IMPERIAL

IMPERIAL COLLEGE LONDON

DEPARTMENT OF MATHEMATICS

SECOND-YEAR GROUP RESEARCH PROJECT

Data-Driven Modelling of Cell Differentiation Trajectories in the Intestine

Author:
SOHAM SUD (CID: 02403389)
JUNYOU LI(CID: 02401343)
BEN THIMBLEBY (CID: 02373998)
GAO YIFAN (CID: 02214998)
SAHIL RAI (CID: 02390048)

Supervisor(s):
Omer Karin

July 20, 2025

Abstract

The study of intestinal cell differentiation is crucial for understanding fundamental biological processes and greatly facilitates the development of clinical and regenerative medicine. In this project, we performed data-driven analysis on single-cell RNA sequencing (scRNA-seq) data to model intestinal cell differentiation. We primarily used the Python package Scanpy [1] for writing the simulation code. Initially, we processed data through filtering and standardisation, and applied Leiden Clustering to identify cell types by matching reference with marker genes and performing enrichment analysis. To infer differentiation progression, we explored various trajectory inference methods including Diffusion Pseudo-time (DPT), CellRank and Palantir - each providing a unique perspective on differentiation trajectories. Finally, we examined the impact of cell-cell communication and interactions on cell differentiation using ligand-receptor interaction analysis. We combined these methods with visualisations, such as gene expression heatmaps and line plots over pseudotime, to present a comprehensive view of intestinal cell differentiation.

Contents

1 Introduction

Understanding how cells differentiate, communicate, and form organised structures are some of the main aims in biology. In the past, a technique called Bulk RNA sequencing (bulk RNA-seq) provided an average gene expression profile across a large number of cells. However, a recently developed technique called single-cell RNA sequencing (scRNA-seq) is able to provide gene expression profiles for individual cells with a high resolution. This development in RNA sequencing has offered many new insights into biological processes. We are now able to better capture cellular heterogeneity and find possible distinctions within cell types, allowing us to discover new cell types and distinguish cells that appear very similar. We can also now track how cells differentiate into different types and trace the origin of each cell fate. Another insight is in cell-cell communication within tissues which can be visualised and allows us to see which cell types are interacting with which other cell types.

In this study, we applied the scRNA-seq workflow on mus musculus (mouse) intestine data, which involves pre-processing and downstream analysis, to make biological inferences and compare with accepted biology. We used data that had already been normalised, and so we mainly continued with the downstream analysis consisting of well-known and now standard methods such as clustering, marker identification, cluster annotation, trajectory inference, gene dynamics, and cell-cell interaction graphs. We explored different clustering and trajectory inference methods, and also explored different scoring methods for cell-cell interactions following the clustering and trajectory methods.

Clustering methods group cells into clusters with similar gene profiles and can provide insight into cell types and transitional states, providing a greater depth of understanding into the cell differentiation cycle. Marker identification then allows us to assign cluster labels of known cell types. Trajectory inference methods produce pseudotime graphs that reconstruct the progression through the cell states throughout differentiation. The methods we explored enable reconstructions of branching trajectories in the differentiation pathways and assign cells a pseudotime reflecting its position along a continuum of differentiation. Cells further along a pseudotime are more committed to a cell fate. Cell-cell communication is known to be heavily involved in coordinating cell self-organisation and cell fate decisions, so insights into cell interactions are essential in understanding the biological processes. We proceeded to explore these interactions via the use of tools such as LIANA.

The focus of this study is an application of the full scRNA-seq workflow, including downstream analysis, to obtain biological inferences for cell pseudotime trajectories and interactions to, in turn, confirm the currently known mechanisms in the small intestine crypt-villi.

2 Biological Background

The small intestine is made up of crypts and villi. Its functions are governed by the many different cell types that line its surface. The main function of the small intestine is to absorb nutrients, which explains why the most common cell fate is the enterocyte cell type, an absorptive cell which can be seen in Figure 1. However, absorption alone is not sufficient for the proper functioning of the small intestine. Secretory cells, which release substances such as mucus and enzymes that help with digestion, are also essential. The secretory lineage includes: Goblet cells, Enteroendocrine cells, Tuft cells, and Paneth cells. Therefore, while most cells differentiate into enterocytes, a balancing of absorptive and secretory cells is required.

Ligand-receptor pairs

Cells interact and coordinate their behaviour through ligand-receptor pairs, which are central to cell-cell interaction. A ligand-receptor pair consists of a signalling molecule, called a ligand, produced by one cell, which binds to a receptor protein on the surface of a receiving cell. This causes the receiving cell to respond and change its behaviour, such as differentiating into a new cell type, migrating, or altering gene expression. Ligand–receptor pairs are fundamental to regulating spatial organisation and cell fate decisions within the intestinal crypt–villus structure, and are thus key to understanding coordinated cell differentiation. Ligands can either be secreted and diffuse through space for longer range interactions or they can be membrane-bound, fixed on the cell surface, requiring direct contact between cells, only allowing for short range contact interactions. These ligand-receptor interactions can be classified into three principal modes of signalling:

- Paracrine signalling: ligands are released into its surroundings and travel a short distance signalling to nearby cells.
- Autocrine signalling: ligands are released into its surroundings and then the sender cell itself responds to it. So this is cells signalling to themselves.
- Juxtacrine signalling: ligands are attached to the surface of the sender cell and can only signal to neighbouring cells directly in contact with it [2].

Cell differentiation in the small intestine

Cells in the small intestine begin as intestinal stem cells (ISCs), which reside at the base of the crypt and are capable of self-renewal and differentiation. ISCs divide and produce transit-amplifying (TA) cells, which are rapidly dividing progenitor cells that sit right above the base of the crypt immediately above the ISCs. These TA cells are an intermediate cell state and further differentiate into either absorptive or secretory cells. This cell fate decision is mainly governed by the Notch signalling pathway. TA cells expressing high levels of Delta ligands activate Notch signalling (via the Notch receptors) in adjacent cells (as Delta ligands are membrane-bound). This signal lead the lateral inhibition of the secretory cell fate for neighbouring cells and allows themselves to commit to the secretory cell fate. Notch signals upregulate Hes1, a transcriptional repressor, which suppresses Atoh1, a pro-secretory transcription factor. A suppressed Atoh1 results in an absorptive cell fate by the prevention of the secretory cell fate and so these neighbouring cells commit to becoming enterocytes. The absorptive fate is considered to be the default cell fate [3].

Most of the differentiated cells migrate up the crypt-villus axis. However, Paneth cells migrate downwards back to the base of the crypt among the stem cells. This direction of migration is

governed by EphB-EphrinB signalling. Paneth cells express EphB receptors, while cells higher up in the crypt-villus axis express EphrinB ligands. This ligand-receptor pair triggers a repulsive signal and causes the Paneth cells to migrate away from regions with higher levels of EphrinB leading to the downwards migration of Paneth cells [4]. In addition to the Notch signalling, where a low Notch activity is required, Wnt signalling is crucial for the differentiation and maturation of Paneth cells. Paneth cells themselves secrete Wnt signals and are essential for stem cell maintenance in the base of the crypt [5].

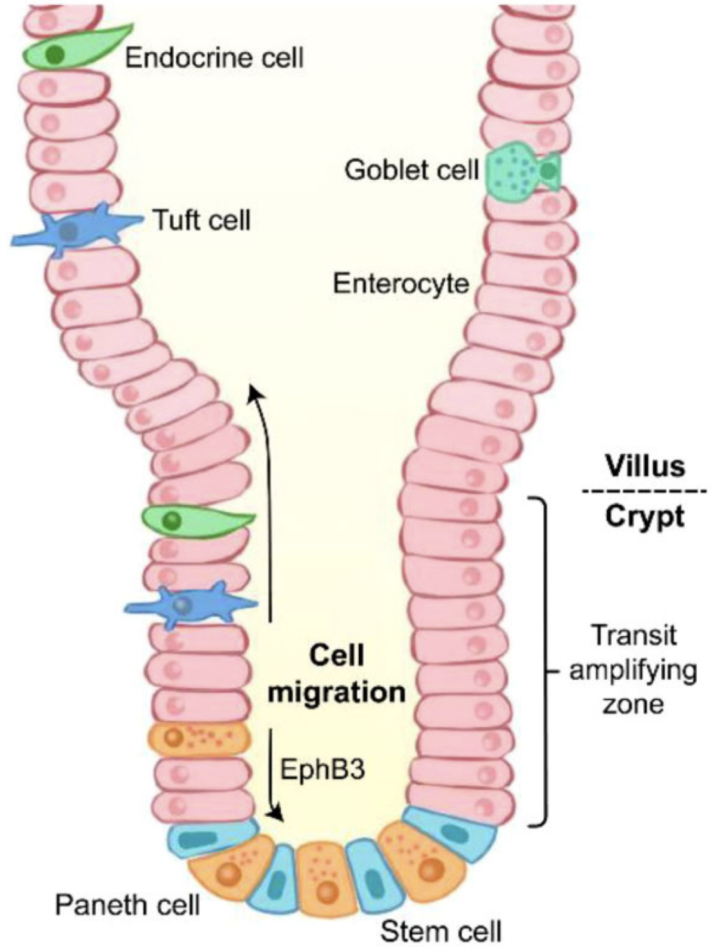


Figure 1 Migration of Paneth cells. Unlike other secretory cells, Paneth cells are located at the base of crypts. After differentiation, Paneth cells migrate downward to the crypt bottom, and they are intercalated between stem cells. The migration of Paneth cells is mediated by EphB3. From Figure 1 of Cui *et al.* [6].

3 Clustering Methods

After preprocessing the data, as in normalisation and dimensional reduction (PCA), we can apply clustering algorithms to group similar cells and these clusters can be interpreted as cell types.

Background Concept: kNN Undirected Graph and Modularity

3.0.1 kNN Undirected Graph

To apply community detection (Clustering) algorithms such as Louvain or Leiden, we first need to construct a cell-cell graph. This is typically done via a k-nearest neighbour (kNN) graph based on PCA-reduced gene expression data.

Step 1: Dimensionality Reduction Let $X \in \mathbb{R}^{n \times d}$ be the gene expression matrix, where n is the number of cells and d the number of genes. We first reduce the dimensionality to $k \ll d$ using PCA:

$$X_{\text{PCA}} = (X - \bar{X})W_k$$

where $W_k \in \mathbb{R}^{d \times k}$ is the matrix whose columns are the top k principal components, which are the eigenvectors corresponding to the largest eigenvalues of the covariance matrix of X. Each column of W_k represents a direction of maximum variance in the gene expression data. Therefore, by multiplying the centered data $(X - \bar{X})$ with W_k, we project the high-dimensional gene expression matrix $X \in \mathbb{R}^{n \times d}$ onto a lower-dimensional subspace of dimension k, capturing the most informative patterns in the data.

Step 2: Neighbourhood Construction For each cell i, we compute the Euclidean distance in PCA space and identify the k nearest neighbours:

$$\mathcal{N}_k(i) = \{j \mid j \text{ is among the } k \text{ closest cells to cell } i\}$$

This forms a directed kNN graph where an edge $i \to j$ exists if $j \in \mathcal{N}_k(i)$.

