\mathrm{PC_v}1$-$\mathrm{PC_v}2$ plane mean value. Finally, after the association of the saved mixture components, PC-lineage-1 and PC-lineage-2 may not describe their respective PC-lineages all the way from start to finish. This phenomenon occurs, as bifurcation detection is only interested in associating PC-lineages up to the saved mixture components. Hence, A.3 (see Section 4.7.5.4) is utilised to mitigate the unassociated cell datapoints. After associating the unassociated cell datapoints and defining the two distinct PC-lineages, each of the two PC-lineages are used as the input dataset to S.1. This step refines the defined PC-lineages to ensure that they do not contain any additional bifurcation points after bifurcating. This process continues until no bifurcation points are detected in any defined PC-lineages (all of the individual defined PC-lineages are best represented with a series of sequential windows where Model 1 is their best fit) implying that all of the possible PC-lineages in space $\mathcal{P}$ have been detected.

4.7.5.2 Data association before bifurcation point

When Bayesian model selection selects Model 1 as the best fit for the cell datapoints of the current window, all of the cell datapoints from that window are associated with one distinct PC-lineage called PC-lineage-1. When global validation is failed, PC-lineage-1 will consist of a series of sequential windows that represent the entire input dataset where Model 1 is their best fit.

4.7.5.3 Data association at bifurcation point

When a global validation is passed and a bifurcation point is detected, BAGEL defines the input dataset as containing two distinct PC-lineages namely PC-lineage-1 and a new PC-lineage-2 by (i) duplicating PC-lineage-1, which consists of a series of sequential windows of the input dataset where Model 1 is their best fit; and (ii) associating the saved mixture components of the three sequential windows[12] to the PC-lineage they most likely represent. This association process is achieved by sequentially executing the following steps.

- Step 1: Associate the first saved mixture component of the first sequential to PC-lineage-1.

[12] When BAGEL operates in terminal state conditions (see Section 4.7.2.3) the total number of windows in global validation may be fewer then three. However, the process of data association (A.2) remains the same and is only limited to the step matching the total number of windows available for association (see Section 4.7.5.3).

- Step 2: Associate the second saved mixture component of the first sequential to PC-lineage-2 and move to the next sequential window.
  - Step 3: Obtain the mean of each mixture component from the previous sequential window, which was associated to PC-lineage-1 and PC-lineage-2 respectively.
  - Step 4: Obtain the mean of each mixture component of the current sequential window.
  - Step 5: Compute the Euclidean distance between each of the current window's mixture component means and the mixture component means of the previous window.
  - Step 6: Associate the saved mixture components of the current window to the PC-lineage, whose previous mixture component mean their mean had the shortest Euclidean distance to.
  - Step 7: Move to next sequential window and repeat from step 3 until there are no more unassociated saved mixture components.
- Step 8: Define the position of the detected bifurcation point in space $\mathcal{P}$ at the maximum pseudo-time value of the window preceding the saved mixture components of the three sequential windows as well as its $\mathrm{PC_v}1$-$\mathrm{PC_v}2$ plane mean value.

4.7.5.4 Data association after bifurcation point

After the association of the saved mixture components, PC-lineage-1 and PC-lineage-2 may not describe their respective PC-lineages from start to finish. This phenomenon occurs, as bifurcation detection is only interested in associating PC-lineages until the end of the three windows. Hence, the remaining cell datapoints after the saved mixture components need to be associated to one of the two PC-lineages. BAGEL resolves this by iterating through the remaining cell datapoints and associating them to their most probable PC-lineage, one cell datapoint at a time. It is assumed that the cell datapoints of a PC-lineage are in close proximity to each other when their pseudo-time values are similar. Therefore, the process starts by calculating the Euclidean distance of the unassociated cell datapoint to the mean of the last 50 cell datapoints of each PC-lineage. A 50 cell cluster is chosen as it is assumed that these cell datapoints will have similar pseudo-times when the dataset is quite large (> 4000 cell datapoints see Table 5.2). The cell datapoint is then associated to the lineage with the shortest distance between the given PC-lineage mean and the cell datapoint. The last 50 cell datapoints of each PC-lineage is then updated after association and the process repeated until all the cell datapoints have been associated to one of the two PC-lineages.

4.7.6 Gaussian process

BAGEL models cell differentiation as a continuous process in space $\mathcal{P}$ by applying a Gaussian process to each obtained PC-lineage. To model cell differentiation as a continuous process in three dimensions, BAGEL concatenates two two-dimensional Gaussian processes with the weight parameter vector of (3.32) set equal to $\mathbf{w} = [1,\ 256,\ 0,\ 0]$. The values of the weight parameter vector was obtained by tuning the Gaussian process. The first Gaussian process used for concatenation, models the dimensional reduced axis $\mathrm{PC_v}1$ and pseudo-time, whereas the second Gaussian process used for concatenation, models the dimensional reduced axis $\mathrm{PC_v}2$ and pseudo-time. Hence, in these two defined Gaussian processes, pseudo-time represents the independent input vector $\mathbf{x}$ and the reduced dimensions of the single-cell gene expression dataset ($\mathrm{PC_v}$) represents the function $f(\mathbf{x})$. This process of continuous modelling is summarised in Algorithm 7.

Algorithm 7 Continuous modelling of PC-lineages

```
Input: All detected PC-lineages.
for $t = 1$ to total number of PC-lineages do        ▷ Iterate through all of the PC-lineages.
    Model $\mathrm{PC_v}1$ and pseudo-time of PC-lineage $t$ utilising Algorithm 2.
    Model $\mathrm{PC_v}2$ and pseudo-time of PC-lineage $t$ utilising Algorithm 2.
    Concatenate the two continuous models.
    Save continuous three-dimensional model of PC-lineage $t$.
end for
Output: Continuous model
```

4.8 ALGORITHM ASSUMPTIONS AND LIMITATIONS

As gene expression modelling is a complex task, BAGEL has to make some assumptions based on (i) BAGEL not having prior knowledge about its input single-cell gene expression dataset/s; and (ii) that most of the sequential steps of BAGEL are dependent on each other. Hence, BAGEL's assumptions are:

- the developed phenotypic manifold accurately represents the true underlying phenotypic manifold of the input single-cell gene expression dataset/s to BAGEL (see Section 4.6);
- the cell-by-gene matrix of counts contains an adequate number of primary single-cell gene expression datapoints (cells > 4000 and genes > 2000) (see Section 4.6.1). This assumption is

based on the datasets that are used in this dissertation (Section 5.2) as they produced credible results;

- the projection of the secondary dataset onto the primary dataset is biologically accurate (see Section 4.6.1);
- the user-defined intervals of Δ_t and Δ_w can be used to obtain a window of cell datapoints which is "sliced" perpendicular to the cell developmental trajectory (see Section 4.7.2);
- obtained sequential windows represent a time series of the cell developmental trajectory in space $\mathcal{P}$ (see Section 4.7.2);
- all the possible terminal states within the single-cell gene expression dataset/s are identified by the Palantir algorithm (see Section 4.7.2);
- the single-cell gene expression dataset/s differentiates at most into two distinct branches at a given instance along its developmental trajectory in space $\mathcal{P}$ (see Section 4.7.3); and
- the cell datapoints of a PC-lineage are in close proximity to each other when their pseudo-time values are similar (see Section 4.7.5).

The limitations of the algorithm due to these assumptions and the nature of biology are:

- there is no mitigation method for when the user-defined intervals of Δ_t and Δ_w are inaccurate;
- there is no mitigation method when a sequential step within BAGEL (Figure 4.2) produces an inaccurate output;
- the instantaneous detection of three or more bifurcations within space $\mathcal{P}$ is not possible;
- the primary single-cell gene expression dataset should contain an increased number of differentiated cells compared to the secondary single-cell gene expression dataset when projecting the secondary dataset onto the primary dataset (see Chapter 6); and
- the primary and secondary single-cell gene expression datasets should express the similar differentiation precursors (see Chapter 6).

CHAPTER 5 SINGLE-CELL GENE EXPRESSION DATA AND EXPERIMENTAL SETUP

5.1 CHAPTER OVERVIEW

This chapter serves as an introductory section to the observed modelling results of Chapter 6. The chapter starts by providing basic knowledge about the three different haematopoietic single-cell gene expression datasets modelled in this dissertation. The discussion about the single-cell gene expression data is followed by the experimental setup of the data collection procedure.

5.2 HAEMATOPOIESIS SINGLE-CELL GENE EXPRESSION DATA

Three different single-cell gene expression datasets are used to illustrate the functional capabilities of BAGEL. The first dataset is Lin–c-Kit+Sca-1+ mouse bone marrow single-cell RNA-seq data derived from [87] consisting of 4423 cells and a total of 2312 genes expressed. The genes in this dataset correspond to cells within the erythroid and myeloid lineages [6]. The second dataset is CD34+ human bone marrow single-cell RNA-seq data consisting of 4142 cells, with a total of 16106 genes expressed [1]. This human dataset will henceforth be referred to as human dataset 1. The third dataset is a dataset from the Faculty of Health Science, which is single-cell RNA-seq data from human umbilical cord blood (UCB). The UCB was obtained from mothers undergoing caesarians at a private hospital in Pretoria, South Africa. CD34+ HSPCs were purified by fluorescent activated cell sorting (FACS) using a FACSAria cell sorter (BD Biosciences, New Jersey, USA) and were directly sorted into C1™ Single-Cell Auto Prep Array Integrated Fluidics Circuit (IFC) plates (Fluidigm, San Francisco, California, USA). Capture, lysis, reverse transcription and amplification of single-cells were performed using the Fluidigm C1™ Single-Cell Auto Prep System (Fluidigm, San Francisco, California, USA) according to the user manual. Library preparation and sequencing were performed by the Agricultural Research Council (ARC) at Onderstepoort Veterinary Research Campus, Onderstepoort, Gauteng, South Africa, using the Nextera XT DNA Library Preparation Kit (Illumina, San Diego, California, USA) and Illumina Hiseq 2500 sequencing system (Illumina, San Diego, California, USA) as per the manufacturer's instructions. The average sequencing depth was $3.67x10^6$ reads per cell.

Trimmomatic was used to remove sequencing adapters and poor-quality reads [88]. The bcbio-nextgen workflow [89] was used to assess the quality of the data using FASTQC; align reads to the human reference genome (GRCh38) using the alignment tool HISAT2 [90]; and perform gene quantification using featureCounts [91].

The use of different datasets require a measurement that determines the correlation between the different datasets. BAGEL uses a percentage of gene overlap between the datasets to determine their correlation. As the datasets used in this dissertation are summarised in Table 5.2 the percentage of gene overlap between the datasets is defined as

$$\text{PO} = \frac{R_x}{C_y} \times 100, \tag{5.1}$$

where R_x is the total number of genes from row x, and C_y is the total number of genes from column y. As seen, in (5.1) the percentage overlap therefore shows the correlation in gene expression between datasets. An example on how to interpret the calculated percentage overlap represented in Table 5.2 is shown with an arbitrary row and column in Table 5.1.

Table 5.1. Visualisation of percentage overlap (5.1).

	Column$_1$	**Column$_2$**	**Column$_3$**
row$_1$			
row$_2$		$\frac{R_2}{C_2} \times 100$	
row$_3$			

Table 5.2. Comparison between single-cell gene expression datasets.

	Mouse dataset	**Human dataset 1**	**Human dataset 2**
Source	Bone marrow	Bone marrow	Umbilical cord blood
Cell type	Lin-cKit+Sca-1+	CD34+	CD34+
Type of data	scRNA-seq	scRNA-seq	scRNA-seq
Sequencing depth	Average of 78,682 reads per cell	-	Average of 3.67 million reads per cell
Number of cells	4423	4142	168
Possible genes	2312	16106	28857
Percentage of overlap with Mouse	100%	12.15%	6.98%
Percentage of overlap with Human 1	84.64%	100%	49.22%
Percentage of overlap with Human 2	87.11%	88.19 %	100%

5.3 WHY IS THE HUMAN UCB SINGLE-CELL GENE EXPRESSION DATASET REFERRED TO AS BEING SUB-SAMPLED?

The Fluidigm C1 system was used to capture a total of 266 single CD34+ HSPCs from UCB from six individual donors. The mean capture efficiency was 46% (±12.1%). The Fluidigm C1 system allows capture of a maximum of 96 cells at a time and with a capture efficiency of 46% on average for the HSPCs, it has been challenging to increase the cell numbers in order to be competitive in the single-cell field. The high cost associated with the consumables for this system is another limitation, especially if experiments need to be repeated in order to reach the cell numbers required in single-cell publications. The various quality control measures along the process further resulted in a decreased total number of cells analysed.

5.4 GIBBS SAMPLES AND BURN-IN PERIOD

In practice there is not a recommend total number of Gibbs samples and a burn-in period when using a Gibbs sampler, as the optimal number of samples is problem specific [17]. In the case of BAGEL, a Gibbs sampler is used to estimate a model evidence whose accuracy is directly proportional to the total number of samples as seen in (3.25) and (3.26). This proportional relationship will intuitively lead to selecting a large number of samples for the Gibbs sampler. However, when a Gibbs sampler was utilised during model selection (see Section 4.7.3) it was noted that an increase in the total number of

Gibbs samples had a negligible effect on the outcome of model selection. It was found that when the data points within a window is best represented with Model 1 or Model 2 respectively, an increase in the number of Gibbs samples will not change the overall outcome. Therefore, the number of Gibbs samples was set to 2000 samples with a burn-in period of 500 samples to allow the Markov chain to converge.

5.5 EXPERIMENTAL SETUP FOR DATA COLLECTION

Data collection of BAGEL is simple, as the modelling of cell differentiation is analogous to a black box model. This analogy is because BAGEL allows several input parameters and converts them to a mathematical representation of single-cell gene expression data. The input parameters are (i) dataset/s (note: the format of a high dimensional dataset should be a cell-by-gene matrix of counts); (ii) early cell[1]; (iii) Δ_t; and (iv) Δ_w. As there are three primary experiments [2], three different setups were required as seen in Table 5.3:

Table 5.3. Experimental setup.

	Result 1	**Result 2**	**Result 3**
Primary dataset	Mouse dataset	Human dataset 1	Human dataset 1
Secondary dataset	-	-	Human dataset 2
Early cell	W30258	Run5_164698952452459	Run5_164698952452459
two_data_set_FLAG [a]	False	False	True
new_manifold_FLAG [b]	True	True	True
Δ_t	200	200	200
Δ_w	150	150	150
Gibbs samples	2000	2000	2000
Burn-in period	500	500	500

[a] two_data_set_FLAG: Flag defining whether one or two datasets are used..

[b] new_manifold_FLAG: This flag defines if a new Palantir phenotypic manifold should be developed. This flag can be set to false after the first iteration of the algorithm as the phenotypic manifold only has to be defined once for operation.

[1] Early cell: Defines a cell at the start of the cell differentiation process required by the Palantir algorithm [1].

[2] The fourth experiment is shown in Appendix B

CHAPTER 6 RESULTS AND DISCUSSION

6.1 CHAPTER OVERVIEW

BAGEL is capable of (i) transforming cell differentiation pseudo-time to a representation of sequential windows that are translated and rotated within a combined pseudo-time-principal-component space using the tangent vectors of window-based Frenet frames; (ii) inferring bifurcation points via a Gibbs sampler and Bayesian model selection; (iii) constructing cell lineage representations of single-cell gene expression data in the combined pseudo-time-principal-component space, that describe the cell developmental trajectory from the start of cell differentiation to a terminal state within a given dataset; (iv) modelling cell differentiation as a continuous process using a Gaussian process; and (v) projecting a related dataset onto the cell developmental trajectory of a primary dataset's phenotypic manifold. Using this projection process, even single-cell gene expression data from a different species can also be compared.

In this chapter three primary modelling results are used to illustrate the operational capabilities of BAGEL namely (i) the modelling of the mouse bone marrow dataset[1]; (ii) the modelling of the human bone marrow dataset; and (iii) the modelling of the projection of the UCB dataset onto the phenotypic manifold of the human bone marrow dataset. All of these obtained results are visualised and interpreted in the following sections. It should be noted, that an additional result of the projection of the human UCB dataset onto the mouse bone marrow dataset is shown in Appendix B.

6.2 LINEAGE IDENTIFICATION

As BAGEL only provide mathematical knowledge about the single-cell gene expression data, prior biological knowledge is used to interpret all of its detected PC-lineages. The prior biological knowledge is that cluster of differentiation 34 (*CD34*) is abundantly expressed in HSPCs, whereas myeloperoxidase

[1] For the sake of brevity single-cell gene expression dataset will also be referred to as dataset.

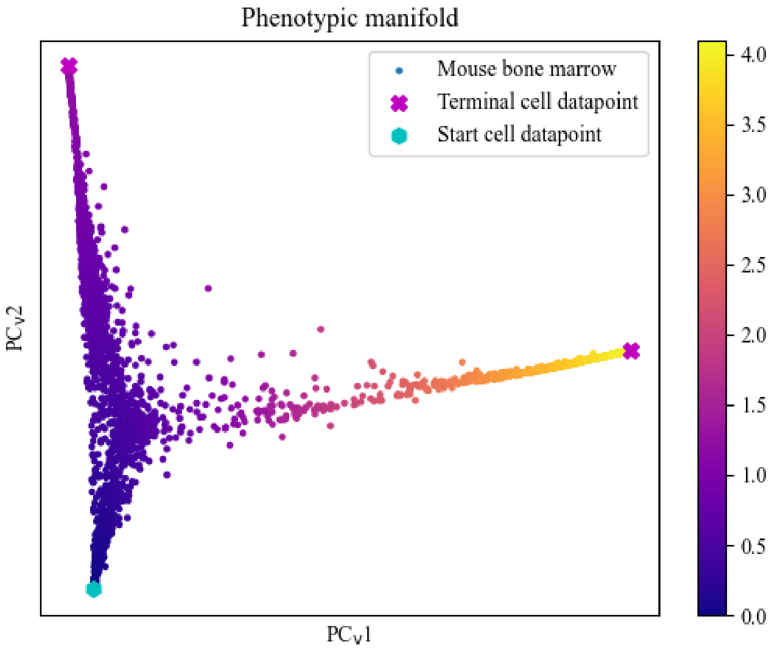
(*MPO*) and GATA-binding factor 1 (*GATA1*) are expressed in cells of myeloid and early erythroid origin, respectively. Hence, the expression of these molecular markers assist in identifying HSPCs, myeloid and erythroid PC-lineages.

6.3 MODELLING OF MOUSE BONE MARROW GENE EXPRESSIONS

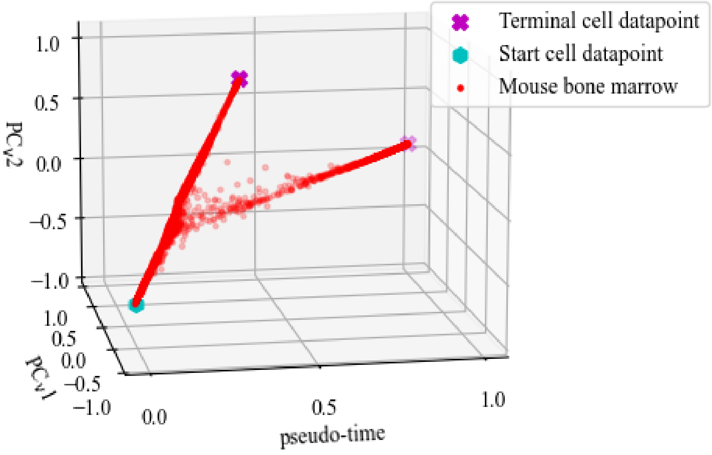
The following results, which are visualised and discussed, are those of the modelling of the mouse bone marrow single-cell gene expression data (see Table 5.2).

6.3.1 Phenotypic manifold

Figure 6.1 shows the obtained phenotypic manifold of the mouse bone marrow dataset. In these figures, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset, is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each cell datapoint is represented with a dot; and (iv) the legend uses blue to indicate the start of the pseudo-time and yellow the end.



(a) The obtained two-dimensional phenotypic manifold of the mouse bone marrow dataset with its pseudo-time.



(b) The obtained three-dimensional phenotypic manifold of the mouse bone marrow dataset in space $\mathcal{P}$.

Figure 6.1. The obtained phenotypic manifold of the mouse bone marrow dataset. In these figures, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset, is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each cell datapoint is represented with a dot; and (iv) the legend uses blue to indicate the start of the pseudo-time and yellow the end.

6.3.2 Gene expressions visualised on the obtained phenotypic manifold

As expected, Figure 6.2 shows increased expression of *CD34* closer to the starting cell, i.e. start of cell differentiation, while expression of *MPO* and *GATA1* indicates the presence of myeloid and erythroid precursor cells, respectively. In Figure 6.2, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each cell datapoint is represented with a dot; and (iv) the level of gene expression is indicated with the colour bar, where blue indicates no expression, red indicates high expression and orange indicates intermediate expression.

