

Multistage Optimal Transport evaluated on scRNA-seq Human Bone Marrow data

Vilhelm Conradi

Hugo Nelsson



LUND
UNIVERSITY

Department of Automatic Control

MSc Thesis
TFRT-6318
ISSN 0280-5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

© 2026 Vilhelm Conradi & Hugo Nelsson. All rights reserved.
Printed in Sweden by Tryckeriet i E-huset
Lund 2026

Abstract

With novel techniques like single-cell RNA sequencing (scRNA-seq) we can look inside single cells in great detail and measure their individual gene activity by counting the mRNA molecules of the cells. In order to model cell differentiation one would naturally turn to longitudinal data of single cells. However, snapshot datasets is the basis of single cell studies due to the way the datasets are collected. The advantage with scRNA-seq is that samples are very informative and in healthy bone marrow, often contain a good representation of cells in different stages of the differentiation process. We can therefore assume that we have sufficient information to create a model for the differentiation process using the snapshot data alongside optimal transport frameworks. Building on the assumption that cell states similar to each other have similar gene expression, and that this gene expression can provide information about the order of these cells in a differentiation process, we use a newly proposed optimal transport algorithm to compute probable transitions between cell states as well as a pseudotime ordering of the cells. We use these transitions to extract subsets of cells which represent differentiation trajectories through human bone marrow samples, and order the subset cells in pseudotime. We then estimate expressions of some common marker genes across the subset's pseudotime in order to verify that the model captures expected biological behaviour. We also apply the method to a sample of cells connected to Acute Myeloid Leukemia (AML) to observe if the changed biological content causes these marker genes to be expressed differently in the sample. The ambition is that this method can aid in further study of the development of AML.

Acknowledgements

We would like to thank our supervisors Anders Rantzer and Karl Rydén for their support during this thesis work. We would also like to thank Magnus Fontes, Thomas Mejier, David Ohlin and Nicolas Captier for valuable discussions during the biweekly meetings held during the course of this thesis. Special thanks to Thoas Fioretos, Rasmus Henningsson, Johan Råde and Henrik Lilljebjörn for invaluable support with both biological and bioinformatic understanding as well as for providing the datasets. We would also like to thank Anders Nilsson for excellent technical support.

We would lastly like to thank our master thesis colleagues at the department for good company as well as our families for the support during this semester.

Use of Artificial Intelligence Tools

Artificial Intelligence has been used during the course of this thesis, more precisely Google Gemini 3.5 Thinking and Gemini 3.1 Pro has been used to assist in code debugging, source search and as writing aid. All suggestions have been carefully reviewed by the authors before implementation and all code and text presented are the authors own.

Keywords: Optimal Transport, scRNA-seq, Hematopoiesis, Markov Chains, Scanpy, Cell Differentiation, Optimization, Pseudotime, Acute Myeloid Leukemia

Contents

1. Introduction	9
2. Background	11
2.1 Biological background	11
2.2 Bioinformatic background	13
2.3 Optimal transport	17
3. Methodology	22
3.1 Dataset selection	22
3.2 Scanpy	24
3.3 Preprocessing	25
3.4 Application of MultistageOT	26
3.5 Evaluating MultistageOT's accuracy	29
4. Results	39
4.1 Incorporating RNA-velocity into the cost function	39
4.2 Lineage inference from optimal transport	40
4.3 Marker gene expression over lineage subsets	52
4.4 Principal Component Analysis on the generated subsets	63
5. Discussion	66
5.1 The complexity of biological data and dimensionality reduction	66
5.2 MultistageOT	68
5.3 Design Parameters	69
5.4 RNA-velocity	70
5.5 PCA on a subset	70
5.6 Application on an AML-positive sample	71
5.7 Further research	71
6. Conclusions	73
A. Cell type descriptions	74
B. Gene expression heatmaps	77
B.1 NBM4-MNC	77
B.2 NBM8-MNC	78

Contents

B.3	NBM10-MNC	79
B.4	NBM8-CD34 + NBM7-MNC + NBM4-MNC	80
B.5	AML155	81
C.	Plots of gene expressions over pseudotime for gene extraction meth-	
	ods	85

1

Introduction

Acute Myeloid Leukemia (AML) is a type of blood cancer where white blood cells develop abnormally during the process of Hematopoiesis, the differentiation of blood cells. This causes uncontrolled crowding in the blood and bone marrow which can potentially be fatal [Leukemia Research Foundation, 2024]. In order to better understand how to treat this disease, as well as other types of cancer, it is of great interest to understand how cells develop over time and to detect abnormalities in gene information.

To study cells and their biological processes, advanced data collection methods such as single-cell RNA sequencing (scRNA-seq) allow us to look inside single cells in great detail and count the amounts of individual genes present in them. Efforts like [Reactome Project, 2026] have in turn allowed us to connect genes to certain functions in the cell. Drawbacks of scRNA-seq are that the samples are very expensive to gather and destroy the measured cells, alongside being a stressful process for the patient. Therefore we cannot follow specific cells and their development over time [Zeng, 2022], making true longitudinal studies of single cell data difficult. We are thus left with highly detailed snapshot datasets. Another type of data that can be extracted from single cells is RNA-velocity, a high dimensional vector that points in the direction of some kind of later cell state. The RNA-velocity of a cell can be estimated through the proportion of spliced, mature mRNA, and unspliced pre-mRNA [La Manno et al., 2018].

It would be of high interest to reconstruct a developmental time axis for cells in scRNA-seq data snapshot data, ordering cells and their stage in the differentiation process in relation to each other. This would allow us to observe this detailed gene activity over time as cells differentiate themselves from stem cells to the the distinct cell types we observe in the body.

To model cell differentiation and construct the inferred temporal ordering, the method of optimal transport has been suggested. The optimal transport problem was first introduced by [Monge, 1781] in 1781 and essentially describes how to move distributions in an optimal way with respect to a given cost function. Originally thought to model the movement of dirt, the theory has proved to be a strong tool when modelling movement of distributions of any kind. Considering cells as dis-

crete distributions in a blood sample, we evaluate how optimal transport can capture the cell movement through the gene space created by scRNA-seq, with a cost function based on Euclidean distance in the gene space. In this thesis, we use the mathematical tools from Multistage Optimal Transport (MultistageOT) [Tronstad et al., 2025] to infer cell differentiation trajectories by finding highly probable transitions between cell states in scRNA-seq samples of healthy blood marrow from humans and analyze the gene expression of these trajectories over the inferred pseudotime. The main research questions are

- Can MultistageOT accurately describe cell lineages in Hematopoiesis?
- Can the MultistageOT algorithm be improved by incorporating RNA-velocity in the cost function?
- Can the method find abnormal gene expressions by comparison with AML-data?

We find that the optimal transport algorithm captures the expected lineage trajectories in the datasets and finds subsets of cells for a given lineage, containing cells and clusters of cells that are expected to be in the specified lineage. Based on these subsets of cells we can extract gene expressions for each lineage and analyze the change in expression over a pseudotime axis, constructed by the algorithm. This pseudotime ordering appears to also be consistent with biological understanding. Incorporating RNA-velocity, we see some improvement in cell fate predictions, but the improvement is very small, suggesting that RNA-velocity might not be significant when studying cell differentiation. Finally, comparing with AML-data we can find gene expressions that differs significantly from what we see in healthy data.

This thesis is a part of a larger research project which is funded by the European Research Council and has been conducted at the Department of Automatic Control in collaboration with research groups from the Department of Clinical Genetics in Lund as well as Institut Roche in Paris. The proposed tools, methodology as well as the structure of the work has been selected through recurrent meetings with representatives and supervisors from these groups.

2

Background

In this chapter, we introduce the theory behind the methods we use to answer the posed research questions. This thesis frequently uses various methods and techniques developed within the fields of biology and bioinformatics. As most readers are assumed to be unfamiliar with these fields, the first two sections will introduce theory and concepts from these fields. The final section is dedicated to the mathematical background of optimal transport as well as Markov chains.

2.1 Biological background

2.1.1 Cell types and Marker genes

Cells are the fundamental building blocks of living organisms. Inside each human cell, there is a nucleus, a biological structure which stores the DNA-molecule, which contain the instructions for constructing different proteins. Proteins are large molecules which are responsible for making your body work correctly. A segment of the DNA which produces a certain protein is called a *gene* [Centers for Disease Control and Prevention, 2024]. Gene instructions are read from the DNA and transported to the Cytoplasm structure in the cell, where the proteins are finally constructed. This transportation is carried out by a biological structure called messenger-RNA (mRNA) [National Human Genome Research Institute, 2024].

To analyse cells in a useful way it is important to be able to categorize them into reasonable groups. Cells which have a similar structure and/or function in the body are categorized as belonging to the same *cell-type* [Zeng, 2022].

Historically and fundamentally, cells were defined by the function of the tissue in which they belong [Clevers et al., 2017] and their morphology [Zeng, 2022]. However, this classification is not definite due to the heterogeneity of the cells in these categories and cells may fall into different classes depending on what context we are examining them in. As cell type classification methods have become more advanced, defining coherent cell-types has also become increasingly more difficult [Zeng, 2022].

Marker genes are a select group of genes which can be used to identify a cell-type or a cell-state based on the explicit expression of these genes in these cells, or lack thereof. There is an ongoing debate on suitable marker genes for different cell types [Trapnell, 2015] and marker genes is not something that in itself can define a cell-type.

2.1.2 Hematopoiesis

Hematopoiesis is the biological process by which blood cells are formed. In humans, this process is primarily localized in the bone marrow though some differentiation steps for certain cell variants occur in the Thymus [Jagannathan-Bogdan and Zon, 2013; Okamoto and Koeffler, 2011]. Hematopoiesis is driven by stem cells in the bone marrow, called Hematopoietic Stem Cells (HSCs) which are capable of self-replication as well as multipotent *differentiation* [Chan Zuckerberg Initiative Foundation, 2024].

Differentiation is the process by which cells undergo mitosis (cell division) and in the process evolve into more specialized cell types with more distinct functions with each division. As the differentiation progresses, the stem cells transform into increasingly specialized progenitor cells. These progenitors gradually lose their multipotency, as they commit to different *cell-type lineages* [Okamoto and Koeffler, 2011]. Broadly speaking, three primary types of blood cells are established during Hematopoiesis:

1. Erythrocytes (red blood cells)
2. Leukocytes (white blood cells)
3. Thrombocytes (blood platelets)

Thesis-relevant cell types adhering to Hematopoiesis are described in more detail in appendix A.

2.1.3 Acute Myeloid Leukemia

Leukemia is cancer developed in blood cells, most commonly in the bone marrow. Acute Myeloid Leukemia (AML) refers to the rapid development in the myeloblast cells, the early progenitors of the Myeloid lineage, which is a sub-lineage of the Leukocytes. This lineage contains cell-types such as Neutrophils, Basophils and Monocytes. These abnormal progenitors divide fast and crowd the blood stream without providing the vital functions of ordinary white blood cells, leaving less room for healthy cells to do their work. This reduced functionality can have consequences such as easy bleeding or anemia. Since the growth is very quick, AML left without fast treatment is very dangerous [Leukemia Research Foundation, 2024]. The cause of AML is thus far rather unknown but it is suggested that mutations of certain genes can be a contributing factor, potentially making it hereditary [Klco and Mullighan, 2021].

2.2 Bioinformatic background

In this section we will describe some central concepts and methods used in the bioinformatic process of translating biological measurements into interpretable data.

2.2.1 Single-cell RNA sequencing

In this thesis, we use datasets which have been acquired using the method of Single-Cell RNA sequencing (scRNA-seq), a high-resolution alternative to traditional bulk sequencing [Hwang et al., 2018; Heumos et al., 2023]. The process is very complex and expensive, but for this thesis it is mainly important to understand that cells are processed through a machine and that their mRNA is extracted and read to a computer. Using scRNA-seq on a sample of cells results in a count matrix with individual cell mRNA barcodes on the rows, and gene names on the columns. Each entry in the matrix is the amount of times each gene was counted up in each cells mRNA. The matrix is typically very large in its dimensions as a sample usually contains thousands of cells, and tens of thousands of genes are measured across them. However, the same genes are not always counted up in every cell. As such, a lot of entries in the matrix are zero, and the matrix can then be saved in a memory-saving *sparse* format.

When looking at scRNA-seq data, we are looking at a snapshot in time of the current gene expression of the cells in the sample. The trade-off in high resolution at the cost of no temporal ordering gives rise to the motivation of optimal transport in section 2.3[Bunne et al., 2024]. With methods like scRNA-seq a great deal of new cells have been discovered and classified [Heumos et al., 2023]. This method is potentially very useful if we want to uncover gene signals that would change over time for specific cell types when studying the cell differentiation process.

2.2.2 Cell states and gene scores

By representing each cell by raw gene counts, each gene count can be thought of as a coordinate in a high-dimensional vector space. Cells can thus be represented as vectors in this "gene-regulation-space", and cells with similar gene counts can thus be thought of as lying closer to each other in this vector space[Trapnell, 2015].

The higher the calculated *gene score* is for a specific gene signature, the more prominently those genes are expressed in a given cell relative to the rest of the sample [Wolf et al., 2018]. Rather than relying purely on raw counts, this score subtracts the background expression of randomly selected control genes with similar expression profiles. This isolates true biological variation, allowing for a robust cell-to-cell comparison of gene signature prominence across the dataset.

This vector space can also be viewed as a state space, where cells move between different *cell-states* as their gene expressions change with perturbations of different sorts, e.g. cell differentiation, exposure to drugs and tissue reparation [Trapnell,

2015]. It is important to take into consideration that the clusters in our dataset actually refer to similar “cell-states” rather than “cell-types”. Cell states need not be unique for a certain cell type.

Just as there are marker genes for the cell types, there are marker genes for these cell states, such as the marker genes of the cell cycle. By looking at marker genes and other gene distributions it is possible to cluster cells with similar gene expressions to then get some notion of a cell type. However, this classification of cell-types is arbitrary, as the gene expression clusters are not always corresponding to previous classifications. It is therefore important for the cell-type labelling to correspond to a well defined set of marker genes for a distribution of cells.

The relationship between the present and future states of a cell is called a *trajectory* [Zeng, 2022].

2.2.3 RNA-velocity

Another type of data that can be extracted from cell samples is the so called RNA-velocity. As suggested in [La Manno et al., 2018], RNA-velocity is the time derivative of the gene expression state and it can be estimated through the proportion of spliced and unspliced mRNA. From this information, a vector can be extracted that points in the direction of the next cell state. The idea is then to use this velocity vector to infer the next possible cell state and thus infer both a lineage trajectory and a notion of pseudotime.

Unspliced mRNA (u) is RNA molecules that have both the intron (non-coding) regions and exon (coding) regions and the spliced (s), mature, RNA molecules only contain the exon regions which are relevant for translation[Heumos et al., 2023]. The suggested dynamical model for RNA-velocity is

$$\frac{du}{dt} = \alpha - \beta u, \quad (2.1)$$

$$\frac{ds}{dt} = \beta u - \gamma s \quad (2.2)$$

where α is the transcription rate, β is the splicing rate and γ is the degradation rate of unspliced RNA. This model implies is that in cells with a high fraction of unspliced RNA, the production of RNA is faster than the processing and vice versa. Progenitor cells should therefore have a higher fraction of unspliced RNA and more terminally differentiated cells should have the opposite.

2.2.4 Preprocessing

The sequencing of single cells is prone to errors and therefore it is important to preprocess the data properly before performing further analysis. The first aspect to look at during preprocessing is the filtering of low quality cells. This can be identified by using a combination of

- The *count depth* of the cells, i.e. the number of mRNA molecules measured.
- The total number of different genes in a cell.
- The percentage of mitochondrial genes present in a cell.

Cells with low count depth and counts are not very useful for the analysis, and mitochondrial genes are an indicator of a broken membrane and thus a dying cell [Heumos et al., 2023, Preprocessing].

In the datasets of [Lilljebjörn et al., 2025], these preprocessing steps had already been taken when identifying the celltypes in the sample. By merging the raw cell-matrices with the team’s annotations, un-annotated cells could then be discarded as outliers. *Normalization* is the next step in the preprocessing. This makes each gene count be relative to the size of the gene library for each cell, e.g. if Cell 1 has 10 000 gene reads in total and 50 reads of gene *A* whereas Cell 2 has 20 000 gene reads in total, and 100 reads of gene *A*, then the result after normalization will be that Cell 1 gets 100 reads of gene *A* as well, as the relative expression is identical.

One typically $\log(x+1)$ transforms the data as a final step of the preprocessing. This makes the variance of the dataset much more balanced.

2.2.5 Graphs

Graphs are central to both processing the data and for the methodology of this thesis.

Definition. A *graph* is a mathematical structure which encodes relationships between objects. The objects are collected as a set $\mathcal{V}$, denoted the *vertices* of the graph, and a corresponding set $\mathcal{E}$ of *edges* tells us how the vertices connect to each other. The graph is then a tuple of these sets, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

A *weighted graph*, denoted $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$, is a graph where each edge e_i has a corresponding number $w_i \in W$ associated with it. w_i is called a *weight*, or *cost*, which is related to the action of moving between two vertices using the connection made with that edge. A *directed graph* is a graph that is not symmetric. [Como and Fagnani, 2021]

kNN-graphs. In Single-Cell analysis, it is standard practice to build a *k*-Nearest-Neighbors-graph based on the distance between any two points x_1 and x_2 using the *Euclidean distance* metric:

$$\|x_1 - x_2\|_2 = \sqrt{(x_1^1 - x_2^1)^2 + (x_1^2 - x_2^2)^2 + \dots + (x_1^n - x_2^n)^2} \quad (2.3)$$

where n is the dimensionality of the data. Each point is then represented as a node in the graph with assigned edges to the k closest nodes with respect to the Euclidean distance between the two vertices. The Euclidean distance is then also used as each edge’s corresponding weight. The graph construction is ran using multiple threads, which makes the resulting graph hardware-dependent. Creating a graph with a small value of k might result in disconnected graphs, but too highly connected graphs

might result in unwanted "shortcuts" between nodes. In the context of this thesis, this could result in a path between a stem cell and a terminally differentiated cell which skips the (expected) connections to intermediate cell stages. k NN-graphs are used as a basis and a tool in further analysis steps.

2.2.6 Leiden Clustering Algorithm

When we don't have available annotations for cell-clusters or want to affect the resolution of the clusters, there are clustering algorithms available to group similar cells together. The most common way to cluster cells in single-cell analysis is by using the Leiden clustering algorithm [Traag et al., 2019]. This algorithm is a graph clustering algorithm and assumes a pre-computed graph, in our case the k NN-graph. The algorithm partitions the nodes(cells) into clusters based on a quality function, often modularity, based on a chosen resolution γ . The idea is that γ decides a density for the connectivity of a cluster, which should be higher than the connectivity between clusters. The algorithm then refines the clusters to find connected subclusters and then aggregates the found clusters as nodes. This process iterates until no improvement can be made.

2.2.7 Dimensionality reduction

In highly dimensional datasets, like those generated through scRNA-seq, we need to reduce the dimensionality to denoise and comprehensively visualize the data, but also to create reasonably sized datasets to feed into the algorithms. Not all genes are important for understanding the dynamics of differentiation either [Heumos et al., 2023, Dimensionality Reduction]. There are multiple methods used for dimensionality reduction. The ones used in the thesis are presented below.

Principal Component Analysis. Principal Component Analysis (PCA) is a linear dimensionality reduction technique which transforms the data into a new coordinate system using orthogonal projection. The basis vectors of this new coordinate system are linear combinations of features in the original data called Principal Components (PCs). Each PC is ranked in decreasing order with respect to how much variance each PC captures from the original dataset [Heumos et al., 2023, Dimensionality Reduction].

By choosing a reasonable number of principal components, we can denoise the data and reduce the dimensionality of the data while keeping the signals that are important for the processes we want to examine. When working with algorithms it is also computationally demanding to work with datasets with thousands of states. Dimensionality reduction by PCA can alleviate this obstacle. Since most of variance in the biological signals are captured by the first two or three principal components, plots incorporating these as spatial dimensions can still give meaningful insights into the data. The PCA is also the basis on which the k NN-graphs are constructed in single-cell analysis.

UMAP. Uniform Manifold Approximation and Projection (UMAP) is a non-linear method of reducing dimensionality in high-dimensional datasets. In essence, it uses a k NN-graph built in the PCA-reduced space and then finds a low-dimensional embedding that best preserves local relationships. The lower-dimensional graph is as structurally similar to the higher-dimensional one as possible [Heumos et al., 2023, Dimensionality Reduction].

As cells which have similar gene expressions are assumed to lie closer in the high dimensional space we can, by using UMAP, retain a sense of how "alike" cells are, and still being able to visualize them in interpretable two-or three dimensional plots. In UMAP coordinates, the axes are not of importance. The relative distances between nodes are the important aspect, indicating "closeness" in gene expression. UMAP is not deterministic unless you fix the randomness seed. UMAP is mainly used for visualization in this thesis.