Step 3: Edge Weighting via Jaccard Similarity We convert the directed kNN graph into a weighted graph using Jaccard similarity. For any pair of nodes i and j, we define the edge weight as:

$$w_{ij} = \frac{|\mathcal{N}_k(i) \cap \mathcal{N}_k(j)|}{|\mathcal{N}_k(i) \cup \mathcal{N}_k(j)|}$$

This reflects the fraction of shared neighbors and yields higher weights for more structurally similar cells.

Step 4: Symmetrization Although the Jaccard similarity itself is symmetric ($w_{ij} = w_{ji}$), computing it only when $j \in \mathcal{N}_k(i)$ (i.e., in one direction), the resulting graph may remain asymmetric. To construct a fully symmetric adjacency matrix A for Louvain or Leiden clustering, we apply one of the following symmetrisation strategies:

- Keep only mutual kNN edges: $i \in \mathcal{N}_k(j)$ and $j \in \mathcal{N}_k(i)$
- Average weights: $w_{ij} = \frac{1}{2}(w_{ij}^{\text{dir}} + w_{ji}^{\text{dir}})$

- Take the maximum: $w_{ij} = \max(w_{ij}^{\text{dir}}, w_{ji}^{\text{dir}})$

This results in a symmetric, weighted graph that can be used for community detection.

3.0.2 Modularity

We then define **modularity**, a measure that depicts whether a network is successfully partitioned into distinct communities, within each of which the nodes are densely connected. Given a weighted graph $G = (V, E, w)$, the modularity Q of a partition is defined as:

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)$$

where:

- A_{ij}: the weighted adjacency matrix (i.e., edge weight between node i and node j)
- $k_i = \sum_j A_{ij}$: the degree (total edge weight) of node i
- $m = \frac{1}{2} \sum_{i,j} A_{ij}$: the total weight of all edges in the graph
- c_i: the community to which node i is assigned
- $\delta(c_i, c_j)$: an indicator function that equals 1 if $c_i = c_j$, and 0 otherwise

The second term $\frac{k_i k_j}{2m}$ represents the expected edge weight between nodes i and j at random in a configuration model. The entire expression then calculates the difference between actual and expected edges within each community.

3.1 Louvain Clustering

The Louvain algorithm is an inspirational, two-stage iterative optimization, aiming to maximize modularity. It is widely implemented in community detection and cell clustering in single cell data analysis [7].

Initiation Consider each node in the network as an individual community.

STEP 1: Local Moving Phase Consider the contribution of each community to the modularity and write:

$$Q = \sum_{\text{clusters}C} Q_C, \quad where \quad Q_C = \frac{1}{2m} \sum_{i,j \in C} \left(A_{ij} - \frac{k_i k_j}{2m} \right) = \frac{1}{2m} \left[\sum_{i,j \in C} A_{ij} - \frac{1}{2m} \left(\sum_{i \in C} k_i \right)^2 \right]$$

We now introduce compact notation to simplify the analysis:

- Σ_{in}: the total weight of edges within the community c' : $\Sigma_{\text{in}} = \sum_{i,j \in c'} A_{ij}$
- Σ_{tot}: the total degree of all nodes in c' : $\Sigma_{\text{tot}} = \sum_{i \in c'} k_i$

- k_i: the degree of node i
- $k_{i,\text{in}}$: the total weight of edges between node i and nodes in c'

Let node i be moved from its original community c to a neighbouring community c'. The change in modularity ΔQ can be computed by considering the modularity contributions before and after the move:

$$\Delta Q = Q_{c'}^{\text{after}} - Q_{c'}^{\text{before}} + Q_c^{\text{after}} - Q_c^{\text{before}}$$

The four terms are computed as follows:

$$\begin{aligned}
&(1) \quad Q_{c'}^{\text{after}} = \frac{1}{2m}\left[\Sigma_{\text{in}} + 2k_{i,\text{in}} - \frac{(\Sigma_{\text{tot}} + k_i)^2}{2m}\right] \\
&(2) \quad Q_{c'}^{\text{before}} = \frac{1}{2m}\left[\Sigma_{\text{in}} - \frac{\Sigma_{\text{tot}}^2}{2m}\right] \\
&(3) \quad Q_c^{\text{before}} = \frac{1}{2m}\left[0 - \frac{k_i^2}{2m}\right] = -\frac{k_i^2}{(2m)^2} \\
&(4) \quad Q_c^{\text{after}} = 0 \quad \text{(assuming the original community becomes empty)}
\end{aligned}$$

Substituting all terms into the ΔQ formula:

$$\Delta Q = \frac{1}{2m}\left[(\Sigma_{\text{in}} + 2k_{i,\text{in}}) - \frac{(\Sigma_{\text{tot}} + k_i)^2}{2m}\right] - \frac{1}{2m}\left[\Sigma_{\text{in}} - \frac{\Sigma_{\text{tot}}^2}{2m}\right] + \frac{k_i^2}{(2m)^2}$$

Now, combine like terms and rearrange, the simplified modularity gain expression becomes:

$$\Delta Q = \left[\frac{\Sigma_{\text{in}} + 2k_{i,\text{in}}}{2m} - \left(\frac{\Sigma_{\text{tot}} + k_i}{2m}\right)^2\right] - \left[\frac{\Sigma_{\text{in}}}{2m} - \left(\frac{\Sigma_{\text{tot}}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right]$$

STEP 2: Aggregation Phase Once an increase in modularity is no longer possible, we finish phase 1 and combine the nodes in the same community as a super node, forming a reduced new network. Now the edges between 2 super nodes have weights equal to the sum of edge weights between all pairs of nodes that lie in these communities in the original network. Moreover, self-loops are formed at each super node, with weights being the sum of edge weights of all internal nodes in the original network. Now repeat phase 1 and iteratively implement the procedure until any movement of nodes into the neighbouring community does not bring any increase in overall modularity.

Remark Even though after the first iteration, the expressions in each individual term of ΔQ change due to self loop A_{ii} of the super node i, the expression for ΔQ remains invariant. This is because:

$$\begin{aligned}
\Delta Q &= \frac{1}{2m}\left[\Sigma_{\text{in}} + 2k_{i,\text{in}} + A_{ii} - \frac{(\Sigma_{\text{tot}} + k_i)^2}{2m}\right] - \frac{1}{2m}\left[\Sigma_{\text{in}} - \frac{\Sigma_{\text{tot}}^2}{2m}\right] - \frac{1}{2m}\left[A_{ii} - \frac{k_i^2}{2m}\right] \\
&= \left[\frac{\Sigma_{\text{in}} + 2k_{i,\text{in}}}{2m} - \left(\frac{\Sigma_{\text{tot}} + k_i}{2m}\right)^2\right] - \left[\frac{\Sigma_{\text{in}}}{2m} - \left(\frac{\Sigma_{\text{tot}}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right]
\end{aligned}$$

3.2 Leiden Clustering

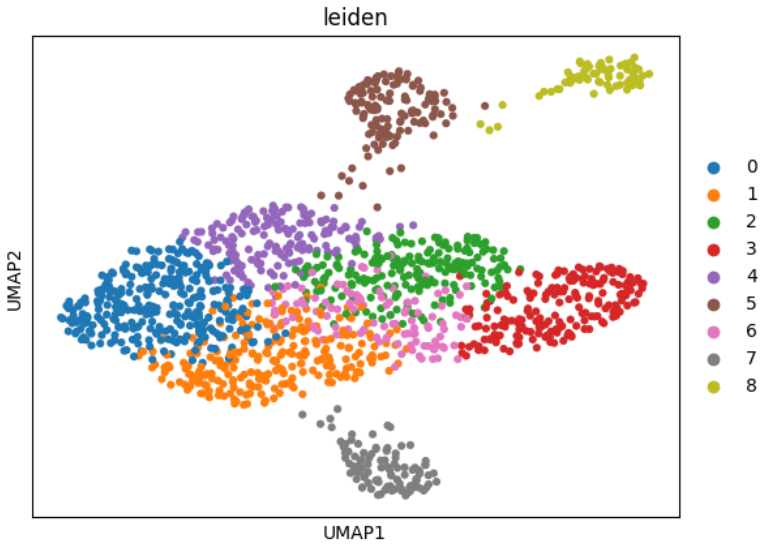
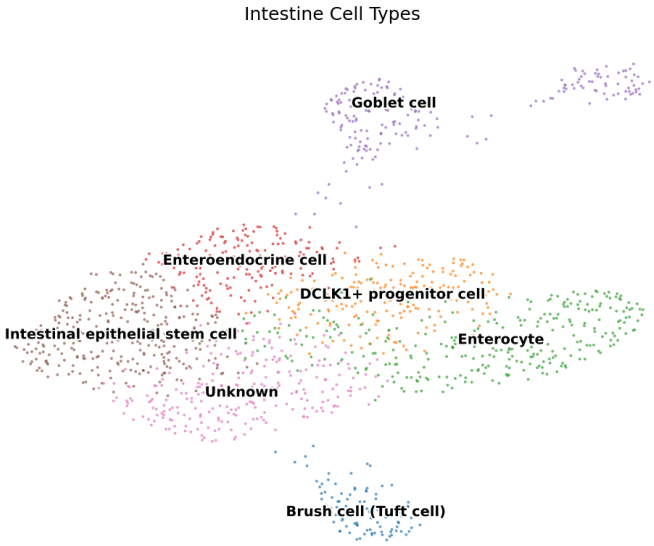


Figure 2 Leiden Clustering of cells based on gene expression. Each colour represents a unique cluster, later annotated with known cell types.

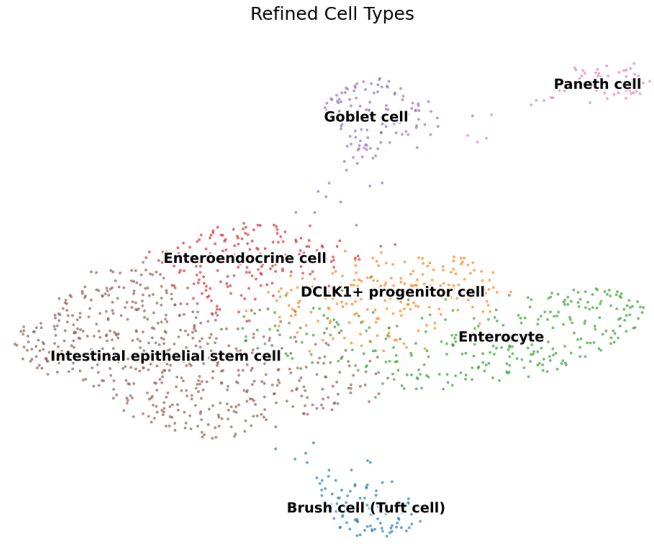
Leiden Clustering essentially shares the same algorithm as Louvain Clustering, aiming to maximize the modularity of a graph. However, a major advantage is that Leiden avoids possible disconnected components within communities by refinement after moving the nodes. Unlike the greedy strategy in Louvain Clustering, Leiden adopts a delayed strategy and it records the potential moves for each node, then performs a connectivity-aware merging process at once (This is called **Fast Local Merge**). Only groups of nodes that can be added to a community without breaking its internal connectivity are actually moved.