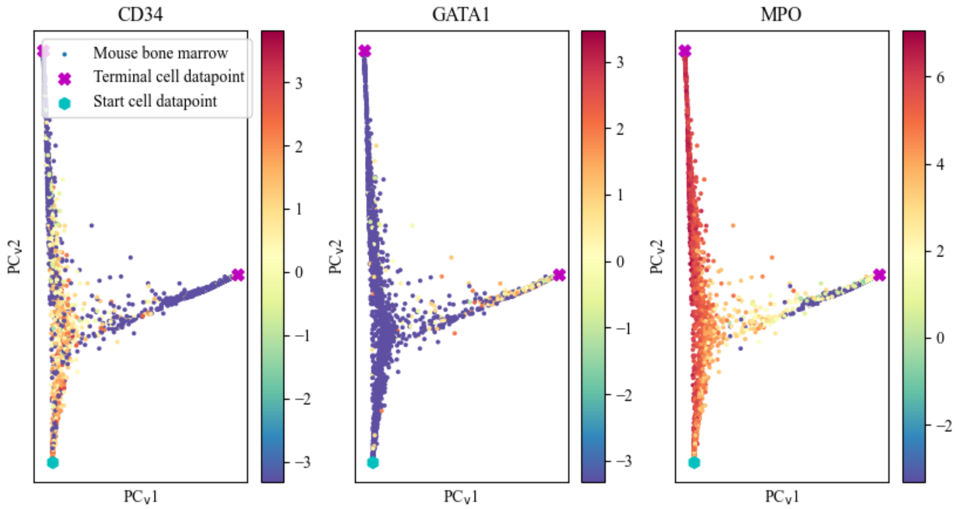
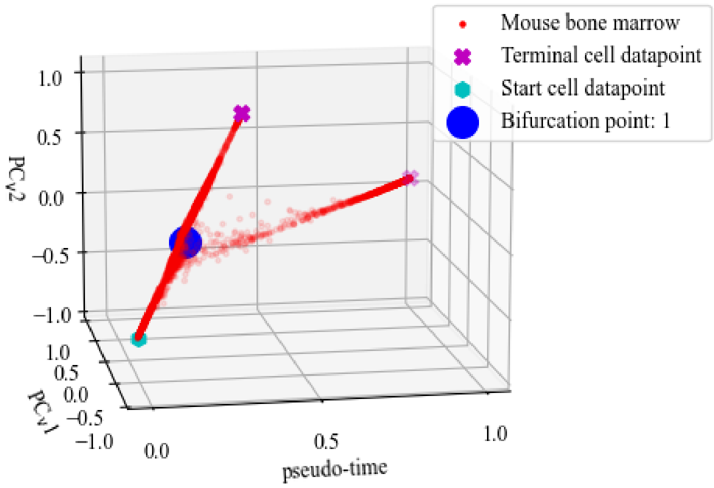


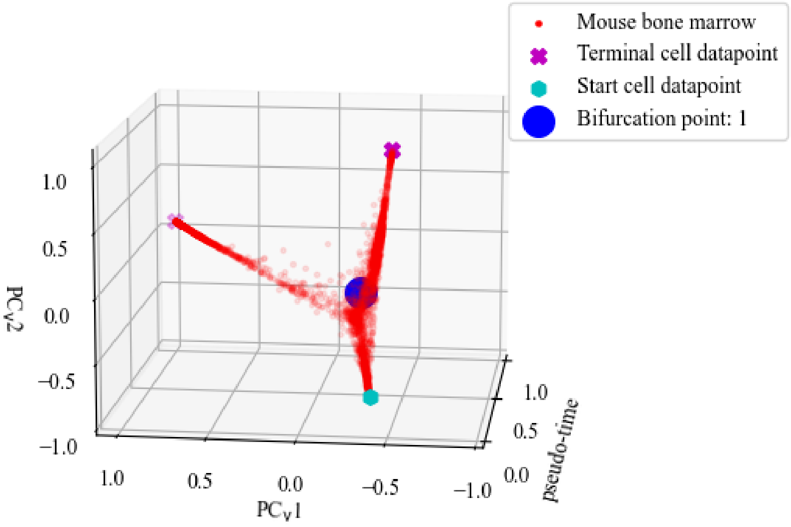
Figure 6.2. Gene expression of lineage-specific genes. Expression of HSPC gene (*CD34*), myeloid- (*MPO*) and erythroid-specific genes (*GATA1*) of the mouse bone marrow dataset. In these figures, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each cell datapoint is represented with a dot; and (iv) the level of gene expression is indicated with the colour bar, where blue indicates no expression, red indicates high expression and orange indicates intermediate expression.

6.3.3 Bifurcation point

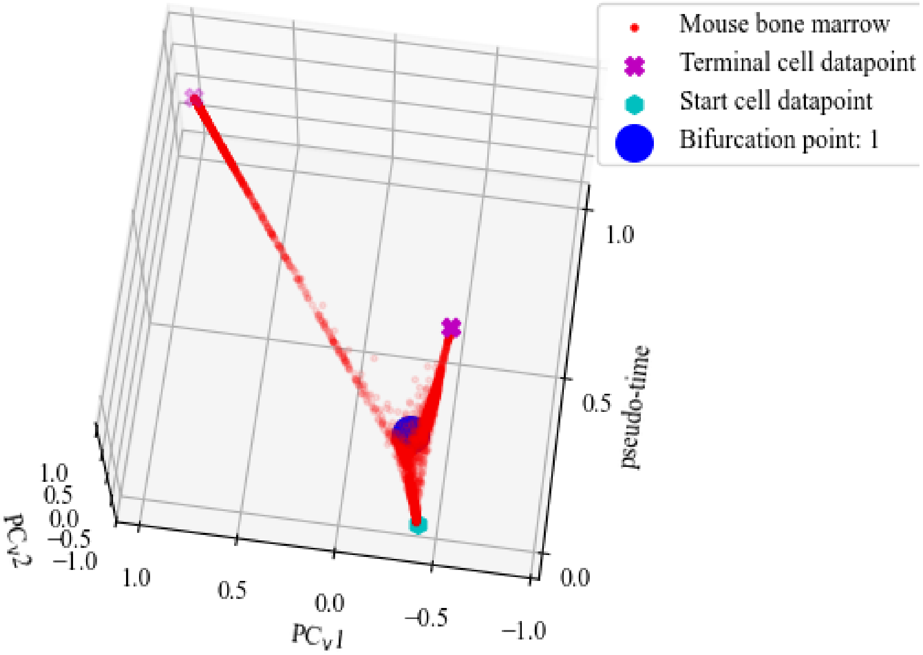
The detected bifurcation point from Bayesian model selection during the modelling of the mouse bone marrow dataset in space $\mathcal{P}$ is shown in Figure 6.3 below. In these figures, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each cell datapoint is represented with a dot; and (iv) the well-defined bifurcation points is indicated with a blue coloured sphere.



(a) The detected bifurcation point of the mouse bone marrow dataset visualised by observing space $\mathcal{P}$ from the PC_v2-pseudo-time axes.



(b) The detected bifurcation point of the mouse bone marrow dataset visualised by observing space $\mathcal{P}$ from the PC_v1-PC_v2 axes.

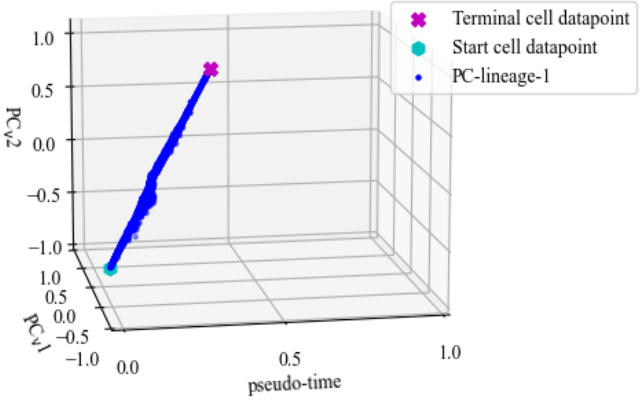


(c) The detected bifurcation point of the mouse bone marrow dataset visualised by observing space $\mathcal{P}$ from the PC$_V$1-pseudo-time axes.

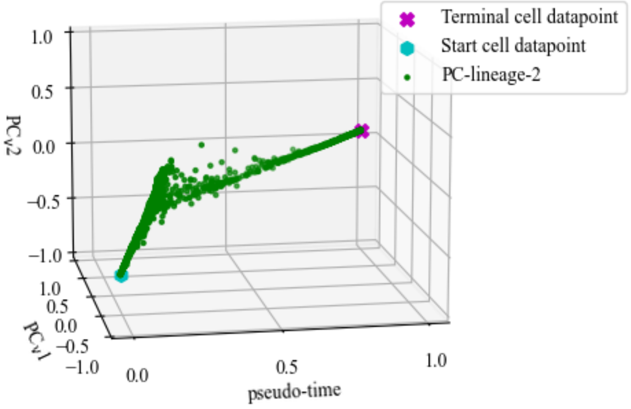
Figure 6.3. The detected bifurcation point from Bayesian model selection during the modelling of the mouse bone marrow dataset in space $\mathcal{P}$. In these figures, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each cell datapoint is represented with a dot; and (iv) the well-defined bifurcation point is indicated with a blue coloured sphere.

6.3.4 Constructed PC-lineages

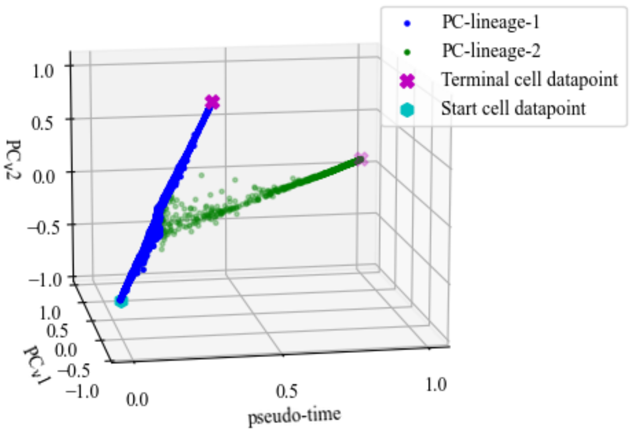
The constructed PC-lineages that indicate the cell developmental trajectory from the start of cell differentiation to a terminal state in space $\mathcal{P}$ of the mouse bone marrow dataset is shown in Figure 6.4. In Figure 6.4, (i) the start cell datapoint, that indicates the starting point of cell differentiation of the dataset is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset is indicated with a pink cross; (iii) each cell datapoint is represented with a dot; and (iv) each detected PC-lineage is represented with a different colour-coded cluster of cell datapoints.



(a) The first constructed PC-lineage of the mouse bone marrow dataset in space $\mathcal{P}$.



(b) The second constructed PC-lineage of the mouse bone marrow dataset in space $\mathcal{P}$.

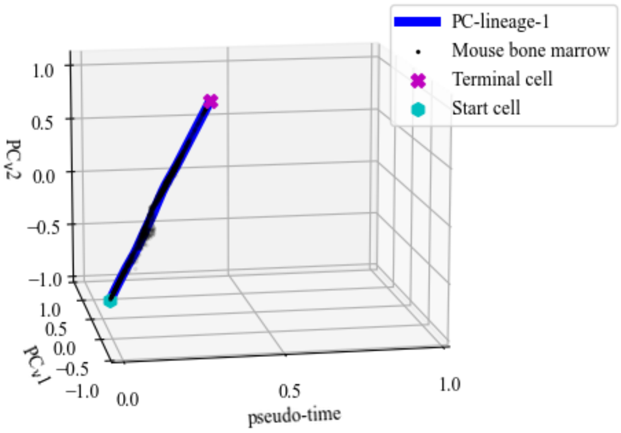


(c) All constructed PC-lineages of the mouse bone marrow dataset in space $\mathcal{P}$.

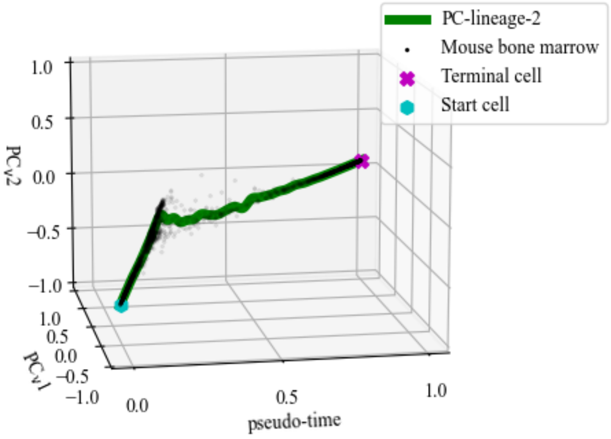
Figure 6.4. The constructed PC-lineages that indicate the cell developmental trajectory from the start of cell differentiation to a terminal state in space $\mathcal{P}$ of the mouse bone marrow dataset. (a) First, (b) second, and (c) all constructed PC-lineages of the mouse bone marrow dataset. In these figures, (i) the start cell datapoint, that indicates the starting point of cell differentiation of the dataset is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset is indicated with a pink cross; (iii) each cell datapoint is represented with a dot; and (iv) each detected PC-lineage is represented with a different colour-coded cluster of cell datapoints.

6.3.5 Continuous modelling of PC-lineages

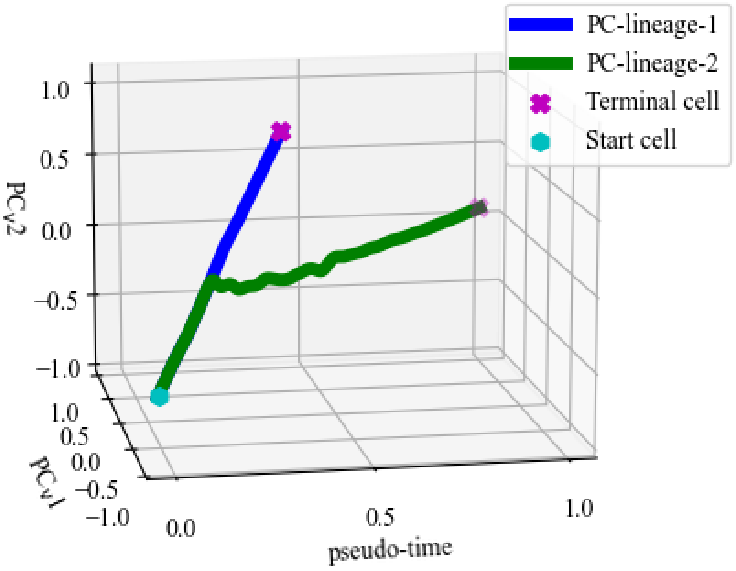
The continuous modelling of the constructed PC-lineages of the mouse bone marrow dataset in space $\mathcal{P}$ is shown in Figure 6.5. In Figure 6.5, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset, is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each cell datapoint is represented with a dot; and (iv) the coloured solid line represents the continuous cell developmental trajectory of a given PC-lineage, also known as the Gaussian process mean.



(a) The first constructed PC-lineage of the mouse bone marrow dataset modelled as a continuous process in space $\mathcal{P}$.



(b) The second constructed PC-lineage of the mouse bone marrow dataset modelled as a continuous process in space $\mathcal{P}$.



(c) All constructed PC-lineages of the mouse bone marrow dataset modelled as a continuous process in space $\mathcal{P}$.

Figure 6.5. The continuous modelling of the constructed PC-lineages of mouse bone marrow dataset in space $\mathcal{P}$. (a) First, (b) second, and (c) all constructed PC-lineages of the mouse bone marrow dataset modelled as a continuous process. In these figures, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset, is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each cell datapoint is represented with a dot; and (iv) the coloured solid line represents the continuous cell developmental trajectory of a given PC-lineage, also known as the Gaussian process mean.

6.3.6 Tangent vectors of window-based Frenet frames

Figure 6.6 shows the tangent vectors of the window-based Frenet frames of the cell developmental trajectory of the mouse bone marrow dataset in space $\mathcal{P}$. In Figure 6.6, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset, is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each cell datapoint is represented with a dot; and (iv) the tangent vectors ($\mathrm{PC_w}1$, see Section 4.7.2) of the window-based Frenet frames are indicated with green arrows.

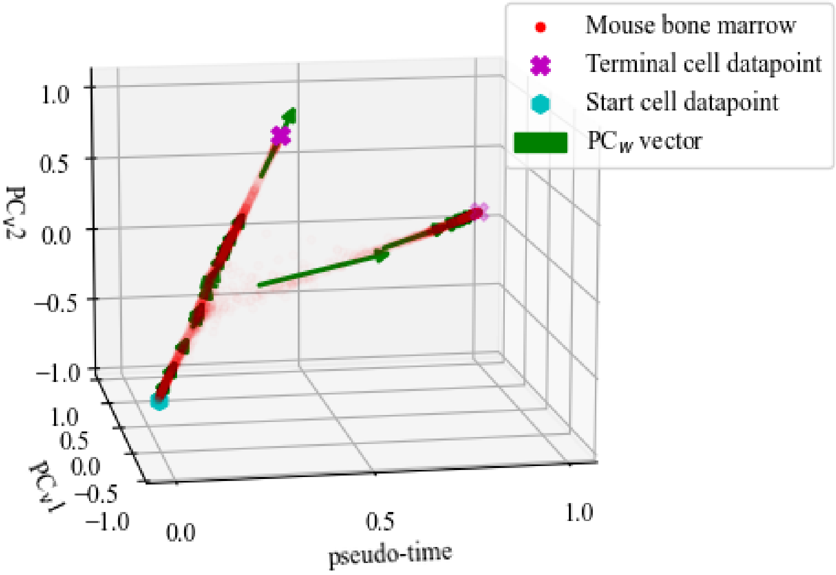


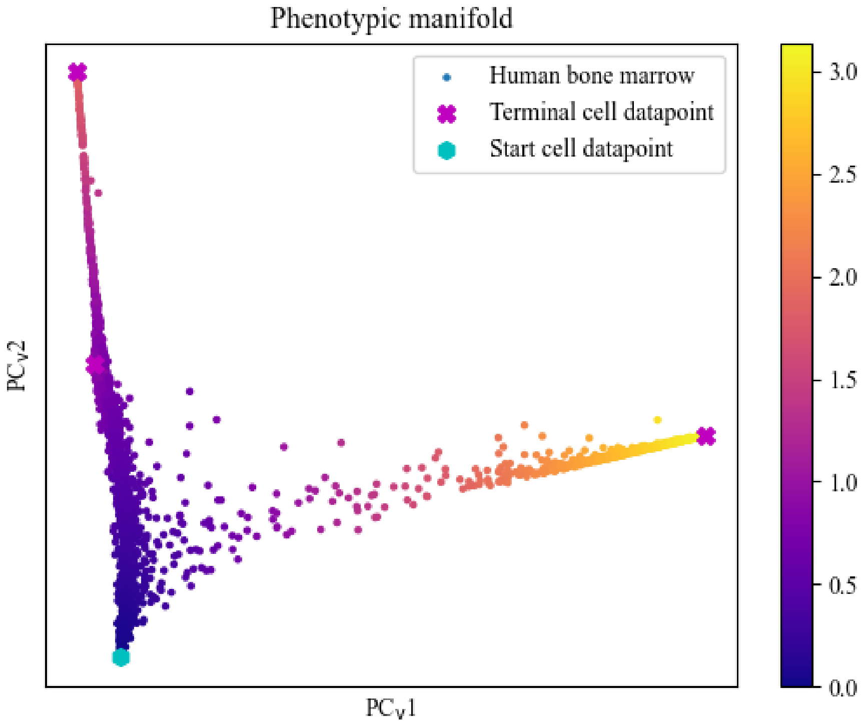
Figure 6.6. Tangent vectors of the window-based Frenet frames of the cell developmental trajectory of the mouse bone marrow dataset in space $\mathcal{P}$. In this figure (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset, is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each cell datapoint is represented with a dot; and (iv) the tangent vectors ($\mathrm{PC_w}1$, see Section 4.7.2) of the window-based Frenet frames are indicated with green arrows.

6.4 MODELLING OF HUMAN BONE MARROW GENE EXPRESSIONS

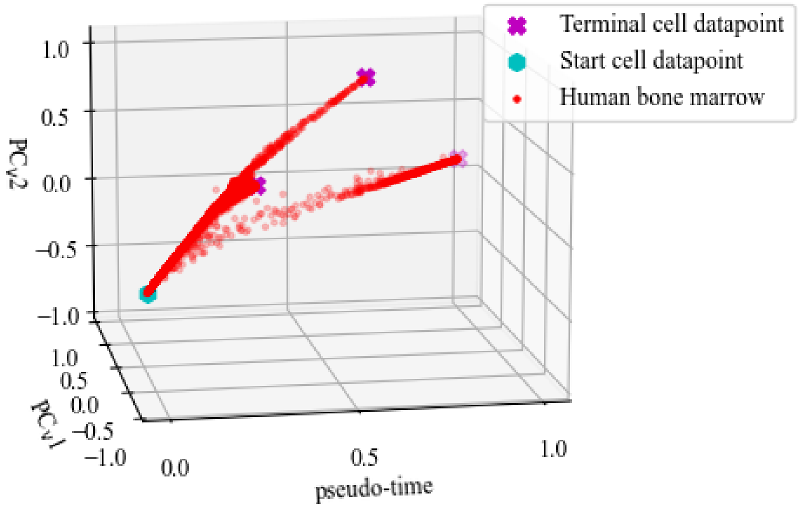
The following results, which are visualised and discussed, are those of the modelling of human bone marrow single-cell gene expression data (see Table 5.2).

6.4.1 Phenotypic manifold

Figure 6.7 shows the obtained phenotypic manifold of the human bone marrow dataset. In these figures, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset, is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each cell datapoint is represented with a dot; and (iv) the legend uses blue to indicate the start of the pseudo-time, and yellow the end.



(a) The obtained two-dimensional phenotypic manifold of the human bone marrow dataset with its pseudo-time.



(b) The obtained three-dimensional phenotypic manifold of the human bone marrow dataset in space $\mathcal{P}$.

Figure 6.7. The obtained phenotypic manifold of the human bone marrow dataset. In these figures, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset, is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each cell datapoint is represented with a dot; and (iv) the legend uses blue to indicate the start of the pseudo-time, and yellow the end.

6.4.2 Gene expressions visualised on the obtained phenotypic manifold

As expected, Figure 6.8 shows increased expression of *CD34* closer to the starting cell, i.e. the start of differentiation, while expression of *MPO* and *GATA1* indicates the presence of myeloid and erythroid precursor cells, respectively. In Figure 6.8, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each cell datapoint is represented with a dot; and (iv) the level of gene expression is indicated with the colour bar, where blue indicates no expression, red indicates high expression and orange indicates intermediate expression.