2.3 Optimal transport

Optimal transport is a field in mathematics which deals with the problem of mapping one probability distribution $\mathcal{X}_1$ to another distribution $\mathcal{X}_2$ in an optimal way with respect to some cost function C [Monge, 1781] [Kantorovich, 1942]. There is both a continuous and a discrete mathematical formulation of the optimal transport problem. In the discrete case, which is the form considered in the thesis, the initial distribution is defined as

$$\mathcal{X}_1 = \sum_{i=1}^{n_1} \mu_1^{(i)} \delta_{x_1^{(i)}}, \quad (2.4)$$

where x_1 is the set of the support points used. n_1 denotes the amount of support points. $\mu_1^{(i)}$ denotes the probability mass in each support point $x_1^{(i)}$. $\delta_{x_1^{(i)}}$ indicates that all mass $\mu_1^{(i)}$ is gathered at exactly the location of the support point.

Likewise, the terminal distribution is defined as

$$\mathcal{X}_2 = \sum_{j=1}^{n_2} \mu_2^{(j)} \delta_{x_2^{(j)}}. \quad (2.5)$$

The basic optimal transport problem then is to create a transport plan matrix M , where each element m_{ij} describes the mass transported from $x_1^{(i)}$ to $x_2^{(j)}$, and minimizes a given cost function $C(x_1, x_2)$. The cost function relates a cost to each mass transport between the two distributions, for example the *squared euclidean distance function*

$$C = \mathbb{R}^{n_1 \times n_2}, \quad c_{ij} = \|x_1^{(i)} - x_2^{(j)}\|_2^2. \quad (2.6)$$

Using optimal transport in a single cell environment, the support points in the distributions are the cells, defined by their position in the gene-regulatory space generated by scRNA-seq.

The optimal transport problem is formulated as an optimization problem where we want to minimize the given cost function to find a transport plan M that conserves mass in an optimal way with respect to the cost function through the transport plan,

$$\min_{M \in \mathbb{R}_+^{n_1 \times n_2}} \langle C, M \rangle, \quad (2.7)$$

$$s.t. \quad M \mathbf{1}_{n_2} = \mu_1, \quad (2.8)$$

$$M^T \mathbf{1}_{n_1} = \mu_2 \quad (2.9)$$

where $\langle \cdot, \cdot \rangle$ is the Frobenius inner product, i.e. the element wise inner product returning a scalar. As all entries in the matrices are real valued, it can be defined as

$$\langle A, B \rangle = \sum_{ij} A_{ij} B_{ij}, \quad (2.10)$$

where A_{ij} and B_{ij} are the entries on row i , column j in each respective matrix.

The motivation behind optimal transport in bioinformatics is the assumption that nature will tend to follow the optimal path from stem cell to terminal cell, optimal being defined as the path with the smallest total cost through gene space. As assumed in section 2.2.2, cells that are closer to each other in the gene space should be more similar than cells with more differing gene expressions. The cost function in the optimal transport problem is therefore based on the squared euclidean distance metric in Equation (2.6) in this gene space [Tronstad et al., 2025]. The amount of probability mass transported between two cells can be interpreted as capturing a probability of transitioning to that cell state in a differentiation process. According to the cost function, the most mass should be transported to states close to the original one. By choosing an initial distribution from the stem cells (HSCs) and terminal states from known differentiated cells, we expect the optimal transport algorithm to find a biologically accurate trajectory through the dataset.

2.3.1 Entropic regularization

The original formulation of optimal transport does not scale well to large datasets, as the computational complexity grows with $\mathcal{O}(n^3 \log(n))$. With the introduction of an entropic regularization term the problem goes from a Linear Programming problem to a strictly convex optimization problem that can be solved with e.g. Sinkhorn-Knopp iterations which greatly reduces the computation time [Cuturi, 2013]. Equations

tion (2.7) then becomes

$$\min_{M \in \mathbb{R}_+^{n_1 \times n_2}} \langle C, M \rangle + \varepsilon D(M|P), \quad (2.11)$$

$$s.t. \quad M \mathbf{1}_{n_2} = \mu_1, \quad (2.12)$$

$$M^T \mathbf{1}_{n_1} = \mu_2 \quad (2.13)$$

where $\varepsilon > 0$ is a weighting parameter and $D(M|P)$ is the entropy term defined as

$$D(M|P) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \left(m_{ij} \log \left(\frac{m_{ij}}{p_{ij}} \right) - m_{ij} + p_{ij} \right) \quad (2.14)$$

where P represents a prior matrix and can be initialized as any positive definite matrix [Tronstad et al., 2025].

2.3.2 The optimal transport problem in MultistageOT

In this thesis we use the optimal transport framework as described by [Tronstad et al., 2025]. Whereas the basic problem formulation of optimal transport only used two distributions, MultistageOT leverages several intermediate transport plans in their algorithm to produce better cell fate prediction results.

In this framework, an initial distribution $\mathcal{X}_0$ consisting of a number n_0 cells is selected of e.g. HSCs, deemed to be early in the differentiation process. Next, a terminal distribution of cells $\mathcal{X}_F$ is selected, containing a number n_F of cells presumed to be at the end of the differentiation process. The remaining cells not selected for $\mathcal{X}_0$ or $\mathcal{X}_F$ are seen as part of an intermediate distribution, $\mathcal{X}$. The algorithm then, for a user-set number T of transport plans, iterates through the mass transport from the initial distribution, through the intermediate cells, to the terminal distribution where the mass exits the system. It thus solves several intermediate optimal transport problems to arrive at a final set of transport plans which collectively describe how mass is transported from $\mathcal{X}_0$ through the cells in $\mathcal{X}$ and finally into $\mathcal{X}_F$.

The first transport plan M_0 describes how mass is transported from cells belonging to $\mathcal{X}_0$ to cells belonging to $\mathcal{X} \cup \mathcal{X}_F$. The transported mass in $\mathcal{X} \cup \mathcal{X}_F$ is then considered as the initial distribution for the next, intermediate, transport plan. This distribution is now mapped once more to other cells belonging to $\mathcal{X} \cup \mathcal{X}_F$. Mass transported to $\mathcal{X}_F$ is absorbed and not used in later transport plans. This process is then repeated, with the constraint that all remaining mass in $\mathcal{X}$ must, in the final transport step, be mapped to only cells in $\mathcal{X}_F$.

The minimization problem in [Tronstad et al., 2025] becomes

$$\min_{\substack{\tilde{M}_t, \tilde{M}_t \geq 0 \\ t=0, \dots, T-1}} \sum_{t=0}^{T-2} \left(\langle \tilde{C}_t, \tilde{M}_t \rangle + \varepsilon D(\tilde{M}_t | \tilde{P}_t) \right) + \sum_{t=0}^{T-1} \left(\langle \hat{C}_t, \hat{M}_t \rangle + \varepsilon D(\hat{M}_t | \hat{P}_t) \right), \quad (2.15)$$

$$s.t. \quad \tilde{M}_0 1_{n_1} + \hat{M}_0 1_{n_F} \geq 1_{n_0}, \quad (2.16)$$

$$\tilde{M}_t 1_{n_{t+1}} + \hat{M}_t 1_{n_f} = \tilde{M}_{t-1}^T 1_{n_{t-1}}, \quad \text{for } t = 1, \dots, T-2, \quad (2.17)$$

$$\hat{M}_{T-1} 1_{n_F} = \tilde{M}_{T-2}^T 1_{n_{T-2}}, \quad (2.18)$$

$$\sum_{t=1}^{T-2} \tilde{M}_t 1_{n_{t+1}} + \sum_{t=1}^{T-1} \hat{M}_t 1_{n_F} \geq 1_n, \quad (2.19)$$

$$\sum_{t=0}^{T-1} \hat{M}_t^T 1_{n_t} \geq 1_{n_F}. \quad (2.20)$$

Here, $\tilde{M}_0$ and $\tilde{C}_0$ denotes the mass and cost from the initial states to the intermediate states respectively and $\hat{M}_0$ and $\hat{C}_0$ denotes the mass sent from the initial states to the terminal states. Likewise, $\tilde{M}_t$ and $\tilde{C}_t$ is the mass and cost between intermediate states and $\hat{M}_t$ and $\hat{C}_t$ is the mass and cost between intermediate states and terminal states. M_T is the final transport plan, which moves all remaining mass in $\mathcal{X}$ to $\mathcal{X}_F$.

2.3.3 Pseudotime inference

Studying cell differentiation, one would hope to have access to longitudinal data where it is possible to track the cells over time to see how they develop. In scRNA-seq, this is not possible since each cell is practically destroyed when the sample is taken. However, even though a bone marrow sample is only ever gathered at one timepoint, it contains many cells in different stages of the differentiation process as cells are created and differentiated continuously at all times in the body. This means that we can assume that one sufficiently large sample has a good enough cell state representation such that it can represent the whole differentiation process. The algorithm infers through the multiple transport plans a *pseudotime* which is an ordering from 0 to 1 for each of the cells where pseudotime 0 is assigned to the HSCs selected as initial distribution and pseudotime 1 is assigned to cells which receive a lot of mass very late in the optimal transport scheme. Each cell then gets a probability of belonging to a pseudotime number by calculating a weighted average of the probability mass received over the intermediate transport plans, and then normalizing. This is an important tool when studying differentiation because we can extract which genes behave in a certain way at a specific pseudotime to gain knowledge of the dynamics that drive the processes.

In the algorithm, pseudotime for each cell is computed based on the mass transported by each cell in each time step. The total mass transported in time step t , $\mu_t^{(i)}$ by an intermediate cell is normalized by the total mass transported $\sum_{k=1}^{T-1} \mu_k^{(i)}$ to get a probability for transporting mass in this step. The same is done for the

mass received by terminal states $\hat{\mu}_t^{(i)}$ at time step t . Based on the expected values of these probabilities, the cells can be ranked and ordered. Initial states is assigned pseudotime 0 and terminal states pseudotime 1.

2.3.4 Markov Chains and random walks

The optimal transport algorithm allows for the computation of a transition matrix with probabilities for each cell to cell transition. To validate the algorithm we want to perform random walks on the matrix to see if the transition matrix correctly describes the biological patterns of cell differentiation.

For a graph $\mathcal{G}$ with edges $\mathcal{E}$, nodes $\mathcal{V}$ and normalized weight matrix P , a Markov chain $X(t), t = 0, 1, \dots$ is a time- discrete stochastic process on the state space $\mathcal{X}$ defined by

$$\mathbb{P}(X(t+1) = j | X(0) = i_0, X(1) = i_1, \dots, X(t-1) = i_{t-1}, X(t) = i_t) = P_{i_t j} \quad (2.21)$$

so that the Markov property describes the process, i.e. the future state only depends on the current state and not the previous states [Como and Fagnani, 2021]. Using the transition matrix computed through MultistageOT, we have a Markov chain on the graph generated by the optimal transport algorithm.

The absorbing probabilities for $s \in \mathcal{S}$ of a Markov chain $X(t)$ is defined as

$$A_{i,s} = \mathbb{P}_i(X(T_{\mathcal{S}}) = s) = \mathbb{P}_i(T_{\mathcal{S}} = T_s) \quad (2.22)$$

for a subset $\mathcal{S} \subseteq \mathcal{X}$ for a given initial state $i \in \mathcal{X}$ [Como and Fagnani, 2021].

3

Methodology

In this chapter, we describe how we processed the dataset, applied the MultistageOT method and finally analyzed the resulting model in order to arrive at our results.

3.1 Dataset selection

The datasets used were provided by Thoas Fioretos team working at the Department of Clinical Genetics of Lund University [Lilljebjörn et al., 2025] which is open to the public and available at <https://doi.org/10.17044/scilifelab.23715648>. The datasets consist of several scRNA-sequenced samples with occasional multiple sample types from different donors. The datasets are imported in the form of Annotated Data (AnnData) objects [Virshup et al., 2024]. The cell types present in the datasets, according to the provided annotations of [Lilljebjörn et al., 2025], are detailed in Table 3.1. A more detailed description of each occurring cell-type is provided in appendix A.

Each dataset is annotated with three levels of "normal" annotations, ranging from the general labels of cell-types, such as "T-cells" and "Progenitors", to the most detailed which is called `predicted.celltype.l2` (l2-annotations), containing detailed labels of subtypes of T-cells, such as "CD4⁺ Memory cells" and "Plasmacytoid Dendritic Cells". For the samples from donors with AML, there are another level of annotations called `celltype.aml` (aml-annotations) with the cell-type "AML Immature", which are the cells which were classified as abnormal white blood cells synonymous with Leukemia. l2-annotations are also available for the AML datasets.

3.1.1 Healthy patients

The healthy datasets consist of 5 samples from healthy donors, with 3 additional samples from healthy donors which have been enriched with cell type progenitors. The healthy measurements are named NBM for Normal Bone Marrow, and are provided in "MNC" and sometimes "CD34" variants. CD34 is a common antigen protein expressed by progenitor cells[Pellin et al., 2013]. The label on the dataset means

Cell type	l2-annotation	aml-annotation
Red Blood Cell Progenitor	Prog_RBC	Erythroid cells
Megakaryocyte Progenitor	Prog_Mk	Megakaryocitic cells
Hematopoietic Stem Cell	HSC	HSC
Lymphoid-Primed Multipotent Progenitor	LMPP	LMPP
Granulocyte Monocyte Progenitor	GMP	GMP
B Cell Progenitors	Prog_B 1, Prog_B 2	B-Cells
Dendritic Cell Progenitors	Prog_DC	Dendritic Cells
Naive B cells	Naive B	B-Cells
Memory B cells	Memory B	B-Cells
Plasmablasts	Plasmablast	B-Cells
CD4 ⁺ Naive T cell	CD4 Naive	T-Cells
CD4 ⁺ Memory T cell	CD4 Memory	T-Cells
CD8 ⁺ Naive T cell	CD8 Naive	T-Cells
CD8 ⁺ Memory T cell	CD8 Memory_1, CD8 Memory_2	T-Cells
CD8 ⁺ Effector T cell	CD8 Effector_1, CD8 Effector_2	T-Cells
Natural Killer T cell (CD14 type)	NK	NK-Cells
Natural Killer T cell (CD56 type)	CD56 bright NK	NK-Cells
Regulatory T-cells	Treg	T-cells
$\gamma\delta$ T-Cells	gdT	T-cells
Mucosal Invariant T-Cells	MAIT	T-Cells
Conventional Dendritic Cell	cDC2	Dendritic Cells
Plasmacytoid Dendritic Cell	pDC	Dendritic Cells
CD14 ⁺ Monocyte	CD14 Mono	Monocytes
CD16 ⁺ Monocyte	CD16 Mono	Monocytes
Immature AML blood cell	HSC,LMPP	AML Immature

Table 3.1 The identified cell types in the datasets and their dataset annotations [Lilljebjörn et al., 2025].

Sample Name	No. datapoints (Raw)	No. datapoints (Filtered)
NBM4-MNC	4941	4636
NBM7-MNC	5852	5215
NBM8-CD34	5458	4509
NBM8-MNC	8247	7701
NBM10-CD34	7626	6867
NBM10-MNC	5597	5224
NBM11-CD34	7487	6810
NBM11-MNC	8298	7694

Table 3.2 The sizes of the healthy bone marrow samples provided by [Lilljebjörn et al., 2025].

Sample Name	No. datapoints (Raw)	No. datapoints (Filtered)
AML155	7605	4645

Table 3.3 The sizes of the AML datasets used in the thesis.

that the sample has been CD34-enriched, thereby containing an increased amount of progenitor cells compared to naturally occurring amounts. MNC variants instead contain mainly mononuclear cells, meaning the sample contains more natural proportions between the different cell types, and thus the samples contain a majority of differentiated cells. Table 3.2 details the healthy samples used in the analysis.

We used the l2-annotations in our subsequent analysis of the healthy samples in order to generate models as biologically accurate as possible.

3.1.2 AML Datasets

There are 38 different samples from donors with AML. Each dataset is named "AML-LXX", where XX is the patient number. In l2-annotations, the datasets contain a massive surplus of LMPP-cells, and in the aml-annotations, the vast majority are classified as "AML Immature". Since these contain an overwhelming amount of AML-cells, many of the datasets lack sufficient amounts of other cell types which we consider containing terminal distributions in the healthy datasets. Table 3.3 details the sizes of the datasets used in the analysis. It should be noted that 10 of the 38 AML-samples were considered for the analysis but only one was deemed to be of sufficient quality with respect to cells considered for initial and terminal distributions, and also managed to converge within the time frame of this thesis.

3.2 Scanpy

In this thesis, we chose to work with the Python single-cell analysis toolkit called Scanpy [Wolf et al., 2018] as MultistageOT is implemented as a Python package. Scanpy is a toolkit designed to handle single-cell analysis of large datasets of up to

several millions of cells and we worked with this toolkit mainly using its capacity for preprocessing, structure of AnnData objects and plotting. The Scanpy environment also has methods for scoring gene expressions that we used to compare the average expressions of genes to find gene dynamics in the samples.

3.3 Preprocessing

The raw datasets are very high-dimensional, consisting of measurements of approximately 33000 different genes across several thousands of cells. In order to be able to work with each dataset, preprocessing is required to remove outliers and reduce the dimensionality of the dataset.

In order to preprocess the raw dataset, we used methods from [Scanpy Development Team, 2026; Heumos et al., 2023]. For all subsets of the dataset, the preprocessing was done in the following way:

1. We removed outlier cells by merging the dataset with the annotations provided by the authors [Lilljebjörn et al., 2025]. This could be done as cells which have not received annotations were classified to be outliers during the annotation step when the datasets were created. The outliers removed in this step are the cells which correspond to the low quality cells mentioned in section 2.2.4, so the initial filtering of the cells is already done.
2. We then normalized the data.
3. Next, we $\log(x + 1)$ transformed the data.
4. We then reduced the genes in the dataset to the 2000 most variable genes. This reduces dimensionality significantly while retaining the most important structure.
5. We then regressed out eventual cell cycle genes and clock genes, as we were only interested in looking at what drives the hematopoiesis. We didn't want to have signals from other processes present in our data. It would therefore be preferable that the genes used for analysis would be relevant for the cell type evolution rather than the cell state. The regression was carried out in accordance with [Tirosh et al., 2016]. We also regressed out "clock" marker genes, associated to the circadian rhythm of the cells.

Principal Component Analysis. After the initial preprocessing step is done the dataset is still very high-dimensional, containing the top 2000 most variable genes for each cell in the dataset. Because of the size of the dataset, it is standard practice as part of the preprocessing to reduce the dimensions to around 10-50 principal components [Heumos et al., 2023]. This filters out much noise yet still retains much of the structure in the data. In our process, after consulting with experts, we chose to

use 50 principal components to retain as much high-dimensional structure as possible while still working within a reasonable number of dimensions for our computers to handle.

Graph construction. Next, a k NN-graph was constructed in the PC-reduced expression space. We used $k = 15$ neighbors as this is standard practice in scanpy[Scanpy Development Team, 2026]. The k NN-graph is the basis for UMAP representations, clustering algorithms and for our terminal distribution selection.

3.4 Application of MultistageOT

In this thesis, we use the recently developed tool called MultistageOT [Tronstad et al., 2025] in order to assign a pseudotime value to each cell and to infer lineage trajectories. Where other optimal transport methods (e.g. StationaryOT [Zhang et al., 2021]) could do the same, they only solved the problem of mapping one distribution onto another. MultistageOT uses several intermediate distributions in order to assign a timestamp to each cell in the dataset.

3.4.1 Choice of initial and terminal distribution

To use the MultistageOT algorithm we need to define the initial ($\mathcal{X}_0$) and terminal ($\mathcal{X}_F$) distribution of cells.

Developing a sufficiently robust method is challenging. To produce the best possible MultistageOT model, the distributions should contain cells that closely mirror an ideal representation of each relevant cell type.

We select the cells for the distributions according to the following algorithm:

1. Compute a k NN-graph on the 50 PCA-reduced space.
2. Calculate the shortest paths between every pair of nodes in the graph.
3. Select a stem cell to act as a starting node for the initial distribution $\mathcal{X}_0$. Select the $n_0 - 1$ cells from the same cluster which lie closest to our initial cell in the k NN graph, and add them to $\mathcal{X}_0$. If we assume that the initial cell is a good representation of a stem cell, then the closest cells of the same cluster to this one in the PCA space should also be good representations of stem cells as reasoned in section 2.2.2.
4. Select a set of clusters (e.g. among the l2-annotations) which should contain terminally differentiated cells.
5. For each of these "terminal clusters":
 - I. Compute which cell in this cluster is the furthest away from the chosen initial stem cell with respect to the k NN graph distances. If similar cells

lie close together in the gene space, then dissimilar cells should lie further away. The cell most dissimilar to the initial stem cell in this cluster should lie the furthest away, and then be a good candidate to represent the terminal cell type. Add it to the terminal distribution $\mathcal{X}_F$.

- II. With respect to this cell, select the $n_F - 1$ closest cells to also act as the terminal distribution of cells for this cell type. If we assume that the terminal cell is a good representation of the cluster cell type, then the closest cells of the same cluster to this one in the PCA space should also be good representations of the terminal cell. Add them to $\mathcal{X}_F$.
- III. Repeat for all terminal clusters.