Refinement For each community, the algorithm extracts its induced subgraph — a subgraph containing only the nodes within the community and all the edges between them that exist in the original graph. It then checks whether this subgraph is connected.

- If the induced subgraph is connected, the community is kept as is.
- If it is not connected, the community is split into its connected components — each of which becomes a separate community.



(a)



(b)

Figure 3 Comparison of initial and refined cell type annotations in UMAP space

In Figure 2 we observe that the Leiden clustering method grouped the cells into different clusters, with little overlap at the boundaries, but still provides us information on similarities between clusters. Using a reference set of cell marker genes at CellMarker we are able to count the overlaps and annotate these clusters by the cell types. Moreover, Enrichment Analysis is adopted to refine the cluster labels by creating a dictionary and matching the Functional Terms to cell types.

From Figure 3b, we can see that the stem cell cluster is closer to the enteroendocrine and enterocyte clusters than it is to the TA cell cluster, suggesting that stem cells may be more transcriptionally similar to these two cell types than they are to the TA cells. This goes slightly against intuition, since TA cells are not fully specialised, but the other two types mentioned are. On the other hand, the other cell types (Tuft, Goblet and Paneth) are separate from the main block of clusters, suggesting they are more transcriptionally distinct from the other types. In particular, the Paneth cells are the furthest away, indicating the least transcriptional similarity to the stem cells.

4 Trajectory Methods

Trajectory inference methods were developed to produce a pseudotime to help us understand cell differentiation trajectories over time. Pseudotime is an ordering of cells in a way that produce an effective timeline based on the gene expression of a cell and can capture the dynamics of the gene expression throughout the pseudotime. These methods achieve this by mapping these cells into a lower-dimensional space that captures their key characteristics and then inferring a trajectory before finally assigning a pseudotime based on a calculated distance from a given root cell, giving that cell its pseudotime. In this section, we examine three methods: Diffusion Pseudo-time (DPT), CellRank and Palantir.

4.1 Diffusion Pseudo-time (DPT)

To capture the continuous process of cell differentiation, reflected by the variations in gene expression, we applied the DPT method which utilises a diffusion map to perform a non-linear dimensional reduction on high-dimensional single cell gene expression data. This embeds it into a lower-dimensional space while preserving the structure and relations of the higher-dimensional data. This method constructs a diffusion transition matrix representing the probability of transitioning from one cell to another. These probabilities are based on expression similarity and derived through a Markov process, which we define with the Gaussian Kernel assigning higher transition probabilities between cells with more similar gene expressions.

STEP 1: Project cells into principal component space Let $X \in \mathbb{R}^{n\times d}$ be the original gene expression matrix for n cells and d genes. To remove noise and irrelevant variations, we reduced the dimensions from d to a smaller number k by selecting highly variable genes and standardizing the data, we then projected the gene expression matrix onto the principal components W_k:

$$X_{\text{PCA}} = (X - \bar{X})W_k, \quad W_k \in \mathbb{R}^{d\times k}$$

STEP 2: Construct similarity matrix using a locally adaptive Gaussian Kernel For each pair of cells i, j, let $x_i, x_j \in \mathbb{R}^k$ be their PCA-reduced coordinates(i.e. i^{th} and j^{th} row of X_{PCA}). We define a local adaptive Gaussian kernel:

$$K_{ij} = \begin{cases} \exp\left(-\dfrac{\|x_i - x_j\|^2}{\sigma_i \sigma_j}\right), & j \in \mathcal{N}_k(i) \\ 0, & \text{otherwise} \end{cases}$$

where σ_i is the average distance from cell i to its k-nearest neighbours (local bandwidth), given by

$$\sigma_i := \frac{1}{k} \sum_{j' \in \mathcal{N}_k(i)} \|x_i - x_{j'}\|$$

This allows the kernel to adapt to varying cell densities and is more flexible. Notice $K_{ij} = 0$ when j is outside of the k nearest neighbours, and this will result in a sparse matrix K and hence greatly enhances efficiency in the implementation of the algorithm.

STEP 3: Normalize to obtain a diffusion transition matrix Higher similarity represents higher transition probability, so for each cell i, we normalize the similarities to its neighbours

$j \in \mathcal{N}_k(i)$ so they sum to 1. Then we can get the Markov transition matrix using the similarity matrix:

$$P_{ij} = \frac{K_{ij}}{\sum_{j' \in \mathcal{N}_k(i)} K_{ij'}}$$

Written more compactly in matrix form:

$$P = D^{-1}K, \quad \text{where} \quad D_{ii} = \sum_{j' \in \mathcal{N}_k(i)} K_{ij'}$$

STEP 4: Eigen-Decomposition of the normalized transition matrix To understand the long-term behaviour of a random walk on the graph of cells represented by P, we perform spectral decomposition:

$$P\psi_k = \lambda_k \psi_k, \quad \lambda_1 > \lambda_2 \geq \cdots \geq \lambda_m$$

where ψ_k are right eigenvectors of P and λ_k are corresponding eigenvalues. These are informative components to understand how cells transit across states over time:

- Each eigenvector represents a direction of variation in the diffusion space. Extracting the largest eigenvectors means extracting the main modes of variation, which help reflect the global cell differentiation pattern.
- The corresponding eigenvalues tell us how important each direction is. The closer λ_k is to 1, the slower it decays over time, and the more stable and informative the associated diffusion direction is.

Note that since P is a transition matrix, $\lambda_1 = 1$ is always the largest eigenvalue (the stationary distribution), and all other eigenvalues $\lambda_k \in [0, 1)$ decay with k, indicating less stable diffusion directions.

STEP 5: Compute diffusion coordinates and pseudo-time We map each cell i into a m-dimensional diffusion space using the top m eigenvectors and their corresponding eigenvalues:

$$\Phi_t(i) = (\lambda_1^t \psi_1(i), \lambda_2^t \psi_2(i), \ldots, \lambda_m^t \psi_m(i))$$

Here, t is a diffusion time parameter, and the resulting coordinates reflect progression along differentiation trajectories. Finally, diffusion pseudotime is computed by measuring the Euclidean distance squared between the diffusion coordinates of each cell and a predefined root cell r:

$$\text{DPT}(i) := \tau(i) = \|\Phi_t(i) - \Phi_t(r)\|^2 = \sum_{k=1}^{m} \lambda_k^{2t} (\psi_k(i) - \psi_k(r))^2$$

This provides a scalar ordering of cells along the inferred diffusion pseudo-time process.

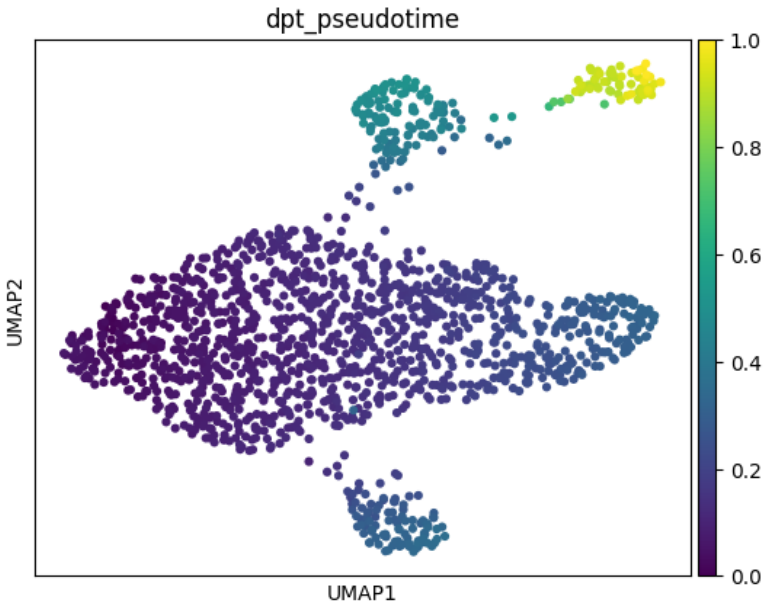


Figure 4 Visualisation of diffusion pseudotime computed using DPT

4.2 RNA Velocity Analysis

RNA velocity analysis uses the ratio of unspliced and spliced RNA counts to infer the direction and speed of changes in gene expression [8, 9]. For example, a high ratio of unspliced to spliced RNA indicates that the gene is being upregulated; a lot of RNA is being produced, but it is mostly not yet spliced, so we predict it is in an early stage. Conversely, a low ratio of unspliced to spliced RNA indicates that the gene is being downregulated, and hence is in a later stage due to more RNA having been spliced. The stages of this process can be expressed as

$$\text{DNA} \xrightarrow[\alpha]{\text{transcription}} \text{unspliced mRNA} \xrightarrow[\beta]{\text{splicing}} \text{spliced mRNA} \xrightarrow[\gamma]{\text{degradation}} \text{degraded mRNA}$$

The dynamics of a specific gene in a cell can be modelled as the following system of ODEs

$$\begin{aligned}\frac{du(t)}{dt} &= \alpha^{(k)}(t) - \beta u(t)\\ \frac{ds(t)}{dt} &= \beta u(t) - \gamma s(t)\end{aligned}$$

where $u(t)$ and $s(t)$ denote the unspliced and spliced mRNA abundance of the gene at time t in a given cell, α, β and γ are the rates of transcription, splicing, and degradation, respectively. $k \in \{\text{induction, repression, steady-state}\}$. During induction, DNA is actively being transcribed, so $\alpha > 0$, whereas during repression, $\alpha = 0$. In the steady state case, $\alpha =$ constant.

Step 1: Steady-state solution Solving the system of ODEs in the steady-state case ($\frac{du}{dt} = \frac{ds}{dt} = 0$) gives the following

$$\tilde{\gamma} := \frac{\gamma}{\beta} = \frac{u(t)}{s(t)}$$

which is the ratio of unspliced/spliced mRNA in the steady-state system. The vectors of unspliced and spliced counts across all cells are denoted as:

$$\mathbf{u} = (u_1, u_2, \ldots, u_n)^\top, \quad \mathbf{s} = (s_1, s_2, \ldots, s_n)^\top$$

In reality, due to noise and variation across cells, we fit a least squares model assuming a linear relation:

$$\mathbf{u} \approx \tilde{\gamma} \cdot \mathbf{s}$$

The optimal fit is obtained by minimizing the residual norm $\|\mathbf{u} - \tilde{\gamma}\mathbf{s}\|^2$, leading to:

$$\hat{\tilde{\gamma}} = \frac{\mathbf{u}^\top \mathbf{s}}{\|\mathbf{s}\|^2}$$

and for each cell i $\nu_i := \frac{ds_i}{dt} = \beta(u_i - \hat{\tilde{\gamma}} s_i)$ is the RNA velocities.