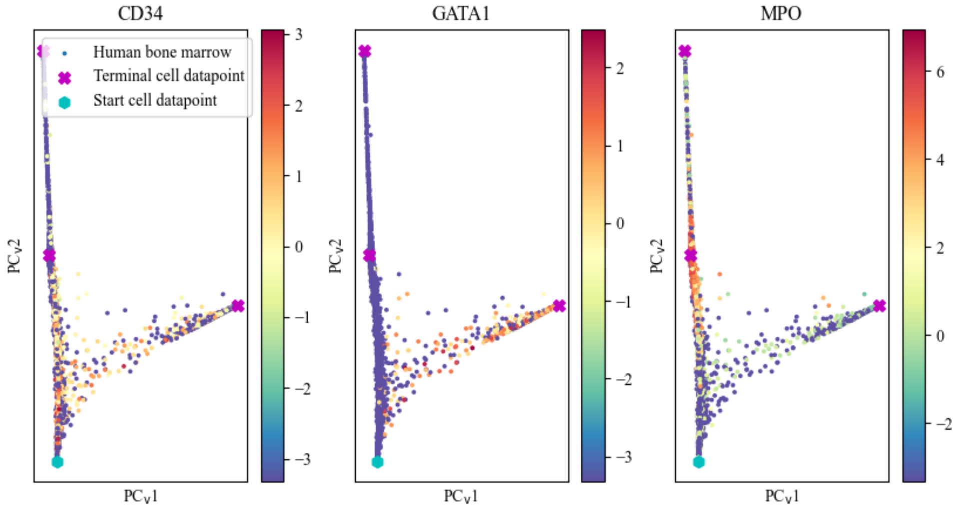
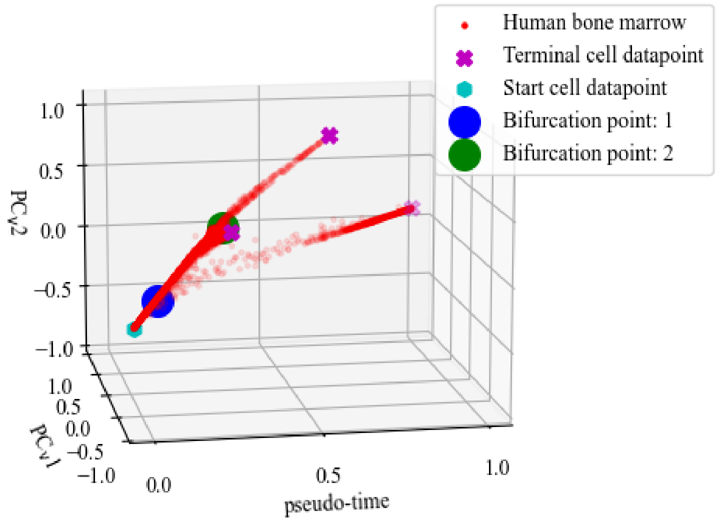


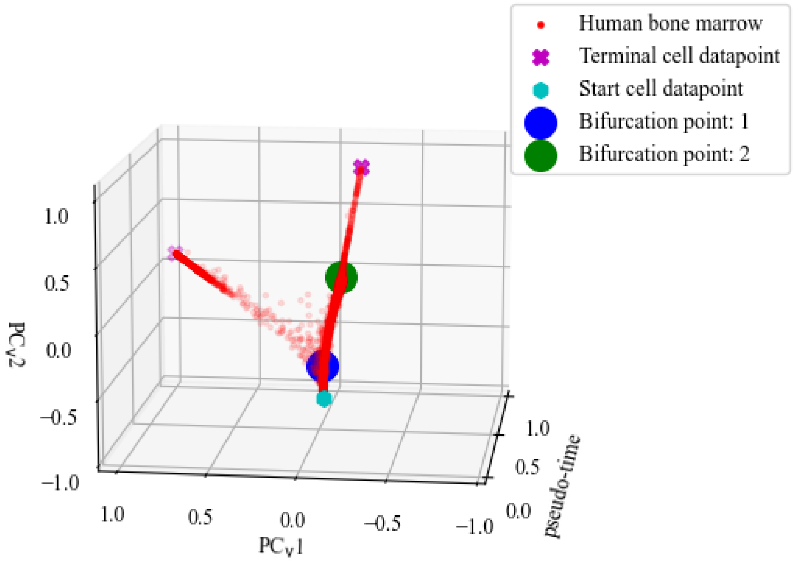
Figure 6.8. Gene expression of lineage-specific genes. Expression of HSPC gene (*CD34*), myeloid- (*MPO*) and erythroid-specific genes (*GATA1*) of the human bone marrow dataset. In these figures, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each cell datapoint is represented with a dot; and (iv) the level of gene expression is indicated with the colour bar, where blue indicates no expression, red indicates high expression and orange indicates intermediate expression.

6.4.3 Bifurcation points

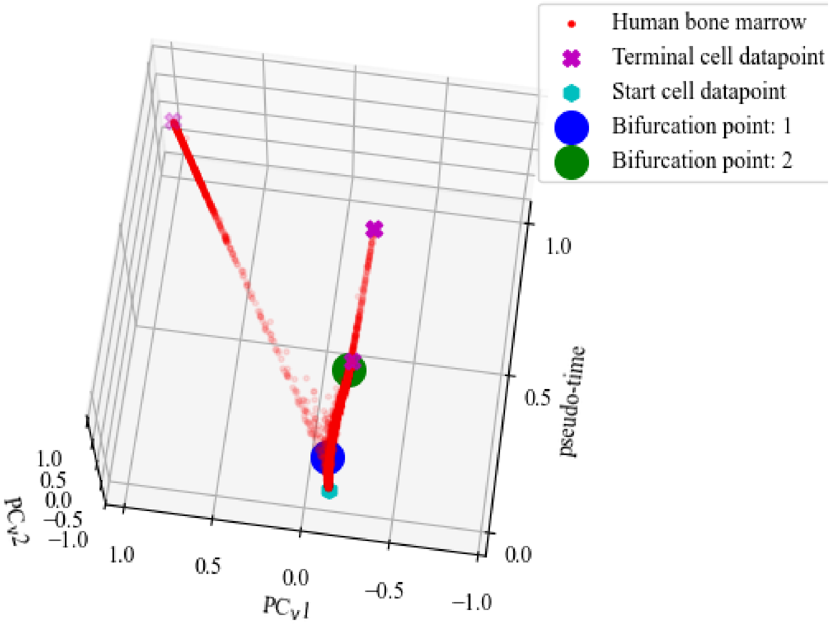
The detected bifurcation points from Bayesian model selection during the modelling of the human bone marrow dataset in space $\mathcal{P}$ is shown in Figure 6.9 below. In these figures, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each cell datapoint is represented with a dot; and (iv) the well-defined bifurcation points are indicated with a blue and a green coloured sphere respectively.



(a) The detected bifurcation points of the human bone marrow dataset visualised by observing space $\mathcal{P}$ from the $\mathrm{PC_v2}$-pseudo-time axes.



(b) The detected bifurcation points of the humna bone marrow dataset visualised by observing space $\mathcal{P}$ from the $\mathrm{PC_v1}$-$\mathrm{PC_v2}$ axes.

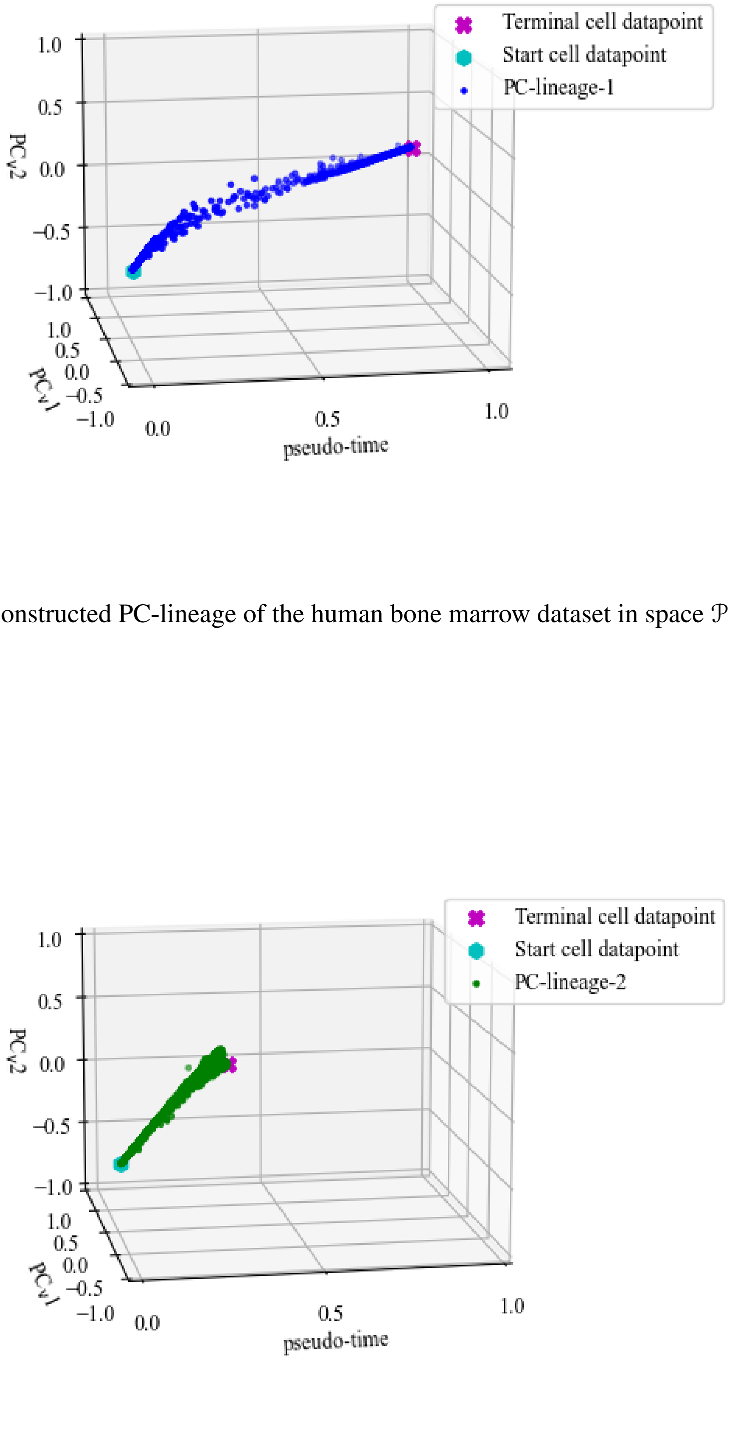


(c) The detected bifurcation points of the human bone marrow dataset visualised by observing space $\mathcal{P}$ from the PC$_v$1-pseudo-time axes.

Figure 6.9. The detected bifurcation points from Bayesian model selection during the modelling of the human bone marrow dataset in space $\mathcal{P}$. In these figures, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each cell datapoint is represented with a dot; and (iv) the well-defined bifurcation points are indicated with a blue and a green coloured sphere respectively.

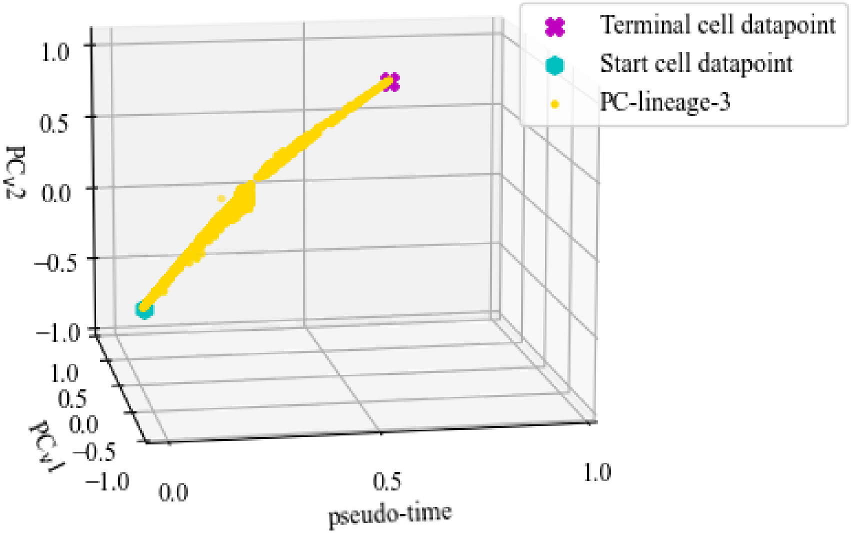
6.4.4 Constructed PC-lineages

The constructed PC-lineages that indicate the cell developmental trajectory from the start of cell differentiation to a terminal state of the human bone marrow dataset in space $\mathcal{P}$ is shown in Figure 6.10. In Figure 6.10, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each cell datapoint is represented with a dot; and (iv) each detected PC-lineage is represented with a different colour-coded cluster of cell datapoints.

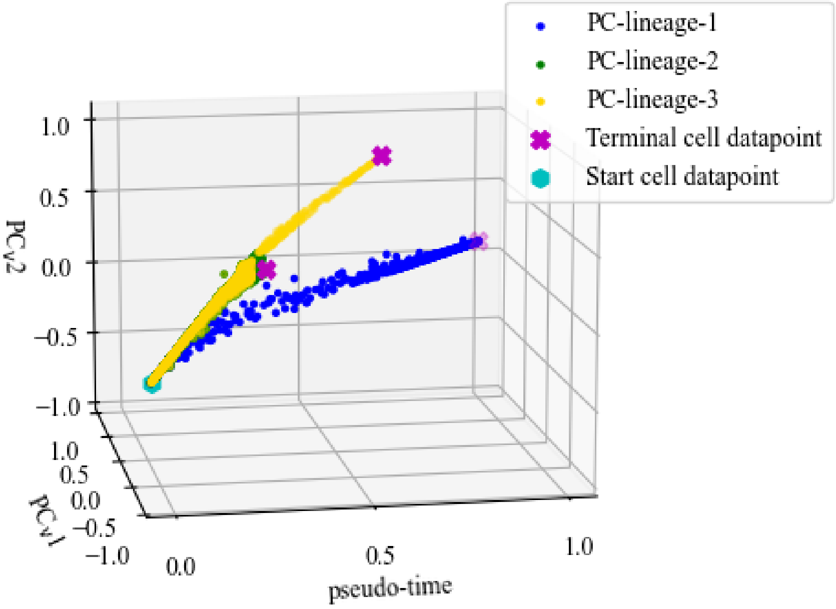


(a) The first constructed PC-lineage of the human bone marrow dataset in space $\mathcal{P}$.

(b) The second constructed PC-lineage of the human bone marrow dataset in space $\mathcal{P}$.



(c) The third constructed PC-lineage of the human bone marrow dataset in space $\mathcal{P}$.

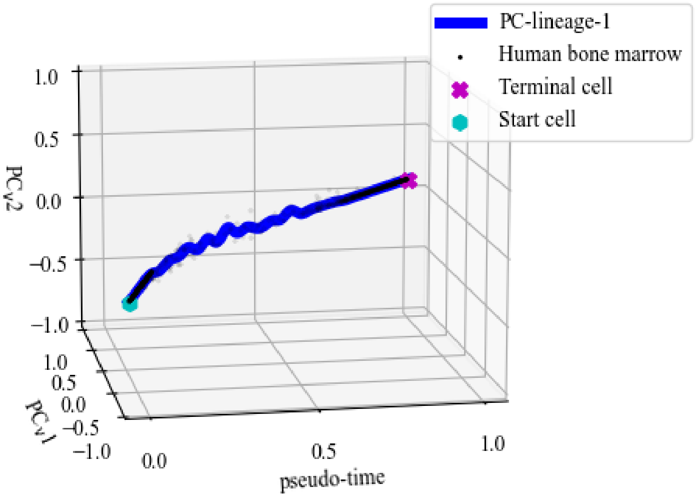


(d) All constructed PC-lineages of the human bone marrow dataset in space $\mathcal{P}$.

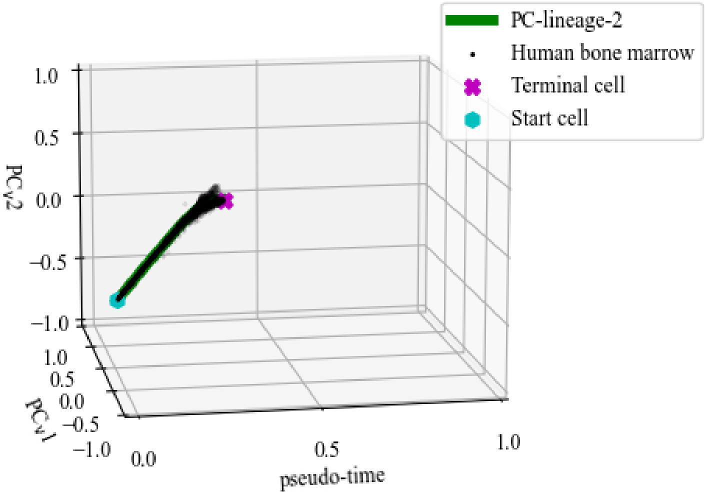
Figure 6.10. The constructed PC-lineages of the human bone marrow dataset. (a) First, (b) second, (c) third and (c) all constructed PC-lineages of human bone marrow dataset. In these figures, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each cell datapoint is represented with a dot; and (iv) each detected PC-lineage is represented with a different colour-coded cluster of cell datapoints.

6.4.5 Continuous modelling of PC-lineages

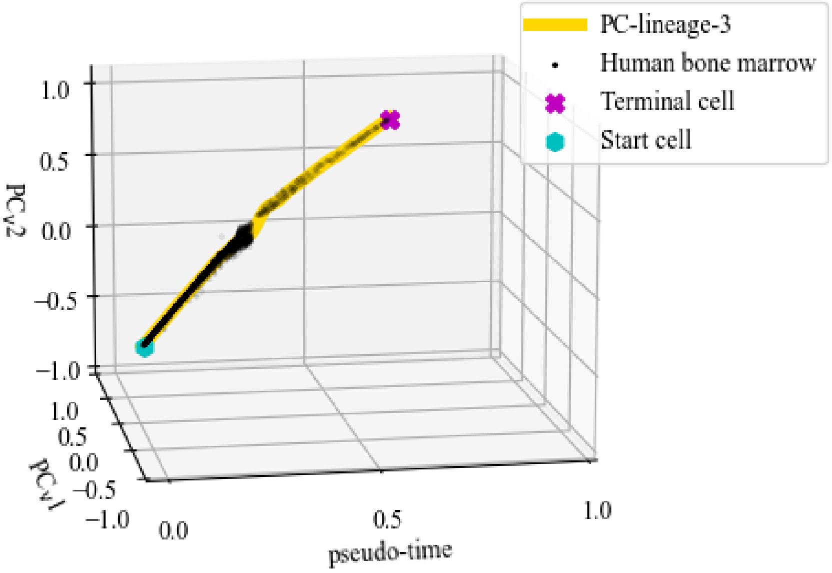
The continuous modelling of the constructed PC-lineages of the human bone marrow dataset in space $\mathcal{P}$ is shown in Figure 6.11. In Figure 6.11, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each cell datapoint is represented with a dot; and (iv) the coloured solid line represents the continuous cell developmental trajectory of a given PC-lineage, also known as the Gaussian process mean.



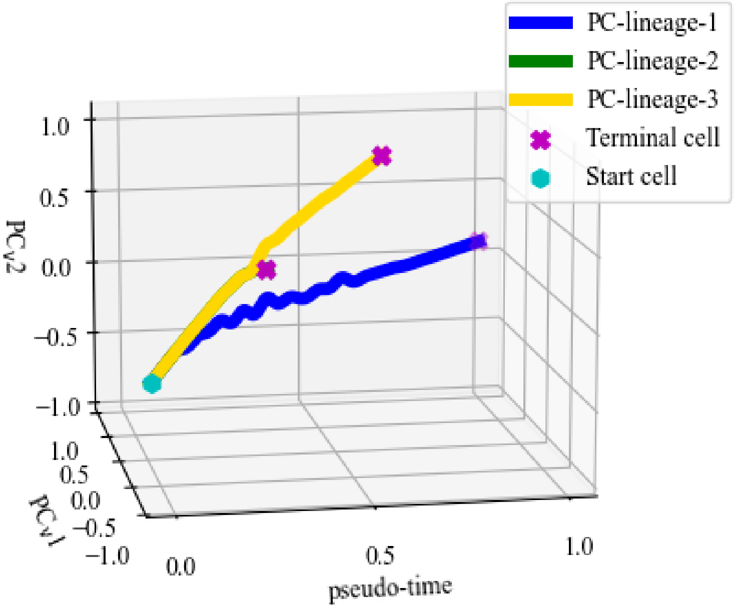
(a) The first constructed PC-lineage of the human bone marrow dataset modelled as a continuous process in space $\mathcal{P}$.



(b) The second constructed PC-lineage of the human bone marrow dataset modelled as a continuous process in space $\mathcal{P}$.



(c) The third constructed PC-lineage of the human bone marrow dataset modelled as a continuous process in space $\mathcal{P}$.



(d) All constructed PC-lineages of the mouse bone marrow dataset modelled as a continuous process in space $\mathcal{P}$.

Figure 6.11. The continuous modelling of the constructed PC-lineages of the human bone marrow dataset in space $\mathcal{P}$. (a) First, (b) second, (c) third and (d) all constructed PC-lineages of human bone marrow dataset modelled as a continuous process. In these figures, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each cell datapoint is represented with a dot; and (iv) the coloured solid line represents the continuous cell developmental trajectory of a given PC-lineage, also known as the Gaussian process mean.

6.4.6 Tangent vectors of window-based Frenet frames

Figure 6.12 shows the tangent vectors of the window-based Frenet frames of the cell developmental trajectory of human bone marrow dataset in space $\mathcal{P}$. In Figure 6.12, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset is indicated with a pink cross; (iii) each cell datapoint is represented with a dot; and (iv) the tangent vectors (PC_w1, see Section 4.7.2) of the window-based Frenet frames are indicated with green arrows.