As AML cells are composed of "abnormal" progenitor cells (see section 3.1.2) which do not evolve to any terminal cell types, we dealt with the MultistageOT models on AML-data by choosing the terminal cell states for the AML distribution by selection from the aml-annotation cluster AML Immature. All other terminal clusters considered when constructing $\mathcal{X}_F$ was selected among the l2-annotations. However, there could be a mix-up for HSCs in the l2-annotations which are really AML Immature cells in the aml-annotations. Thus, we also used the aml-annotations when selecting the initial HSC cells for $\mathcal{X}_0$. Every time cells were selected, their ID:s were checked such that no double-labelling occurred. For these datasets, l2-annotated cell types containing less than 5 cells were excluded from the set of terminal distributions.

3.4.2 Choice of entropy parameter value

The introduction of the entropy term in Equation (2.11) not only makes the computations easier but also permits the flow of mass to more states [Tronstad et al., 2025]. The choice of the entropy parameter ε then essentially determines the sparsity of the matrix.

To accurately model biology we would want to have a good value on ε that captures the 'randomness' exhibited in cells. Finding ε is a part of the optimization algorithm and is implemented by [Tronstad et al., 2025] and the method is labelled `proximal_sinkhorn()`. The proximal Sinkhorn algorithm was ran down to the recommended threshold of $\varepsilon = 0.006$.

3.4.3 Computer specifications and MultistageOT parameters

All MultistageOT parameters were kept the same as provided in the initial template by [Tronstad et al., 2025]. The parameter values are detailed in Table 3.4. The models were computed using a department computer with specifications detailed on the thesis Github: <https://github.com/vilhelmconradi-prog/MSOTonNBM>. As previously mentioned in section 3.1.1, due to the intense RAM memory required for large datasets as well as the limited processor power on the department computer, we were not able to generate models for datasets of size ≥ 14500

Parameter name	Value
n_0	10
n_F (per cluster)	5
epsilon	0.055
n_groups	20
patience	1000
tolerance	0.0001
epsilon_threshold	0.006

Table 3.4 Parameter values used in the MultistageOT model creation process. `patience` refers to the amount of iterations for which the user is updated on the process. `tolerance` is the convergence limit.

cells, as they caused overload of the Jupyter kernel which immediately disconnected the computer. Some smaller datasets still failed to converge due to time-outs in the remote computer connection. Reducing the parameter ε to too small values resulted in NaN entries in the transport plan, as the Sinkhorn terms $e^{-\frac{\lambda}{\varepsilon}}$ became smaller than could be stored in the computer memory. Finally, we discovered that if the `epsilon_threshold` parameter was tweaked even lower in the `proximal_sinkhorn` method, the extensive computations kept going for so long that the remote computer connection once again timed out, erasing progress.

3.4.4 Expansion of the cost function incorporating RNA-velocity

Aside from improving selection of $\mathcal{X}_0$ and $\mathcal{X}_F$, we also wanted to expand on the work done in [Tronstad et al., 2025] by incorporating RNA-velocity in the cost function of the algorithm. This has been suggested in [Tronstad et al., 2025; Zhang et al., 2021].

Thus we modified the cost function to incorporate a term based in on the cosine similarity in a fractional relation to the original cost described in

$$C = \alpha C_{sq-euc} + (1 - \alpha) C_{vel} \quad (3.1)$$

where

$$(C_{vel})_{ij} = \frac{1}{2} \left(1 - \frac{\langle x_j - x_i, v_i \rangle}{\|x_j - x_i\| \|v_i\|} \right). \quad (3.2)$$

Here, α is the specified fraction of the cost function being based on distance (with $(\alpha - 1)$ being the fraction of RNA-velocity in the cost function), x_i, x_j are the cell coordinates in the gene space and v_i is the velocity for cell x_i . The two cost functions are normalized separately to be in the interval of 0 – 1, to be able to control the relation between them. Unfortunately, we did not have access to the RNA velocity of the dataset provided by [Lilljebjörn et al., 2025]. The hypothesis

Sample Name	No. datapoints
<code>scvelo.datasets.bonemarrow()</code>	7264

Table 3.5 The size of the dataset used in the RNA-velocity evaluation

Cell type	Annotation
Hematopoietic stem cells	HSC_1, HSC_2
Unspecified precursors	Precursors
Common lymphoid progenitors	CLP
Erythroid cells	Ery_1, Ery_2
Monocytes	Mono_1, Mono_2
Dendritic cells	DCs
Megakaryocytic cells	Mega

Table 3.6 The identified cell types in the datasets and their dataset annotations [Setty et al., 2019]

was instead evaluated on a bone marrow dataset from [Setty et al., 2019] provided by `scvelo` [Bergen et al., 2020]. The analysis of RNA-velocity was restricted to this dataset and is not combined with the other datasets used in this thesis. Information about the dataset is found in Table 3.5 and Table 3.6.

To evaluate the effect of combining the two cost functions, both are computed with the same number of principal components. The total variation of the prediction is computed according to

$$\frac{1}{2} \sum |P_i - A_i| \quad (3.3)$$

where P is the probability of the cell to adhere to a certain cell type and A is the annotation that the model is trying to predict. The probabilities P are computed with the MultistageOT provided python method `cell_fate_probabilities()` based on absorbing probabilities, section 2.3.4.

To analyse RNA velocity we use the python package `scvelo` proposed by [Bergen et al., 2020].

3.5 Evaluating MultistageOT:s accuracy

As explained in section 2.3.3, the MultistageOT algorithm assigns a pseudotime value between 0 and 1 to each cell, as well as transition probabilities between all cells in the dataset. To validate that the resulting pseudotime order and the transition probabilities from MultistageOT are reflecting actual biological processes we would expect that

- different cell types to be ordered in pseudotime according to known biological processes.

- highly probable trajectories from stem cell to final differentiated cells pass through known intermediate progenitor clusters.

We propose two methods to validate the accuracy of the model and use these to extract information about the gene dynamics. The goal of these methods is, for a given terminal cell type:

- To extract subsets of cells which describe the likeliest trajectory from stem cell to terminal cell types using the transition probabilities provided by MultistageOT. If these contain cell types associated with the differentiation of the terminal cell type, it would be a strong indication that the transition probabilities are working correctly.
- To verify that the pseudotime ordering of cells in these subsets causes:
 1. Marker genes to be ordered according to known biology. Marker genes of stem cells and progenitors should have high expression in early stages of pseudotime, whereas marker genes for distinct cell types should have high expression towards the end of pseudotime.
 2. Genes which change heavily across the pseudotime axis should be associated with the differentiation of the terminal cell type.

The lineage analysis in sections 3.5.1 and 3.5.2 was evaluated on 5 different MultistageOT models utilizing the proximal Sinkhorn scheme and the variables in Table 3.4, trained on the following datasets:

- The individual MNC datasets: NBM4-MNC, NBM8-MNC, NBM10-MNC to observe effects on healthy individual samples,
- A combined dataset made up of NBM8-CD34 + NBM4-MNC + NBM7-MNC to observe effects of larger datasets,
- The AML dataset AML155 to observe discrepancies in the cell pathways and gene expressions between healthy and AML samples. AML155 is one of the few datasets with a well enough representation of other cell-types that the MultistageOT model can be somewhat compared to the healthy variants with respect to selection of the terminal distribution.

More datasets were attempted, but we only managed to compute models on the datasets above on our hardware, see section 3.4.3.

3.5.1 Random Walk lineage analysis

To extract likely differentiation trajectories from the datasets we performed 5000 random walks on the transition matrices generated by the MultistageOT algorithm. Each walk originated from a randomly selected cell in the HSC cluster and a new starting cell was chosen at random for each walk. This was done to enforce the randomness of the walk. Each walk was performed until it ended up in a cell belonging to $\mathcal{X}_F$ or took more than 200 steps on the matrix, in which case the walk was discarded. This number was chosen so that the algorithm explored far enough in the graph while still being fast. Since a trajectory from a starting node to an end node had an average length of 20 steps, this number felt reasonable. Each cell visited during each walk was logged once it ended in a terminal cell, allowing for individual cell visits to be extracted for walks which ended in specific terminal cell types. We chose 5000 walks to not overextend the subsets but still cover most cells which could be included in the lineage analysis.

A subset containing a likely trajectory could then be extracted for each terminal cell type, containing all cells visited at some point during the random walks towards that type. As the walks are only based on transition probabilities, and as all cells have a majority of non-zero probabilities to transition to most other cells in the dataset, it is expected for the random walks to visit cells not involved in the expected cell differentiation lineage yet still terminating in the selected terminal cell type of interest. To account for this, we used the method `cell_fate_probabilities()` already implemented by MultistageOT to compute how likely cells in the subset were to end up in the specified terminal state. The probabilities computed are the absorbing probabilities for a Markov chain model on the transition matrix. We then filtered out the bottom 5% of cells with the lowest cell fate probabilities (i.e. removing cells falling below the 5th percentile in terms of likelihood to become the terminal cell type specified). We chose 5% as an arbitrary number to have a consistent way to achieve a "desired" subset. Such a method might be expanded upon in future work. An example of applying the filter can be viewed in Figure 3.1.

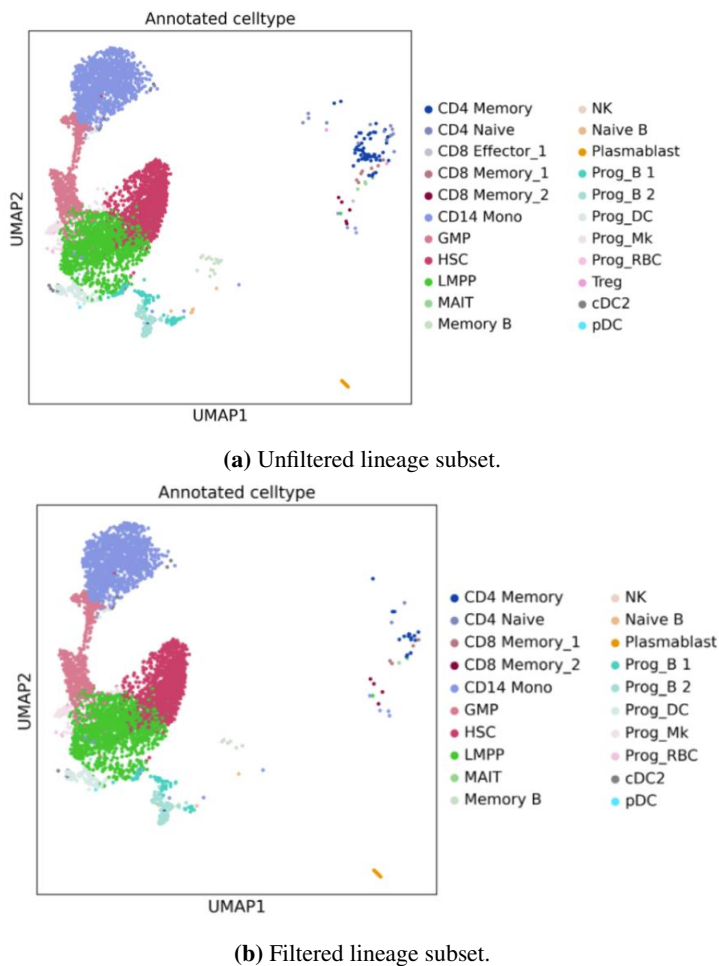
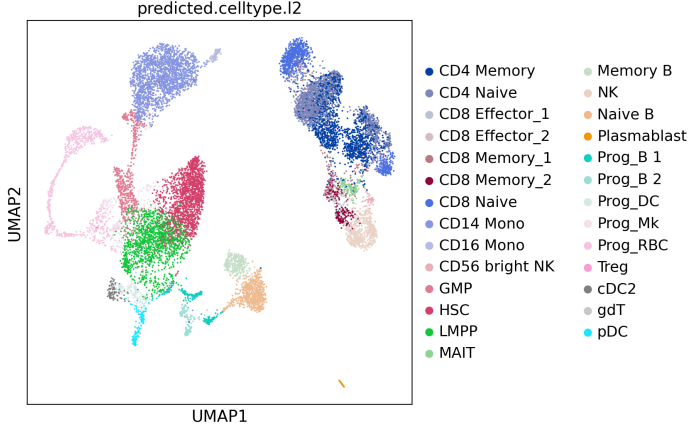


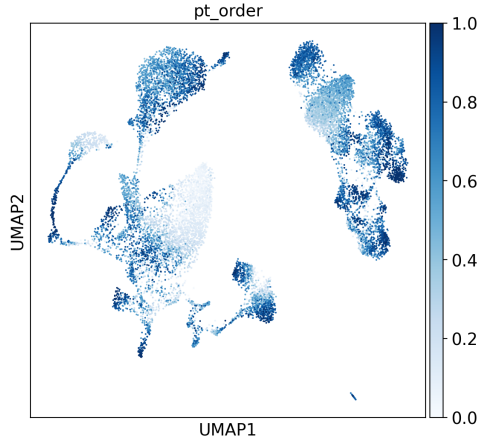
Figure 3.1 (a) Unfiltered and (b) filtered CD14⁺ Mono trajectory subset for the combined NBM8-CD34+NBM4-MNC+NBM7-MNC dataset. Filtering out the bottom 5th percentile of the subset reduced the subset size from 5024 cells to 4772 cells. The CD8⁺ effector cell type as well as Regulatory T-cells are removed from the subset by this process.

To visualize the trajectories we can look at the cells most frequently visited by the random walks for the chosen lineage. This can either be done by directly plotting the amount of visits each cell has across all of the specified random walks or optionally, and perhaps more easily interpretable, by summing visits across leiden clusters of high resolution. For easier visualization, we logarithmized the counts in both cases. Cell and cluster based count density is visualized in Figures 3.2a and 3.3. The resolution used was $\gamma = 2.5$ in order to have more clusters than the

l2-annotations while still having clusters containing a significant amount of cells. Gray dots are cells from the original dataset which were not visited in the filtered random walk subset.

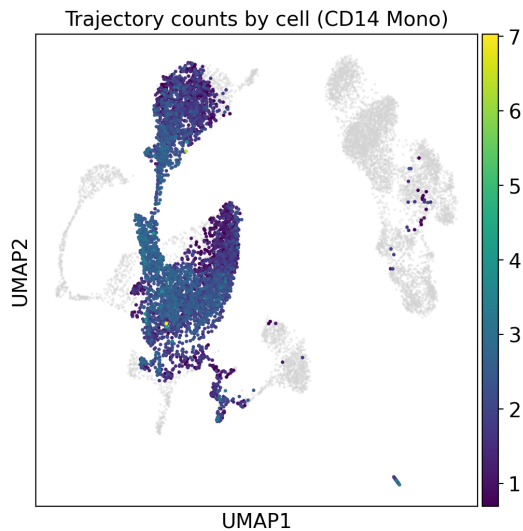


(a) l2-annotations.

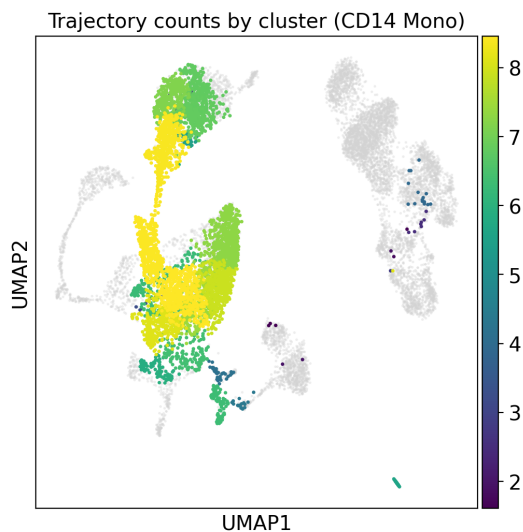


(b) Pseudotime ordering.

Figure 3.2 (a) l2-annotations and (b) Pseudotime ordering for the combination dataset NBM8-CD34 + NBM7-MNC + NBM4-MNC.



(a) Cell count density ($\log(x+1)$)-transformed.



(b) Cluster count density ($\log(x+1)$)-transformed.

Figure 3.3 Resulting subset from Figure 3.1b containing the cells which were visited on a random walk terminating in the $CD14^+$ Monocyte cells of $\mathcal{X}_F$ for the dataset NBM8-CD34 + NBM7-MNC + NBM4-MNC. **(a)** Shows the logarithmized visits for each cell. **(b)** Shows the logarithmized visits per Leiden cluster, as the visits were summed for the cells in each Leiden cluster.

Target Cell Type	Marker Genes
HSC	<i>AVP</i>
LMPP	<i>SPINK2</i>
GMP	<i>MPO, AZU1</i>
Granulocytic Progenitor	<i>ELANE</i>
Monocytes	<i>CD14, VCAN, FCN1</i>

Table 3.7 Marker genes used in the lineage analysis.

3.5.2 Gene lineage analysis

The final thing we evaluate is the behaviour of gene levels across the pseudotime. In order to do this, we subdivided the pseudotime axis into discrete bins where we gathered cells according to how their MultistageOT computed pseudotime value. We then take the expression of all 2000 genes across all of the cells in each bin and compute their mean value and standard deviation in order to compute boxplots for every gene average expression in every bin. A simple visualizing example is provided in Figure 3.4.

To evaluate the trajectories calculated by using the MultistageOT framework, we establish a baseline of canonical cell-type marker genes retrieved from the Cell-Marker 2.0 database [Hu et al., 2023]. The marker genes are listed in Table 3.7.

As can be seen, the common HSC marker gene *AVP* decreases in early pseudotime, whereas a GMP marker gene *AZU1* increases, only to decrease later as the marker gene associated with Monocytes, *FCN1* steadily increases, reaching its highest expression towards the end of the pseudotime axis. An analysis of the viability of the gene expression levels is then possible by boxplotting each gene expression for each bin, as is done in Figure 3.5, and by analyzing the composition of the cell types present in each bin, as is done in Figure 3.6.

We can then observe the accuracy of the gene estimates in a boxplot and evaluate the uncertainty based on what cells lie in each gene, such as in Figures 3.5 and 3.6.

In this thesis, we used 10 bins in order for improved readability and to have a sufficient amount of cells represented in each bin for the averages, yet still managing to subdivide most cell types into distinct locations in time.

With gene levels computed, we were able look at several metrics to extract potentially interesting and important genes, as gene levels which change drastically over time can be reasoned to be associated with ongoing differentiation processes. Such changes were computed, quite simply in this thesis, in the following ways:

1. PCA was performed on a lineage subset, and the top 100 genes which vary the most in the top 3 PC:s were extracted. A visualization of each cells coordinate along this PC is also generated to observe cell clusters which are separated along this PC.
2. The top 10 highest expressed genes in every pseudotime bin was extracted.

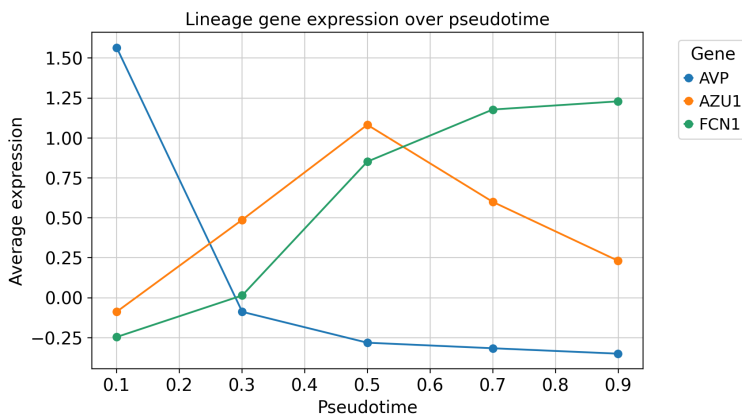
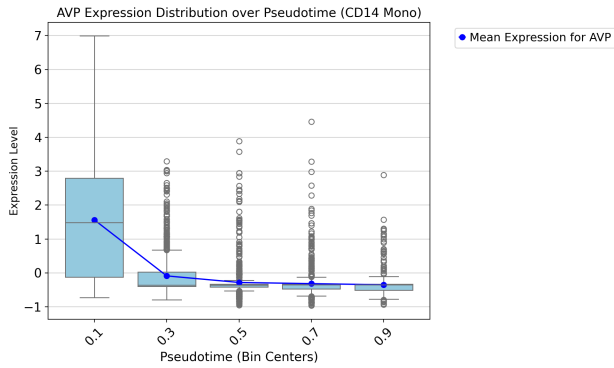


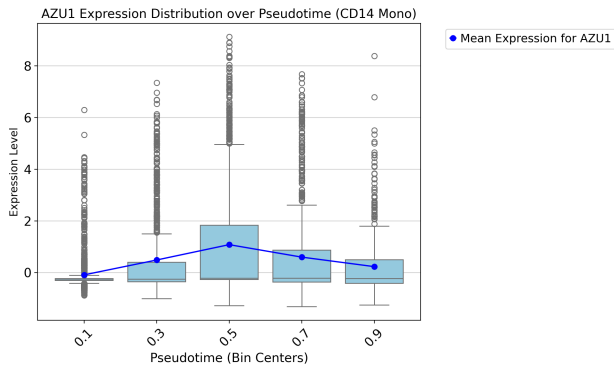
Figure 3.4 Example plot of mean gene expressions across 5 pseudotime bins for three different marker genes associated with cell types involved in the CD14⁺ Monocyte lineage, computed on the cells from Figure 3.2a.