Step 2: Derive expressions for τ The full model can be solved to give

$$u(t) = u_0 e^{-\beta\tau} + \frac{\alpha^{(k)}(t)}{\beta}(1 - e^{-\beta\tau}),$$
$$s(t) = s_0 e^{-\gamma\tau} + \frac{\alpha^{(k)}(t)}{\gamma}(1 - e^{-\gamma\tau}) + \frac{\alpha^{(k)}(t) - \beta u_0}{\gamma - \beta}(e^{-\gamma\tau} - e^{-\beta\tau}),$$

where $\tau = t - t_0^{(k)}$ is the latent time and $t_0^{(k)}$ denote the time at which the gene transitions into state k.
Combining the two equations and set $\alpha^{(k)}(t) = \alpha$, the spliced counts can be expressed as a function of unspliced counts

$$s(t) = \tilde{\beta}u(t) + \frac{\alpha}{\gamma} - \tilde{\beta}\frac{\alpha}{\beta} + (s_0 - \frac{\alpha}{\gamma} - \tilde{\beta}(u_0 - \frac{\alpha}{\beta}))e^{-\gamma\tau}$$

where $\tilde{\beta} = \frac{\beta}{\gamma - \beta}$. Defining $\tilde{s}(t) = s(t) - \tilde{\beta}u(t)$, we can rewrite this exepression as:

$$\tilde{s}(t) = \frac{\alpha}{\gamma} - \tilde{\beta}\frac{\alpha}{\beta} + (\tilde{s}_0 - (\frac{\alpha}{\gamma} - \tilde{\beta}\frac{\alpha}{\beta}))e^{-\gamma\tau} = \tilde{s}_\infty + (\tilde{s}_0 - \tilde{s}_\infty)e^{-\gamma\tau}$$

solve this equation to obtain (in the case $\beta > \gamma$)

$$\tau = \frac{1}{\gamma}\log\left(\frac{\tilde{s}_0 - \tilde{s}_\infty}{\tilde{s}(t) - \tilde{s}_\infty}\right)$$

Step 3: EM Algorithm The Expectation Maximization Algorithm is an iterative method to find the maximum likelihood estimator of a probabilistic model in presence of latent variables (variables that cannot be observed but have impact on the outcome).

Given observed data x, latent variables z, and model parameters θ, the goal of the EM algorithm is to maximize the log-likelihood function $\log p(x \mid \theta)$. By using marginalization and by introducing an arbitrary pdf of latent variable z, we write:

$$\log p(x \mid \theta) = \log \sum_z p(x, z \mid \theta) = \log \sum_z q(z) \cdot \frac{p(x, z \mid \theta)}{q(z)}$$

Then by using Jensen's inequality on the concave logarithm function, we obtain the following lower bound:

$$\log \sum_z q(z) \cdot \frac{p(x, z \mid \theta)}{q(z)} \geq \sum_z q(z) \log \frac{p(x, z \mid \theta)}{q(z)}$$

We define this lower bound as:

$$\mathcal{L}(q, \theta) = \sum_z q(z) \log p(x, z \mid \theta) + H(q)$$

where the entropy term $H(q)$ is defined as:

$$H(q) = -\sum_z q(z) \log q(z)$$

Here we choose $q(z) = p(z \mid x, \theta)$ and this ensures the inequality turns into an equality and obtains a tight lower bound, since Jenson Inequality obtains equality iff all terms are equal in the expectation:

$$q(z) = p(z \mid x, \theta) = \frac{p(x, z \mid \theta)}{p(x \mid \theta)}$$

$$\frac{p(x, z \mid \theta)}{q(z)} = p(x \mid \theta) = \text{constant} \quad \text{w.r.t } z$$

E-Step

In each iteration, let $q(z) = p(z \mid x, \theta^{(t)})$, then the lower bound for $\log p(x \mid \theta)$ under $q(z)$ reads:

$$\log p(x \mid \theta) \geq \mathcal{L}(q, \theta) = \underbrace{\sum_z p(z \mid x, \theta^{(t)}) \log p(x, z \mid \theta)}_{Q(\theta, \theta^{(t)})} + \underbrace{H(q)}_{\text{Entropy term}}$$

Since:

$$\frac{p(x, z \mid \theta^{(t)})}{q(z)} = \frac{p(x, z \mid \theta^{(t)})}{p(z \mid x, \theta^{(t)})} = p(x \mid \theta^{(t)}) \quad \text{(a constant)}$$

Therefore, Jensen's inequality reaches lower bound $\mathcal{L}(q, \theta)$ at $\theta = \theta^{(t)}$:

$$\log p(x \mid \theta^{(t)}) = \mathcal{L}(q, \theta^{(t)}) = \sum_z p(z \mid x, \theta^{(t)}) \log p(x, z \mid \theta^{(t)}) + H(q) = Q(\theta^{(t)}, \theta^{(t)}) + H(q)$$

M-Step

Maximize the lower bound $\mathcal{L}(q, \theta)$ with respect to the parameters θ. Since the entropy function is independent of θ, so it suffices to maximize $Q(\theta, \theta^{(t)})$ and ignore the entropy term. using the $q(z)$ we defined in each iteration gives:

$$\theta^{(t+1)} = \arg\max_\theta \sum_z p(z \mid x, \theta^{(t)}) \log p(x, z \mid \theta) = \arg\max_\theta Q(\theta, \theta^{(t)})$$

The expression on the right hand side after argmax is usually written as the expectation of the complete data log-likelihood under the posterior distribution of z:

$$Q(\theta \mid \theta^{(t)}) := \mathbb{E}_{z \sim p(z \mid x, \theta^{(t)})} \left[\log p(x, z \mid \theta) \right]$$

Hence we see that by the choice of $\theta^{(t+1)}$ and by considering the lower bound we could see an improvement in the log-likelihood over each iteration:

$$\log p(x \mid \theta^{(t+1)}) \geq Q(\theta^{(t+1)}, \theta^{(t)}) + H(q) \geq Q(\theta^{(t)}, \theta^{(t)}) + H(q) = \log p(x \mid \theta^{(t)})$$

In the dynamical model, the kinetic parameters β and γ are part of the model parameter set

$$\boldsymbol{\theta} = (\alpha, \beta, \gamma),$$

which is estimated iteratively via the EM algorithm. The latent time τ is treated as a hidden variable in this framework. The EM algorithm alternates between:

- **E-step:** Estimate τ given the current parameters $\boldsymbol{\theta}$;
- **M-step:** Update $\boldsymbol{\theta}$ given the current estimates of τ;

This process is repeated until convergence. Finally, the RNA velocity for each cell i is computed by plugging the estimated parameters into the original ODE:

$$\nu_i = \frac{ds_i(t)}{dt} = \hat{\beta} u_i - \hat{\gamma} s_i$$

Step 4: Compute transition probabilities Now we have the RNA velocities, we want to compute transition probabilities. We use neighbouring cells to estimate the future direction; we define $\delta_{ij} := x_j - x_i$ as the displacement between cells i and j. Intuitively, if the displacement is in a similar direction to the RNA velocity, the probability of a transition is greater. We use cosine similarity to capture this behaviour

$$\tilde{\pi}_{ij} = \frac{\delta_{ij} \cdot \nu_i}{\|\delta_{ij}\| \, \|\nu_i\|}, \quad \text{only if } j \in \mathcal{N}_k(i)$$

We use an exponential kernel to transform the values $\tilde{\pi}_{ij}$, since these are cosine values and as such lie in the interval $[-1, 1]$, so

$$\pi_{ij} = \frac{1}{Z_i} \exp(\frac{\tilde{\pi}_{ij}}{\sigma})$$

where Z_i is a normalisation constant and σ is a global parameter set by softmax scale.

4.3 CellRank

Construct a directed transition matrix respecting Pseudo-time

CellRank is a probabilistic model for calculating cell fate decisions based on Markov Chains. It originally uses RNA-velocity, but in the absence of spliced/unspliced single-cell data, we use the Pseudo-time kernel constructed in the DPT method. When constructing the transition matrix for CellRank, directions are introduced by modifying the original transition matrix to respect the forward progression of pseudo-time.

Starting from the diffusion kernel K constructed in Section 4.1 using an adaptive Gaussian kernel, CellRank ensures that transitions can only take place from earlier to later pseudo-time, ensuring biological order in lineage progression. $\tau(i)$. We define a directed kernel $\tilde{K}$ by:

$$\tilde{K}_{ij} = \begin{cases} K_{ij}, & \text{if } \tau(j) > \tau(i) \\ 0, & \text{otherwise} \end{cases}$$

The directed transition matrix this time is obtained by normalizing each row:

$$\tilde{P}_{ij} = \frac{\tilde{K}_{ij}}{\sum_{j'} \tilde{K}_{ij'}}$$

If using RNA velocity rather than pseudotime, the transition matrix uses the values π_{ij} defined in the RNA velocity section.

Defining an absorbing Markov Chain

To compute fate probabilities, by reordering of the cells and relabelling, the cells are partitioned into 2 types: - Transient cells $\mathcal{T}$: intermediate states - Terminal cells $\mathcal{A}$: absorbing states (terminal fates) The transition matrix $\tilde{P}$ is reordered into a block structure:

$$\tilde{P} = \begin{bmatrix} Q & R \\ 0 & I \end{bmatrix}$$

- $Q \in \mathbb{R}^{t \times t}$: transitions within transient states
- $R \in \mathbb{R}^{t \times m}$: transitions from transient to absorbing states

Fundamental Matrix construction and Fate Probability

Define a binary random variable $X_{ij}^{(k)} \in \{0, 1\}$, indicating whether starting at transient state i, is in transient state j at time step k. The expected value of this variable is:

$$\mathbb{E}[X_{ij}^{(k)}] = (Q^k)_{ij}$$

Summing over all time steps gives the expected total number of visits from state i to j:

$$N_{ij} := \mathbb{E}\left[\sum_{k=0}^{\infty} X_{ij}^{(k)}\right] = \sum_{k=0}^{\infty} \mathbb{E}[X_{ij}^{(k)}] = \sum_{k=0}^{\infty} (Q^k)_{ij}$$

In matrix form, we define the **fundamental matrix**:

$$N := (I - Q)^{-1} = I + Q + Q^2 + Q^3 + \cdots$$

This expansion is valid when the **spectral radius** of Q denoted by $\rho(Q)$, satisfies $\rho(Q) < 1$, which is guaranteed for absorbing Markov chains.

Fate Probability Matrix

To compute the fate probabilities of each terminal state, let R be the matrix of transition probabilities from transient to absorbing states. Our goal is to compute the fate probability Matrix F, where each row F_i gives the probabilities that a transient cell i will eventually be absorbed into each terminal fate.

$$F_{ia} = \mathbb{P}(\text{starting at } i,\ \text{eventually absorbed in absorbing state } a \in \mathcal{A})$$

However, a path from i to a is not necessarily a single step. In fact, a typical trajectory will: | wander among transient states $j \in \mathcal{T}$, possibly visiting some of them multiple times, and on each visit to j, have a certain chance of making a one-step transition from $j \to a$. So by calculating the expected counts from i to j , the total chance to end up in a is the sum over all intermediate transient states j of:

$$\underbrace{\mathbb{E}[\#\text{ visits from } i \text{ to } j]}_{N_{ij}} \times \underbrace{\mathbb{P}(j \to a)}_{R_{ja}}$$

That is,

$$F_{ia} = \sum_{j \in \mathcal{T}} N_{ij} R_{ja} \quad \Rightarrow \quad F = NR = (I - Q)^{-1} R$$

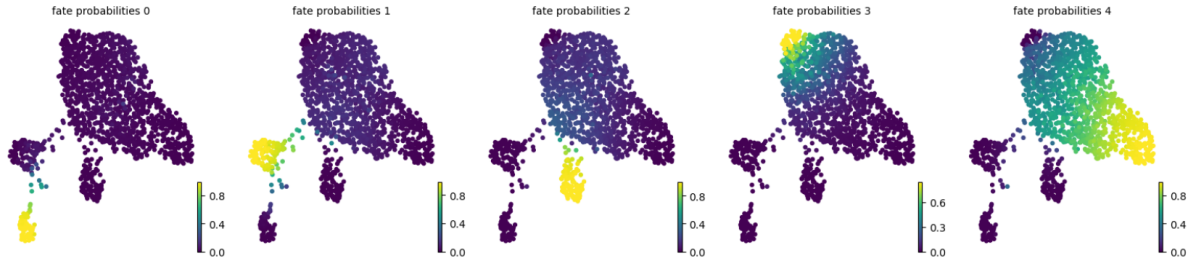


Figure 5 Cellrank branch probabilities toward each terminal fate across pseudotime

4.4 Palantir

Similar to DFT, we again use a diffusion map to perform a nonlinear dimensional reduction to capture the non-linear structures of the gene expression data. As the cells differentiate and their gene expressions vary, they follow curved and branching paths rather than straight lines, so using non-linear dimensional reduction methods can help uncover and preserve this structure within the data.