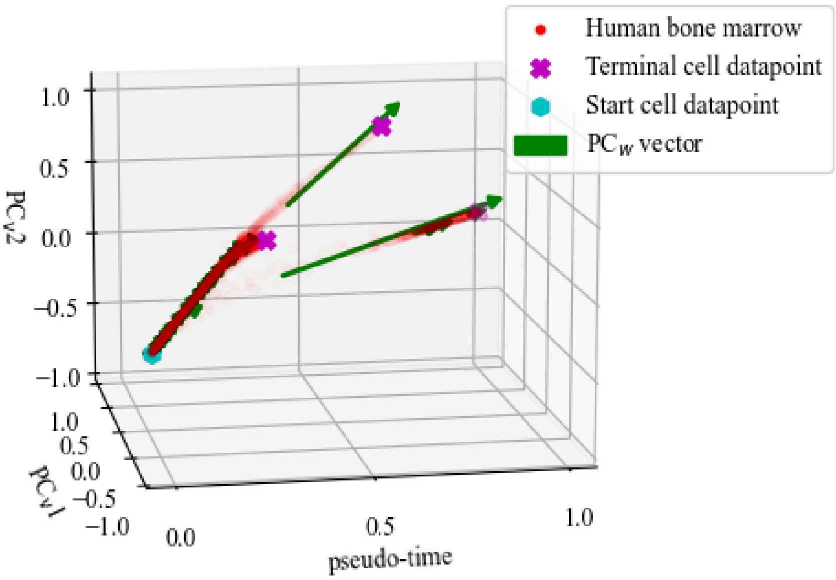


Figure 6.12. Tangent vectors of the window-based Frenet frames of the cell developmental trajectory of human bone marrow in space $\mathcal{P}$. In this figure, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset is indicated with a pink cross; (iii) each cell datapoint is represented with a dot; and (iv) the tangent vectors (PC_w1, see Section 4.7.2) of the window-based Frenet frames are indicated with green arrows.

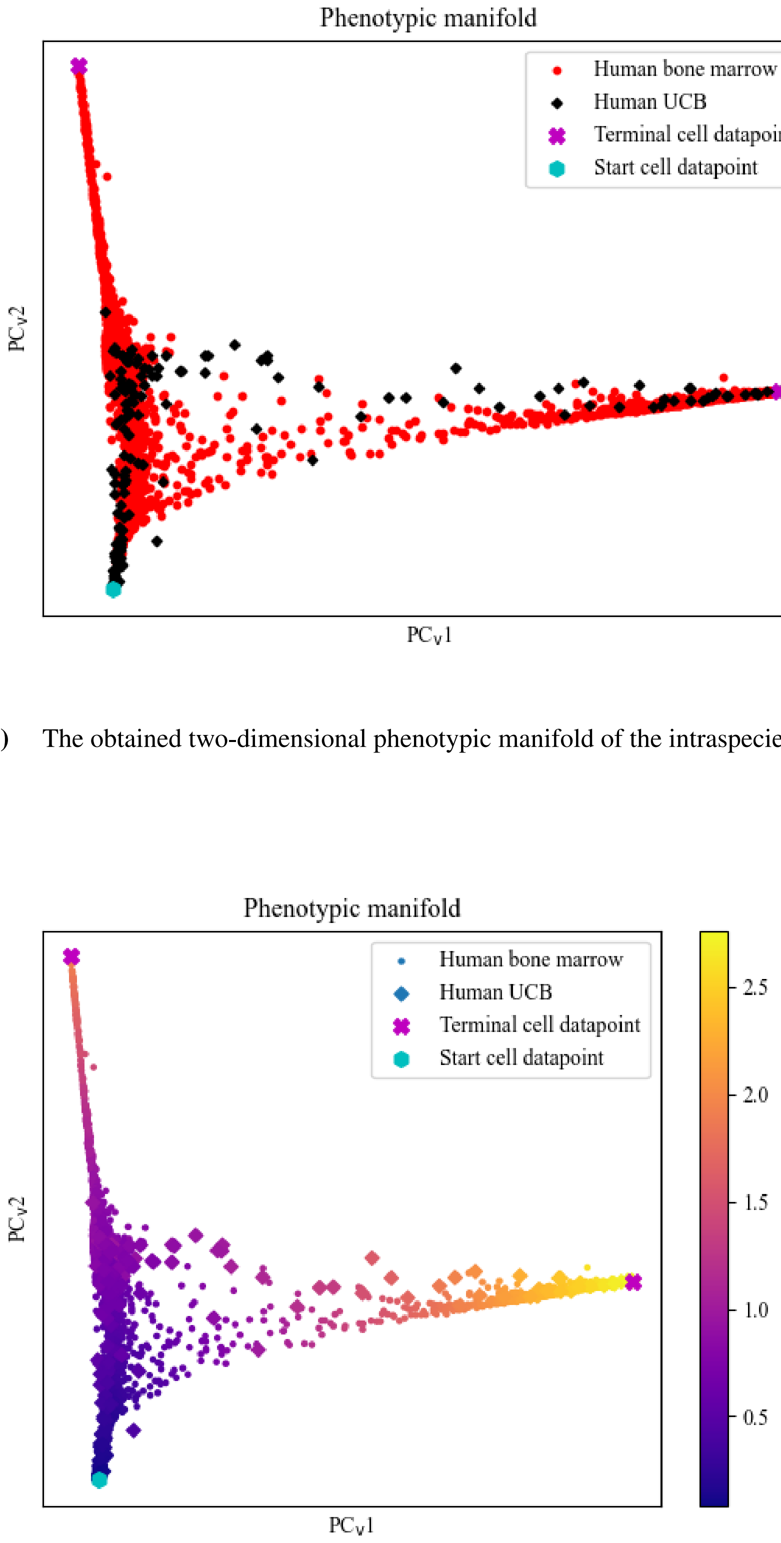
6.5 MODELLING OF INTRASPECIES GENE EXPRESSIONS

The following results, which are visualised and discussed, are those of the modelling of the projection of human UCB single-cell gene expression data onto the phenotypic manifold of the human bone marrow single-cell gene expression data (see Table 5.2). For the sake of brevity this data projection will be referred to as the intraspecies dataset.

6.5.1 Phenotypic manifold

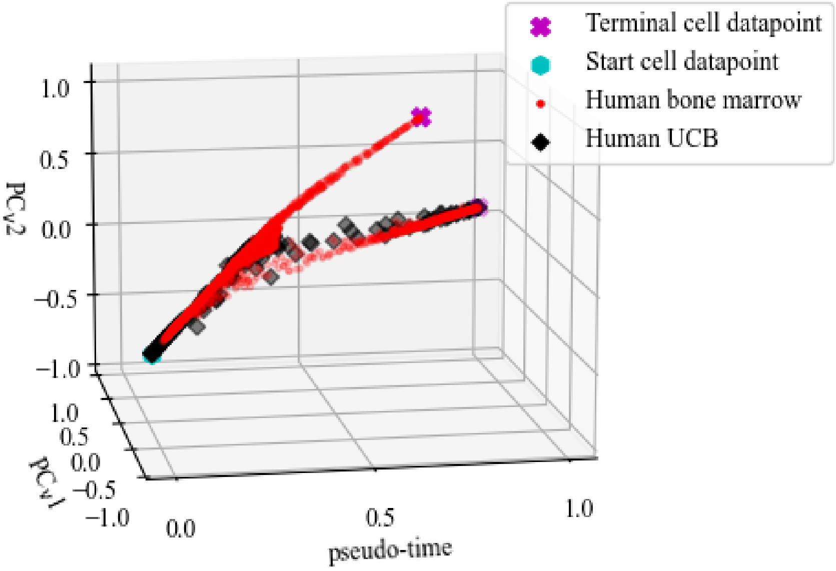
Figure 6.13 shows the obtained phenotypic manifold of the intraspecies dataset. In Figure 6.13, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset, is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each dot represents a human bone marrow cell datapoint; (iv) whereas the projected human UCB cell datapoints onto the phenotypic manifold is represented with diamonds; and (iv) the legend uses blue to indicate the start of the pseudo-time and yellow the end. When comparing the developed manifolds of human bone marrow in Figure 6.7 and Figure 6.13 it is clear that the latter has one less terminal cell datapoint. The terminal cell datapoint is missing due to the Palantir algorithm approaching terminal state detection statistically. The statistical approach leads to the developed Markov chain of the projected data not converging to all of the same extrema of the diffusion components as the non-projected data (Appendix: A.4.1).

The projection method is also deemed to be accurate as the differentiation of the human UCB dataset corresponds to the biology of haematopoiesis found in literature. An elaboration of why this projection corresponds to the literature and why it is deemed to be accurate is discussed more in-depth in Section 6.6.



(a) The obtained two-dimensional phenotypic manifold of the intraspecies dataset.

(b) The obtained two-dimensional phenotypic manifold of the intraspecies dataset with its pseudo-time.



(c) The obtained three-dimensional phenotypic manifold of the intraspecies dataset in space $\mathcal{P}$.

Figure 6.13. The obtained phenotypic manifold of the intraspecies dataset. In these figures, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset, is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each dot represents a human bone marrow cell datapoint; (iv) whereas the projected human UCB cell datapoints onto the phenotypic manifold is represented with diamonds; and (iv) the legend uses blue to indicate the start of the pseudo-time and yellow the end.

6.5.2 Gene expressions visualised on the obtained phenotypic manifold

As expected, Figure 6.14 shows increased expression of *CD34* closer to the starting cell, i.e. start of cell differentiation, while expression of *MPO* and *GATA1* indicates the presence of myeloid and erythroid precursor cells, respectively. In Figure 6.14, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each dot represents a human bone marrow cell datapoint; (iv) whereas projected human UCB cell datapoints onto the phenotypic manifold is represented with diamonds; and (v) the level of gene expression is indicated with the colour bar, where blue indicates no expression, red indicates high expression and orange indicates intermediate expression.

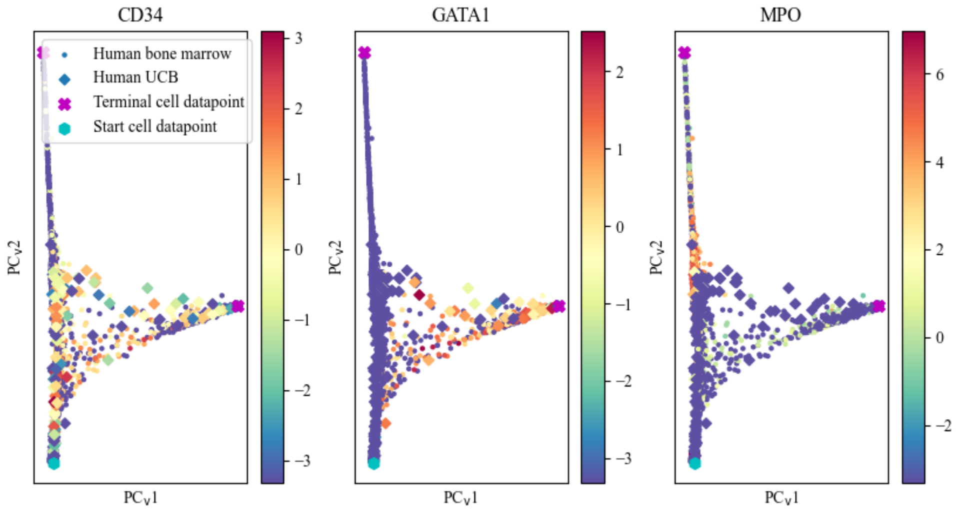
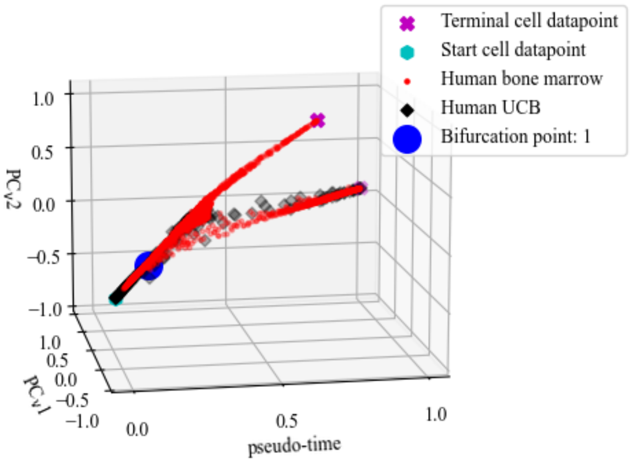


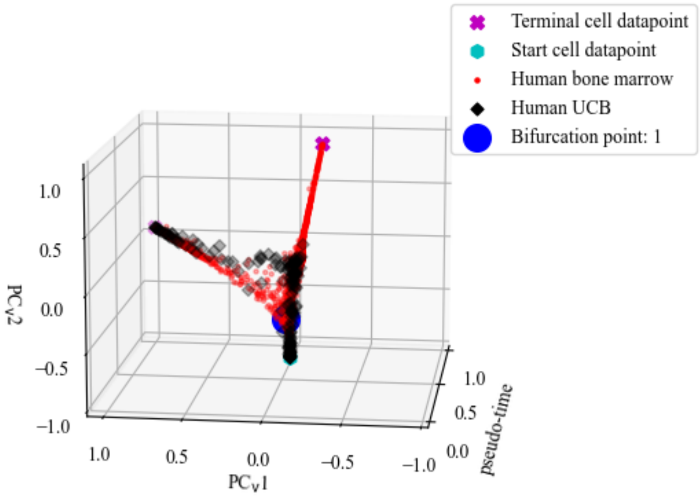
Figure 6.14. Gene expression of lineage-specific genes. Expression of HSPC gene (*CD34*), myeloid- (*MPO*) and erythroid-specific genes (*GATA1*) of the mouse bone marrow dataset. In these figures, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each dot represents a human bone marrow cell datapoint; (iv) whereas projected human UCB cell datapoints onto the phenotypic manifold is represented with diamonds; and (v) the level of gene expression is indicated with the colour bar, where blue indicates no expression, red indicates high expression and orange indicates intermediate expression.

6.5.3 Bifurcation point

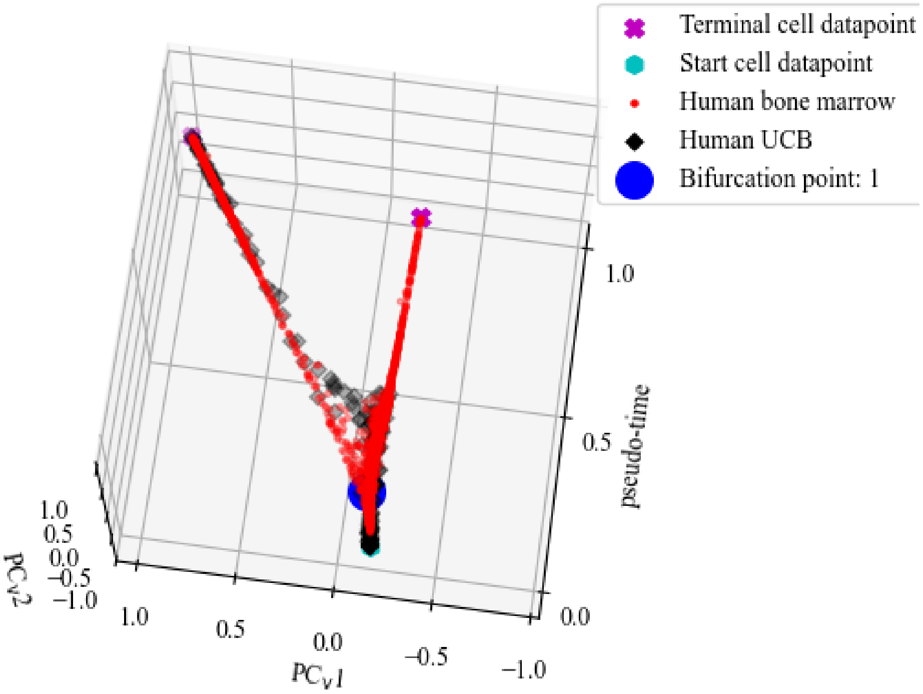
The detected bifurcation point from Bayesian model selection during the modelling of the intraspecies dataset in space $\mathcal{P}$ is shown in Figure 6.15 below. In these figures, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each dot represents a human bone marrow cell datapoint; (iv) whereas the projected human UCB cell datapoints onto the phenotypic manifold is represented with diamonds; and (v) the well-defined bifurcation point is indicated with a blue coloured sphere.



(a) The detected bifurcation points of the intraspecies dataset visualised by observing space $\mathcal{P}$ from the PC_v2-pseudo-time axes.



(b) The detected bifurcation points of the intraspecies dataset visualised by observing space $\mathcal{P}$ from the PC_v1-PC_v2 axes.

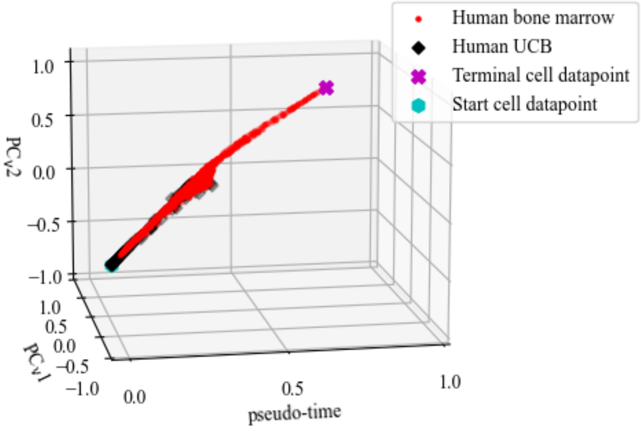


(c) The detected bifurcation points of the intraspecies dataset visualised by observing space $\mathcal{P}$ from the PC$_V$1-pseudo-time axes.

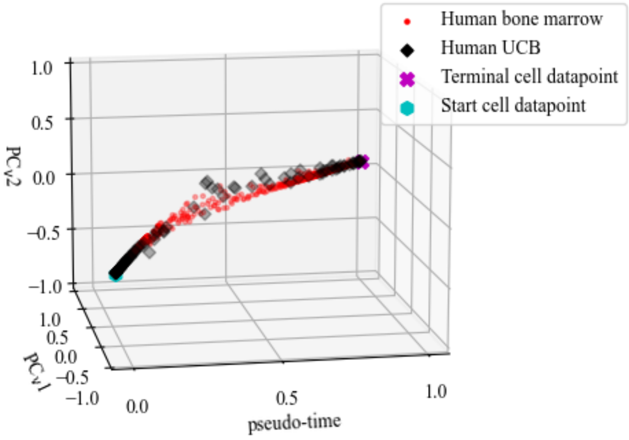
Figure 6.15. Bifurcation points detected of the intraspecies dataset in space $\mathcal{P}$. In these figures, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each dot represents a human bone marrow cell datapoint; (iv) whereas the projected human UCB cell datapoints onto the phenotypic manifold is represented with diamonds; and (v) the well-defined bifurcation point is indicated with a blue coloured sphere.

6.5.4 Constructed PC-lineages

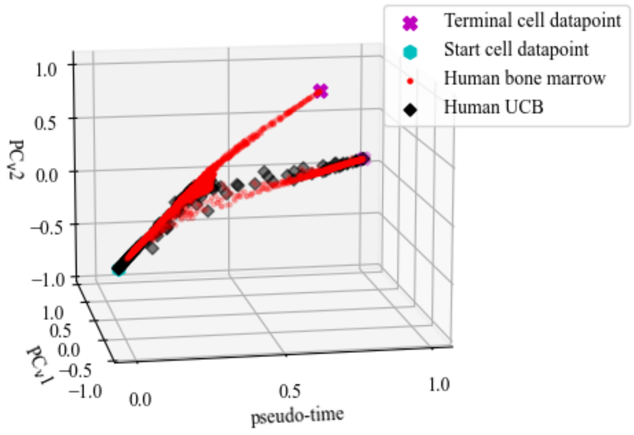
The constructed PC-lineages that indicate the cell developmental trajectory from the start of cell differentiation to a terminal state of the intraspecies dataset in space $\mathcal{P}$ is shown in Figure 6.10. In Figure 6.10, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset, is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each dot represents a human bone marrow cell datapoint; (iv) whereas projected human UCB cell datapoints onto the phenotypic manifold is represented with diamonds.



(a) The first constructed PC-lineage of the intraspecies dataset in space $\mathcal{P}$.



(b) The second constructed PC-lineage of the intraspecies dataset in space $\mathcal{P}$.

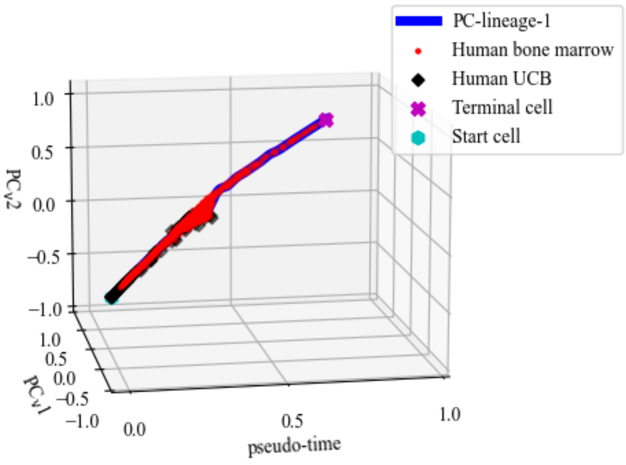


(c) All constructed PC-lineages of the intraspecies dataset in space $\mathcal{P}$.

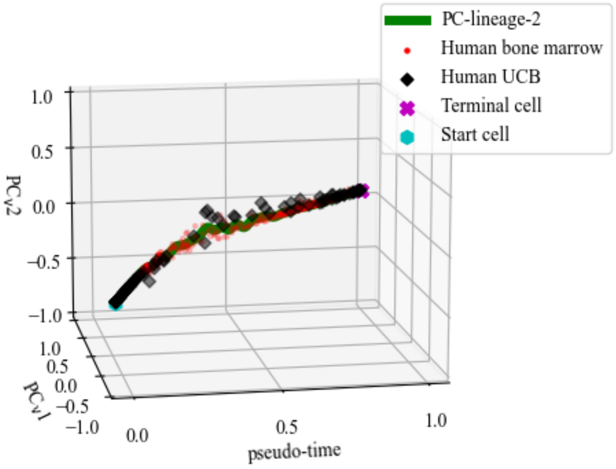
Figure 6.16. The constructed PC-lineages of the intraspecies dataset in space $\mathcal{P}$. (a) First, (b) second, and (c) all constructed PC-lineages of the intraspecies dataset. In these figures, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset, is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each dot represents a human bone marrow cell datapoint; (iv) whereas projected human UCB cell datapoints onto the phenotypic manifold is represented with diamonds.