3. The top 100 genes with the highest gene expression difference across the entire pseudotime is extracted.
4. The difference in gene expression between every pair of pseudotime bins to extract what genes are changing fast between cells. These four cases were considered:
 - a) The top 100 genes which had the largest increase in gene expression across all time bins
 - b) The top 100 genes which had the largest decrease in gene expression across all time bins
 - c) The top 10 genes which had the largest increase between every bin, totalling a list of 100 genes.
 - d) The top 10 genes which had the largest decrease between every bin, totalling a list of 100 genes.

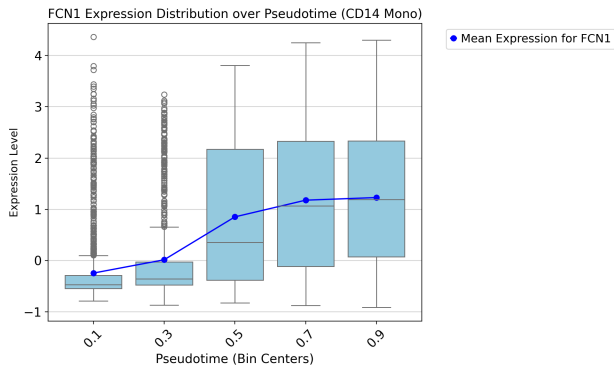
Several genes could be extracted this way, and a subset of genes generated from each method is plotted in appendix C. The full gene lists are available on the github, where we also performed this analysis on Memory B-cells, pDCs and NK cells.



(a) AVP.

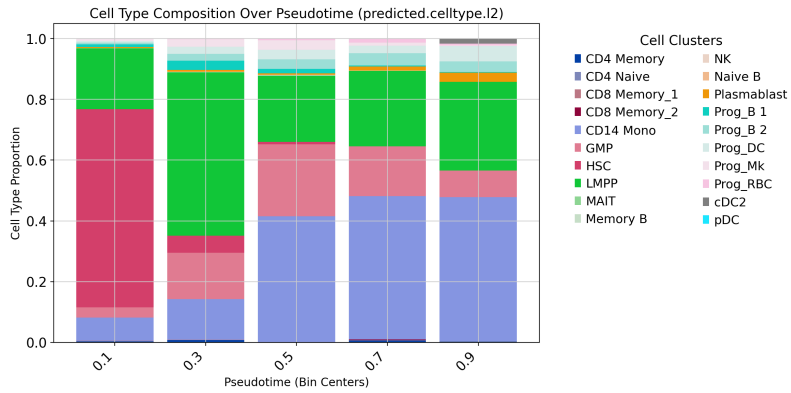


(b) AZU1.

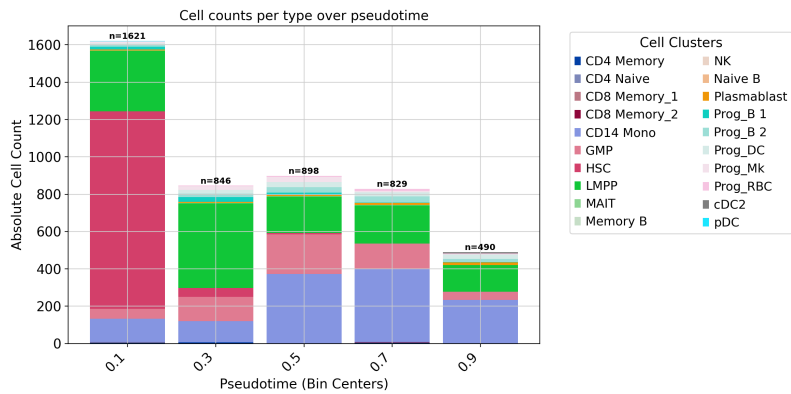


(c) FCN1.

Figure 3.5 Boxplots for the three average gene levels of (a) AVP, (b) AZU1 and (c) FCN1 from Figure 3.4. Perceived outliers are included as small circles.



(a) Percentual composition per bin.



(b) Absolute composition per bin.

Figure 3.6 Cell type composition in each bin from the dataset in Figure 3.4. (a) shows how large proportions of each l2-annotated cell are present in each pseudotime bin. (b) shows the absolute proportion of cells between bins. The number of cells in each bin is indicated with black text on top of each column.

4

Results

In this chapter we present our results from using the methodology in chapter 3. The lineage analysis will be conducted on the cell type CD14⁺ Monocytes. Supplementary figures are provided in appendices B and C. The results section is structured into three parts:

1. Evaluating RNA-velocity effects using the total variation in the fate predictions by MultistageOT.
2. Random walk lineage evaluation on normal bone marrow data.
3. Random walk lineage evaluation on AML bone marrow data.

4.1 Incorporating RNA-velocity into the cost function

The results from expanding the cost function to incorporating RNA-velocity from section 3.4.4 is presented in Table 4.1.

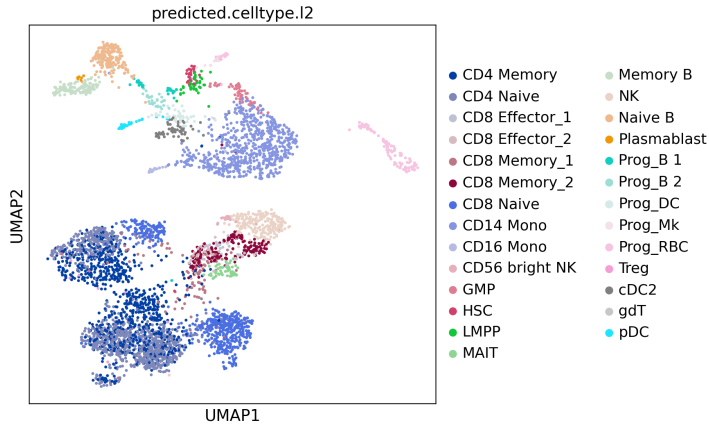
α	Total variation(+/- Std.)
0	0.4407 +/- 0.0023
0.1	0.4378 +/- 0.0023
0.2	0.4359 +/- 0.0023
0.3	0.4353 +/- 0.0022
0.4	0.4369 +/- 0.0021
0.5	0.4412 +/- 0.0020

Table 4.1 Table detailing the resulting total variation in cell fate predictions by weighting the RNA-velocity function by different amounts. The original MultistageOT cost function which only includes the squared euclidean distance as cost is represented by the $\alpha = 0$ entry.

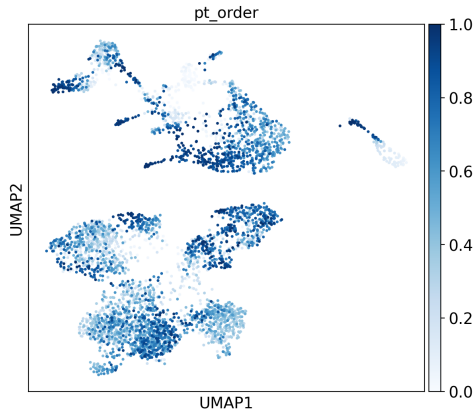
4.2 Lineage inference from optimal transport

To be sure that our method to extract the trajectory from a dataset is robust and consistent between datasets we can look at the average gene expressions of a set of marker genes plotted against pseudotime. The expression levels should be similar for subsets describing the same lineage. In Figures 4.1, 4.3, 4.5 and 4.7, the annotations and pseudotime ordering of the datasets of NBM4-MNC, NBM8-MNC and NBM10-MNC are presented, respectively. In Figures 4.2, 4.4, 4.6 and 4.8, the logarithmized visit density per cell and per leiden cluster are presented for each healthy dataset for random walks terminating in the CD14⁺ Monocyte cluster.

4.2.1 Healthy datasets



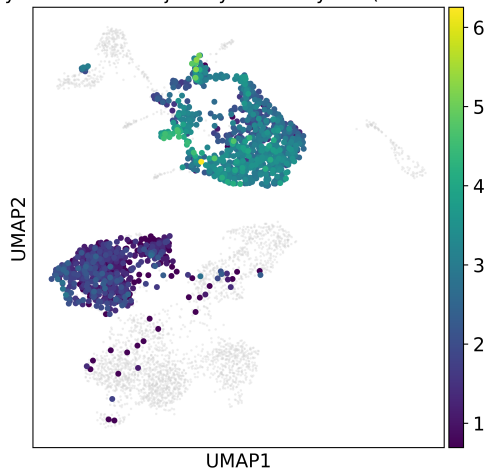
(a) l2-annotations.



(b) Pseudotime ordering.

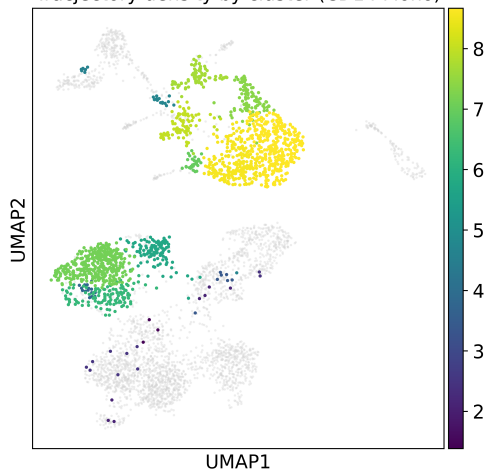
Figure 4.1 (a) l2-annotations and (b) pseudotime ordering for the NBM4-MNC dataset.

Layered UMAP - Trajectory counts by cell (CD14 Mono)



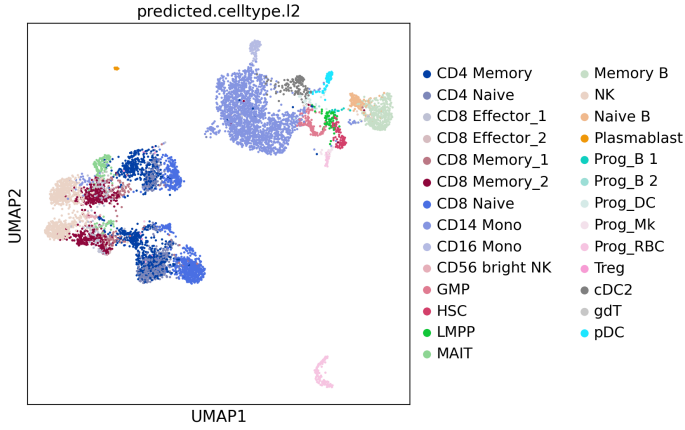
(a) Log-scaled visit density.

Trajectory density by cluster (CD14 Mono)

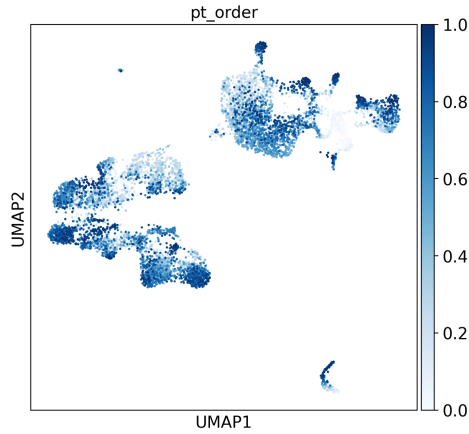


(b) Log-scaled visit density per cluster.

Figure 4.2 Random walk visit densities for the NBM4-MNC dataset. (a) Log-scaled visit frequency per cell. (b) Average log-scaled visits mapped per Leiden subcluster. Gray dots denote cells excluded from the random walk trajectory.



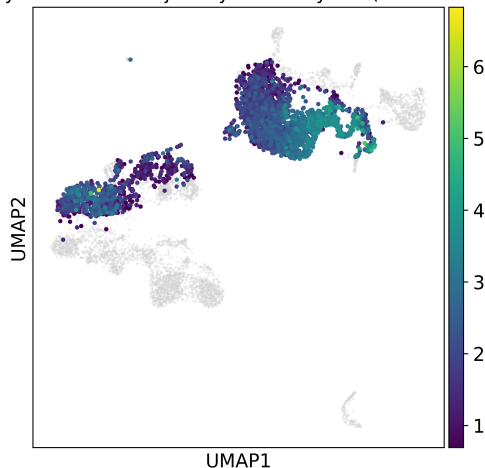
(a) l2-annotations.



(b) Pseudotime ordering.

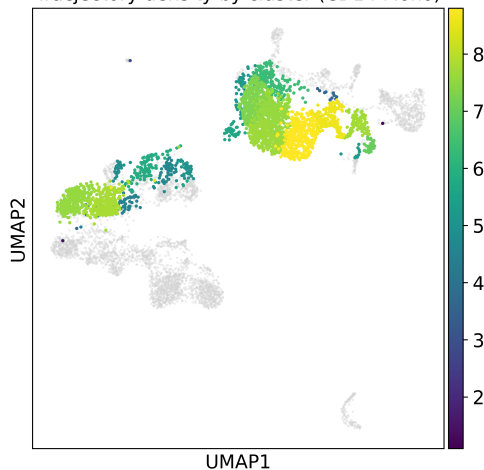
Figure 4.3 (a) l2-annotations and (b) pseudotime ordering for the NBM8-MNC dataset.

Layered UMAP - Trajectory counts by cell (CD14 Mono)



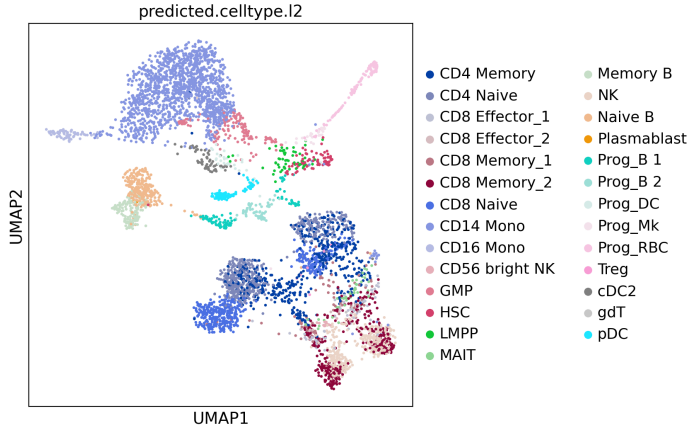
(a) Log-scaled visit density.

Trajectory density by cluster (CD14 Mono)

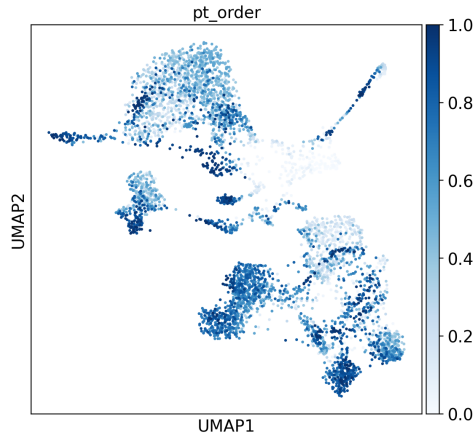


(b) Log-scaled visit density per cluster.

Figure 4.4 Random walk visit densities for the NBM8-MNC dataset. (a) Log-scaled visit frequency per cell. (b) Average log-scaled visits mapped per Leiden subcluster. Gray dots denote cells excluded from the random walk trajectory.



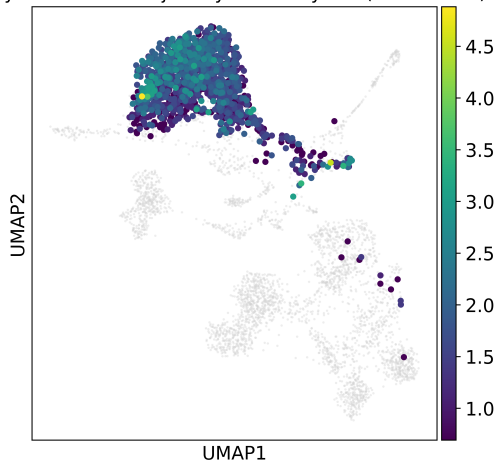
(a) l2-annotations.



(b) Pseudotime ordering.

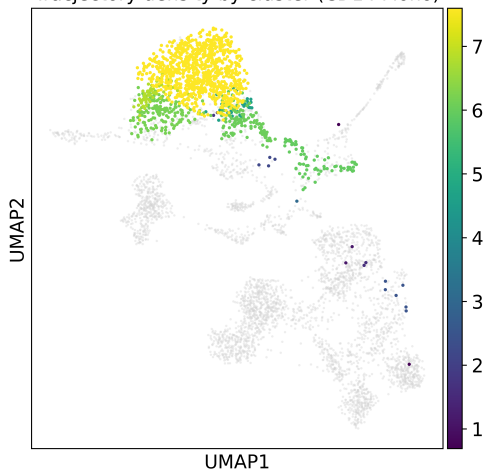
Figure 4.5 (a) l2-annotations and (b) pseudotime ordering for the NBM10-MNC dataset.

Layered UMAP - Trajectory counts by cell (CD14 Mono)



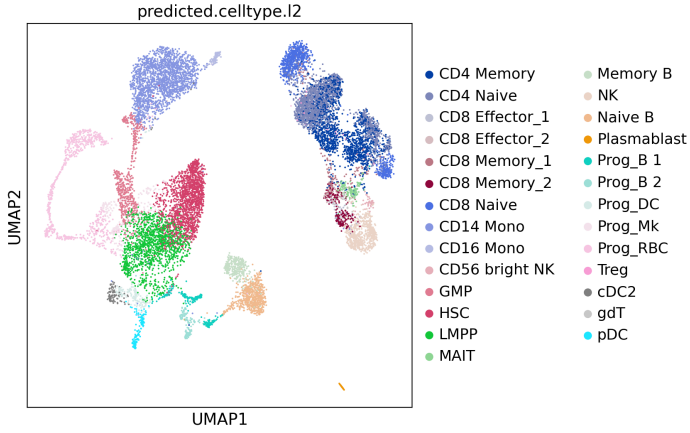
(a) Log-scaled visit density.

Trajectory density by cluster (CD14 Mono)

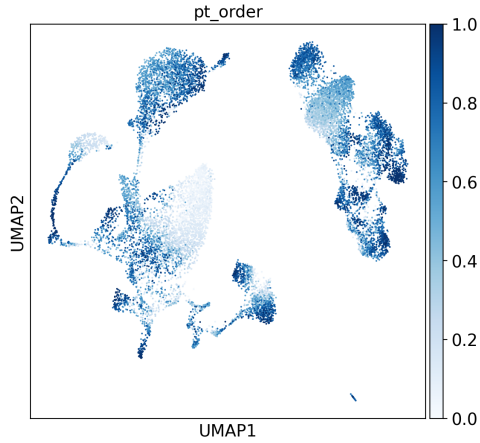


(b) Log-scaled visit density per cluster.

Figure 4.6 Trajectory extraction and random walk visit densities for the NBM10-MNC dataset. **(a)** Log-scaled visit frequency per cell. **(b)** Average log-scaled visits mapped per Leiden subcluster. Gray dots denote cells excluded from the random walk trajectory.

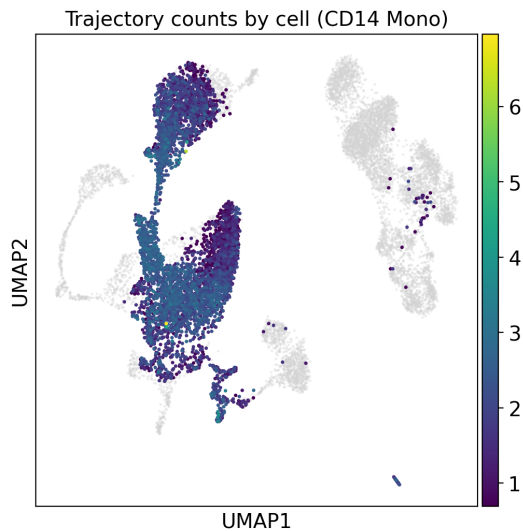


(a) l2-annotations.

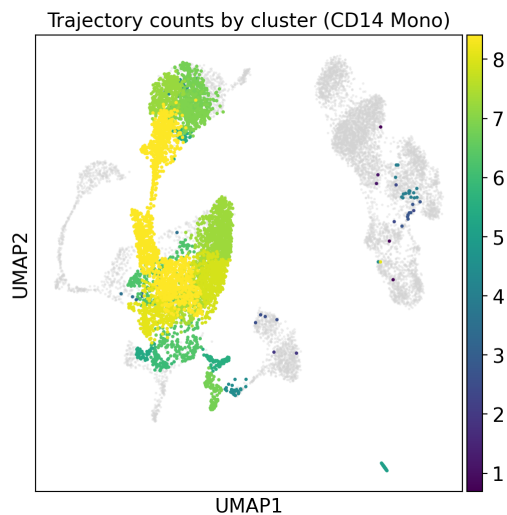


(b) Pseudotime ordering.

Figure 4.7 (a) l2-annotations and (b) pseudotime ordering for the NBM8-CD34 + NBM7-MNC + NBM4-MNC dataset.



(a) Log-scaled visit density.



(b) Log-scaled visit density per cluster.

Figure 4.8 Trajectory extraction and random walk visit densities for the NBM8-CD34 + NBM7-MNC + NBM4-MNC dataset. (a) Log-scaled visit frequency per cell. (b) Average log-scaled visits mapped per Leiden subcluster. Gray dots denote cells excluded from the random walk trajectory.

4.2.2 AML dataset

The annotations are presented in Figure 4.9 and the pseudotime ordering is visualized in Figure 4.10. The cell-and cluster count densities are presented in Figure 4.11.