Palantir is a probabilistic method that is designed to model continuous and branching trajectories. It builds almost directly from the DFT method by the more explicit use of Markov chains to simulate random walks starting from a chosen cell to calculate pseudotime and cell fate probabilities [10].

Step 1: Build transition matrix based on Gaussian Kernel (same as DPT)

Let $x_i, x_j \in \mathbb{R}^k$ be the PCA-reduced coordinates of cells i and j, and let $\mathcal{N}_k(i)$ denote the k-nearest neighbors of cell i. Define the local adaptive kernel that forms a sparse matrix K:

$$K_{ij} = \begin{cases} \exp\left(-\dfrac{\|x_i - x_j\|^2}{\sigma_i \sigma_j}\right), & j \in \mathcal{N}_k(i) \\ 0, & \text{otherwise} \end{cases}$$

where the local bandwidth is

$$\sigma_i = \frac{1}{k} \sum_{j' \in \mathcal{N}_k(i)} \|x_i - x_{j'}\|$$

The diffusion transition matrix P is defined by normalizing the kernel:

$$P_{ij} = \frac{K_{ij}}{\sum_{j' \in \mathcal{N}_k(i)} K_{ij'}}, \quad j \in \mathcal{N}_k(i)$$

Step 2: Calculate Shannon Entropy and identify terminal cells

The Shannon entropy of the transition probability distribution from cell i is given by:

$$H_i = -\sum_{j' \in \mathcal{N}_k(i)} P_{ij'} \log P_{ij'}$$

This measures the uncertainty in the possible next states for cell i in the Markov process. We use entropy to locate cells that represent terminal states, and lower entropies usually represent higher stability.

Step 3: Modify transition matrix to form an Absorbing Markov Chain

Let:

- $r \in \{1, \ldots, n\}$ denote the root cell (early progenitor),
- $\mathcal{A} = \{a_1, a_2, \ldots, a_m\} \subset \{1, \ldots, n\}$ be the set of terminal cells (absorbing fates),
- $\mathcal{T} := \mathcal{A}^c = \{1, \ldots, n\} \setminus \mathcal{A}$ be the set of transient cells.

After identifying the terminal cells and label them as terminal fates set $\mathcal{A}$, We define an absorbing Markov chain $\tilde{P} \in \mathbb{R}^{n \times n}$ by modifying the rows corresponding to A in P as:

$$\tilde{P}_{ij} = \begin{cases} 1, & \text{if } i = j \in \mathcal{A} \\ 0, & \text{if } i \in \mathcal{A}, j \neq i \\ P_{ij}, & \text{if } i \in \mathcal{T} \end{cases}$$

Step 4: Define Pseudo-time as expected number of steps needed to visit a cell

Let the initial distribution $\pi_0 = e_r \in \mathbb{R}^n$ be a one-hot vector centered at the root cell. The state distribution after t steps of the Markov chain is:

$$\pi_t = \pi_0 \tilde{P}^t$$

We define the **Palantir pseudotime** $\tau(i)$ for each cell i as the expected number of steps needed to visit cell i:

$$\tau(i) = \sum_{t=1}^{T} t \cdot \pi_t(i)$$

This is the average number of steps of visiting node i.

Step 5: Calculate Fate Probabilities

Exactly the same as CellRank, we could reorder $\tilde{P}$ so that transient states $\mathcal{T}$ appear before absorbing states $\mathcal{A}$, yielding the block matrix:

$$\tilde{P} = \begin{bmatrix} Q & R \\ 0 & I \end{bmatrix}$$

- $Q \in \mathbb{R}^{t \times t}$: transitions within transient states,
- $R \in \mathbb{R}^{t \times m}$: transitions from transient to absorbing states.

We define the fundamental matrix:

$$N = (I - Q)^{-1} = \sum_{k=0}^{\infty} Q^k$$

Then as derived in CellRank, the fate probability matrix $F \in \mathbb{R}^{t \times m}$ is:

$$F = NR$$

Each row $F_i \in \mathbb{R}^m$ gives the probability that cell $i \in \mathcal{T}$ is eventually absorbed by each fate $a \in \mathcal{A}$.

Step 5: Alternative computation of Fate Probability, converging to the same result

We propagate the fate probabilities $\pi_i^{(k)} \in \mathbb{R}^m$ (where m is the number of terminal states) iteratively using the following rule:

$$\pi_i^{(k+1)} = \sum_j \widetilde{P}_{ij} \cdot \pi_j^{(k)} \quad \text{for all } i \in \mathcal{T} \text{ (transient cells)}$$

This update rule means:

- Sort all cells $i_1, i_2, \ldots, i_n$ by their pseudo-time $\tau(i)$;
- Update fate probabilities in pseudo-time order:

$$\pi_{i_1}^{(k+1)} \to \pi_{i_2}^{(k+1)} \to \cdots \to \pi_{i_n}^{(k+1)}$$

- For terminal cells $i \in \mathcal{A}$, the fate probabilities remain constant during all iterations. Each such cell is initialized with a standard basis vector $e_i \in \mathbb{R}^m$, where the i-th entry corresponds to its assigned fate:

$$\pi_i^{(k)} = \pi_i^{(0)} = e_i, \quad \text{for all } k \geq 0,\ i \in \mathcal{A}$$

Proof of Convergence Let the absorbing Markov chain transition matrix $\widetilde{P} \in \mathbb{R}^{n \times n}$ be decomposed into block form:

- $Q \in \mathbb{R}^{t \times t}$: transitions among transient states $\mathcal{T}$
- $R \in \mathbb{R}^{t \times m}$: transitions from transient states to absorbing states $\mathcal{A}$

Then we iteratively propagate fate probabilities and update them according to pseudo-time across transient cells using the recurrence :

$$\pi_i^{(k+1)} = \sum_j \tilde{P}_{ij} \cdot \pi_j^{(k)}$$

can be decomposed as:

$$\pi_i^{(k+1)} = \sum_{j \in \mathcal{T}} Q_{ij} \cdot \pi_j^{(k)} + \sum_{j \in \mathcal{A}} R_{ij} \cdot \pi_j^{(k)} = \sum_{j \in \mathcal{T}} Q_{ij} \cdot \pi_j^{(k)} + R_i$$

Stacking all $\pi_i^{(k)} \in \mathbb{R}^m$ for $i \in \mathcal{T}$ yields the matrix recurrence:

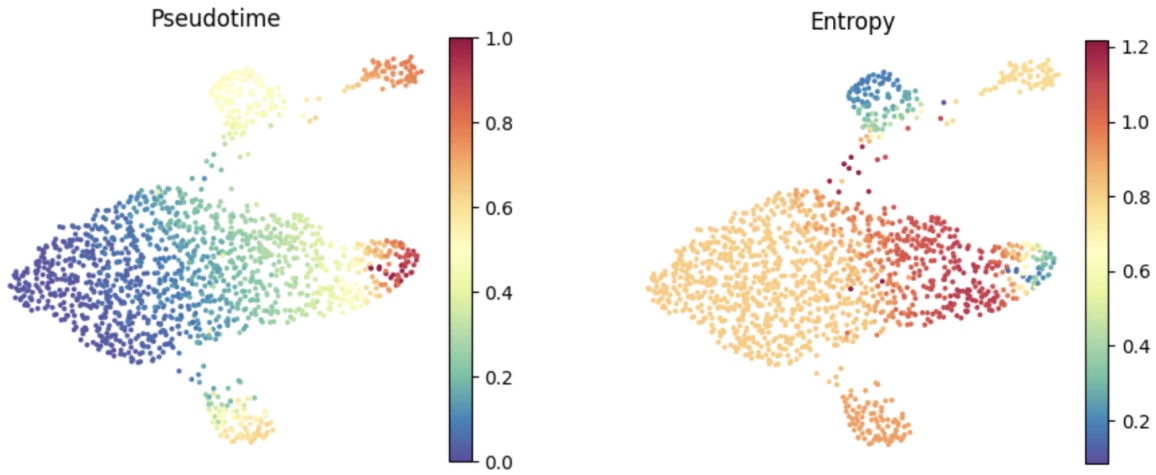
$$\pi_{\mathcal{T}}^{(k+1)} = Q \cdot \pi_{\mathcal{T}}^{(k)} + R$$

where $\pi_{\mathcal{T}}^{(k)} \in \mathbb{R}^{t \times m}$ concatenates the fate probability of each transient cell at iteration k row-wise. By definition:

$$\pi_{\mathcal{T}}^{(0)} = R$$

And thus, it converges to the analytical solution:

$$\pi_{\mathcal{T}}^{(\infty)} = \sum_{k=0}^{\infty} Q^k R = (I - Q)^{-1} R = F$$



(a) **(b)**

Figure 6 Visualisation of Palantir pseudotime and entropy in UMAP space

4.5 Method Comparisons

CellRank with RNA velocity is, in general, superior to CellRank constructed using a pseudotime kernel [11]. This is because directionality has to be inferred artificially from static data in the pseudotime case, as opposed to using the direction inherent to RNA velocity. However, using RNA velocity requires additional data, namely the counts of spliced and unspliced RNA, so depending on the data available may not be viable. Moreover, CellRank identifies fate-driven genes and tracks their gene expression across all cells, which helps to identify terminal states by locating the region where these driver genes are highly expressed (since usually driver genes are also marker genes of corresponding terminal states).

Additionally, Palantir produces an entropy plot in Figure 6b which represents the differentiation potential of each cell. We can see that the ISCs have a high differential potential which supports our current biological understanding that ISCs lie at the base of the crypt along with Paneth cells which send stem cell maintenance signals, ensuring they stay stem cells. However, stem

cells are inherently cells that differentiate into more specialised cell types and replenish cells, which is reflected in the high entropy shown in the graph. However, we see that there is a group of cells with a much higher differentiation potential, and this clearly represents the TA-cells in the crypt, which are known to proliferate rapidly and differentiate into either absorptive or secretory cells and so compared to the ISCs which are receiving signals to maintain themselves. ISCs have a much longer cell cycle time than TA-cells as well.