6.5.5 Continuous modelling of PC-lineages

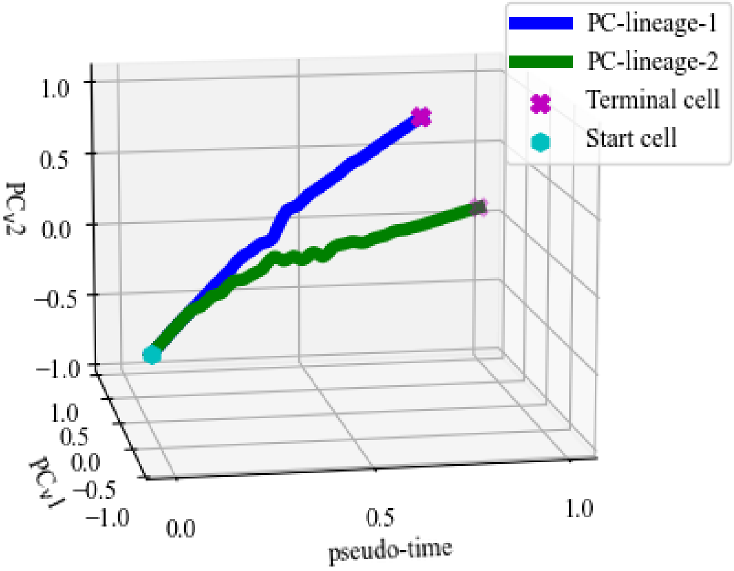
The continuous modelling of the constructed PC-lineages of the intraspecies dataset in space $\mathcal{P}$ is shown in Figure 6.17. In Figure 6.17, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each dot represents a human bone marrow cell datapoint; (iv) whereas the projected human UCB cell datapoints onto the phenotypic manifold is represented with diamonds; and (v) the coloured solid line represents the continuous cell developmental trajectory of a given PC-lineage, also known as the Gaussian process mean.



(a) The first constructed PC-lineage of the intraspecies dataset modelled as a continuous process.



(b) The second constructed PC-lineage of the intraspecies dataset modelled as a continuous process.



(c) All constructed PC-lineages of the intraspecies dataset modelled as a continuous process.

Figure 6.17. The continuous modelling of the constructed PC-lineages of the intraspecies dataset in space $\mathcal{P}$. (a) First, (b) second, and (c) all constructed PC-lineages in intraspecies dataset modelled as a continuous process. In these figures, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each dot represents a human bone marrow cell datapoint; (iv) whereas the projected human UCB cell datapoints onto the phenotypic manifold is represented with diamonds; and (v) the coloured solid line represents the continuous cell developmental trajectory of a given PC-lineage, also known as the Gaussian process mean.

6.5.6 Tangent vectors of window-based Frenet frames

Figure 6.18 shows the tangent vectors of the window-based Frenet frames of the cell developmental trajectory of the intraspecies dataset in space $\mathcal{P}$. In Figure 6.18, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset, is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each dot represents a human bone marrow cell datapoint; (iv) whereas the projected human UCB cell datapoints onto the phenotypic manifold is represented with diamonds; and (v) the tangent vectors ($\mathrm{PC_w1}$, see Section 4.7.2) of the window-based Frenet frames are indicated with green arrows.

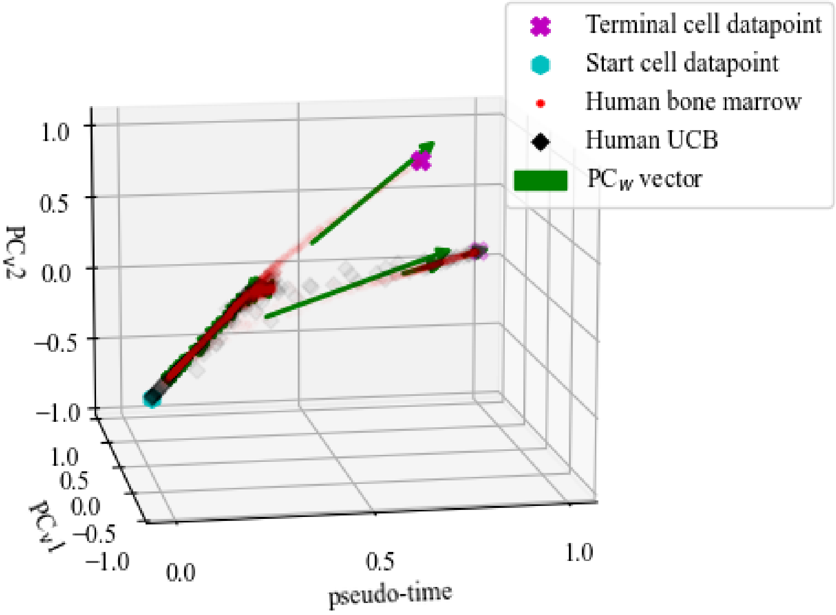


Figure 6.18. Tangent vectors of the window-based Frenet frames of the cell developmental trajectory of the intraspecies dataset in space $\mathcal{P}$. In this figure, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset, is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each dot represents a human bone marrow cell datapoint; (iv) whereas the projected human UCB cell datapoints onto the phenotypic manifold is represented with diamonds; and (v) the tangent vectors ($\mathrm{PC_w1}$, see Section 4.7.2) of the window-based Frenet frames are indicated with green arrows.

6.6 SUMMARY OF OBSERVED RESULTS

Three primary modelling results are used to illustrate the operational capabilities of BAGEL namely (i) the modelling of the mouse bone marrow gene expression dataset; (ii) the modelling of the human bone marrow gene expression dataset; and (iii) the modelling of the projection of the UCB gene expression dataset onto the phenotypic manifold of the human bone marrow gene expression dataset. The robustness of BAGEL is proven by its ability to accurately model all these single-cell gene expression datasets with the same modelling parameters (see Table 5.3). This implies that BAGEL requires minimal model tuning to model single-cell gene expression data.

The first computational step in modelling cell differentiation data, before attempting to identify and characterise cell populations, is to overcome the “curse of dimensionality”. BAGEL follows a similar approach as the Palantir algorithm to reduce the dimensionality of single-cell gene expression data, but changes the visualisation step from t-SNE to $\mathrm{PCA_v}$. This change from t-SNE to $\mathrm{PCA_v}$ is due to $\mathrm{PCA_v}$ performing mathematically-based (static) transformations and allows for the same result with each simulation. Hence, cell differentiation can be observed in a combined pseudo-time-principal-component space (space $\mathcal{P}$) as seen in Figure 6.1, Figure 6.7 and Figure 6.13.

BAGEL also provides a novel manner in which to project a secondary single-cell gene expression dataset onto the phenotypic manifold of a primary single-cell gene expression dataset. This projection is accomplished by manipulating the first step, which is $\mathrm{PCA_d}$, of the Palantir algorithm’s dimensionality reduction. The innovative manipulation of the $\mathrm{PCA_d}$ step is to project the secondary dataset onto the manifold on the primary dataset, one cell at a time. The technique of using $\mathrm{PCA_d}$ to project the secondary dataset onto the primary dataset is effective because the influence of the secondary dataset is directly proportional to the ratio of the number of cell datapoints in the secondary dataset to the number of cell datapoints in the primary dataset. In most cases the primary dataset will include more than 4000 cell datapoints [1,6] which implies that the influence of a single secondary cell datapoint is negligible during the $\mathrm{PCA_d}$ calculations. A visual validation of the projection method, and the limiting visual distortion it has on a developed phenotypic manifold of a primary dataset can be seen, when comparing the original phenotypic manifold of Figure 6.7 to the projected phenotypic manifold in Figure 6.13.

Owing to the novel data projection method, the Palantir algorithm does not detect all of the same terminal cell datapoints in human bone marrow single-cell gene expression data after projection, as seen in Figure 6.13. The terminal cell datapoint is missing because the Palantir algorithm approach to terminal state detection is statistical. This statistical approach develops a Markov chain and assigns terminal states based on converging extremes of the diffusion components (Appendix: A.4.1). Hence, the projected data does not converge to all the same extremes as the non-projected data and therefore does not have all the same statistical terminal cell datapoints.

The expression of lineage-specific genes, *CD34*, *MPO* and *GATA1* assisted in identifying PC-lineages detected by BAGEL of the mouse bone marrow dataset, the human bone marrow dataset and the projection of the human UCB datset onto the human bone marrow dataset, as shown in Figure 6.2, Figure 6.8 and Figure 6.14.

The second validation of the projection method and the most important is a biological validation. The projection output of BAGEL in Figure 6.14 is deemed biologically accurate as it is in line with other studies (see Section 2.4), where early HSPCs (blue hexagon) show direct differentiation into the megakaryocyte-erythroid PC-lineage through the expression of GATA1. The presence of myeloid-biased HSPCs are identified through expression of MPO. The projected UCB CD34+ HSPCs (black dots) on the contrary show minimal direct differentiation from early HSPCs towards the megakaryocyte/erythroid PC-lineage. An increased number of UCB-derived HSPCs is seen towards the myeloid PC-lineage and capable of differentiating into the megakaryocyte-erythroid PC-lineage. This suggests that UCB CD34+ HSPCs consist of heterogeneous MPPs or CMPs with PC-lineage-biased potential towards the megakaryocyte-erythroid PC-lineage.

The biology is also confirmed in Figure 6.13, where human UCB *CD34*+ cells are spread out in a triangle-like arrangement with cells closer to the starting cell express higher levels of *CD34* (more primitive), whereas *CD34*+ progenitor cells are already committed to either myeloid or erythoid PC-lineages. As the developed method passed the visual and biological validation in ideal conditions (intraspecies single-cell gene expression projection of the same biological processes), it is assumed that the method of single-cell gene expression projection is sound.

BAGEL is dependent on the output of the Palantir algorithm. The Palantir algorithm produces (i) a well-defined phenotypic manifold of the cell developmental trajectory; (ii) pseudo temporal ordering of the cells; and (iii) terminal states of a given dataset. The first output of BAGEL (after the phenotypic manifold) is well-defined bifurcation points of the cell developmental trajectory. As seen in Figure 6.3, Figure 6.9 and Figure 6.15, the bifurcation points defined by a Gibbs sampler and Bayesian model selection in space $\mathcal{P}$ are at the exact instance where PC-lineages branch from each other. It should be noted that the capability of BAGEL to identify bifurcation points is dependent on well defined terminal states within the single-cell gene expression data. This dependency can be seen when comparing the developed manifolds of human bone marrow in Figure 6.7 and Figure 6.13 where the latter has one less terminal state. Owing to this undefined terminal state in the latter, BAGEL did not identify all of the same PC-lineages within the single-cell gene expression data as seen when comparing Figure 6.9 and Figure 6.15.

Owing to these well-defined bifurcation points in space $\mathcal{P}$, BAGEL was able to construct PC-lineages, as seen in Figure 6.4, Figure 6.10 and Figure 6.16. The significance of these constructed PC-lineages are that all of them can be modelled with individual Gaussian processes to obtain an overall continuous model of cell differentiation in space $\mathcal{P}$. The continuous models of cell differentiation can be seen in Figure 6.5, Figure 6.5 and Figure 6.17. It should be noted that the continuous modelling of the cell developmental trajectory performs well with many of cell datapoints, as this will decrease the uncertainty of a Gaussian process.

The tangent vector of the window-based Frenet frame of the cell developmental trajectory stems from the manner in which BAGEL estimates bifurcation points: (i) by dividing the data into windows, which can be thought of as sequential pseudo-time steps of the cell developmental trajectory; and (ii) by determining if a window is best defined with one or a mixture of two multivariate Gaussian distributions. During the window defining step, BAGEL defines a tangent vector parallel to the cell developmental trajectory of each window, as seen in Figure 4.5. When all these tangent vectors are concatenated, a tangent vector window-based Frenet frame of the entire cell developmental trajectory in space $\mathcal{P}$ is obtained, as seen in Figure 6.6, Figure 6.12 and Figure 6.18. These tangent vectors of the window-based Frenet frames are important as they provided an overview of the most likely future fate of cell differentiation in space $\mathcal{P}$ along its developmental path.

As the projection of the human UCB datast onto the human bone marrow dataset (inter-species projection) was successful, the question of single-cell gene expression data projection between different species was inevitable. The projection of the human UCB dataset onto the mouse bone marrow dataset is therefore shown in Appendix B. Although this projection showed promise, it is outside the scope of this research as intrerspecies research require a lot more investigation and will be included in future work, as seen in Chapter 7.

CHAPTER 7 CONCLUSIONS AND FUTURE WORK

7.1 SUMMARY AND CONCLUSION

Cell differentiation is a complex process that is fundamental in biology. To shed a light on this process, scientists are turning to single-cell gene expression data. Therefore, in this dissertation, a mathematical model was developed named BAGEL: **B**ayesian **A**nalysis of **G**ene **E**xpression **L**ineages. The purpose of BAGEL was to provide novel insights about cell differentiation by modelling single-cell gene expression data. The focus of this study was on haematopoiesis, which is the process of cell differentiation for manufacturing blood cells. Although this study was focused on haematopoiesis, BAGEL should hold for all other forms of single-cell gene expression data as BAGEL is independent of its input data.

The three datasets utilised in this dissertation to investigate haematopoiesis are (i) Lin–c-Kit+Sca-1+ mouse bone marrow single-cell RNA-seq data; (ii) CD34+ human bone marrow single-cell RNA-seq data; and (iii) CD34+ human umbilical cord single-cell RNA-seq data. As these single-cell gene expression datasets all consist of high dimensions, the first part of BAGEL was to address the “curse of dimensionality". Hence, the novelty of BAGEL starts by addressing dimensionality reduction.

BAGEL was able to observe cell differentiation in a combined pseudo-time-principal-component space called space $\mathcal{P}$, by utilising dimensionality reduction techniques and the pseudo-time ordering of cell datapoints. While reducing the dimensionality of single-cell gene expression data for visualisation purposes, BAGEL was able to achieve cell datapoint projection. Projection refers to the process whereby a sub-sampled secondary single-cell gene expression dataset is projected onto the phenotypic manifold of a primary optimally sampled single-cell gene expression dataset, one cell datapoint at a time. This process of projection was accomplished by manipulating PCA_d during the phenotypic manifold development of the primary dataset. To ensure that the implemented projection method is accurate, the method was verified visually and biologically. This verification process was completed under the following conditions (i) the single-cell gene expression data of both the secondary and the primary datasets are from the same species; and (ii) both the secondary and the primary datasets are from the same biological process. The reasoning for these verification conditions are that it was assumed that when the projection capabilities of BAGEL passes with ideal condition it should hold for any condition within BAGEL's limits as stated in Chapter 4.

The projection method was verified visually by observing (i) that the secondary dataset has a limited influence on the phenotypic manifold of the primary dataset; and (ii) that the datasets fit really well with each other. The more important biological verification showed that the projected cell datapoints showed similar gene expressions to their peers of the primary dataset, which was expected from the literature. Therefore, the projection method was deemed biologically sound. Owing to this conclusion, interspecies single-cell gene expression data projection was also investigated, and showed promise, as seen in Appendix B.

BAGEL extends its novelty in the continuous manner it models single-cell gene expression data. Bayesian inference techniques were used to produce highly accurate bifurcation points within a cell's developmental trajectory in space $\mathcal{P}$. Although the assumption of this dissertation was that cell differentiation only bifurcates at most into two different lineages at a given instance, Bayesian inference can be extended to allow for multiple bifurcations at any instance. Owing to the manner in which bifurcation points are determined the tangent vector of a window-based Frenet frame of the cell developmental trajectory was obtained. The tangent vectors of the window-based Frenet frames provided the most likely direction of cell differentiation in space $\mathcal{P}$ after an observed window.

Finally, cell lineages within the space $\mathcal{P}$ that describe the cell developmental trajectory from the start of cell differentiation, till a given terminal state, known as PC-lineages was constructed. BAGEL was able to construct these PC-lineages due to the well-defined bifurcation points. These PC-lineages were effectively modelled with a Gaussian process to provide a continuous view of cell differentiation in space $\mathcal{P}$.

7.2 SUMMARY OF CONTRIBUTIONS

1. Cell differentiation is visualised in a combined pseudo-time-principal-component space known as space $\mathcal{P}$ by using (i) dimensionality reduction techniques on single-cell gene expression datasets; and (ii) the Palantir algorithm.
2. Cell differentiation pseudo-time-intervals are transformed to a representation of sequential windows that are translated and rotated within space $\mathcal{P}$ using the tangent vectors of window-based Frenet frames.
3. Bifurcation points are inferred via Bayesian model selection and a Gibbs sampler along the transformed pseudo-timeline in space $\mathcal{P}$.
4. Cell lineages are constructed within the space $\mathcal{P}$ that describe the cell developmental trajectory from the start of cell differentiation, till a given terminal state, known as PC-lineages.
5. Cell differentiation is represented as a continuous process by modelling the obtain PC-lineages with a Gaussian process.
6. A biologically sound projection method is defined, that can project a sub-sampled secondary single-cell gene expression dataset onto the phenotypic manifold of a primary optimally sampled single-cell gene expression dataset by manipulating PCA_d during dimensionality reduction.
7. The door is opened to visualise and investigate the similarities and differences between intra- and inter-species single-cell gene expression datasets.

7.3 FUTURE WORK

This dissertation has highlighted the promise of utilising Bayesian inference in modelling single-cell gene expression data to better understand cell differentiation. As the modelling of gene expression remains a complex task, there are several different avenues available for future research, including:

1. Eliminate the human factor of BAGEL by implementing a statistical method to initialise the two intervals Δ_t and Δ_w when obtaining a window. This will completely eliminate model tuning.
2. Develop a novel Bayesian approach to obtain the pseudo-time ordering of cells, to allow for a complete Bayesian model.
3. Increase the dimensionality of the mixture models, by increasing the visualisation step PC_v as this will most likely lead to even more accurate bifurcation points.

4. Increase the total number of possible models of Bayesian model selection. The increase in models will remove the assumption that cell differentiation only bifurcates into a maximum of two lineages at a given instance.
5. An exciting field to explore is the projection of human single-cell gene expression data onto mouse single-cell gene expression data, and vice versa. An example of these types of projections is discussed in Appendix B where a human UCB dataset is projected onto the phenotypic manifold of a mouse bone marrow dataset. This field shows a lot of promise and is in high demand as mice are usually used in biomedical research due to their similarities to humans [18]. This projection will contribute to the body of knowledge by providing a more in-depth understanding of the similarities of the biology that exists between humans and mice.
6. Compare the novel BAGEL algorithm to algorithms such as Palantir [1], Slingshot [3], Wishbone [6] etc. to clearly show how BAGEL improves on state-of-the-art algorithms in literature. This analysis should be performed on synthetic datasets as the ground truth for real world data are not known. Finally, this analysis should include visual and empirical information regarding the novelty that BAGEL provides as opposed to other algorithms.

REFERENCES

[1] M. Setty, V. Kiseliovas, J. Levine, A. Gayoso, L. Mazutis, and D. Pe'er, "Characterization of cell fate probabilities in single-cell data with Palantir," *Nature biotechnology*, vol. 37, no. 4, pp. 451–465, 2019.

[2] E.-a. D. Amir, K. L. Davis, M. D. Tadmor, E. F. Simonds, J. H. Levine, S. C. Bendall, D. K. Shenfeld, S. Krishnaswamy, G. P. Nolan, and D. Pe'er, "viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia." *Nature biotechnology*, vol. 31, no. 6, pp. 545–52, Jun 2013.

[3] K. Street, D. Risso, R. B. Fletcher, D. Das, J. Ngai, N. Yosef, E. Purdom, and S. Dudoit, "Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics," *BMC genomics*, vol. 19, no. 1, pp. 477–492, 2018.

[4] M. Plass, J. Solana, F. A. Wolf, S. Ayoub, A. Misios, P. Glažar, B. Obermayer, F. J. Theis, C. Kocks, and N. Rajewsky, "Cell type atlas and lineage tree of a whole complex animal by single-cell transcriptomics," *Science*, vol. 360, no. 6391, pp. 1723–1734, 2018.

[5] A. L. Haber, M. Biton, N. Rogel, R. H. Herbst, K. Shekhar, C. Smillie, G. Burgin, T. M. Delorey, M. R. Howitt, Y. Katz *et al.*, "A single-cell survey of the small intestinal epithelium," *Nature*, vol. 551, no. 7680, pp. 333–339, 2017.

[6] M. Setty, M. D. Tadmor, S. Reich-Zeliger, O. Angel, T. M. Salame, P. Kathail, K. Choi, S. Bendall, N. Friedman, and D. Pe'er, "Wishbone identifies bifurcating developmental trajectories from single-cell data," *Nature biotechnology*, vol. 34, no. 6, pp. 637–645, 2016.

[7] F. Buettner, K. N. Natarajan, F. P. Casale, V. Proserpio, A. Scialdone, F. J. Theis, S. A. Teichmann, J. C. Marioni, and O. Stegle, “Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells,” *Nature biotechnology*, vol. 33, no. 2, pp. 155–162, 2015.

[8] C. Trapnell, D. Cacchiarelli, J. Grimsby, P. Pokharel, S. Li, M. Morse, N. J. Lennon, K. J. Livak, T. S. Mikkelsen, and J. L. Rinn, “The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells,” *Nature biotechnology*, vol. 32, no. 4, pp. 381–391, 2014.

[9] S. C. Bendall, K. L. Davis, E.-a. D. Amir, M. D. Tadmor, E. F. Simonds, T. J. Chen, D. K. Shenfeld, G. P. Nolan, and D. Pe’er, “Single-cell trajectory detection uncovers progression and regulatory coordination in human b cell development,” *Cell*, vol. 157, no. 3, pp. 714–725, 2014.

[10] M. A. Rieger and T. Schroeder, “Hematopoiesis,” *Cold Spring Harbor perspectives in biology*, vol. 4, no. 12, p. a008250, 2012.

[11] R. Bacher and C. Kendziorski, “Design and computational analysis of single-cell RNA-sequencing experiments,” *Genome biology*, vol. 17, no. 1, p. 63, 2016.

[12] L. Zhu, J. Lei, B. Devlin, and K. Roeder, “A unified statistical framework for single cell and bulk RNA sequencing data,” *The annals of applied statistics*, vol. 12, no. 1, p. 609, 2018.