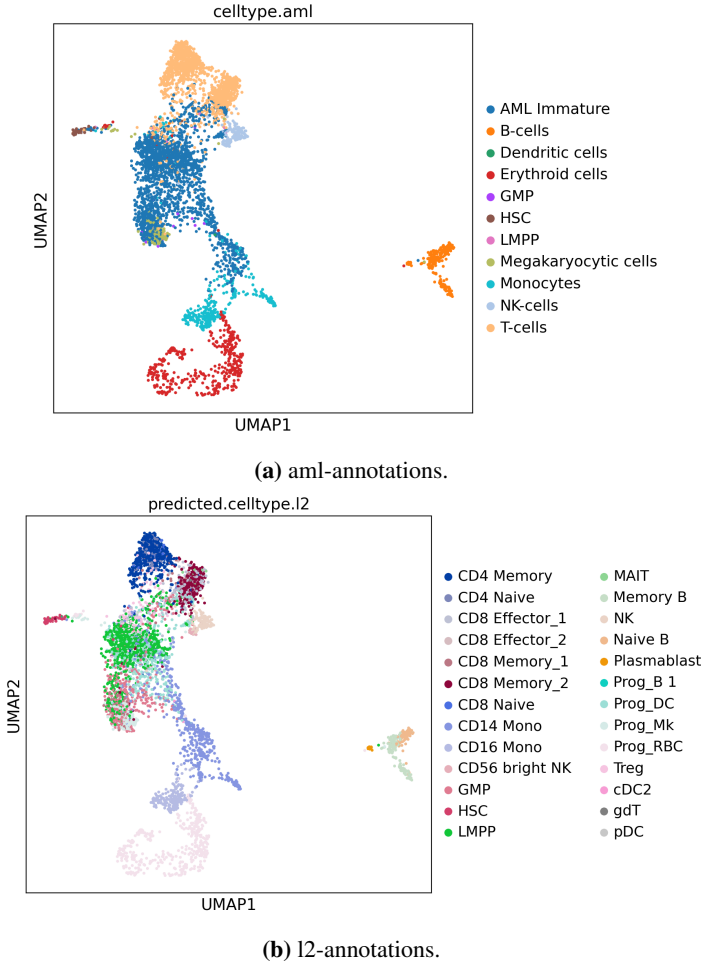


Figure 4.9 (a) aml-annotations and (b) l2-annotations for the AML155 dataset.

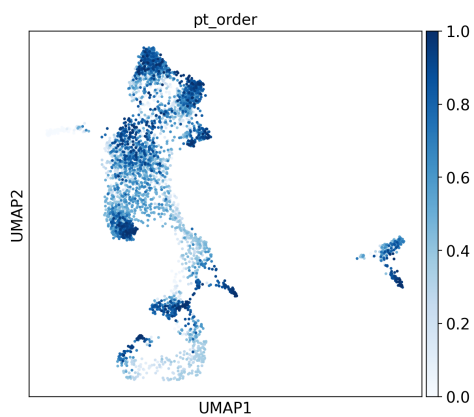
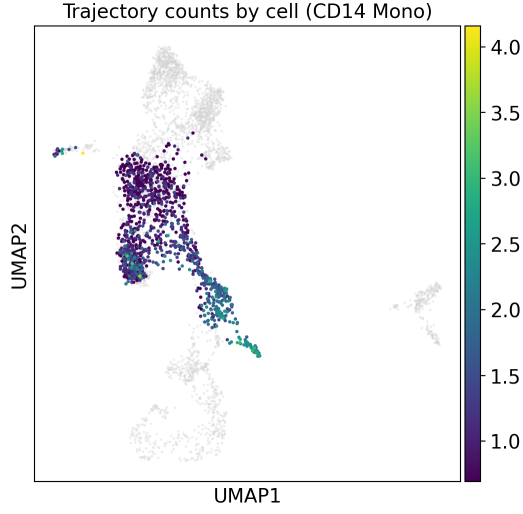
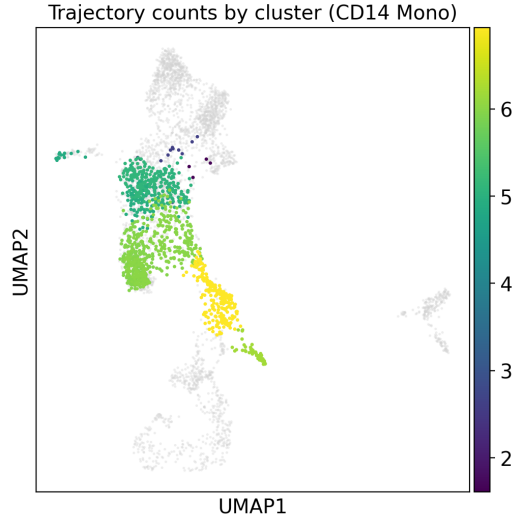


Figure 4.10 Pseudotime ordering of the AML155 dataset.



(a) Log-scaled visit density.



(b) Log-scaled visit density per cluster.

Figure 4.11 Trajectory extraction and random walk visit densities for the AML155 dataset. (a) Log-scaled visit frequency per cell. (b) Average log-scaled visits mapped per Leiden subcluster. Gray dots denote cells excluded from the random walk trajectory.

4.3 Marker gene expression over lineage subsets

4.3.1 Healthy datasets

In Figure 4.14, we have compared the marker gene expression of the datasets NBM4-MNC, NBM8-MNC, NBM10-MNC and NBM8-CD34 + NBM7-MNC + NBM4-MNC over pseudotime. Plots of the gene expression of these marker genes for the different cell types in the datasets can be seen in appendix B.

4.3 Marker gene expression over lineage subsets

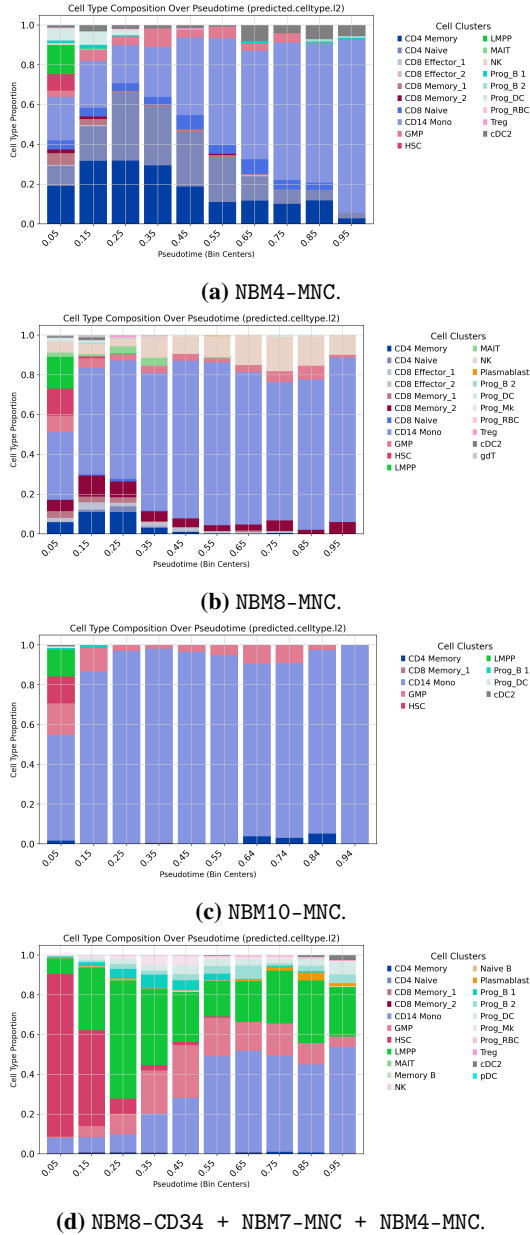


Figure 4.12 Proportional comparison of the l2-annotated cell type compositions for the healthy datasets across the pseudotime bins. As NBM8-CD34 + NBM7-MNC + NBM4-MNC is progenitor-enriched, it has much greater representation of progenitor cells.

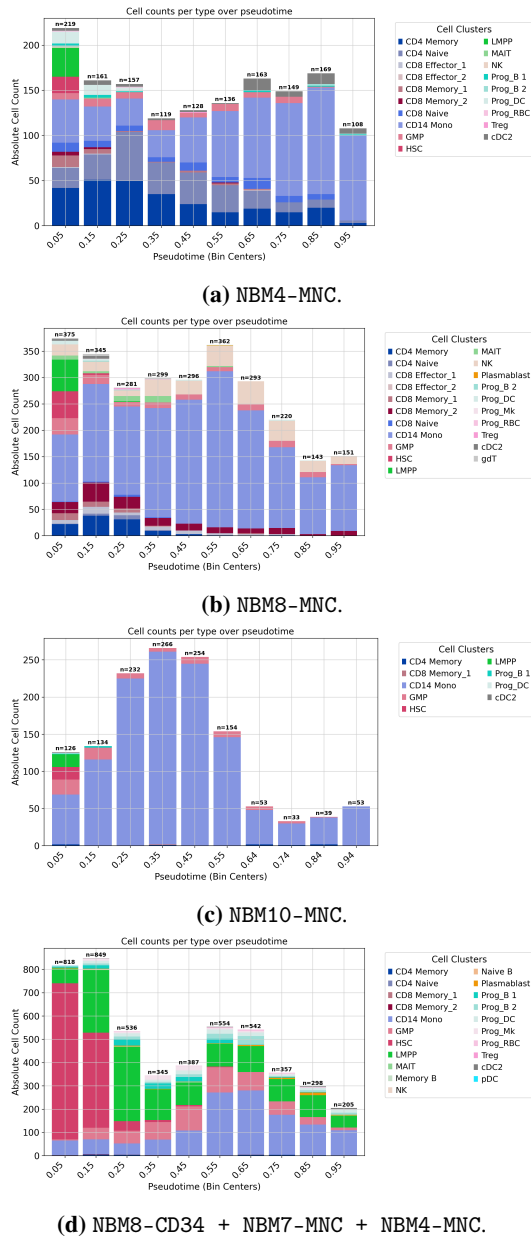
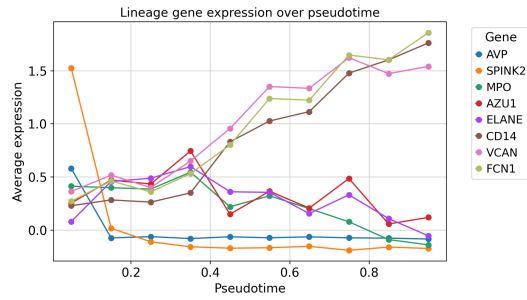
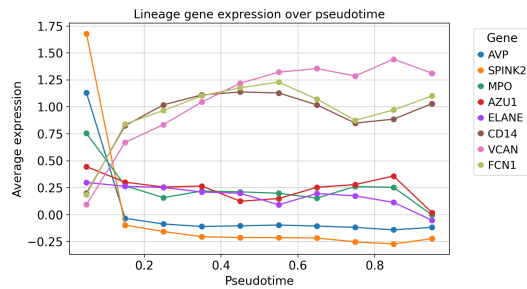


Figure 4.13 Absolute count comparison of the l2-annotated cell type compositions for the healthy datasets. The number of cells in each bin is indicated on the y-axis as well as in black text on top of each bar.

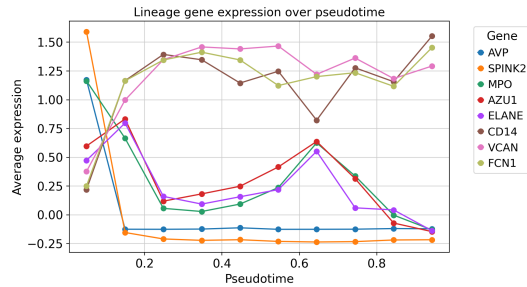
The average marker gene value plots in all healthy datasets are provided in Figure 4.14. Their gene level estimates is reviewed with boxplots of the gene expressions in each cell bin, without outliers, in Figures 4.15 to 4.18. Plots with outliers are included in appendix C.



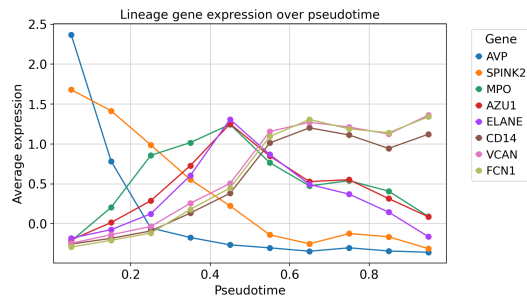
(a) NBM4-MNC.



(b) NBM8-MNC.



(c) NBM10-MNC.



(d) NBM8-CD34 + NBM7-MNC + NBM4-MNC.

Figure 4.14 Comparison of the marker gene expression between datasets.

4.3 Marker gene expression over lineage subsets

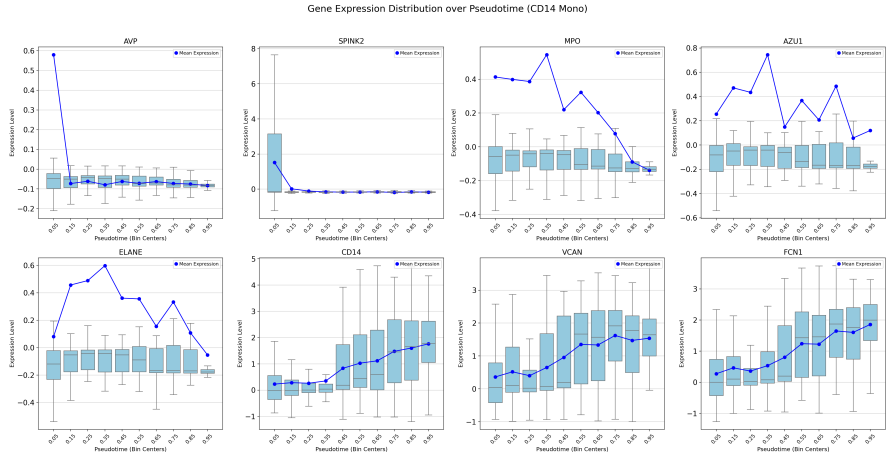


Figure 4.15 Boxplot visualization for each marker gene estimate for the dataset NBM4-MNC.

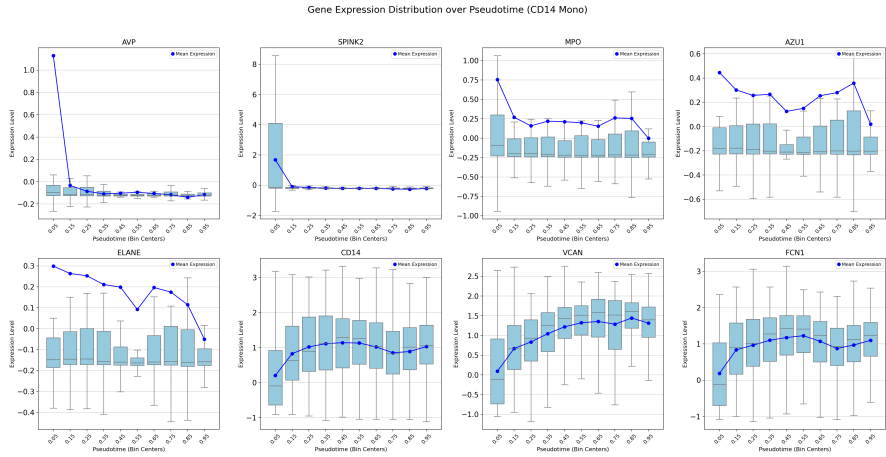


Figure 4.16 Boxplot visualization for each marker gene estimate for the dataset NBM8-MNC.

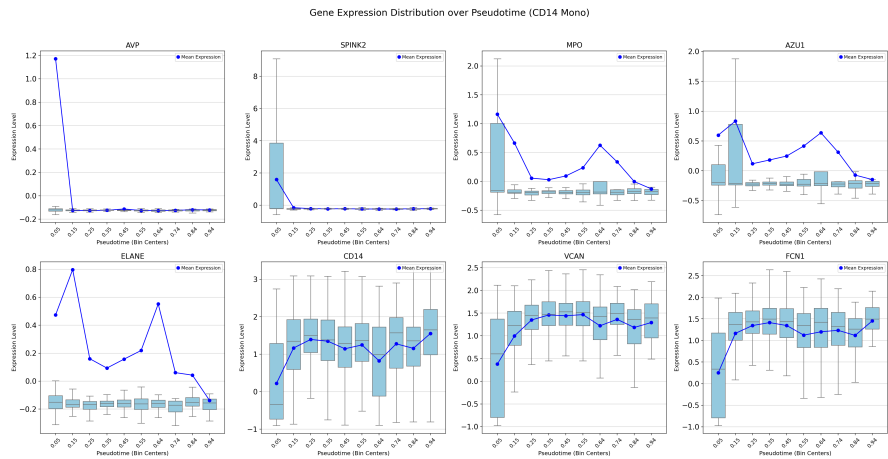


Figure 4.17 Boxplot visualization for each marker gene estimate for the dataset NBM10-MNC.

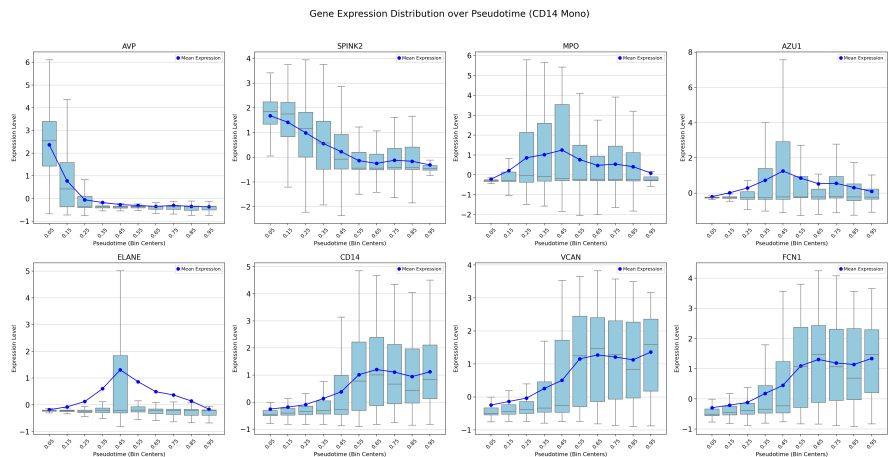
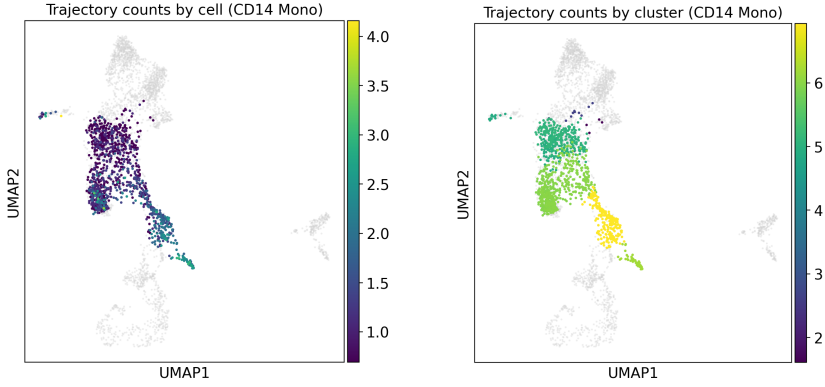


Figure 4.18 Boxplot visualization for each marker gene estimate for the dataset NBM8-CD34+NBM7-MNC+NBM4-MNC.

4.3.2 AML dataset

The AML cell composition is illustrated in Figure 4.20. Running Multistage optimal transport and extracting from the random walks, the subset corresponding to the $CD14^+$ Monocyte lineage we get the trajectory represented in Figure 4.19. The cell counts and composition of the subset can be viewed in Figure 4.20. The marker genes for the lineage is presented in Figure 4.22. The marker gene boxplots are

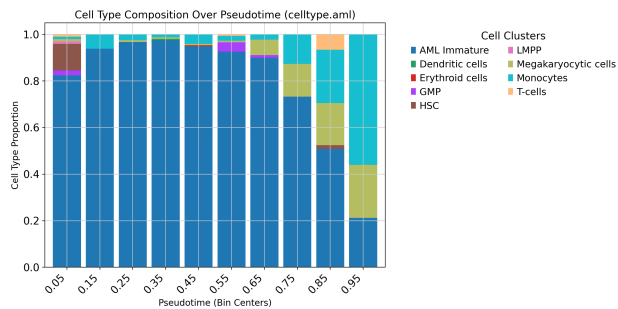
presented in Figure 4.23. A plot of quickly changing genes in early pseudotime is presented in Figure 4.24, and the expression distribution plots of two genes from the list in Figure 4.24 is presented in Figure 4.25.



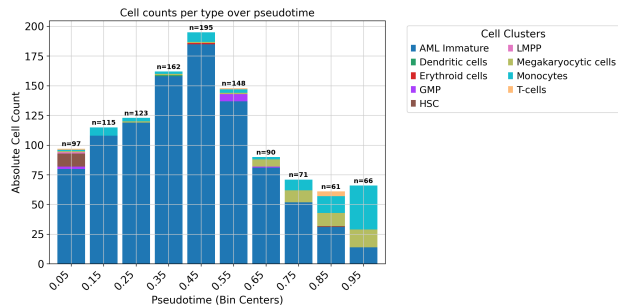
(a) Log-scaled visit density per cell.