Unlike other methods, DPT requires a specific start point to be selected as a reference for distances. We choose ISCs as the start point, as biologically we know that these are the start point of cell differentiation. DPT can have the advantage being resistant to noise, but the selection of a root cell his has the potential to introduce bias if the incorrect cell is chosen [12].

5 Cell-Cell Interactions

5.1 LIANA (Ligand-Receptor Analysis Framework)

LIANA is a unified interface tool for combining and integrating communications between different cell types, aiming to provide a standardized platform for inferring and comparing the interactions between Ligand-Receptor pairs. There are 5 methods embedded in the LIANA algorithm.

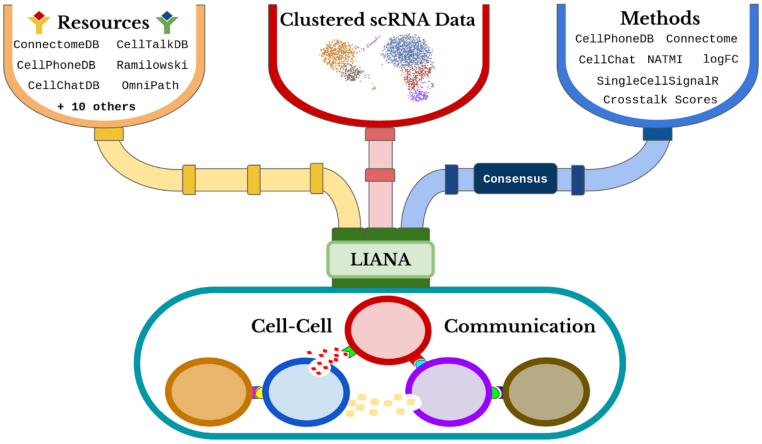


Figure 7 LIANA Procedure and Methods

Connectome

- **Expression proportion threshold:** Before computing interaction scores between cell types, Connectome applies two key filtering steps, and interactions are only computed for Ligand-Receptor pairs whose expressions pass the expression threshold. The expression level of ligand L in sender cell type s is based on average expression and is defined as:

$$\text{Expr}_s(L) = \begin{cases} \frac{1}{N_s} \sum_{i \in s} x_i(L), & \text{if } \frac{1}{N_s} \sum_{i \in s} \mathbf{1}[x_i(L) > 0] \geq p \\ 0, & \text{otherwise} \end{cases}$$

Similarly for the expression of receptor R in receiver cell type r:

$$\text{Expr}_r(R) = \begin{cases} \frac{1}{N_r} \sum_{j \in r} x_j(R), & \text{if } \frac{1}{N_r} \sum_{j \in r} \mathbf{1}[x_j(R) > 0] \geq p \\ 0, & \text{otherwise} \end{cases}$$

Given a sender–receiver cell pair (s, r) and a Ligand–Receptor pair (L, R), the Connectome scoring function computes:

$$\text{Score}_{s \to r}^{(L,R)} = \log_2 \left(\text{Expr}_s(L) \cdot \text{Expr}_r(R) + \epsilon \right)$$

where $\text{Expr}_s(L)$ and $\text{Expr}_r(R)$ are the average expression levels of ligand L and receptor R in cell types s and r respectively, and ϵ is a small constant to avoid $\log(0)$.

- **Score threshold:** After scoring, weak interactions (e.g. those below a cutoff) can be filtered out to reduce noise.

NATMI

NATMI scores each ligand–receptor pair based on the product of their mean expression values in sender and receiver cell types, this is the simplest method LIANA uses to score the expression of Ligand-Receptor pairs.

Given a pair (s, r) and ligand–receptor (L, R), the NATMI score is:

$$\text{Score}_{s \to r}^{(L,R)} = \text{Expr}_s(L) \cdot \text{Expr}_r(R)$$

This method captures the extent to which both ligand and receptor are co-expressed in their respective cell types. No transformation or normalization is applied.

CellPhone DB (Permutation-based Differential Communication)

To assess whether a ligand–receptor pair (L, R) exhibits specific signaling between two cell types s (sender) and r (receiver), CellPhoneDB defines the observed interaction score as:

$$\text{Score}_{s \to r}^{(L,R)} = \text{Expr}_s(L) \cdot \text{Expr}_r(R)$$

where $\text{Expr}_s(L)$ and $\text{Expr}_r(R)$ are the average expression levels of ligand L and receptor R in cell types s and r, respectively, subject to an expression threshold.

To evaluate statistical significance, a permutation test is performed:

- Cell type labels are randomly shuffled among all cells;
- For each permutation, new sender/receiver assignments $s^{(\text{perm})}, r^{(\text{perm})}$ are generated;
- The permuted score is then computed as:

$$\text{Score}_{s^{(\text{perm})} \to r^{(\text{perm})}}^{(L,R)} = \text{Expr}_{s^{(\text{perm})}}(L) \cdot \text{Expr}_{r^{(\text{perm})}}(R)$$

After k permutations, the p-value is computed as:

$$p = \frac{\#\left\{\text{PermutedScore} \geq \text{ObservedScore}\right\} + 1}{k + 1}$$

This non-parametric test evaluates whether the interaction score observed between s and r is significantly higher than expected under random assignment of cell type labels, indicating signalling for specific cell types.

log2FC

The log2FC method evaluates whether a ligand or receptor gene is specifically overexpressed in a particular cell type, compared to all other cell types.

For a ligand L in sender cell type s, the log fold-change is defined as:

$$\log_2 FC_s(L) = \log_2 \left(\frac{\text{Expr}_s(L) + \epsilon}{\frac{1}{|S| - 1} \sum_{s' \in S \setminus \{s\}} \text{Expr}_{s'}(L) + \epsilon} \right)$$

where:

- $\text{Expr}_s(L)$ is the mean expression of ligand L in sender cell type s;
- S is the set of all cell types;
- ϵ is a small positive constant to avoid division by zero;
- The denominator represents the average expression of L in all cell types except s.

A positive log2FC implies that ligand L is specifically overexpressed in cell type s, relative to background, and thus may be functionally relevant in intercellular communication.

Similar computation is applied to receptor R in receiver cell type r, and the final interaction score is computed as:

$$\text{Score}_{s \to r}^{(L,R)} = \log_2 FC_s(L) + \log_2 FC_r(R)$$

Single Cell Signal R

This method estimates the strength of interaction between ligand L in sender cell type s and receptor R in receiver cell type r based on an internally defined LRscore function:

$$\text{LRscore}_{L,R}^{s \to r} = \frac{\sqrt{\text{Expr}_s(L) \cdot \text{Expr}_r(R)}}{\sqrt{\text{Expr}_s(L) \cdot \text{Expr}_r(R)} + \mu}$$

where μ is some parameter used for saturating normalization, to prevent the dominating effect of some high expressing genes.

5.2 Construction and Interpretation of Cell–Cell Communication Maps

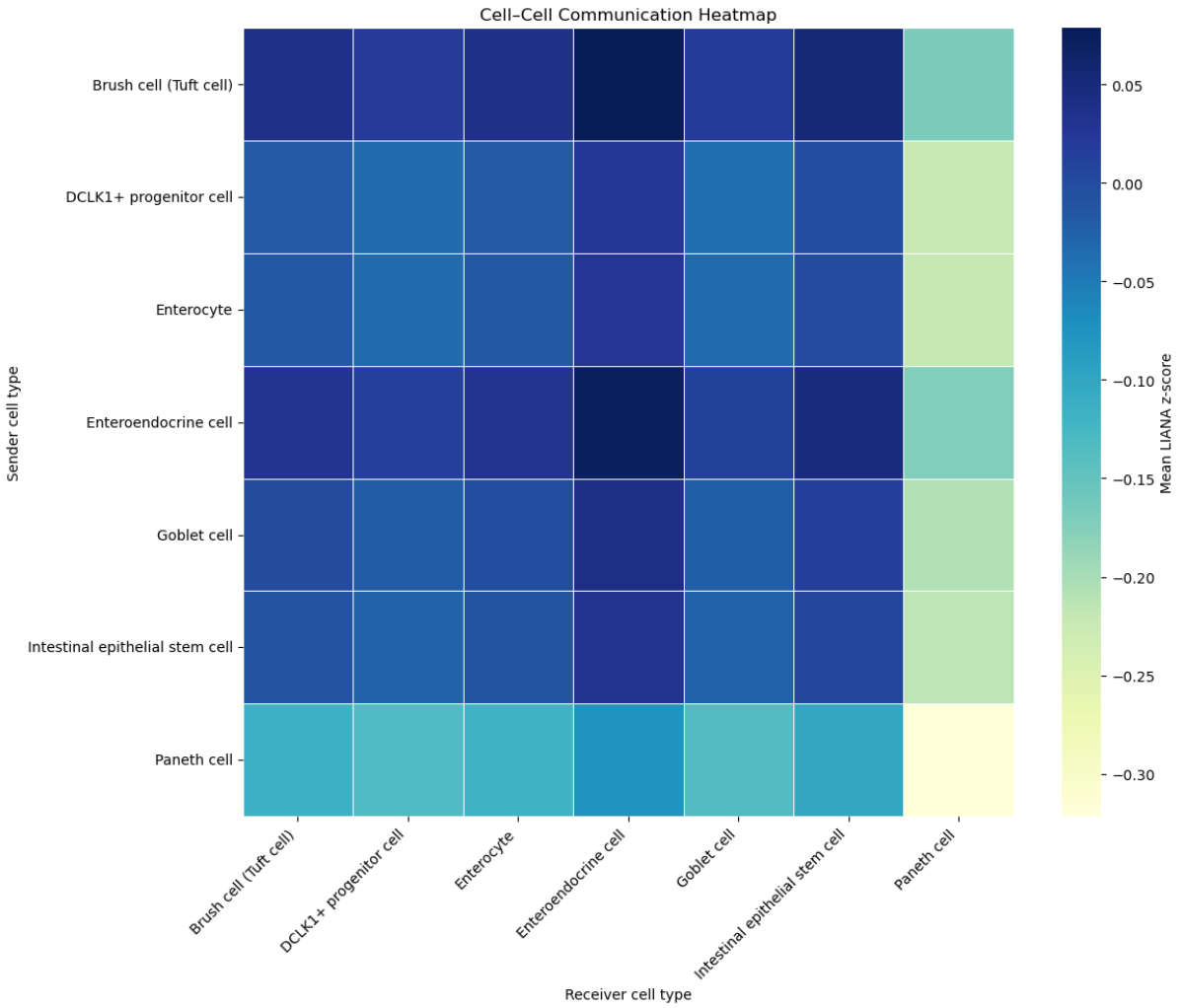


Figure 8 Heatmaps for overall interactions between cell types

Figure 8 summarises the directed ligand-receptor traffic between all the pairs of cell types found from clustering. Using the LIANA pipeline, we first scored each individual ligand-receptor pair and then averaged the resulting z-standardised weights for every sender/receiver combination. Therefore, this heatmap reads as a two-dimensional bar plot where the horizontal axis indicates how strongly a given cell type receives signals from all other lineages and the vertical axis shows how strongly that same type sends signals to its neighbours.