[13] N. Friedman, M. Linial, I. Nachman, and D. Pe’er, “Using Bayesian networks to analyze expression data.” *Journal of computational biology : a journal of computational molecular cell biology*, vol. 7, no. 3-4, pp. 601–20, 2000.

[14] N. Campbell, J. Reece, L. Urry, M. Cain, and S. Wasserman, *Biology: A Global Approach, Global Edition*. Harlow, Essex, England: Pearson Education Limited, 2014.

[15] P. M. Magwene, P. Lizardi, and J. Kim, “Reconstructing the temporal ordering of biological samples using microarray data,” *Bioinformatics*, vol. 19, no. 7, pp. 842–850, 2003.

[16] T. Lönnberg, V. Svensson, K. R. James, D. Fernandez-Ruiz, I. Sebina, R. Montandon, M. S. Soon, L. G. Fogg, A. S. Nair, U. Liligeto *et al.*, "Single-cell RNA-seq and computational analysis using temporal mixture modelling resolves Th1/Tfh fate bifurcation in malaria," *Science immunology*, vol. 2, no. 9, 2017.

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

[18] E. C. Bryda, "The Mighty Mouse: the impact of rodents on advances in biomedical research," *Missouri medicine*, vol. 110, no. 3, p. 207, 2013.

[19] J. M. Irish, R. Hovland, P. O. Krutzik, O. D. Perez, Ø. Bruserud, B. T. Gjertsen, and G. P. Nolan, "Single cell profiling of potentiated phospho-protein networks in cancer cells," *Cell*, vol. 118, no. 2, pp. 217–228, 2004.

[20] J. H. Levine, E. F. Simonds, S. C. Bendall, K. L. Davis, D. A. El-ad, M. D. Tadmor, O. Litvin, H. G. Fienberg, A. Jager, E. R. Zunder *et al.*, "Data-driven phenotypic dissection of AML reveals progenitor-like cells that correlate with prognosis," *Cell*, vol. 162, no. 1, pp. 184–197, 2015.

[21] G. Sanguinetti *et al.*, *Gene Regulatory Networks*. New York, NY: Springer, 2019.

[22] A.-E. Saliba, A. J. Westermann, S. A. Gorski, and J. Vogel, "Single-cell RNA-seq: advances and future challenges," *Nucleic acids research*, vol. 42, no. 14, pp. 8845–8860, 2014.

[23] X. Wang, J. Park, K. Susztak, N. R. Zhang, and M. Li, "Bulk tissue cell type deconvolution with multi-subject single-cell expression reference," *Nature communications*, vol. 10, no. 1, p. 380, 2019.

[24] F. Tang, C. Barbacioru, Y. Wang, E. Nordman, C. Lee, N. Xu, X. Wang, J. Bodeau, B. B. Tuch, A. Siddiqui *et al.*, "mRNA-Seq whole-transcriptome analysis of a single cell," *Nature methods*, vol. 6, no. 5, pp. 377–382, 2009.

[25] B. Hwang, J. H. Lee, and D. Bang, "Single-cell RNA sequencing technologies and bioinformatics pipelines," *Experimental & molecular medicine*, vol. 50, no. 8, pp. 1–14, 2018.

[26] L. Haghverdi, M. Buettner, F. A. Wolf, F. Buettner, and F. J. Theis, "Diffusion pseudotime robustly reconstructs lineage branching," *Nature methods*, vol. 13, no. 10, p. 845, 2016.

[27] A. A. Pollen, T. J. Nowakowski, J. Shuga, X. Wang, A. A. Leyrat, J. H. Lui, N. Li, L. Szpankowski, B. Fowler, P. Chen, N. Ramalingam, G. Sun, M. Thu, M. Norris, R. Lebofsky, D. Toppani, D. W. Kemp, M. Wong, B. Clerkson, B. N. Jones, S. Wu, L. Knutsson, B. Alvarado, J. Wang, L. S. Weaver, A. P. May, R. C. Jones, M. A. Unger, A. R. Kriegstein, and J. A. A. West, "Low-coverage single-cell mRNA sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex." *Nature biotechnology*, vol. 32, no. 10, pp. 1053–1058, Oct 2014.

[28] T. S. Andrews and M. Hemberg, "Identifying cell populations with scRNASeq," *Molecular aspects of medicine*, vol. 59, pp. 114–122, 2018.

[29] J. H. L. Byungjin Hwang and D. Bang, "Single-cell RNA sequencing technologies and bioinformatics pipelines," *Experimental and Molecular Medicinevolume*, vol. 50, 2018.

[30] J. Farrell, Robert E., *RNA methodologies : a laboratory guide for isolation and characterization*, 4th ed. Amsterdam: Elsevier/Academic Press, 2010.

[31] A. Bergkvist, V. Rusnakova, R. Sindelka, J. M. A. Garda, B. Sjögreen, D. Lindh, A. Forootan, and M. Kubista, "Gene expression profiling–Clusters of possibilities," *Methods*, vol. 50, no. 4, pp. 323–335, 2010.

[32] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns." *Proceedings of the National Academy of Sciences of the United States of America*, vol. 95, no. 25, pp. 14 863–8, Dec 1998.

[33] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher, "Comprehensive identification of cell cycle–regulated genes of the

yeast Saccharomyces cerevisiae by microarray hybridization," *Molecular biology of the cell*, vol. 9, no. 12, pp. 3273–3297, 1998.

[34] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine, "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *Proceedings of the National Academy of Sciences*, vol. 96, no. 12, pp. 6745–6750, 1999.

[35] A. Ben-Dor, R. Shamir, and Z. Yakhini, "Clustering gene expression patterns," *Journal of computational biology*, vol. 6, no. 3-4, pp. 281–297, 1999.

[36] N. Vijesh, S. K. Chakrabarti, and J. Sreekumar, "Modeling of gene regulatory networks: a review," *Journal of Biomedical Science and Engineering*, vol. 6, no. 02, pp. 223–231, 2013.

[37] R. Cannoodt, W. Saelens, and Y. Saeys, "Computational methods for trajectory inference from single-cell transcriptomics," *European journal of immunology*, vol. 46, no. 11, pp. 2496–2506, 2016.

[38] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker, "Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps," *Proceedings of the national academy of sciences*, vol. 102, no. 21, pp. 7426–7431, 2005.

[39] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. Cambridge, London, England: The MIT Press, 2005.

[40] J. E. Reid and L. Wernisch, "Pseudotime estimation: deconfounding single cell time series," *Bioinformatics*, vol. 32, no. 19, pp. 2973–2980, 2016.

[41] R. B. Gramacy and H. K. H. Lee, "Bayesian treed Gaussian process models with an application to computer modeling," *Journal of the American Statistical Association*, vol. 103, no. 483, pp. 1119–1130, 2008.

[42] W. Chu and Z. Ghahramani, "Gaussian processes for ordinal regression," *Journal of machine learning research*, vol. 6, no. Jul, pp. 1019–1041, 2005.

[43] K. Campbell and C. Yau, "Bayesian Gaussian Process latent variable Models for pseudotime inference in single-cell RNA-seq data," *bioRxiv*, p. 026872, 2015.

[44] K. R. Campbell and C. Yau, "Orinty: robust differential expression analysis using probabilistic models for pseudotime inference," *PLoS computational biology*, vol. 12, no. 11, p. e1005212, 2016.

[45] S. Ahmed, M. Rattray, and A. Boukouvalas, "GrandPrix: scaling up the bayesian GPLVM for single-cell data," *Bioinformatics*, vol. 35, no. 1, pp. 47–54, 2019.

[46] A. Damianou, C. Ek, M. Titsias, and N. Lawrence, "Manifold relevance determination," *arXiv preprint arXiv:1206.4610*, 2012.

[47] S. Eleftheriadis, O. Rudovic, and M. Pantic, "Discriminative shared gaussian processes for multiview and view-invariant facial expression recognition," *IEEE transactions on image processing*, vol. 24, no. 1, pp. 189–204, 2014.

[48] J. D. Welch, A. J. Hartemink, and J. F. Prins, "MATCHER: manifold alignment reveals correspondence between single cell transcriptome and epigenome dynamics," *Genome biology*, vol. 18, no. 1, pp. 1–19, 2017.

[49] A. Boukouvalas, J. Hensman, and M. Rattray, "BGP: identifying gene-specific branching dynamics from single-cell data with a branching Gaussian process," *Genome biology*, vol. 19, no. 1, p. 65, 2018.

[50] N. Lawrence and A. Hyvärinen, "Probabilistic non-linear principal component analysis with gaussian process latent variable models." *Journal of machine learning research*, vol. 6, no. 11, 2005.

[51] M. Lázaro-Gredilla, S. Van Vaerenbergh, and N. D. Lawrence, "Overlapping mixtures of gaussian processes for the data association problem," *Pattern recognition*, vol. 45, no. 4, pp. 1386–1395, 2012.

[52] M. Kondo, "Lymphoid and myeloid lineage commitment in multipotent hematopoietic progenitors," *Immunological reviews*, vol. 238, no. 1, pp. 37–46, 2010.

[53] A. E. Rodriguez-Fraticelli, S. L. Wolock, C. S. Weinreb, R. Panero, S. H. Patel, M. Jankovic, J. Sun, R. A. Calogero, A. M. Klein, and F. D. Camargo, "Clonal analysis of lineage fate in native haematopoiesis," *Nature*, vol. 553, no. 7687, pp. 212–216, 2018.

[54] A. Sanjuan-Pla, I. C. Macaulay, C. T. Jensen, P. S. Woll, T. C. Luis, A. Mead, S. Moore, C. Carella, S. Matsuoka, T. B. Jones *et al.*, "Platelet-biased stem cells reside at the apex of the haematopoietic stem-cell hierarchy," *Nature*, vol. 502, no. 7470, pp. 232–236, 2013.

[55] R. Yamamoto, Y. Morita, J. Ooehara, S. Hamanaka, M. Onodera, K. L. Rudolph, H. Ema, and H. Nakauchi, "Clonal analysis unveils self-renewing lineage-restricted progenitors generated directly from hematopoietic stem cells," *Cell*, vol. 154, no. 5, pp. 1112–1126, 2013.

[56] C. Gekas and T. Graf, "CD41 expression marks myeloid-biased adult hematopoietic stem cells and increases with age," *Blood, The Journal of the American Society of Hematology*, vol. 121, no. 22, pp. 4463–4472, 2013.

[57] J. Carrelha, Y. Meng, L. M. Kettyle, T. C. Luis, R. Norfo, V. Alcolea, H. Boukarabila, F. Grasso, A. Gambardella, A. Grover *et al.*, "Hierarchicalles of multipotent haematopoietic stem cells," *Nature*, vol. 554, no. 7690, pp. 106–111, 2018.

[58] S. Hu, M. Lundgren, and A. J. Niemi, "Discrete Frenet frame, inflection point solitons, and curve visualization with applications to folded proteins," *Physical Review E*, vol. 83, no. 6, p. 061908, 2011.

[59] K. Li, W. Su, and L. Chen, "Performance analysis of three-dimensional differential geometric guidance law against low-speed maneuvering targets," *Astrodynamics*, vol. 2, no. 3, pp. 233–247,

2018.

[60] K. T. W, *Vectors, pure and applied: a general introduction to linear algebra*. Cambridge: Cambridge University Press, 2013.

[61] K. Bury, *Statistical distributions in engineering*. Cambridge University Press, 1999.

[62] K. P. Murphy, "Conjugate bayesian analysis of the gaussian distribution," *def*, vol. 1, no. 2σ2, p. 16, 2007.

[63] S. J. Prince, *Computer vision: models, learning, and inference*. New York, NY: Cambridge University Press, 2012.

[64] J. Franzén, "Bayesian inference for a mixture model using the gibbs sampler," *MResearch Report*, vol. 1, 2006.

[65] A. M. Ellison, "Bayesian inference in ecology," *Ecology letters*, vol. 7, no. 6, pp. 509–520, 2004.

[66] J. VanderPlas, "Frequentism and Bayesianism: A Python-driven Primer," *Proceedings of the 13th Python in Science Conference (SciPy 2014)*, pp. 85–93, 6 - 12 July 2014.

[67] P. B. Herbert Hoijtink, Irene Klugkist, *Bayesian evaluation of informative hypotheses*. New York, NY: Springer, 2008.

[68] N. Friel and J. Wyse, "Estimating the model evidence: a review," *Statistica Neerlandica*, Early view online.

[69] A. R. Webb, *Statistical pattern recognition*, 2nd ed. Chichester, United Kingdom: John Wiley, 2003.

[70] L. Wasserman *et al.*, "Bayesian model selection and model averaging," *Journal of mathematical psychology*, vol. 44, no. 1, pp. 92–107, 2000.

[71] H. Jeffreys, "Theory of probability (3rd edt.) oxford university press," *MR0187257*, 1961.

[72] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian data analysis*. New York, New York: CRC press, 2013.

[73] J. M. Bernardo and A. F. Smith, *Bayesian theory*. United Kingdom, Chichester: John Wiley & Sons, 2009.

[74] S. Chib, "Marginal Likelihood from the Gibbs Output," *Journal of the american statistical association*, vol. 90, no. 432, pp. 1313–1321, 1995.

[75] J. Friedman, T. Hastie, R. Tibshirani *et al.*, *The elements of statistical learning*. Springer series in statistics New York, 2001, vol. 1, no. 10.

[76] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning : with applications in R*, ser. Springer texts in statistics. New York: Springer, 2013.

[77] K. V. Gris, J.-P. Coutu, and D. Gris, "Supervised and unsupervised learning technology in the study of rodent behavior," *Frontiers in behavioral neuroscience*, vol. 11, p. 141, 2017.

[78] K. M. Abadir and J. R. Magnus, *Matrix algebra*. Cambridge: Cambridge University Press, 2005, vol. 1.

[79] F. Wood and M. J. Black, "A nonparametric Bayesian alternative to spike sorting," *Journal of neuroscience methods*, vol. 173, no. 1, pp. 1–12, 2008.

[80] M. Bartcus, "Bayesian non-parametric parsimonious mixtures for model-based clustering," Ph.D. dissertation, Université de Toulon, 2015.

[81] C. Shalizi, "Advanced data analysis from an elementary point of view," *UK: Cambridge University Press*, 2016.

[82] J. De la Porte, B. Herbst, W. Hereman, and S. Van Der Walt, "An introduction to diffusion maps," in *Proceedings of the 19th Symposium of the Pattern Recognition Association of South Africa (PRASA 2008), Cape Town, South Africa*, 2008, pp. 15–25.

[83] R. Talmon, I. Cohen, S. Gannot, and R. R. Coifman, "Diffusion maps for signal processing: A deeper look at manifold-learning techniques based on kernels and graphs," *IEEE signal processing magazine*, vol. 30, no. 4, pp. 75–86, 2013.

[84] M. D. Luecken and F. J. Theis, "Current best practices in single-cell RNA-seq analysis: a tutorial," *Molecular systems biology*, vol. 15, no. 6, 2019.

[85] A. Platzer, "Visualization of SNPs with t-SNE," *PloS one*, vol. 8, no. 2, p. e56883, 2013.

[86] R. C. Penney, *Linear algebra: Ideas and applications*. United States - electronic book: John Wiley & Sons, 2021.

[87] F. Paul, Y. Arkin, A. Giladi, D. A. Jaitin, E. Kenigsberg, H. Keren-Shaul, D. Winter, D. Lara-Astiaso, M. Gury, A. Weiner *et al.*, "Transcriptional heterogeneity and lineage commitment in myeloid progenitors," *Cell*, vol. 163, no. 7, pp. 1663–1677, 2015.

[88] A. M. Bolger, M. Lohse, and B. Usadel, "Trimmomatic: a flexible trimmer for Illumina sequence data," *Bioinformatics*, vol. 30, no. 15, pp. 2114–2120, 2014.

[89] R. V. Guimera, "bcbio-nextgen: Automated, distributed next-gen sequencing pipeline," *EMBnet. journal*, vol. 17, no. B, p. 30, 2011.

[90] D. Kim, B. Langmead, and S. L. Salzberg, "HISAT: a fast spliced aligner with low memory requirements," *Nature methods*, vol. 12, no. 4, pp. 357–360, 2015.

[91] Y. Liao, G. K. Smyth, and W. Shi, "featureCounts: an efficient general purpose program for assigning sequence reads to genomic features," *Bioinformatics*, vol. 30, no. 7, pp. 923–930, 2014.

APPENDIX A THE PALANTIR ALGORITHM

A.1 PALANTIR ALGORITHM OPERATION

A flow diagram of the Palantir algorithm's approach to model cell differentiation is shown in Figure A.1. The algorithm starts by developing a phenotypic manifold (PM.1) on the cell differentiation data. This is used to develop a pseudo-time (PT.1) of the entire process from a start cell to all possible final fates. The pseudo-time is used to develop a directed manifold of the data. The directed data is then used to define an absorbing Markov chain (MP.1) with branch probabilities. These probabilities define the likelihood of a given cell to differentiate into possible final fates. Finally, gene expression trends (GE.1) are estimated with generalised additive models (GAMs).

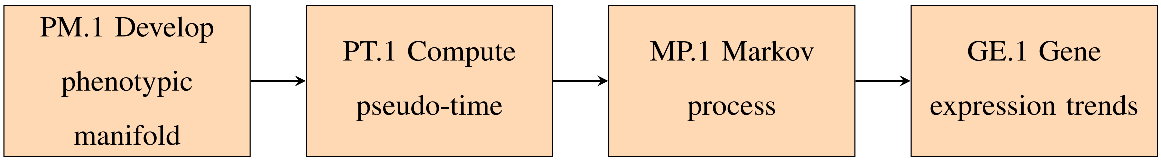


Figure A.1. The Palantir algorithm operational flow diagram. The Palantir algorithm starts by defining a phenotypic manifold (PE.1) of the data set. This phenotypic manifold is then used to develop a pseudo-time ordering (PT.1) of the cell differentiation process. Finally, a directed Markov chain (MC.1) is inferred from the pseudo-time to develop gene expression trends (GE.1).

A.2 DEVELOPMENT OF PHENOTYPIC MANIFOLD

To develop the phenotypic manifold, it is first required to construct a nearest neighbour graph based on the robust trends of the data. This is accomplished by utilising diffusion maps which approximate the differentiation landscape of the data by projecting them onto a low-dimensional manifold. Diffusion maps require the definition of a measure that represents the similarity between cells. In the Palantir algorithm, the measure implemented is the Euclidean distance.

Consider a cell-by-gene matrix of counts single-cell gene expression data set $X \in \Re^{N \times M}$ where N denotes the cells and M denotes the number of genes. This can be used to construct a k-nearest neighbour graph $G_x \in \Re^{N \times N}$ which utilises Euclidean distances to represent the similarity between cells. The calculated distances are then converted to an affinity matrix, which ensures an exponential decrease between the similarity of cells with an increase of distance. This is accomplished with an adaptive (width) Gaussian kernel defined as

$$K(x_i,\, x_j) = \frac{1}{\sqrt{|2\pi(\sigma_i + \sigma_j)|}} \cdot \exp\left(-0.5 \frac{(x_i - x_j)(x_i - x_j)^T}{(\sigma_i + \sigma_j)} \right), \tag{A.1}$$

where x_i is the i^{th} cell's gene expression vector with a scaling factor of the i^{th} cell defined as

$$\sigma_i, \tag{A.2}$$

where σ_i is the distance to the l^{th} neighbour and $l < k$. The final step in developing the manifold is to take the Laplacian of the obtained affinity matrix $K \in \Re^{N \times N}$. This allows for the development of a diffusion operator $T \in \Re^{N \times N}$, which defines the probability of reaching cell j in one step from cell i as T_{ij}. The eigenvectors of T are known as the top diffusion components and are used to define the low-dimensional embedding that approximates the phenotypic manifold.

A.3 DEVELOPMENT OF PSEUDO-TIME ORDERING OF CELLS

The Palantir algorithm uses multiple diffusion components to develop a pseudo-time ordering of cells. These diffusion components are then used to construct a more reliable nearest-neighbour graph $G_E \in \Re^{N \times N}$ with multi-scale distances. The implementation of multi-scale distances removes most of the noise that was obtained from the original neighbour graph. Next, the pseudo-time of cell differentiation is determined by using the shortest path distance of G_E. The obtained pseudo-time may accumulate noise with an increase in distance. A method to counteract the possible noise is to implement waypoints. Waypoints are used as guides where the waypoint closest to a cell gets the highest vote in estimating the cell's position along the pseudo-time. The sampling of the waypoints is determined with max-min sampling to ensure that the complete landscape of the dataset is covered.

It should be noted that the pseudo-time is initialised with the shortest path distance from a user-defined start cell. It then converges to the extremes of the diffusion components which depict the boundaries of the phenotypic space. This means that steps of determining the pseudo-time can be defined by using the steps below.

A.3.1 Multi-scale distances

In order to calculate multi-scale distances, it is necessary to define the manifold as $E \in \Re^{N \times L}$, with the embedding dimension denoted by L. The dimension of the embedding (L) is obtained by the eigengap between the top eigenvectors. This is accomplished by defining the eigenvalues associated with the eigenvectors of the diffusion components as $\lambda_1,\, \lambda_2,\, \cdots,\, \lambda_L$. By definition, of eigenvectors these eigenvalues uphold the following condition: $1 > \lambda_1 > \lambda_2 > \cdots > \lambda_L > 0$. This allows for the multi-scale distances between cells to be defined as

$$MS(e_i,\, e_j)^2 = \sum_{l=1}^{L} \left(\frac{\lambda_l}{1-\lambda_l} \right)^2 \times (e_i^{(l)} - e_j^{(l)})^2, \tag{A.3}$$

where $e_i^{(l)}$ and $e_j^{(l)}$ represent the embedding of cells i and j respectively along the diffusion component l.