(b) Log-scaled visit density per cluster.

Figure 4.19 UMAP representation of AML-positive sample with extracted trajectory subset for CD14⁺ Monocyte lineage. (a) shows, for each cell, the logarithmized number of random walk visits for random walks which terminate in the Monocyte cluster. (b) shows the same metric, but averaged over subclusters in the AML dataset.



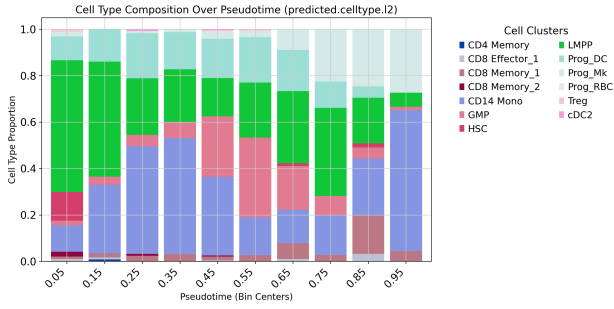
(a) AML155 subset composition.



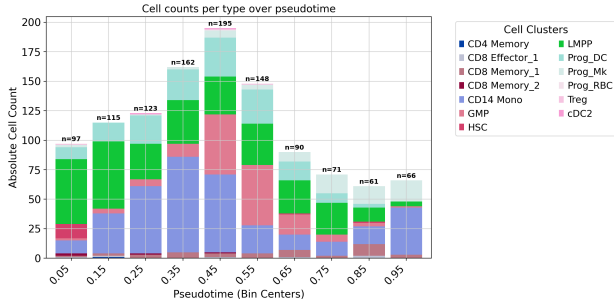
(b) AML155 subset bin cell counts.

Figure 4.20 AML155 subset composition and bin cell count with aml-annotations.

4.3 Marker gene expression over lineage subsets



(a) AML155 subset composition.



(b) AML155 subset bin cell counts.

Figure 4.21 AML155 subset composition and bin cell count with l2-annotations.

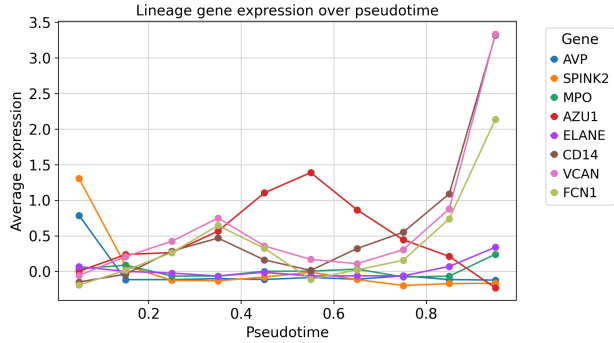


Figure 4.22 Marker genes for the CD14⁺ Monocyte lineage subset in the AML155 dataset.

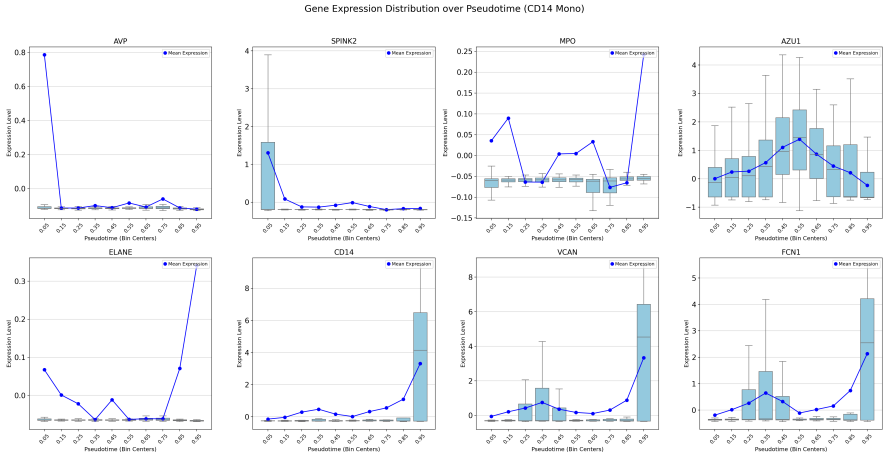


Figure 4.23 Boxplot estimates of the common marker genes for AML155.

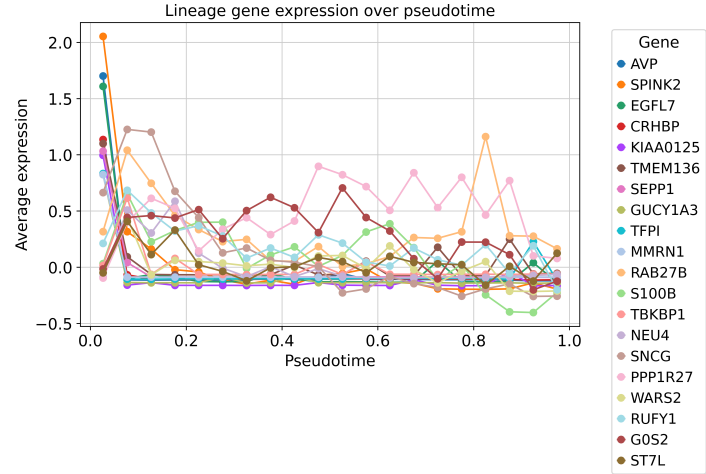
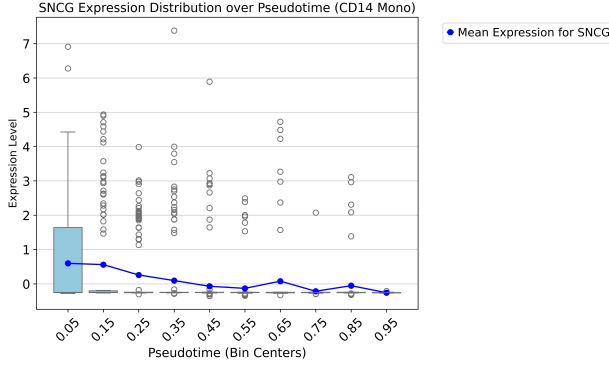
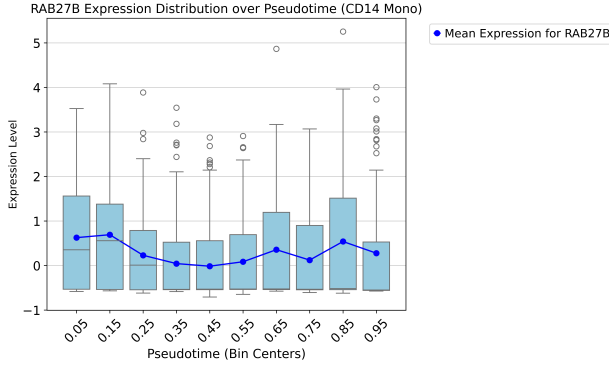


Figure 4.24 Rapidly changing genes in early pseudotime for the CD14⁺ Monocyte lineage subset in the AML155 dataset.

4.4 Principal Component Analysis on the generated subsets



(a) Expression estimate of SNCG.



(b) Expression estimate of RAB27B.

Figure 4.25 Expression distribution of abnormal genes in AML155.

4.4 Principal Component Analysis on the generated subsets

4.4.1 Healthy datasets

To extract further information about the dynamics of the healthy bone marrow cells we can perform PCA on the subset and visualize the expression of the components in a UMAP-representation. The PCA score for the cells in the CD14⁺ Monocyte subset is visualized for the top three PCA components in Figures 4.26 to 4.29 for the considered healthy datasets. The different principal components then separate different parts of the subset and thus the different parts of the differentiation stages of the cells. We can extract the PCA loadings, lists of genes driving the positive and negative direction of the genes, and these are presented on the thesis Github.

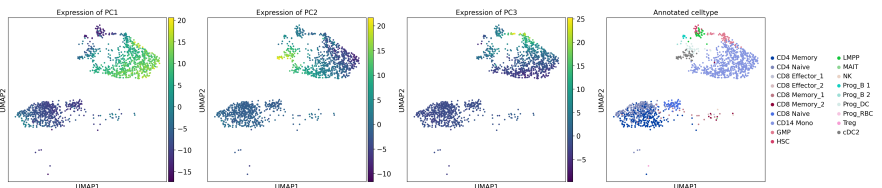


Figure 4.26 Cell score along the top 3 principal components of NBM4-MNC expressed in UMAP together with the l2-annotations of the cells which make up the subset.

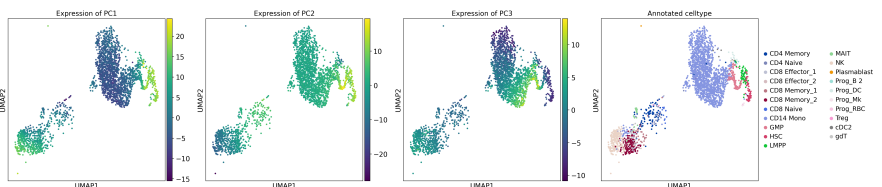


Figure 4.27 Cell score along the top 3 principal components of NBM8-MNC expressed in UMAP together with the l2-annotations of the cells which make up the subset.

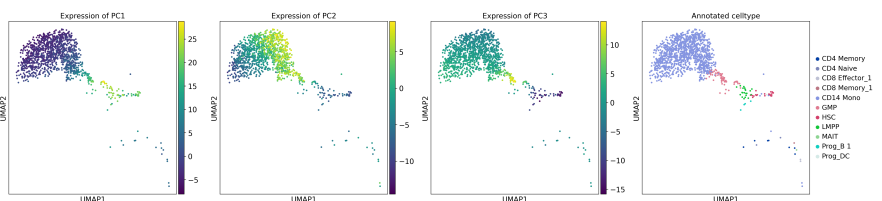


Figure 4.28 Cell score along the top 3 principal components of NBM10-MNC expressed in UMAP together with the l2-annotations of the cells which make up the subset.

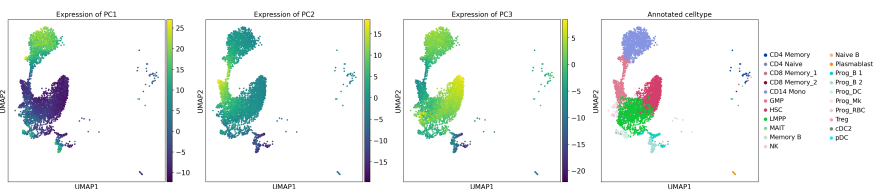


Figure 4.29 Cell score along the top 3 principal components of NBM8-CD34 + NBM7-MNC + NBM4-MNC expressed in UMAP together with the l2-annotations of the cells which make up the subset.

4.4.2 AML Dataset

The scores of the top three PCs for the AML155 dataset is visualized in Figure 4.30.

4.4 Principal Component Analysis on the generated subsets

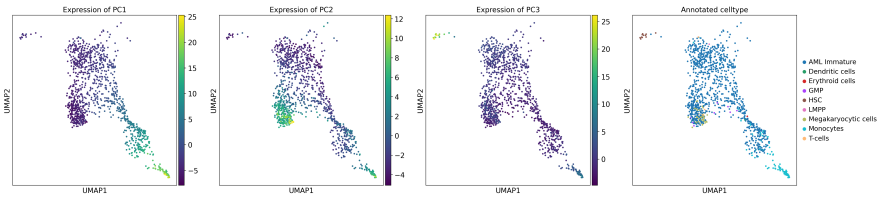


Figure 4.30 Cell score along the top 3 principal components of AML155 expressed in UMAP together with the aml-annotations of the cells which make up the subset.

5

Discussion

5.1 The complexity of biological data and dimensionality reduction

Biological data is quite hard to work with and draw conclusions from, especially in processes when we only have snapshot data such as the ones gathered by scRNA-sequencing. From a biological perspective, it is hard to define exactly what cell-type or cell-state a cell belongs to, or what it even is. We can use a definition based on gene expression data but classification may still differ between samples. We can perform our models and extract our subsets but as we can see, for example, in Figure 4.4, the subset generated by the random walk mostly captures the annotated cell-types known to be of the CD14⁺ monocyte lineage yet we see some cells from other annotated cell-types included as well. Comparing this with the known marker genes for Monocytes, Figure 4.14, we can see that these marker genes also show up in some areas of the T-cells, seen in appendix B. Since the models run on computed transition probabilities in the gene space, it would seem natural that these T-cells would show up in the random walks. Defining cells by just gene expression has this drawback, and this is why we choose to threshold based on fate probabilities, and include these differently annotated cells in the subsets as they still had sufficiently large probabilities to remain after the filtering step. This shows that the boundaries between cell-types, or cell-states, are not as rigid as one would hope. The true cell-type of a single cell could also be very fluid and change over time. There are some reported cases where cells could even differentiate backward in their trajectories.

We can not be fully confident in the annotations of the cells, so it could be possible that the annotated cells which are not classified as monocytes still, in a gene expression sense, are close to being monocytes. Looking at UMAP-representations, such as in Figure 4.2, Figure 4.4 and Figure 4.6, we can see that single cell annotations can deviate from the surrounding cells. This can either mean that they are annotated wrong or that they are rightfully annotated but show up in another cell cluster for some other reason we do not know about.

This brings us to the limitations of the dimensionality reduction methods we use in this thesis to make the data useful. When representing very high dimen-

sional data such as the ones gathered via scRNA-seq in lower 2- or 3-dimensional space, we will naturally lose information about our data. There might be some high dimensional dynamic that places Monocytes in a T-cell cluster which allows the mass transition to other T-cells. The UMAP and our algorithms are based on 50-dimensional PC reduction, which takes 30000-dimensional data and projects it onto 50 which is then projected onto 2 dimensions. One should therefore be careful of drawing conclusions based on these dimensionality reduced presentations.

To get around the problems adhering from the complexity of the data, more rigorous methods of preprocessing could be applied but also a deeper analysis of the dimensionality reduction. Analysis of bigger datasets averages out the small anomalies present in smaller dataset. However, this complexity and uncertainty is something that is inherent to biology and will be present in all single-cell analysis.

5.1.1 Choice of initial and terminal states

A consequence of the difficulties of defining cell-types is how we choose our initial and terminal distributions.

We chose distributions based on the assumption that the difference in cell functionality between the cells could be based on their distance from each other in the gene space and used the k NN graph distance for the selection of terminal states. However, the initial states are just chosen in the stem cell cluster, based on a UMAP representation and the I2-annotations. We cannot be sure that this is the best interpretation and that these biological processes would adhere to euclidean metrics. Another method, as discussed above, could be based on marker gene expression. But as discussed earlier, and looking at appendix B, marker genes are not solely expressed in the cell types which they are markers for and cannot be interpreted as a ground truth.

Further improvement of the model should also include how to choose an initial state for the algorithm.

5.1.2 Batch Effects

The size and composition of the dataset has an impact on our results. Comparing smaller datasets, such as the MNC datasets, with bigger combinations of datasets, including CD34 datasets, we get different compositions and pseudotime orderings. This can be seen in Figure 4.12 where the dataset NBM8-CD34 + NBM7-MNC + NBM4-MNC has a higher resolution among the progenitor cells, that reaches further into pseudotime. It is therefore important to note that the datasets needs to be similar in structure if the plots are to reflect the same gene levels and pseudotime orderings. For CD34 enriched datasets, we can expect more information in early pseudotime in the plots. The marker genes for progenitors and more differentiated cells differs more in Figure 4.14 in the NBM8-MNC dataset due to it being larger (see Table 3.2) and therefore has more progenitor cells. For example, NBM8-MNC has 121 instances of annotated *GMP* cells while NBM4-MNC has 62. This effect can be seen in the com-

position and raw count plots in Figure 4.12 and Figure 4.13. It can also be seen in Figure 4.4 that the density is higher for progenitors in this dataset. This could also be an artefact of the dataset being bigger as the same thing is visible in Figure 3.3.

This implies that bigger datasets are likely to lead to better results for the methods we use, and combining datasets so that the methods are run on both MNC and CD34 samples gives more resolution in both early and late differentiation stages. For an even better analysis, combining even more datasets would be an interesting way to continue, however, we couldn't do this because of the limitations of our hardware.

5.1.3 Gene expression certainty

Another interesting aspect of dataset sizes is to observe how it affects the certainty of the gene level estimates. Due to the smaller amount of progenitor cells in the -MNC samples, the progenitor gene levels are highly affected by "outliers" in the boxplots, as can be seen in Figures B.11 to B.13. The average levels of the Monocyte marker genes instead lands well within the boxplot boundaries, Though we can still observe a large spread in the monocyte marker gene expression for the datapoints, about 50% of all datapoints are contained well within a range of ± 0.4 of the average gene expression level in the final pseudotime bin, indicating the gene levels have somewhat stabilized among the cells in the bin.

It is only in the combined dataset in Figure 4.18 where estimates look reliable across all marker genes, likely due to the progenitor cells being better represented. These progenitors do end up present in almost every bin, and we think this can be attributed to the way pseudotime is computed, as well as the size of the bins. Due to the large bins, cell types are bound to be grouped together, and the LMPP:s in the final pseudotime bins in Figure 4.18 might be far away from easy mass transport paths in MSOT in the high-dimensional space. As pseudotime is not computed by amount of mass, but rather using proportions between mass in the multiple transport plans to estimate a pseudotime, the cells might get ordered late by receiving some small bit of mass towards the end of the trajectory.

Surprisingly, the gene levels of AZU1 for AML data in Figure 4.23 are somewhat reliable, maybe due to the increase of LMPP-like cells in the dataset, as is evident in Figure 4.21.

5.2 MultistageOT

5.2.1 Entropy parameter

The value of the entropy parameter ϵ is important for the results of the MultistageOT algorithm. We looked at what would happen if we didn't run the extra optimization of ϵ and then doing random walks on the transition matrix. The result could not be interpreted as something representative of the biological processes. The optimization of ϵ is therefore crucial for the algorithm to describe cell differentiation. Since this optimization algorithm is very computationally heavy, it would be interesting

to see if an even lower ϵ could make the results even better, though as explored in [Tronstad et al., 2025], the value of ϵ converges to around the same value for a big selection of datasets, which is the motivation as to why we chose our value and did not explore this further.

5.2.2 Pseudotime

MultistageOT orders the cells in an expected way in pseudotime. Looking at marker gene plots, such as Figure 4.14 we can see that the pseudotime ordering is reflective of when marker genes should be expressed. Since MultistageOT is computed in multiple steps based on the Euclidean distance cost function, and if the assumption that Euclidean distance is a good measure for difference in cell states, the pseudotime ordering should be accurate.

There are many methods for inferring pseudotime ordering in scRNA-seq data. We do not claim that any method is better than any other, and using other pseudotime methods for the lineage subsets in our results might be interesting. A comparison of different methods is relevant, since pseudotime does not reflect actual time and would probably differ between methods.

5.2.3 Trajectory Inference

MultistageOT combined with the Random Walks finds the expected trajectories according to the annotations of the data and knowledge of biological processes. The cells included in the subsets can be reasonably assumed to represent the lineages. Looking at the plots in Figure 4.14, the marker genes corresponding to different parts of the lineage are ordered as expected in pseudotime. In the subsets, some cells and clusters are more frequently visited, as seen in Figures 4.2, 4.4, 4.6 and 4.8, and could be interpreted as important in the development of the lineage. Creating smaller subsets based on the visits of cells or in clusters, one could extract gene lists that could describe finer processes within the lineage subsets, and could be a further point of investigation.

Using the gene extraction methods, one could perform a much deeper analysis by studying genes which vary a lot in PCA directions as well as changes a lot over the course of the pseudotime axis through some sort of lineage analysis. We have extracted such lists for our datasets but did not feel we had the time nor the background to perform such an analysis. It could further verify the effectiveness of MultistageOT, if these gene lists prove to contain a majority of genes associated with differentiation of the different lineages. The lists are provided in C for further analysis.

5.3 Design Parameters

In this thesis, and in bioinformatics in general, there are a lot of arbitrary design parameters that needs to be chosen. This could be, for example, the amount of prin-

cial components to use in the preprocessing of the data and how many neighbors are reasonable in a k NN-graph. We get suggestions from [Heumos et al., 2023] about several common parameters in single cell analysis, but these are not set values and we are often given an interval instead of fixed values to work with. In the MultistageOT algorithm there are also a lot of parameters that need to be decided, such as the number of intermediate stages in between the initial and terminal distributions or the dimensions of the data fed into it. This will affect the results of the algorithm in different ways, for example the resolution in pseudotime, but at the cost of longer computations. We can base our choices on how the authors [Tronstad et al., 2025] have demonstrated their method and on our computer limitations. In our subsequent analysis we also need to choose parameters for our random walks. The purpose of this thesis is not to optimize the parameters to get the best results so tuning of these parameters is also something that can be further explored.

5.4 RNA-velocity

As demonstrated in Table 4.1, incorporating RNA-velocity to drive the gene seems to have some positive effect but the resulting impact is very small. It could therefore be interesting to see to what extent RNA-velocity could be a contribution in optimal transport frameworks, by using other cost functions or processing the data in other ways.