Notice that since signalling is intrinsically directional, the matrix is non-symmetric. Since each entry in the heatmap is a mean of z-scored ligand receptor hits, the plot primarily demonstrates the breadth of signalling and not absolute potency. Deep blues (>0) indicate sender and receiver pairs whose average ligand-receptor strength is above the global baseline whereas the yellow/green (<0) mark pairs that are below the baseline. Crucially, a negative entry does not correspond to a lack of signalling but in fact indicates that after all expressed and non-expressed ligand and receptor pairs are pooled together, this particular sender receives or emits fewer or more narrowly focused interactions than the dataset's mean. The diagonal lets us see whether a cell type primarily communicates in an autocrine or paracrine fashion. Figure 9 confirm that some specialised and low-abundance populations such as Paneth cells, have some of the strongest overall interactions, despite showing cooler entries overall in Figure 8.

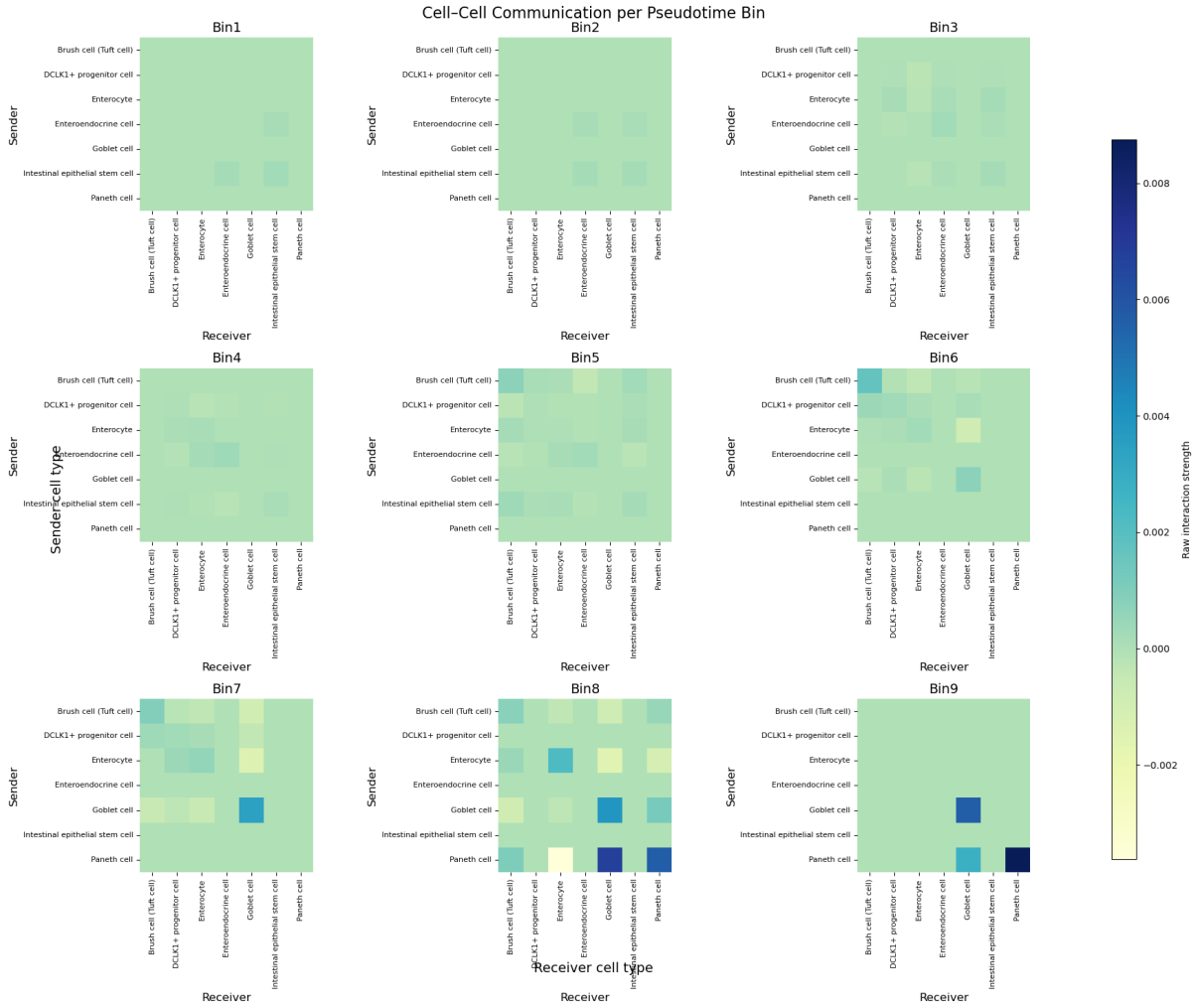


Figure 9 Heatmaps for cell-cell interactions across pseudo-time bins

Our simulation had the first pseudotime bin to only include stem and Enteroendocrine cells and each bin is such that at least two cell types exist within that pseudotime subsection. Then, by LIANA, we calculated the raw strength of interactions between the cell types that exist within each bin and plotted a heatmap for each bin seen in Figure 9. These heatmaps provide a clear view of the cell types that exist throughout different sections of pseudotime and their interaction strengths with cells at a similar point in the pseudotime. In the early bins, only interactions with stem cells and Enteroendocrine cells are present. Interactions involving Goblet, Paneth, Enterocyte and Tuft cells can clearly be seen in bin 8. This is because they are terminal states and so appear later in the pseudotime. It could potentially allow us to single out interactions between physically close, spatially in the crypt-villus axis, cells as even though we are looking at pseudotime, we know there is a general upwards movement of the cells as they differentiate. We know that Paneth cells only exist at the base of the crypt, so in bin 8 from Figure 9, if we ignore the Paneth cells, we get a clearer visualisation of how the secretory cells may interact in the upper regions of the crypt-villus axis.

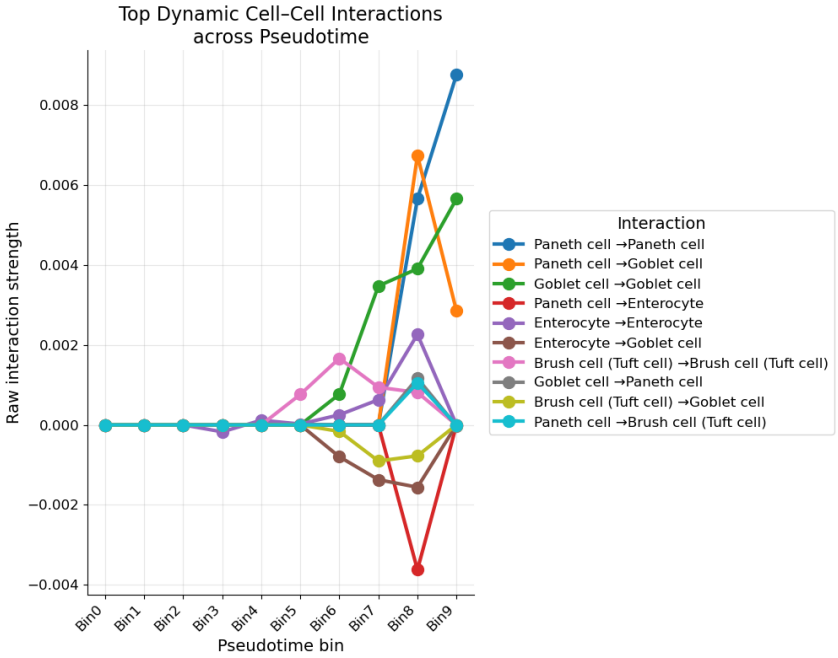


Figure 10 Line plots for top interactions between cell types

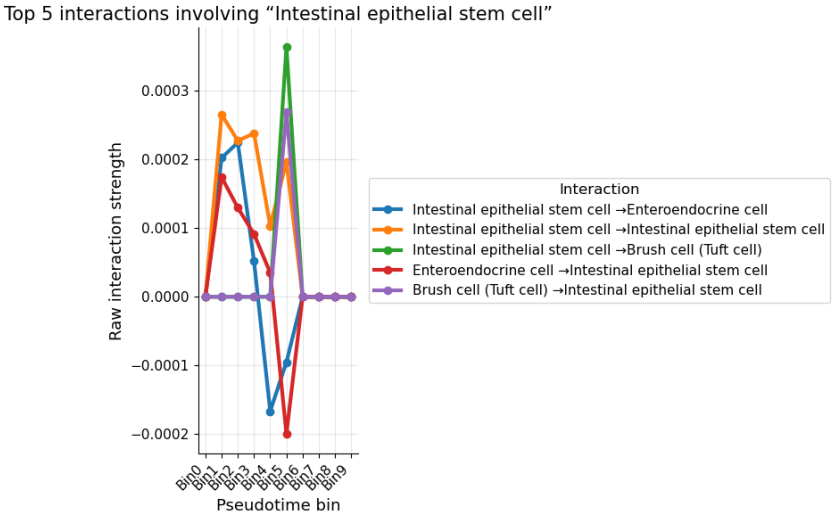


Figure 11 Line plots for Interactions with stem cells

For interactions involving the stem cells, notice that the interaction strength is strong at the start and middle along the pseudo-time, and drastically decrease to zero after bin 6. This is because there are no stem cells at later stages, so no interactions at all. Notice that the interaction between stem cells and Brush cells is initially zero because Brush cells are produced in later bins.

For the top 10 cell-cell interactions selected, notice there are no interactions with stem cells. This is because the interactions with stem cells usually have low strength, as can be seen from the above heatmaps (weak contrast), so they are not chosen to be top 10 interactions. And the interaction strength between other cells is initially around 0 and begin to increase from bin 4. We could anticipate this because Paneth, Goblet, Enterocyte and Brush cells are all terminal states and are produced in later stages, so this follows from biological order.