A.3.2 Max-min waypoint sampling

The max-min waypoint sampling is initialised by randomly sampling a cell from a given diffusion component with

$$WS^{(l)} = random(N,\, 1), \tag{A.4}$$

where $WS^{(l)}$ is the waypoint set of diffusion component l, initialised by the python function sample.random(N,1), which returns one random sampled cell from all the cells within the data set (N). The next step is to compute the distances along the diffusion component to the current waypoint set, for each cell with

$$wd_{ij} = \left(e_i^{(l)} - e_i^{(l)} \right)^2 \forall j \in WS^{(l)}. \tag{A.5}$$

Next, the minimum waypoint distance of the current distances is computed for each cell i with

$$md_i = min(wd_{ij}) |\, j \in WS^{(l)}. \tag{A.6}$$

The final step is to add the cell to the waypoint set that has the maximum of these minimum distances with

$$WS^{(l)} = \cup \left(WS^{(l)},\, argmin(md) \right). \tag{A.7}$$

This process is repeated until a predetermined number of waypoints is sampled.

A.3.3 Iterative pseudo-time computation

This is the main step in estimating the pseudo-time for the cell differentiation process. The iterative approach to the pseudo-time first requires the estimation of the boundary cells within the differentiation process with

$$C = \cup_{l=1}^{L} (argmin\, E^{(l)}, argmax\, E^{(l)}), \tag{A.8}$$

where E represents the manifold and $E^{(l)}$ corresponds to the l^{th} diffusion component of the manifold. Next, the initialisation of the pseudo-time $\tau_i^{(0)}$ is enabled by determining the shortest path distance from a start cell s'. The start cell is defined with a user-defined early cell s as

$$s' = argmin_{i \in C} MS(e_s,\, e_i). \tag{A.9}$$

The start cell is then used to compute waypoint perspectives. This is to ensure that the closest waypoints to a given cell influence the pseudo-time the most. The perspective V_{wi} of cell i with respect to waypoint w is computed with the multi-scale distance to the start cell s' with

$$V_{wi} := \begin{cases} \tau_w^{(0)} + D_{wi} & \text{if } \tau_i^{(0)} > \tau_w^{(0)}, \\ \tau_w^{(0)} - D_{wi} & \text{otherwise,} \end{cases} \tag{A.10}$$

where D_{wi} is defined as the shortest path distance from cell i to waypoint w and $\tau_i^{(0)}$ is the pseudo-time of cell i, which is initialised as the shortest path distance from s' and $\tau_w^{(0)}$ is the pseudo-time of the waypoint w. These perspectives are then used to calculate a weighted average to refine the pseudo-time. The weighted average is an exponentially decreasing function that decreases with an increase in distance calculated with

$$\tau_i^{(1)} = \sum_{w \in WS} V_{wi} \times W_{wi}, \tag{A.11}$$

where the weights are defined as

$$W_{wi} = exp\left(\frac{-D_{wi}^2}{\sigma}\right) \Big/ \sum_{k=1:N} exp\left(\frac{-D_{wi}^2}{\sigma}\right), \tag{A.12}$$

and where the standard deviation of the distance matrix D is defined as σ.

A.4 DEVELOPMENT OF MARKOV CHAIN

As cell differentiation is a directed process, it is first necessary to convert the undirected graph of the pseudo-time (G'_E) to a directed graph ($G_D \in \Re^{N \times N}$). The directed graph is calculated with

$$G_{D_{ij}} := \begin{cases} G'_{E_{ij}} & \text{if } \tau_i < \tau_j, \\ G'_{E_{ij}} & \text{if } \tau_i > \tau_j \text{ and } \tau_i - \tau_j < \sigma_i, \\ 0 & \text{if } \tau_i > \tau_j \text{ and } \tau_i - \tau_j > \sigma_i, \end{cases} \tag{A.13}$$

where an undirected edge is converted to a directed edge between cell i and its neighbouring cell j, when the inferred pseudo-time of cell i (τ_i) is smaller than its neighbours' inferred pseudo-time of cell j (τ_j). The edges are pruned between cell i and cell j when; (i) $\tau_i > \tau_j$; and (ii) the estimated distances between cell i and cell j exceed the scaling factor of cell i, which is calculated with Equation 2.

The distances of the directed graph are then used to determine an affinity matrix $Z \in \Re^{nW \times nW}$, with Equation A.1, where nW is the number of waypoints. This provides the transition probabilities for the construction of the Markov chain. The affinity matrix is used to construct a transition probability matrix P of the Markov chain with a state transition matrix A, with

$$P_{ij} = \frac{Z_{ij}}{\sum_k Z_{ik}}, \tag{A.14}$$

where the probability of reaching cell state j from cell state i in one step is denoted by P_{ij}. Finally, terminal states (final fate) of the cell differentiation process can be set by prior knowledge or they can be computed directly from the Markov chain. This is accomplished by converting the directed Markov chain to an absorbing Markov chain with

$$A_{ij} = 0 | i \in TS\,;\; j = 1, \cdots, nW, \tag{A.15}$$

where TS is a set of terminal states. The subsequent sections provide a more complete description of the terminal states estimation and the characterisation of a cell's fate/differentiation potential.

A.4.1 Identifying terminal states

Terminal states in the Markov chain can be identified by random walks, as these walks are directed towards the extrema of the diffusion components (boundary cells, C). The Markov chain is time-invariant, implying that it is a steady-state distribution. This means that a stationary distribution can be defined as the probability distribution over the states of the chain. A Gaussian percent point function ($\mathcal{N}_{ppf}$) with the assistance of a median absolute deviation utilised as the scale of the stationary distribution can be used to determine outliers (extrema of the diffusion components) within the distribution. The outliers can therefore be computed with

$$TS^o = \{i | \pi_i > \mathcal{N}_{ppf}(0.9999,\, Median(\pi), sc)\}, \tag{A.16}$$

where TS^o is the calculated terminal states outliers, π the stationary distribution and the median absolute deviation computed with

$$sc = Median(\pi_i - Median(\pi)). \tag{A.17}$$

The set of states TS^{cands} that correspond to diffusion component extremes is chosen as the terminal states with

$$TS = \cap(TS^o,\, C). \tag{A.18}$$

A.4.2 Characterisation of a cell fate/differentiation potential

The aim of this section is to calculate a branch probability vector B_i for each cell. The branch probability vector denotes the probability that each cell might reach the absorbing terminal state b. This can be computed with random walks through the Markov chain, which denotes the probability of a cell starting at an intermediate state and reaching a specific terminal state. Owing to the implementation of an absorbing Markov chain, the state transition matrix can be represented by

$$A = \begin{bmatrix} Q & R \\ 0 & I \end{bmatrix}, \tag{A.19}$$

where the transition probabilities between intermediate states (probability between transient states) are represented with Q which is a $(nW-b)\times(nW-b)$ sub-matrix, the probabilities between intermediate states and terminal states (probability from transient states to absorbing states) are represented with R which is a $(nW-b)\times(b)$ sub-matrix and I is an identity matrix $b\times b$. This is used to determine the fundamental matrix of the Markov. The fundamental matrix is used to provide the expected number of times that the process (cell differentiation) is in the transient state j (cell state j) if it started in the transient state i (cell state i) and is defined as

$$F = (1-Q)^{-1}, \tag{A.20}$$

where F_{ij} is the probability of reaching the terminal state j from an intermediate stage i in steps $1,\ ,2\ ,\cdots,\ \infty$. The fundamental matrix is then used to compute the differentiation probabilities with

$$B = F\times R, \tag{A.21}$$

where the probability of a cell in intermediate stage i reaching the terminal state j in steps $1,\ ,2\ ,\cdots,\ \infty$ is denoted with B_{ij}. The distribution of B is multinomial, which means that $\sum_j B_{ij} = 1$. This implies that the branch probabilities of the terminal states (when the cell is at a terminal state) can be defined as

$$B_{ij} = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j, \end{cases} \tag{A.22}$$

which implies that the probability of the terminal state is 1 when cell i is the same as cell j and zero otherwise. The final step in calculating the branch probability vector is to multiply the waypoint weighted average calculated with Equation A.11. This means that the branch probability vector for cell differentiation (for each individual cell in the manifold) can be defined as

$$B_{ij} = \sum_{w\in WS} B_{wj}\times W_{wi}. \tag{A.23}$$

A.5 GENE EXPRESSION PREDICTION

The Palantir algorithm uses generalised additive models (GAMs) to estimate gene expression trends along specific lineages. The gene expression of gene g in a given branch b can therefore be defined as

$$y_{gi} = \beta_0 + f(\tau_i)\ for\ i \in B_{ib} > 0, \tag{A.24}$$

where the expression of gene g in cell i is defined as y_{gi}, the pseudo-time ordering of cell i is denoted as τ_i and f is a non-parametric function.

APPENDIX B MODELLING OF INTERSPECIES GENE EXPRESSIONS

B.1 APPENDIX OBJECTIVES

This Appendix provides an overview of the projection of human UCB single-cell gene expression data onto the phenotypic manifold of the mouse bone marrow single-cell gene expression data. These results show promise to investigate the similarities between mice and human but were excluded from Chapter 6 as they do not fall in the scope of this dissertation.

B.2 EXPERIMENTAL SETUP FOR DATA COLLECTION

Data collection of BAGEL is simple, as the modelling of cell differentiation is analogous to a black box model. This analogy is because BAGEL allows several input parameters and converts them to a mathematical representation of single-cell gene expression data. The input parameters are (i) dataset/s (note: the format of a high dimensional dataset should be a cell-by-gene matrix of counts); (ii) early cell[1]; (iii) Δ_t; and (iv) Δ_{w}. Hence, the data collection and visualisation of the projection of the human UCB dataset[2] onto the mouse bone marrow dataset is obtained by setting the input parameters of BAGEL to the values of Table B.1.

[1] Early cell: Defines a cell at the start of the cell differentiation process required by the Palantir algorithm [1].

[2] For the sake of brevity single-cell gene expression dataset will also be referred to as dataset.

Table B.1. Experimental setup for Appendix B

	Result 4
Primary dataset	Mouse dataset
Secondary dataset	Human dataset 2
Early cell	W30258
two_data_set_FLAG [a]	True
new_manifold_FLAG [b]	True
Δ_t	200
Δ_w	150
Gibbs samples	2000
Burn-in period	500

[a] two_data_set_FLAG: Flag defining whether one or two datasets are used..

[b] new_manifold_FLAG: This flag defines if a new Palantir phenotypic manifold should be developed. This flag can be set to false after the first iteration of the algorithm as the phenotypic manifold only has to be defined once for operation.

B.3 LINEAGE IDENTIFICATION

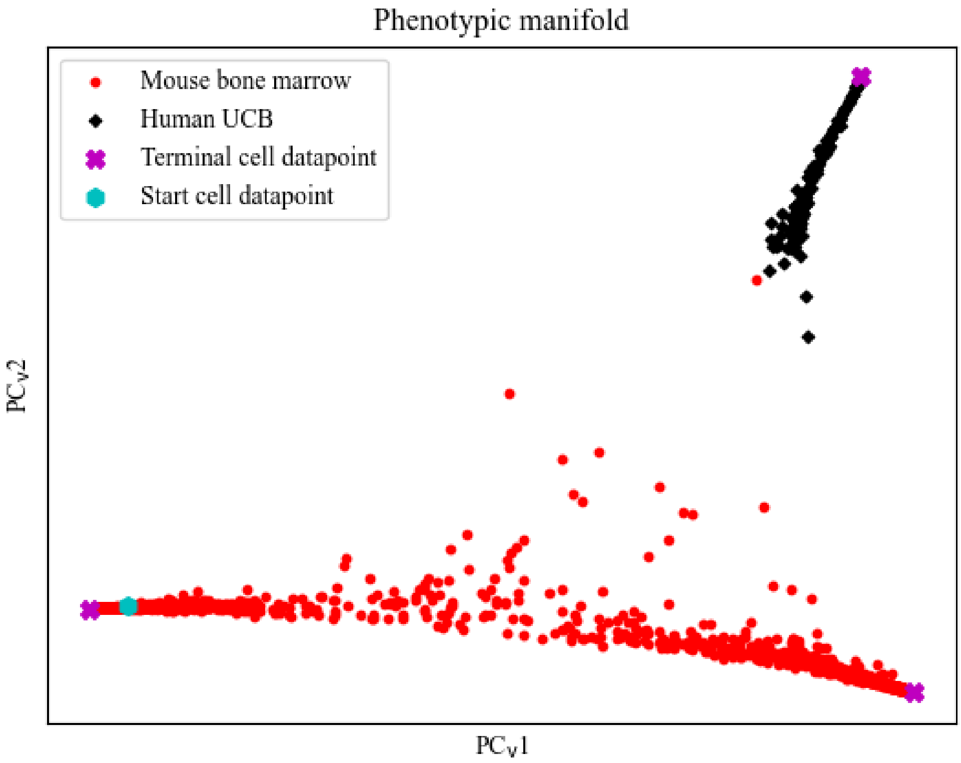
As BAGEL only provide mathematical knowledge about the single-cell gene expression data, prior biological knowledge is used to interpret all of its detected PC-lineages. The prior biological knowledge is that cluster of differentiation 34 (*CD34*) is abundantly expressed in HSPCs, whereas myeloperoxidase (*MPO*) and GATA-binding factor 1 (*GATA1*) are expressed in cells of myeloid and early erythroid origin, respectively. Hence, the expression of these molecular markers assist in identifying HSPCs, myeloid and erythroid PC-lineages.

B.4 MODELLING OF INTERSPECIES GENE EXPRESSIONS

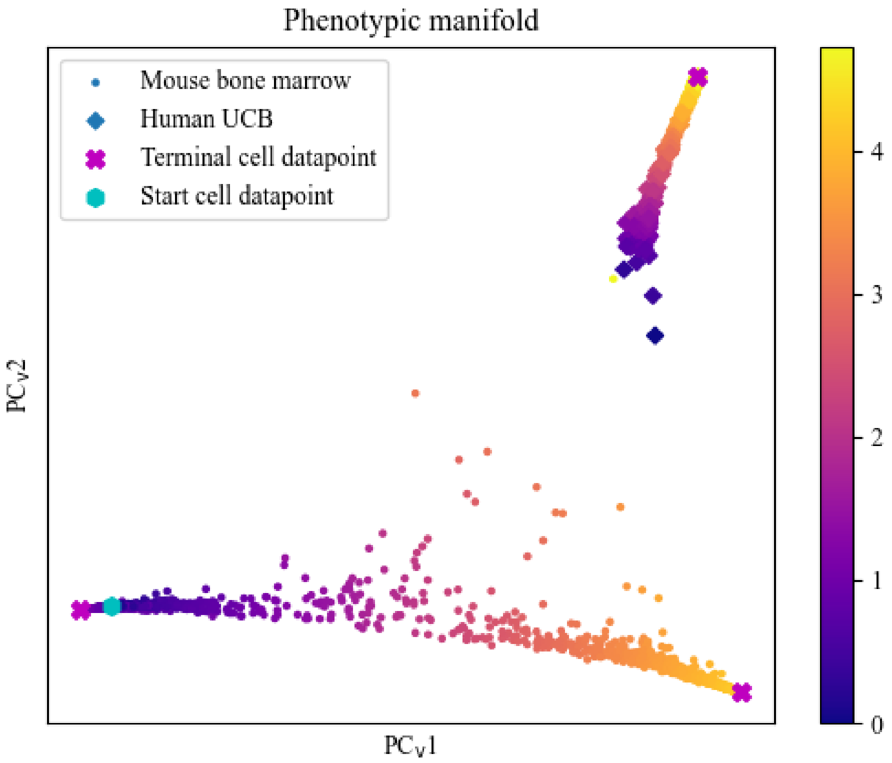
The following results, which are visualised and discussed, are those of the modelling of the projection of human UCB single-cell gene expression data onto mouse bone marrow single-cell gene expression data (see Table B.1). For the sake of brevity this data projection will be referred to as the iterspecies dataset.

B.4.1 Phenotypic manifold

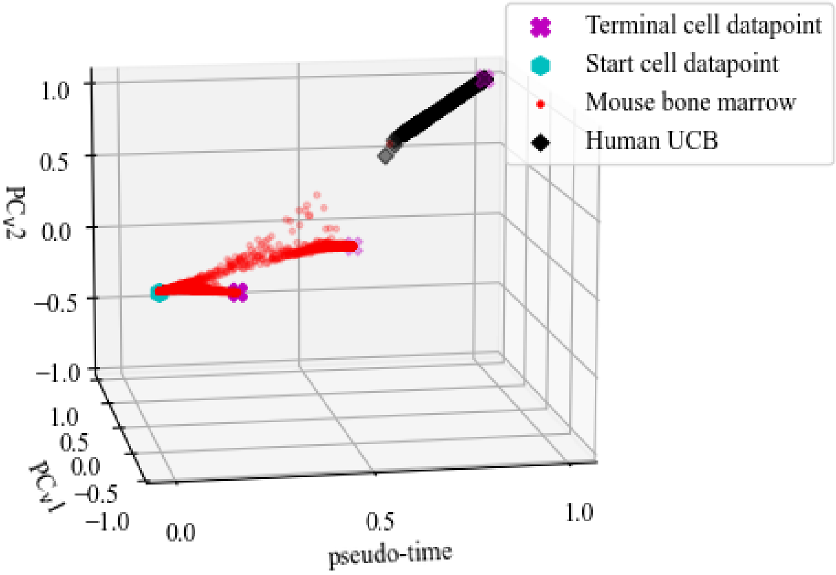
Figure B.1 shows the phenotypic manifold of the iterspecies dataset. In Figure B.1, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each dot represents a mouse bone marrow cell datapoint; (iv) whereas the projected human UCB cell datapoints onto the phenotypic manifold is represented with diamonds; and (iv) the legend uses blue to indicate the start of the pseudo-time, and yellow the end.



(a) The obtained two-dimensional phenotypic manifold of the iterspecies dataset.



(b) The obtained two-dimensional phenotypic manifold of the iterspecies dataset with its pseudo-time.



(c) The obtained three-dimensional phenotypic manifold of the iterspecies dataset in space $\mathcal{P}$.

Figure B.1. The obtained phenotypic manifold of human of the iterspecies dataset. In these figures, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each dot represents a mouse bone marrow cell datapoint; (iv) whereas the projected human UCB cell datapoints onto the phenotypic manifold is represented with diamonds; and (iv) the legend uses blue to indicate the start of the pseudo-time, and yellow the end.

B.4.2 Gene expressions visualised on the obtained phenotypic manifold

As expected, Figure B.2 shows increased expression of CD34 closer to the starting cell, i.e. start of cell differentiation, while expression of MPO and GATA1 indicates the presence of myeloid and erythroid precursor cells, respectively. In Figure B.2, (i) each dot represents a mouse bone marrow cell datapoint; (ii) whereas projected human UCB cell datapoints onto the phenotypic manifold is represented with diamonds;(iii) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset is indicated with a blue hexagon; (iv) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; and (v) the level of gene expression is indicated with the colour bar, where blue indicates no expression, red indicates high expression and orange indicates intermediate expression.

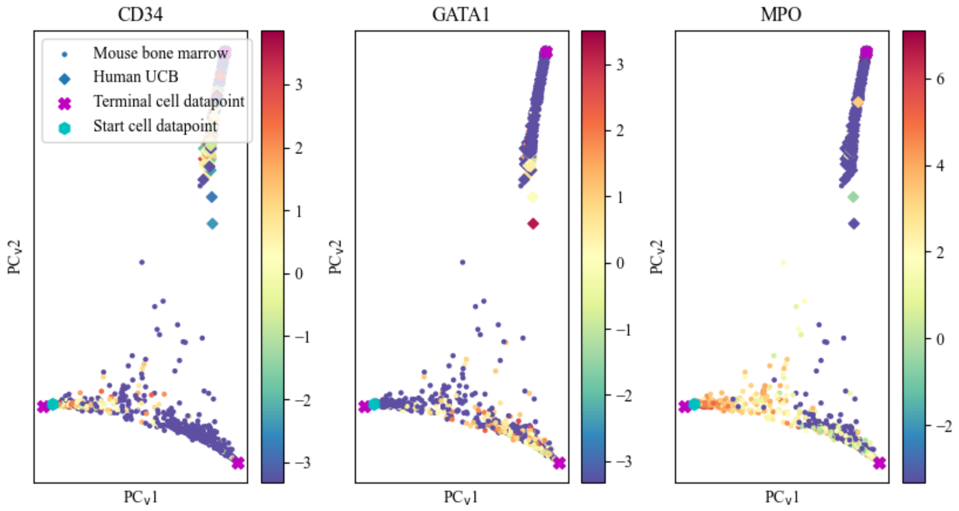
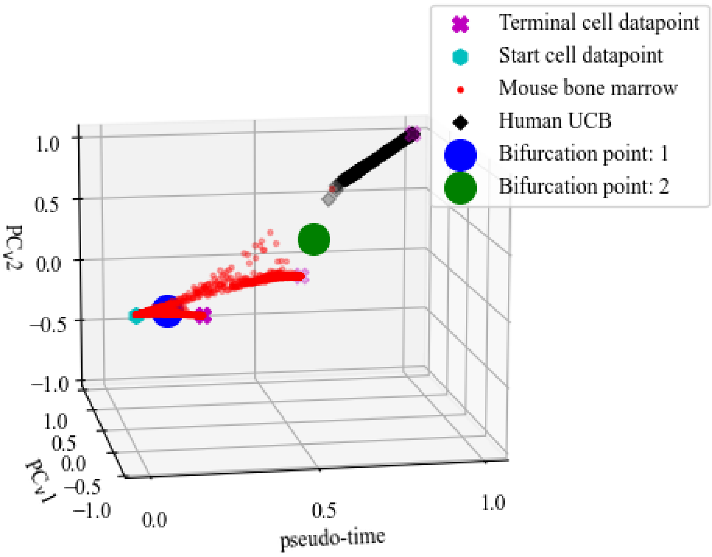


Figure B.2. Gene expression of lineage-specific genes. Expression of HSPC gene (CD34), myeloid- (MPO) and erythroid-specific genes (GATA1) of the mouse bone marrow dataset. In these figures, (i) each dot represents a mouse bone marrow cell datapoint; (ii) whereas projected human UCB cell datapoints onto the phenotypic manifold is represented with diamonds; (iii) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset is indicated with a blue hexagon; (iv) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; and (v) the level of gene expression is indicated with the colour bar, where blue indicates no expression, red indicates high expression and orange indicates intermediate expression.