However, since the improvements were minimal, RNA-velocity data might not be too helpful in the study of trajectory inference. For example, it does not specify what the next cell state is, to which the velocity vector points to. Since there are many biological processes occurring at the same time in a sample, the next state is not necessarily one of a differentiation trajectory, but maybe a state in the cell cycle or circadian cycle. Therefore, one should not put too much weight into RNA-velocity in the sense of inferring specifically differentiation trajectories without knowing the necessary signals from these different processes.

5.5 PCA on a subset

Doing a separate principal component analysis on the extracted subset might give further information into the gene dynamics in the subset. In section 4.4 we see the first three principal components of a subset visualized in a UMAP representation. Looking at the plots, the PC's are highlighted in very distinct parts of the subset, and assuming that the subset describes differentiation of the lineage, extracting the genes that drive the PC's, we can get information about which genes are important for this specific part of the differentiation process.

5.6 Application on an AML-positive sample

We can see that in order for the algorithm to reach the terminal states in the CD14⁺ Monocyte cluster, mass needs to go through many cells annotated as *AML Immature* which overlaps with annotations for many progenitor states according to the NBM equivalent annotations. That the *AML Immature* affects the progenitors can be seen in the corresponding marker gene plot for this lineage on the AML-dataset, where progenitor marker genes for Monocytes, *ELANE* and *MPO*, are not present (see Figure 4.22) as they were in the NBM-samples (see Figure 4.14).

Looking at Figure 4.22 and Figure 4.20, we can see that the transition from *HSC* to *AML Immature* should happen rather early in pseudotime. Between the pseudo-time bins where *HSC* markers, *SPINK2* and *AVP*, drop we can plot the genes that most rapidly heightens their expressions, see Figure 4.24. Among these upregulated genes we find, for example, *SNCG* and *RAB27B* which are known markers for development of different types of cancer [National Library of Medicine (US), 2026], [Ren et al., 2023].

The future of how this method is applied to cancer data needs to be further established, also since MultistageOT is mainly used to describe healthy hematopoiesis. Hopefully, the methodology here can be further developed in order to understand the underlying gene expressions in AML samples.

5.7 Further research

5.7.1 Unsupervised Pipeline

Using optimal transport in the supervised way described above, aims to reconstruct the already expected lineages in the dataset, based on the initial and terminal states chosen by the given annotations. For further investigation of the data, we suggest that the pipeline is done unsupervised with a cluster resolution higher than the one provided by the annotations. If MultistageOT accurately captures the dynamics of the process, we hope that finer results could be found inside the annotated clusters if they are split up, and the genes responsible. The proposed way to chose terminal states is to use clustering algorithms to define new, smaller clusters and choose end states corresponding to these clusters instead.

Clustering. Since the annotations provided only are so detailed, we thought the Leiden clustering might allow be able to find cell subgroups which better capture differentiation trajectories between different cell types. As an example, consider the LMPP cluster. It consists of cells which might differentiate into Monocytes, Dendritic cells, T-cells and its subgroups, or B cells and its subgroups. By performing MultistageOT models on datasets where the terminal distributions are drawn from smaller subgroups of cells, we can hope to find even more specific cell subgroups involved with the differentiation of cell types. This would in turn also allow us to have better cell representation when computing average gene expression levels in

the bins as we hopefully now only include progenitor cells which are heading towards certain cell types, giving us better measurements of gene expression levels. Furthermore, methods from machine learning might be able to identify good terminal clusters and states.

Challenges. Since we cluster not based on annotations, we cannot know based on the clusters alone what would be stem cells and terminally differentiated cells. The clusters chosen to represent initial and terminal cell types must therefore be decided in another way. The choice of an initial cluster could be based on known marker genes for stem cells or progenitors where good choices would be AVP or CD34. To choose end states, we could base this off of marker genes as well but if we want to capture dynamics inside cell-type clusters, marker genes might not be enough to identify these differences. We could choose every cluster, with the exception of the one we define as initial, with the motivation that we do not know which clusters are terminal. The problem with this is that clusters that correspond to progenitors or cells in the middle of the differentiation process will be chosen as terminal states and mass that reaches this state will exit the system, which would not be biologically accurate. Another method could be choosing the N clusters furthest from our chosen initial state and validate this on the UMAP, but this method could fail to capture important end state cell-types.

5.7.2 Doublets

We have not preprocessed the occurrence of doublets in our datasets. This could be an important aspect to look at since single cells can give differing signals which might affect the algorithms run. After consulting with the bioinformatics team we chose not to take doublets into account since they filtered out based on an arbitrary threshold. We were not comfortable deciding this with our limited biological knowledge, since a too high threshold might filter out normal data points. However, with the correct knowledge of how this is done, it might improve the results of the algorithms and is worth looking into.

6

Conclusions

MultistageOT generates a transition matrix that describes the differentiation of cells in hematopoiesis in an expected way based on common biological knowledge and the cell-type annotations of the data.

Incorporating RNA-velocity in the cost function of MultistageOT improves the predictions only marginally, suggesting that RNA-velocity describes some of the development of biological processes. It is however not a very strong tool. The implementation of this data could probably be improved but seems to be limited in how it contributes to optimal transport descriptions of cell differentiation.

We can then capture a subset of cells which accurately describes a lineage from the cluster annotations by doing random walks on the transition matrix and registering the visited cells. The labels of these cells are in accordance with the expected cell-types associated with the lineage we look at. Due to the complexity of biological data some discrepancies appear and effective filtering of these discrepancies would be interesting in future work. From the subsets we can extract lists of genes based on their expression in pseudotime and different principal components. These methods can be applied to AML-positive samples, and upon comparison with healthy data, we can isolate genes that differ from the usual expression. Since the optimal transport algorithm can be used to find relevant subsets of cells for gene analysis and to infer pseudotime that tracks quickly changing genes, applying this to cancer data could aid in finding abnormalities in gene expression.

In order to increase the accuracy of the gene level predictions it is likely that larger datasets need to be used in the application of MultistageOT. This poses challenges in data collection as well as computer performance, but the potential benefits are large enough to warrant such an investment.

A

Cell type descriptions

In this thesis, the datasets studied are samples of human bone marrow and the cells present are of types adhering to hematopoiesis [Lilljebjörn et al., 2025].

Due to the vast variety of cell types encountered in the datasets, we need to also specify that the Leukocytes are split into two distinct lineages:

- The Myeloid lineage (Monocytes)
- The Lymphocyte lineage (T-Cells, B-Cells, Dendritic Cells and Natural Killer Cells)

A short overview of the different cell types identified in the datasets are provided here. The descriptions were gathered from [Diehl et al., 2011; Chan Zuckerberg Initiative Foundation, 2024]. The "Acute Myeloid Leukemia Immature cells" entry was written using [Leukemia Research Foundation, 2024].

Hematopoietic Stem Cells (HSC). HSCs are multipotent cells which are capable of self-renewal as well as differentiating into all cell types which are encountered in the bloodstream.

Lymphoid-Primed Multipotent Progenitors (LMPP). LMPPs have begun to differentiate themselves towards the Lymphocyte lineage but still possess a multipotency to become any of the lymphocytes.

Granulocyte Monocyte Progenitors (GMP). GMPs are progenitor cells for the Myeloid lineage.

Erythroid Progenitor Cells. These progenitor cells have begun differentiating towards Erythroids, red blood cells which are responsible for carrying oxygen around the body.

Megakaryocyte Progenitor Cells. These progenitor cells have begun differentiating towards Megakaryocytes, which are responsible for the production of blood platelets.

Common Dendritic Progenitors. These progenitor cells have begun committing to the dendritic cell lineage.

Conventional dendritic cells (cDC). Conventional dendritic cells present antigens from ongoing infections in the body in order to activate naive T cells, initiating the immune response in the lymph node.

Plasmacytoid dendritic cells (pDC). These dendritic cells are primarily associated with early-stage immune responses.

B-Cell progenitors. These cells have begun committing to becoming B-cells.

B-Cells. A *Naive B cell* is a mature B cell which has not yet been activated in an immune system response. They primarily produce antibodies to aid in the immune response against bacterial and virus infections.

A *Memory B cell* is a mature B cell which was activated in an immune system response and remains in the lymph node to respond quicker against similar infections in the future.

The B in B-cell indicates the location in the body where they mature (Bone Marrow).

Plasmablasts. Plasmablasts are activated Naive and Memory B cells. They secrete antibodies to aid in an immune system response.

Natural Killer Cells (NK). NK-Cells are lymphocytes which provide rapid response to virus infected cells in the body. They come in two primary subtypes, based on proteins which are presented on their cellular surface: CD16⁺ or CD56⁺. The first subtype is primarily cytotoxic (damaging to other cells), aiming to destroy cells which exhibit typical virus infection behaviour. The second subtype fulfills a more regulatory role and interacts frequently with other parts of the immune system.

CD4/CD8 T-Cells. These are the typical T Cells. They primarily come in two subtypes, based on proteins presented on their surface: CD4⁺ and CD8⁺.

CD4⁺ cells play a large role in the activation of other Lymphocytes, whereas CD8⁺ cells play a cytotoxic role, being equipped to harm other malignant and malfunctioning cells.

Naive variants of these cells are yet to be activated in the event of an infection. Memory variants have experienced an infection and remain after the event in order to provide a quick response in case of a similar infection.

In the case that a CD8⁺ T-Cell has been activated, it is called a CD8⁺ effector cell while it deals with the infection.

The T in T-Cells indicates the location in the body where they mature (Thymus).

Regulatory T-Cells (Treg). Treg cells prevent excessive immune responses by limiting the proliferation and activation of other T cells.

$\gamma\delta$ T-Cell. These cells are a subtype in the T-Cell lineage.

Mucosal Invariant T-Cell (MAIT). MAIT cells are T-Cells which interact primarily with microbes.

CD14/CD16 Monocytes. Monocytes devour harmful pathogens in the body. They come in two subtypes, based on proteins which are presented on their cellular surface: CD14⁺ and CD16⁺. CD14⁺ are the classical monocytes, primarily navigating to the site of infection and devouring pathogens and debris.

CD16⁺ are non-classical monocytes which also migrate to the site of infection but appear play a bigger role in inflammatory diseases.

Acute Myleoid Leukemia Immature cells. These cells are abnormal versions of the Myleoid cells, which do not fully develop into functioning white blood cells yet are produced in enormous numbers. They crowd the bloodstream, preventing healthy blood cells from fulfilling their duties.

B

Gene expression heatmaps

Here, we plot the gene expressions for the different marker genes across all considered datasets. We also provide the boxplots of the marker gene estimates with outliers included.

B.1 NBM4-MNC

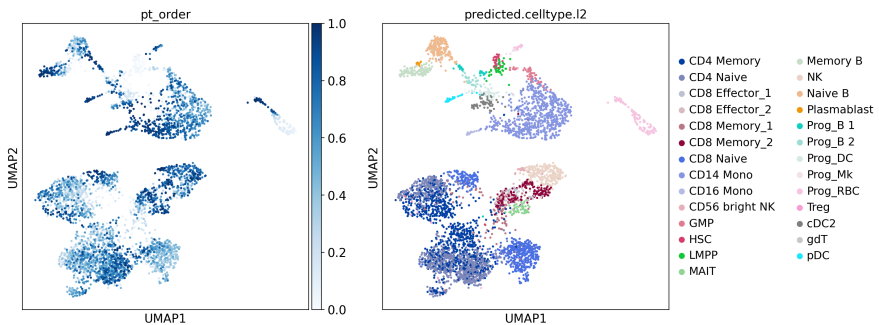


Figure B.1 Pseudotime ordering and l2-annotations for NBM4-MNC.

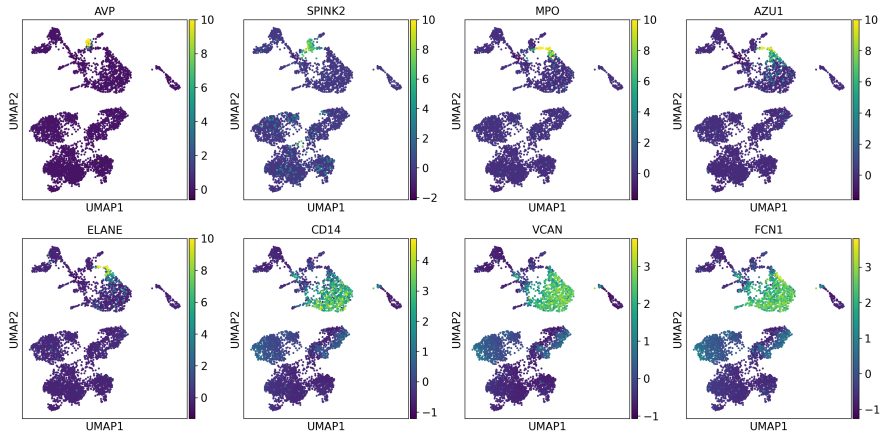


Figure B.2 CD14⁺ Monocyte marker gene expression for NBM4-MNC.

B.2 NBM8-MNC

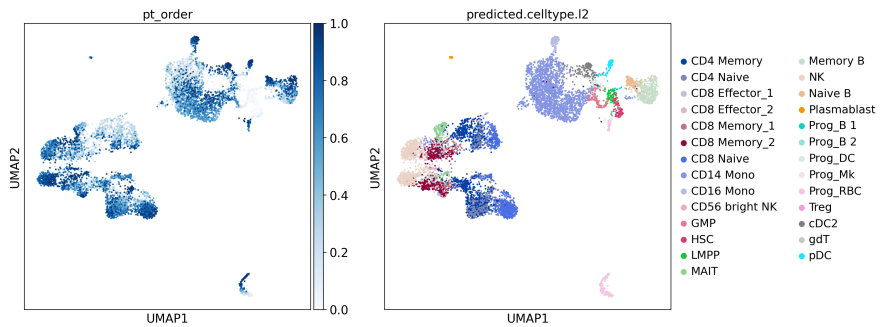


Figure B.3 Pseudotime ordering and l2-annotations for NBM8-MNC.

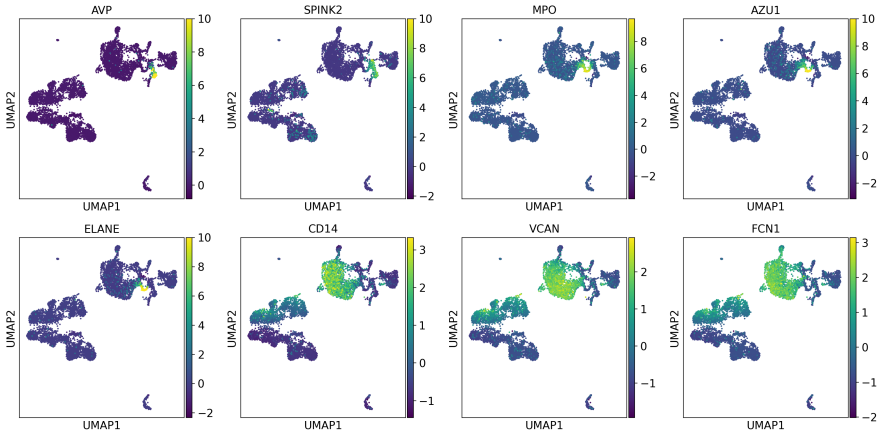


Figure B.4 CD14⁺ Monocyte marker gene expression for NBM8-MNC.

B.3 NBM10-MNC

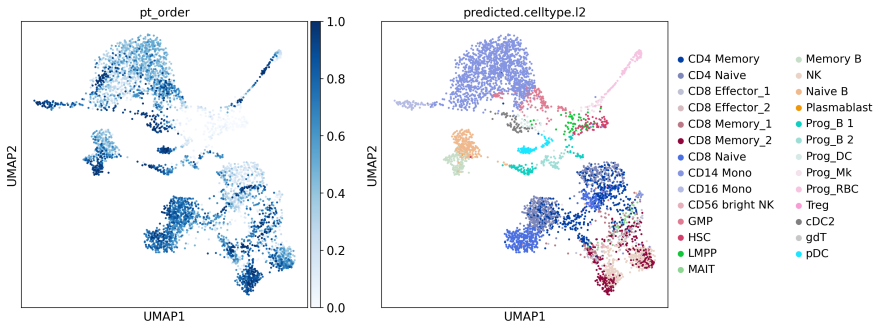


Figure B.5 Pseudotime ordering and l2-annotations for NBM10-MNC.

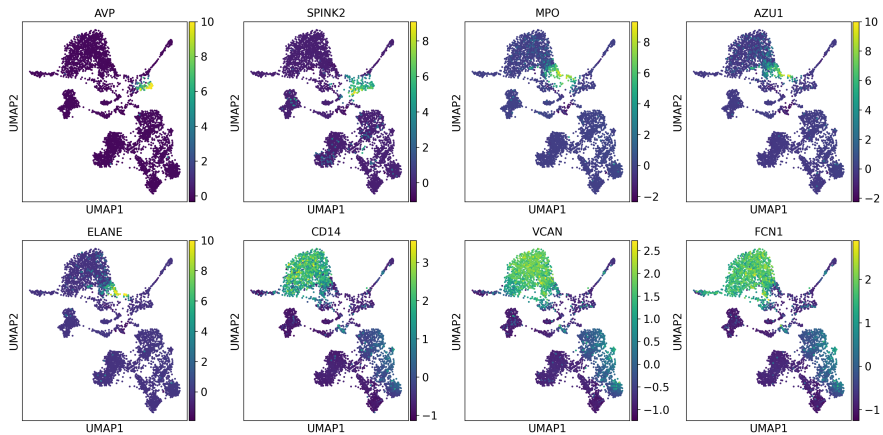


Figure B.6 CD14⁺ Monocyte marker gene expression for NBM10-MNC.

B.4 NBM8-CD34 + NBM7-MNC + NBM4-MNC

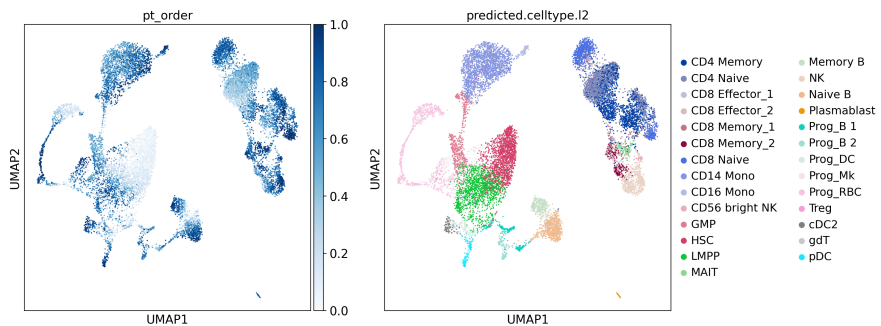


Figure B.7 Pseudotime ordering and l2-annotations for NBM8-CD34 + NBM7-MNC + NBM4-MNC.

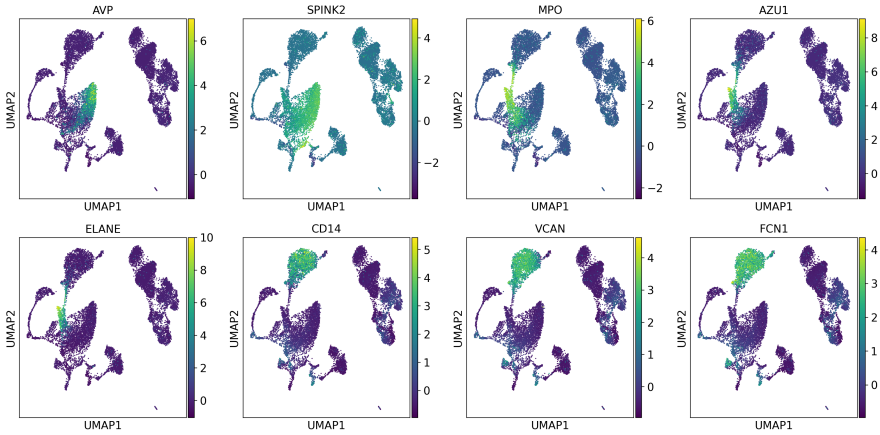


Figure B.8 CD14⁺ Monocyte marker gene expression for NBM8-CD34 + NBM7-MNC + NBM4-MNC.

B.5 AML155

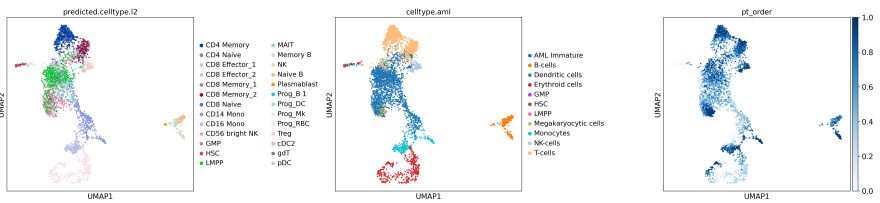


Figure B.9 Pseudotime ordering and l2-annotations for AML155.

Appendix B. Gene expression heatmaps

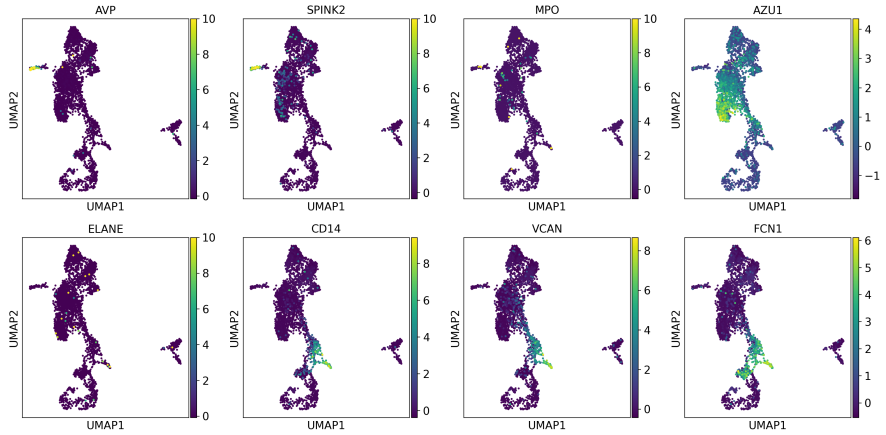


Figure B.10 CD14⁺ Monocyte marker gene expression for AML155.

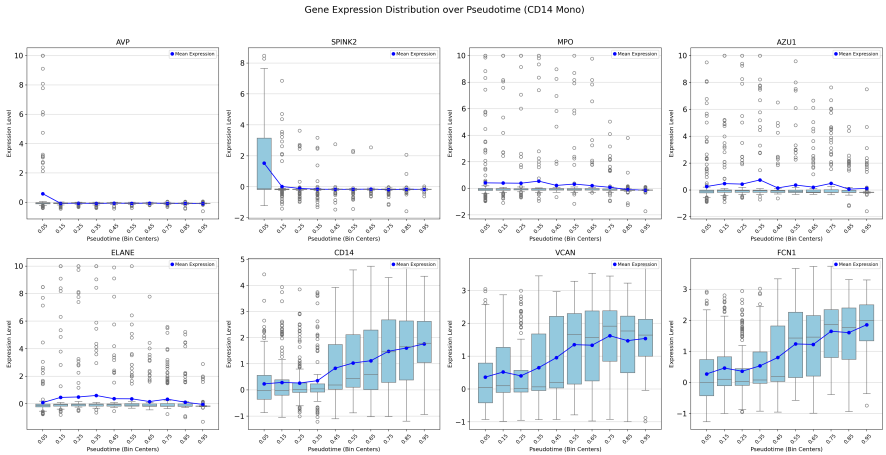


Figure B.11 Boxplot visualization for each marker gene estimate for the dataset NBM4-MNC.

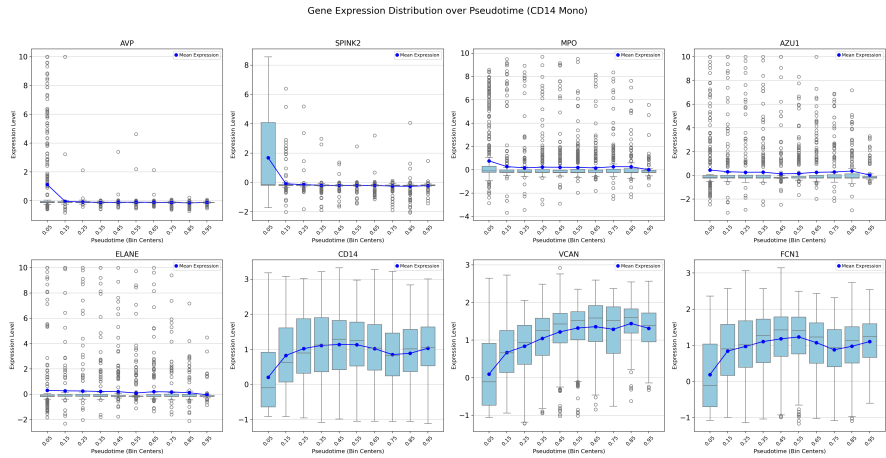


Figure B.12 Boxplot visualization for each marker gene estimate for the dataset NBM8-MNC.

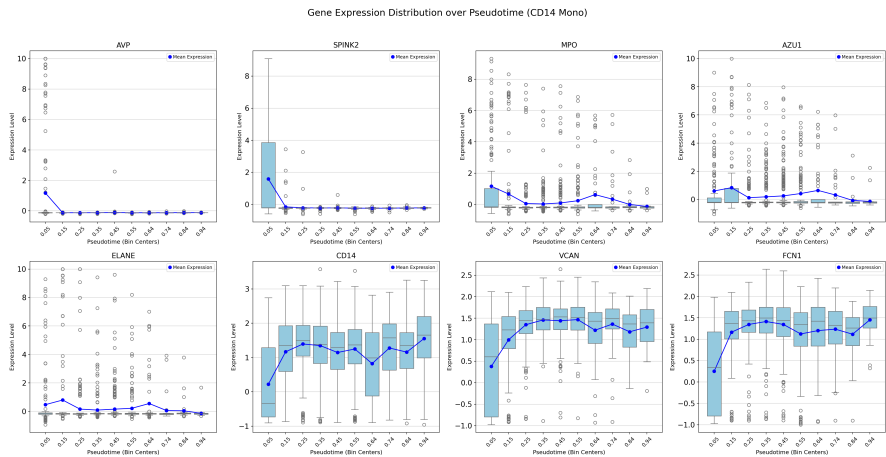


Figure B.13 Boxplot visualization for each marker gene estimate for the dataset NBM10-MNC.

Appendix B. Gene expression heatmaps

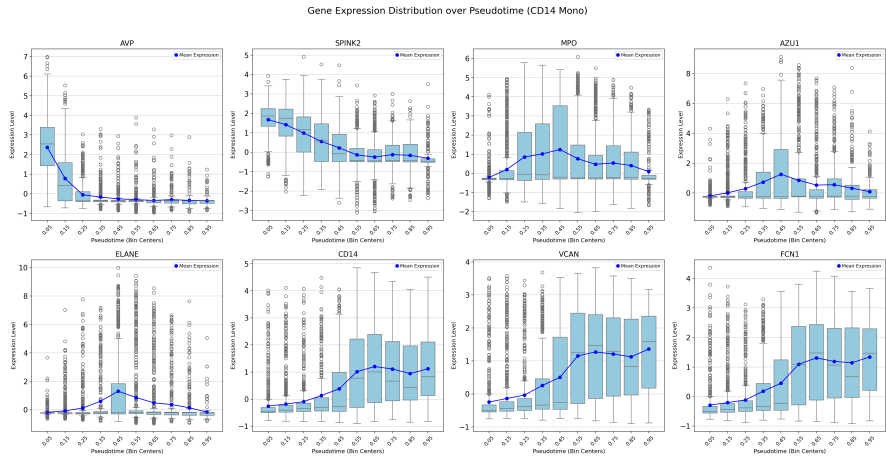


Figure B.14 Boxplot visualization for each marker gene estimate for the dataset NBM8-CD34+NBW7-MNC+NBW4-MNC.

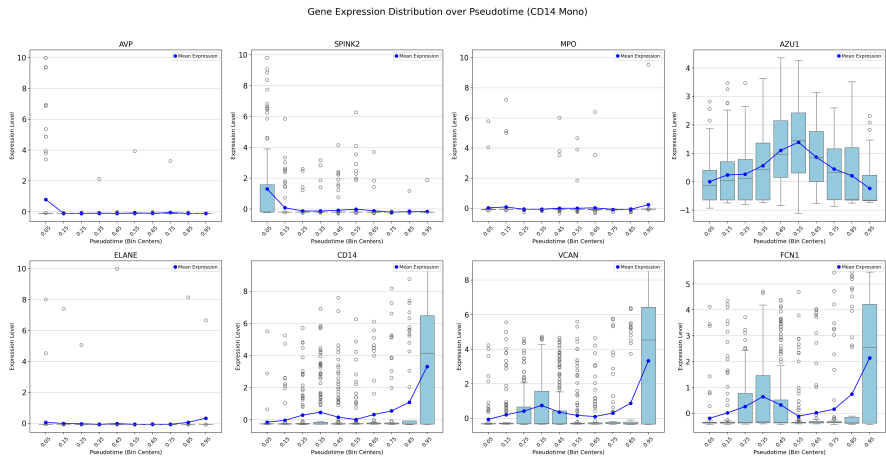


Figure B.15 Boxplot visualization for each marker gene estimate for the dataset AML155.

C

Plots of gene expressions over pseudotime for gene extraction methods

Here we detail the gene lists for CD14⁺ Monocytes as well as B-Cells, NK-Cells and pDCs extracted as described in section 3.5.2. The full list of extracted genes are available on the <https://github.com/vilhelmconradi-prog/MSOTonNBM>.

Below we present plots of the gene list mentioned in section 3.5 for the lineages subsets in Figure 4.2, Figure 4.4 and Figure 4.6. For reference, the percentage composition is presented in Figure 4.12. Here, the size of the gene lists in the plots are kept small to avoid cluttered plots and thus, only the highest scored genes for each method are presented..

Largest change in gene expression over all timesteps. These results are presented in Figures C.1 and C.2.

Appendix C. Plots of gene expressions over pseudotime for gene extraction methods

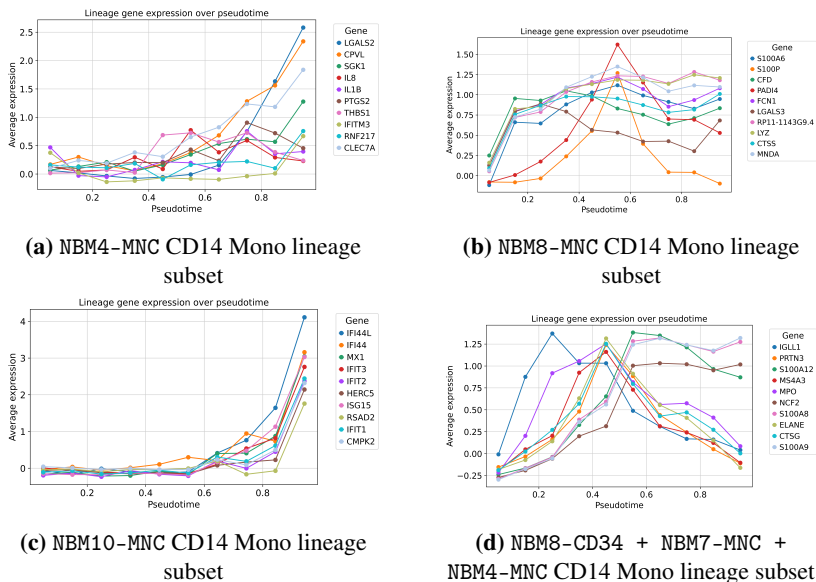


Figure C.1 Largest increase in gene expression across all time bins

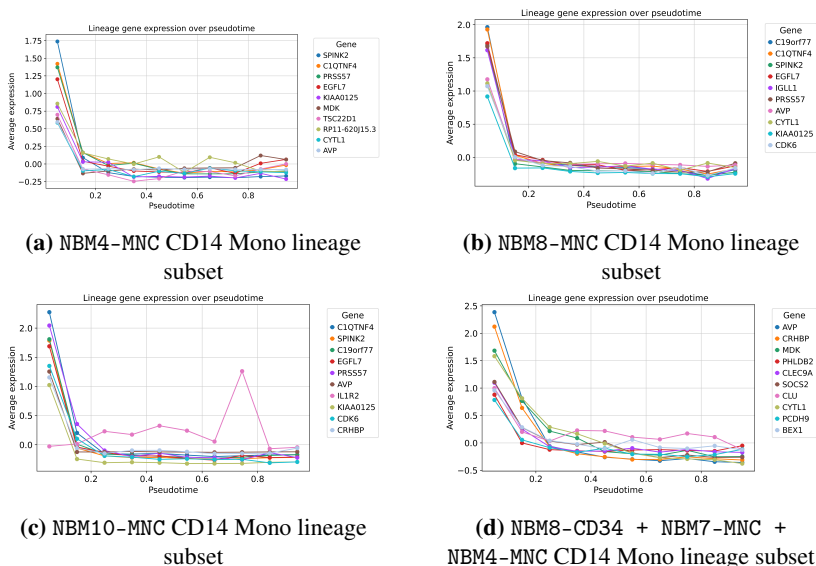


Figure C.2 Largest decrease in gene expression across all time bins

Highest gene expression at each time step. These results are presented in Figure C.3.

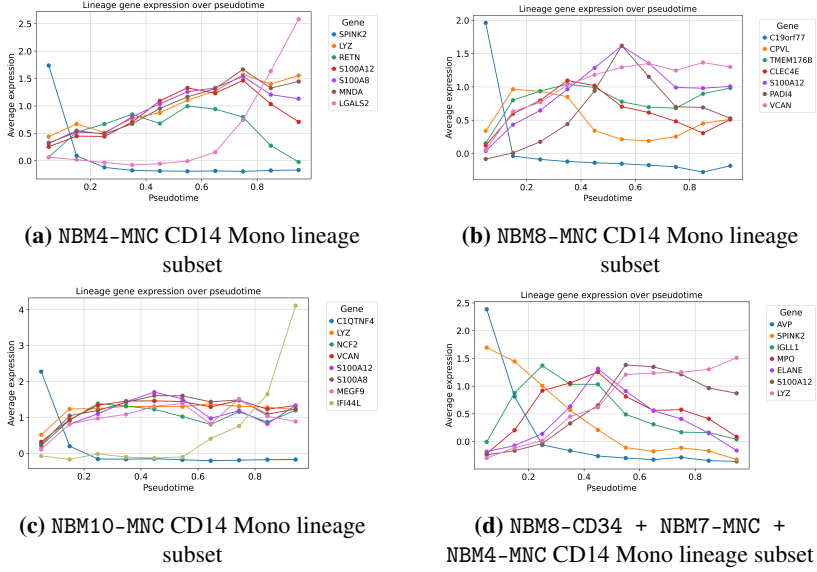


Figure C.3 Highest gene expression for each bin

Highest absolute change over pseudotime. These results are presented in Figure C.4.

Appendix C. Plots of gene expressions over pseudotime for gene extraction methods

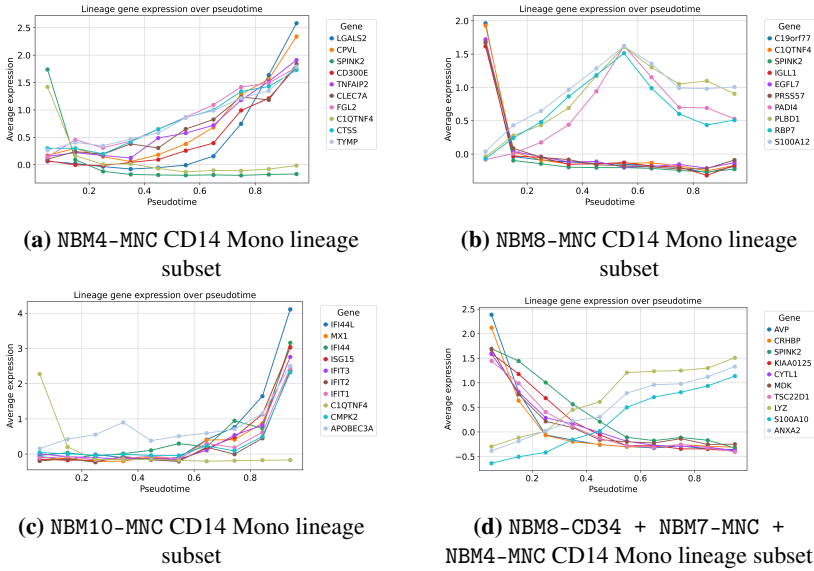


Figure C.4 Highest absolute change over pseudotime

Steepest change between each timesteps. These results are presented in Figures C.5 and C.6.

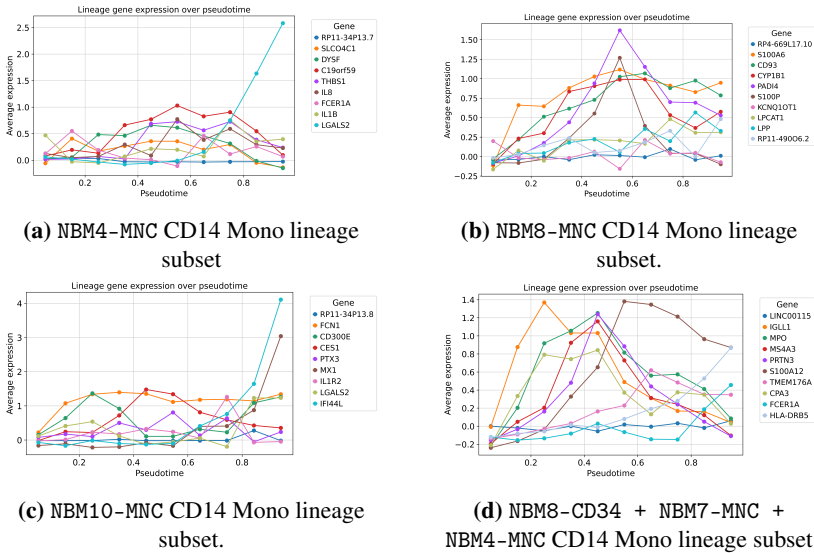
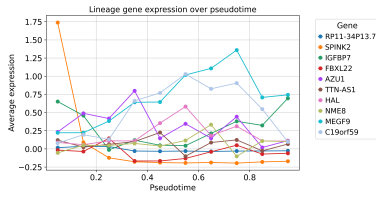
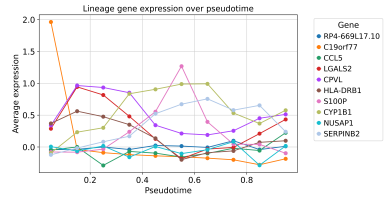


Figure C.5 Steepest jump between each timestep.

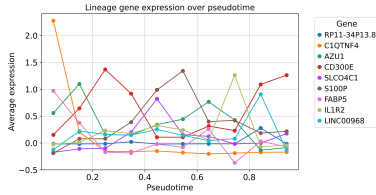
Appendix C. Plots of gene expressions over pseudotime for gene extraction methods



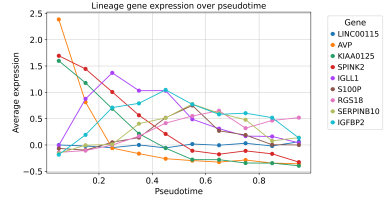
(a) NBM4-MNC CD14 Mono lineage subset



(b) NBM8-MNC CD14 Mono lineage subset.



(c) NBM10-MNC CD14 Mono lineage subset.



(d) NBM8-CD34 + NBM7-MNC + NBM4-MNC CD14 Mono lineage subset

Figure C.6 Steepest drop between each timesteps

Lund University Department of Automatic Control Box 118 SE-221 00 Lund Sweden		<i>Document name</i> MASTER'S THESIS	
		<i>Date of issue</i> June 2026	
		<i>Document Number</i> TFRT-6318	
<i>Author(s)</i> Vilhelm Conradi Hugo Nelsson		<i>Supervisor</i> Karl Rydén, Dept. of Automatic Control, Lund University, Sweden Anders Rantzer, Dept. of Automatic Control, Lund University, Sweden Magnus Fontes, Dept. of Automatic Control, Lund University, Sweden (examiner)	
<i>Title and subtitle</i> Multistage Optimal Transport evaluated on scRNA-seq Human Bone Marrow data			
<i>Abstract</i> With novel techniques like single-cell RNA sequencing (scRNA-seq) we can look inside single cells in great detail and measure their individual gene activity by counting the mRNA molecules of the cells. In order to model cell differentiation one would naturally turn to longitudinal data of single cells. However, snapshot datasets is the basis of single cell studies due to the way the datasets are collected. The advantage with scRNA-seq is that samples are very informative and in healthy bone marrow, often contain a good representation of cells in different stages of the differentiation process. We can therefore assume that we have sufficient information to create a model for the differentiation process using the snapshot data alongside optimal transport frameworks. Building on the assumption that cell states similar to each other have similar gene expression, and that this gene expression can provide information about the order of these cells in a differentiation process, we use a newly proposed optimal transport algorithm to compute probable transitions between cell states as well as a pseudotime ordering of the cells. We use these transitions to extract subsets of cells which represent differentiation trajectories through human bone marrow samples, and order the subset cells in pseudotime. We then estimate expressions of some common marker genes across the subset's pseudotime in order to verify that the model captures expected biological behaviour. We also apply the method to a sample of cells connected to Acute Myeloid Leukemia (AML) to observe if the changed biological content causes these marker genes to be expressed differently in the sample. The ambition is that this method can aid in further study of the development of AML.			
<i>Keywords</i> Optimal Transport, scRNA-seq, Hematopoiesis, Markov Chains, Scanpy, Cell Differentiation, Optimization, Pseudotime, Acute Myeloid Leukemia			
<i>Classification system and/or index terms (if any)</i>			
<i>Supplementary bibliographical information</i>			
<i>ISSN and key title</i> 0280-5316			<i>ISBN</i>
<i>Language</i> English	<i>Number of pages</i> 1-89	<i>Recipient's notes</i>	
<i>Security classification</i>			

<http://www.control.lth.se/publications/>