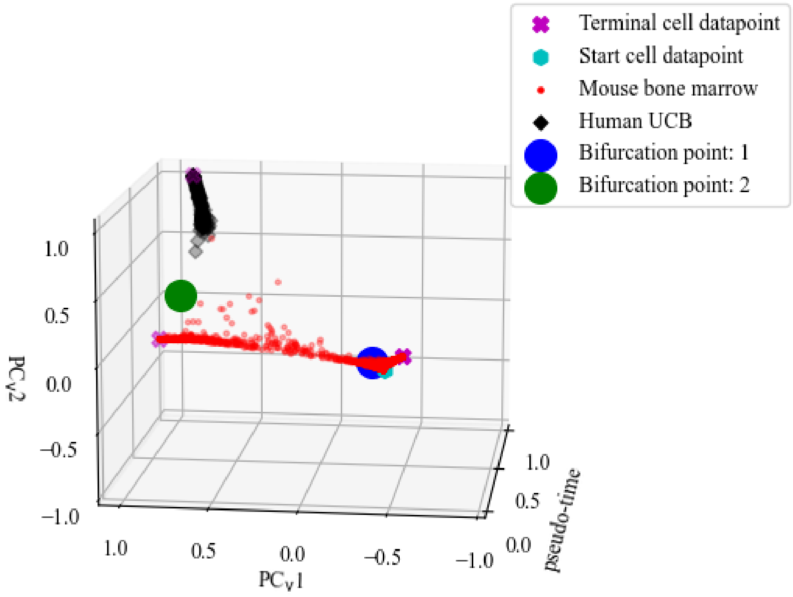
B.4.3 Bifurcation points

The detected bifurcation points from Bayesian model selection during the modelling of the iterspecies dataset in space $\mathcal{P}$ is shown in Figure B.3 below. In Figure B.3, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each dot represents a mouse bone marrow cell datapoint; (iv) whereas the projected human UCB cell datapoints onto the phenotypic manifold is represented with diamonds;

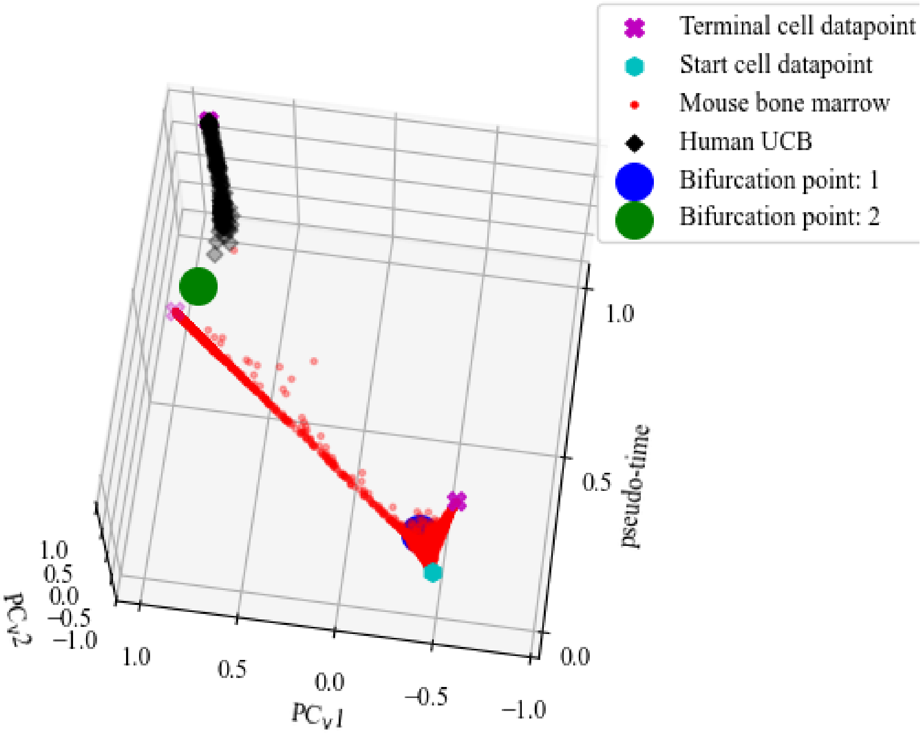
and (v) the well-defined bifurcation points are indicated with a blue and a green coloured sphere respectively.



(a) The detected bifurcation points of the iterspecies dataset visualised by observing space $\mathcal{P}$ from the PC_v2-pseudo-time axes.



(b) The detected bifurcation points of the iterspecies dataset visualised by observing space $\mathcal{P}$ from the PC_v1-PC_v2 axes.

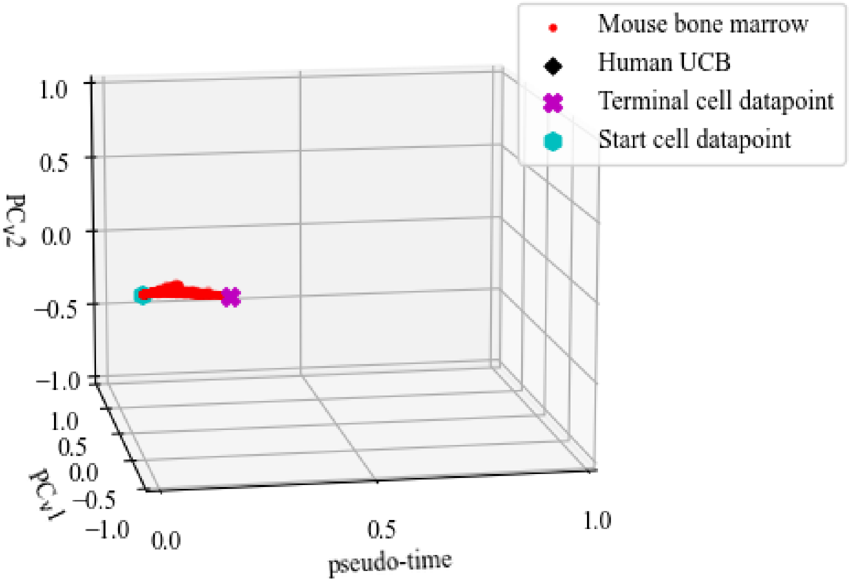


(c) The detected bifurcation points of the iterspecies dataset visualised by observing space $\mathcal{P}$ from the PC_v1-pseudo-time axes.

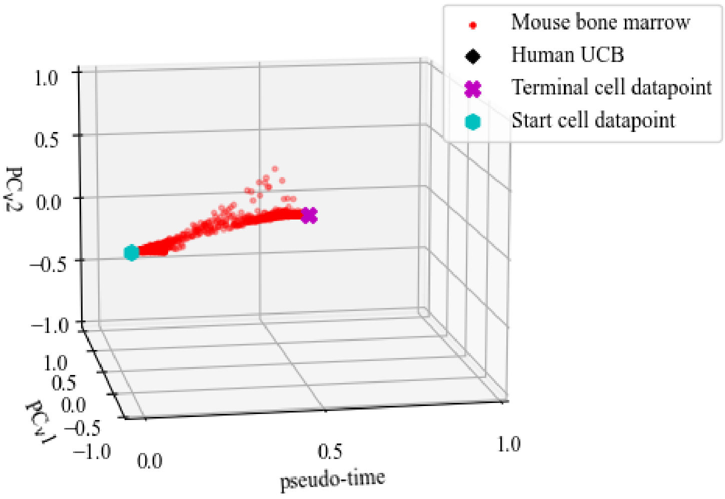
Figure B.3. Bifurcation points detected of the iterspecies dataset in space $\mathcal{P}$. In these figures, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each dot represents a mouse bone marrow cell datapoint; (iv) whereas the projected human UCB cell datapoints onto the phenotypic manifold is represented with diamonds; and (v) the well-defined bifurcation points are indicated with a blue and a green coloured sphere respectively.

B.4.4 Constructed PC-lineages

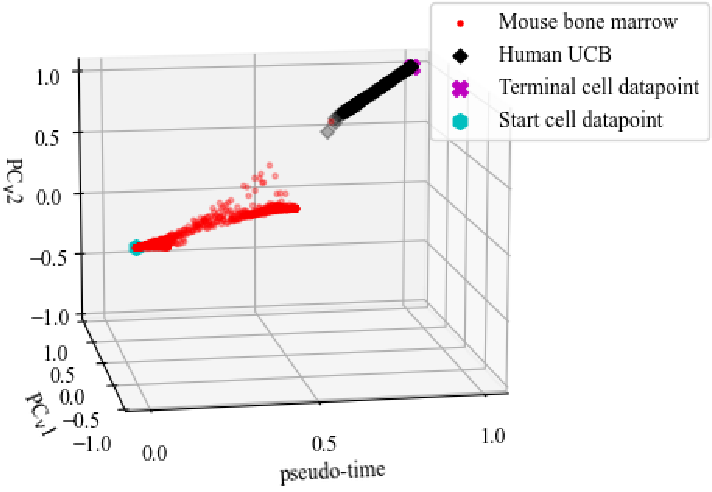
The constructed PC-lineages that indicate the cell developmental trajectory from the start of cell differentiation to a terminal state of the iterspecies dataset in space $\mathcal{P}$ is shown in Figure B.4. In Figure B.4, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset, is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each dot represents a mouse bone marrow cell datapoint; (iv) whereas projected human UCB cell datapoints onto the phenotypic manifold is represented with diamonds.



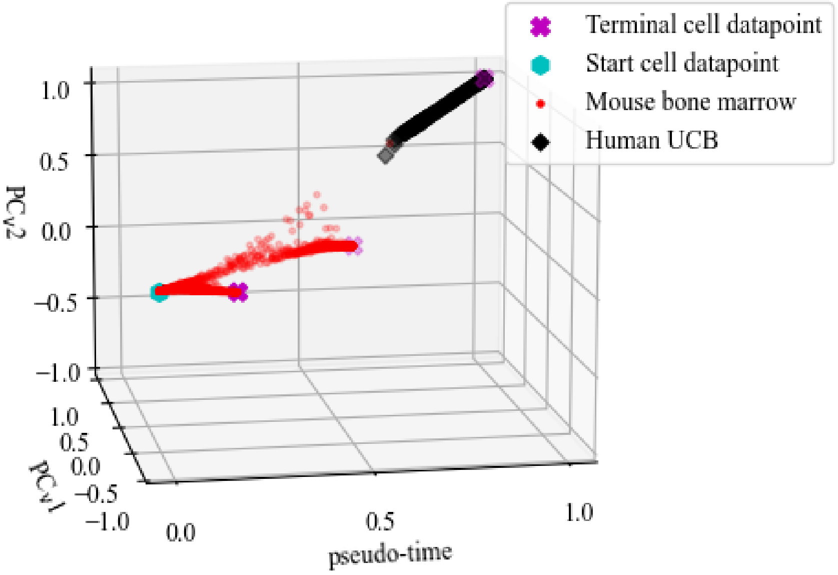
(a) The first constructed PC-lineage of the iterspecies dataset in space $\mathcal{P}$.



(b) The second constructed PC-lineage of the iterspecies dataset in space $\mathcal{P}$.



(c) The third constructed PC-lineage of the iterspecies dataset in space $\mathcal{P}$.

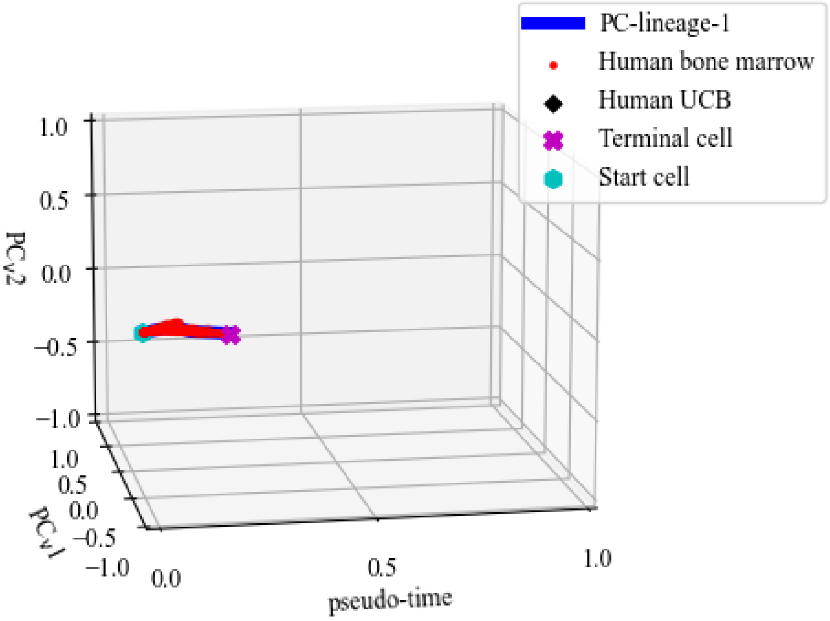


(d) All constructed PC-lineages of the iterspecies dataset in space $\mathcal{P}$.

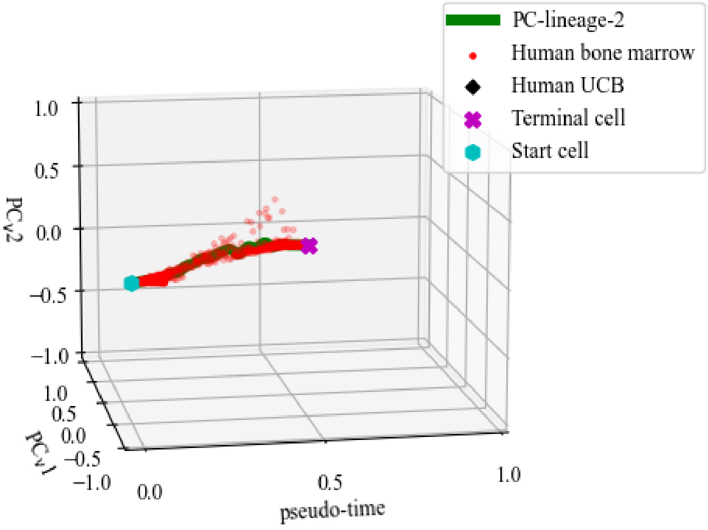
Figure B.4. The constructed PC-lineages of the iterspecies dataset in space $\mathcal{P}$. (a) First, (b) second, and (c) all constructed PC-lineages of the iterspecies dataset. In these figures, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset, is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each dot represents a mouse bone marrow cell datapoint; (iv) whereas projected human UCB cell datapoints onto the phenotypic manifold is represented with diamonds.

B.4.5 Continuous modelling of PC-lineages

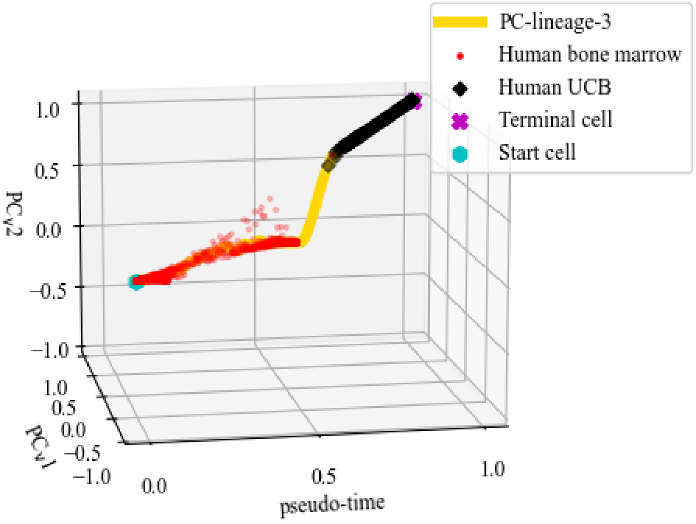
The continuous modelling of the constructed PC-lineages of the iterspecies dataset in space $\mathcal{P}$ is shown in Figure B.5. In Figure B.5, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each dot represents a mouse bone marrow cell datapoint; (iv) whereas the projected human UCB cell datapoints onto the phenotypic manifold is represented with diamonds; and (v) the coloured solid line represents the continuous cell developmental trajectory of a given PC-lineage, also known as the Gaussian process mean.



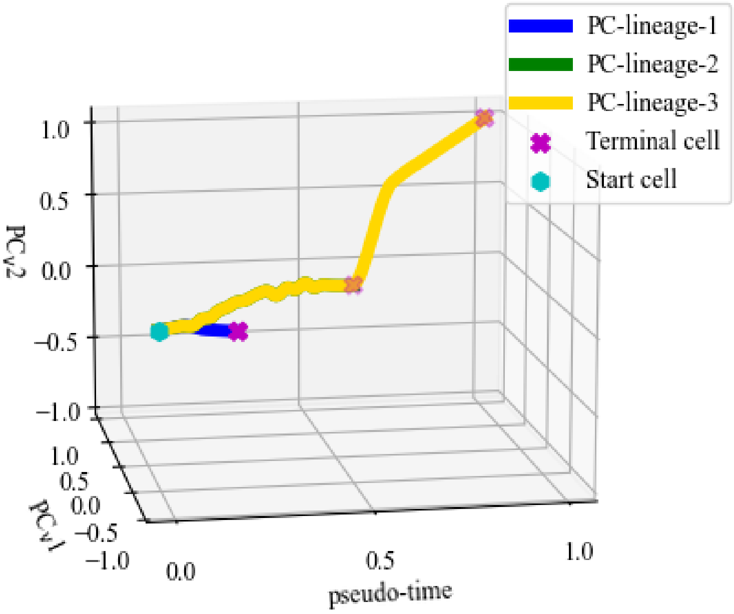
(a) The first constructed PC-lineage of the iterspecies dataset modelled as a continuous process.



(b) The second constructed PC-lineage of the iterspecies dataset modelled as a continuous process.



(c) The third constructed PC-lineage of the iterspecies dataset modelled as a continuous process.



(d) All constructed PC-lineages of the iterspecies dataset modelled as a continuous process.

Figure B.5. The continuous modelling of the constructed PC-lineages of the iterspecies dataset. (a) First, (b) second, and (c) all constructed PC-lineages in mouse bone marrow data modelled as a continuous process. In these figures, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each dot represents a mouse bone marrow cell datapoint; (iv) whereas the projected human UCB cell datapoints onto the phenotypic manifold is represented with diamonds; and (v) the coloured solid line represents the continuous cell developmental trajectory of a given PC-lineage, also known as the Gaussian process mean.

B.4.6 Tangent vectors of window-based Frenet frames

Figure B.6 shows the tangent vectors of the window-based Frenet frames of the cell developmental trajectory of the iterspecies dataset in space $\mathcal{P}$. In Figure B.6, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each dot represents a mouse bone marrow cell datapoint; (iv) whereas the projected human UCB cell datapoints onto the phenotypic manifold is represented with diamonds; and (v) the tangent vectors (PC$_w$1, see Section 4.7.2) of the window-based Frenet frames are indicated with green arrows.

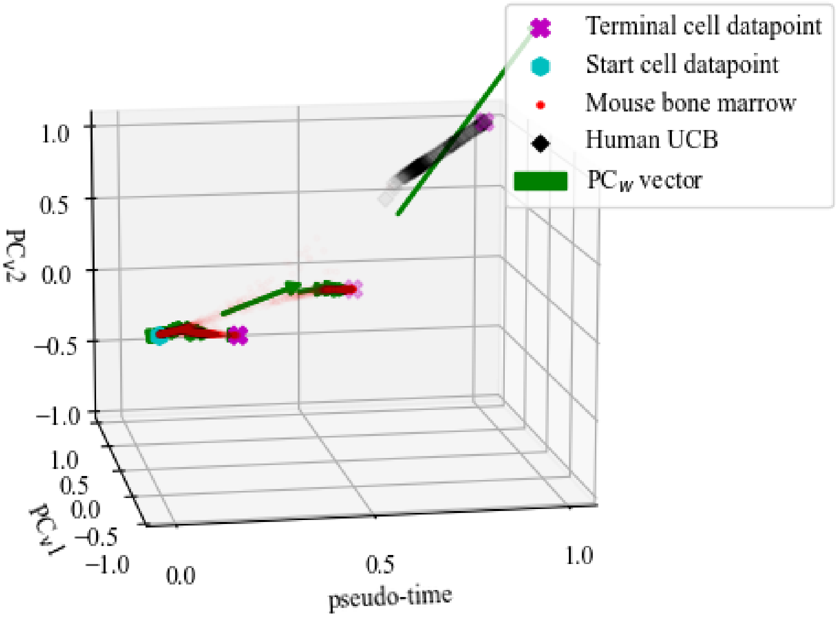


Figure B.6. Tangent vectors of window-based Frenet frames of the cell developmental trajectory of the iterspecies dataset in space $\mathcal{P}$. In this figure, (i) the start cell datapoint, which indicates the starting point of cell differentiation of the dataset is indicated with a blue hexagon; (ii) the terminal cell datapoint, which indicates the end of cell differentiation of the dataset, is indicated with a pink cross; (iii) each dot represents a mouse bone marrow cell datapoint; (iv) whereas the projected human UCB cell datapoints onto the phenotypic manifold is represented with diamonds; and (v) the tangent vectors (PC$_w$1, see Section 4.7.2) of the window-based Frenet frames are indicated with green arrows.

B.5 SUMMARY OF OBSERVED RESULTS

BAGEL was able to project human UCB single-cell gene expression data onto the mouse bone marrow single-cell gene expression data, as seen in Figure B.1. As seen in Figure B.1, there seems to be a jump discontinuity between the human and mouse single-cell gene expression data. This discontinuity is due to the biological difference between mouse and human, summarised in Table 5.2. Although the projection of datasets is not perfect, some introductory biological insight can still be obtained from the simulation. The insight is that gene expression of lineage-specific genes, as seen in Figure B.2, also confirm that the human UCB gene expression data best corresponds to the erythroid PC-lineage of mouse bone marrow.

The rest of the simulation results in this Appendix are further confirmation of the operational capabilities of BAGEL as BAGEL was able to correctly (i) detect bifurcation points, as seen in Figure B.3; (ii) detect PC-lineages, as seen in Figure B.4; (iii) model cell differentiation as a continuous process, as seen in Figure B.5; and (iv) present the tangent vectors of window-based Frenet frames of the cell developmental trajectory as seen in Figure B.6. For an in-depth discussion regarding the operational capabilities of BAGEL see Chapter 6.