6 Results

6.1 Cluster Result

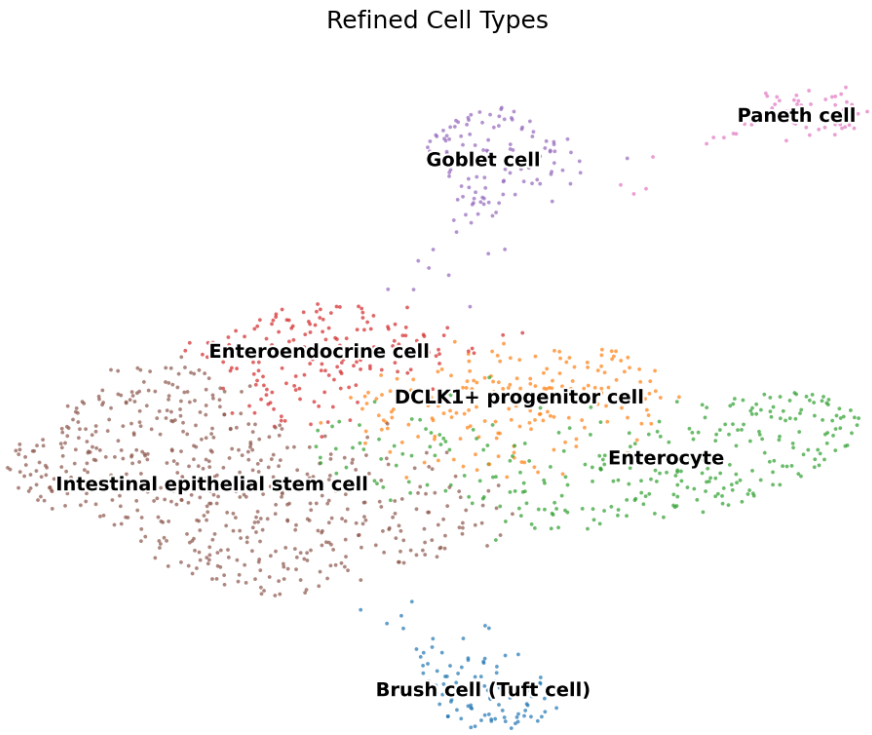


Figure 12 Clearer illustration of cell types on clusters

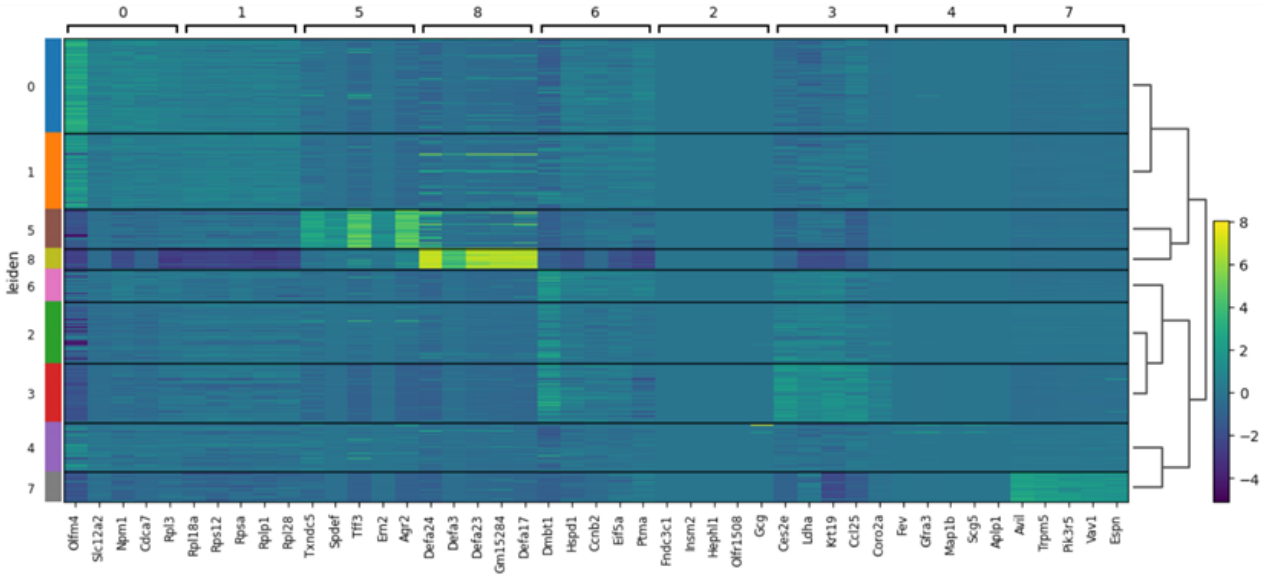


Figure 13 Heatmap for marker genes expression in each cluster

The horizontal axis represents the top 5 marker genes within each cluster, and colour means the expression of these marker genes in every cluster by matching the number labels from the vertical axis. Notice the marker genes expression of cluster 8 in itself is significantly higher than

in other clusters, and this is verified by the UMAP of Leiden Clustering as cluster 8 is far from the major clusters, and vice versa for clusters 3, 5 and 7. This is biologically reasonable since they are labelled as terminal states in later stages, so their gene expressions must be special and distinctive.
Also, by inspecting the row corresponding to cluster 8, we find that the marker gene expressions for cells in clusters 1, 6, and 3 are very low in cluster 8. This could be anticipated because these clusters are far away from cluster 8 in the UMAP, so they share completely different marker gene expressions. The marker gene expression similarity could also be seen from the hierarchical clustering on the right.

Notice Transit Amplifying cells are not shown in the cluster labels, because they are intermediate states between stem cells and mature differentiated cells. And since their gene expressions cannot over-weight other cell types [10], no cluster is labelled as a TA cell.

6.2 Trajectory method graphs

These graphs produced by the trajectory methods show a clear branching trajectory from the ISC cluster to the absorptive lineage (Enterocyte) and secretory lineages (Paneth, Goblet and Enteroendocrine). This pseudotime ordering can reveal the dynamics of gene expression through a given branch for a specific cell fate, enabling us to better see and understand the gene dynamics.

6.2.1 Cell Rank Inferences

Figure 14 Paneth driver genes expression distribution

The UMAP projection (Figure 14) displays the expression levels of Paneth fate driver genes across individual cells. The colour intensity represents the expression level of the given gene in the cells. By cross-referencing the cluster label graph, the bottom left region corresponding to the Paneth cluster appears with the highest colour intensity. This elevated expression confirms that these genes remain highly expressed in mature Paneth cells and implies that they are actively driving the differentiation toward the Paneth fate.

Similarly, in the UMAP of Enterocyte driver gene expressions (Figure 15), cells in the Enterocyte cluster display elevated expression of driver gene expression compared to other clusters. This indicates that these genes serve as markers or functional regulators specific to the Enterocyte lineage.

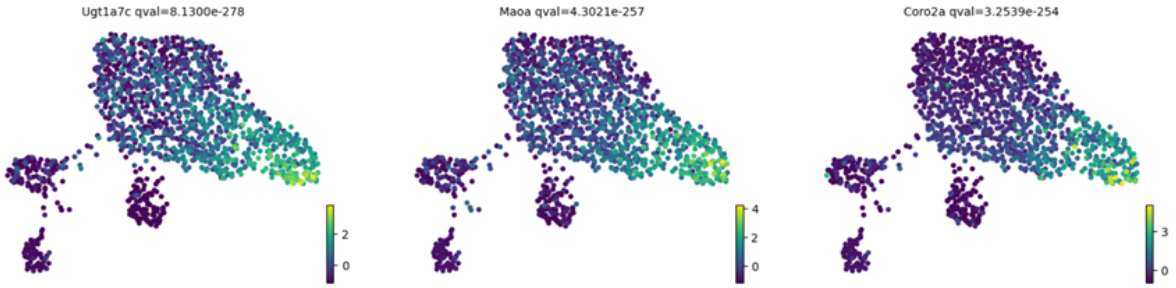


Figure 15 Enterocyte driver genes expression distribution

We could also visualize the line plot of driver gene expression in Enterocyte cells (Figure 16). In that figure, these genes show a rapid increase in expression during early stages of pseudotime, followed by a sharp drop below average levels. This pattern indicates they function as early fate-driven genes, actively regulating differentiation toward the Enterocyte lineage at the beginning. Once the cells commit to their fate, these genes are repressed and act as transient regulators. This is not contradictory to the UMAP distribution plot (Figure 15), which shows relative expression levels across cell clusters. Rather, Figure 16 focuses on the magnitude and temporal dynamics of expression over pseudotime, clarifying how these genes behave dynamically during Enterocyte differentiation.

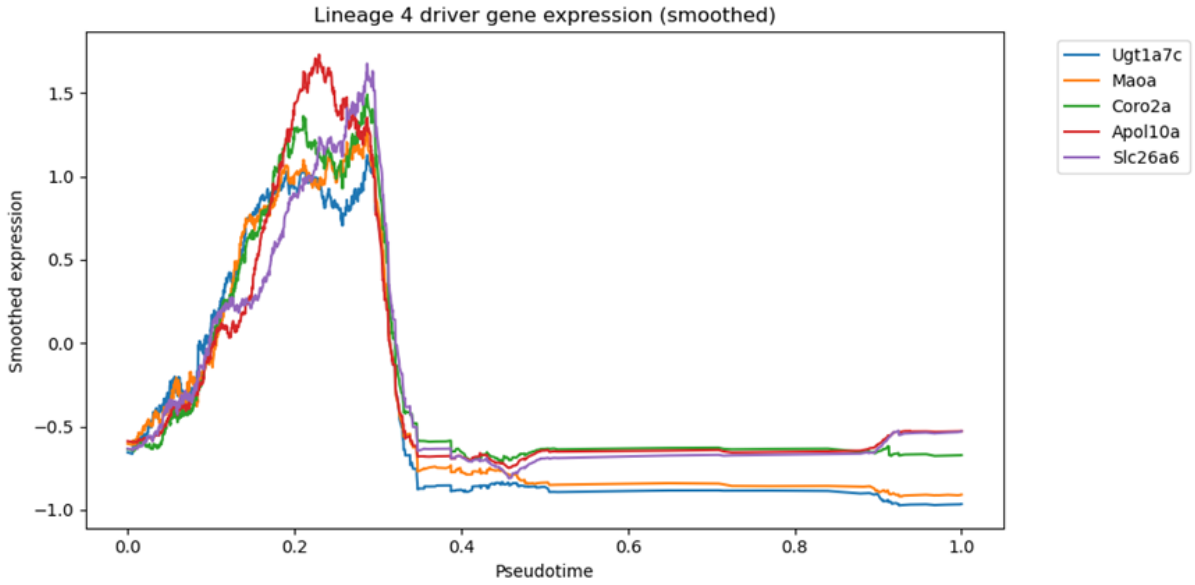


Figure 16 Enterocyte driver genes expression line plot

6.2.2 Palantir Inferences

From the fate probability distribution of each terminal state (Figure 17), we concluded that Enterocyte, Goblet cell and Paneth cell are very strong terminal states because the cells in corresponding region are marked as dark red, showing a definite trend towards ending states. In contrast, the Enteroendocrine and Brush (Tuft) states display a siginicant proportion of cells in red or orange, with fate probabilities around 0.4 to 0.5. This suggests ambiguity in fate assignment for these states, implying they are not strongly defined by Palantir.

This conclusion could be further supported by the entropy graph in Figure 6b , where cells with higher entropy correspond to Enteroendocrine and Brush cell states. Elevated entropy near these cell types indicates less stability.

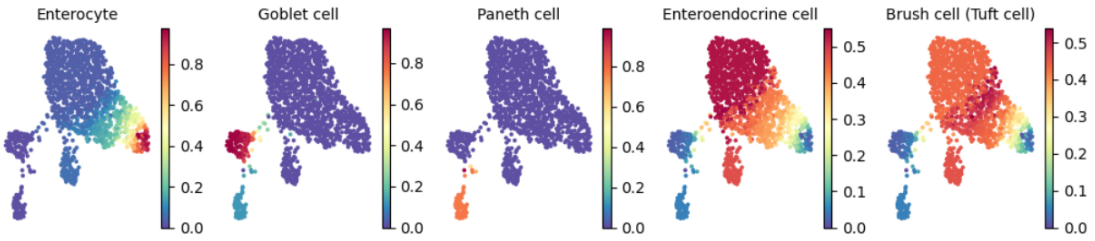


Figure 17 Fate probabilities of individual cells of each terminal state

We also verified these findings using the branch probability visualization provided by Palantir (Figure 18). In the left-hand plots, cells that increase in fate probability towards a specific terminal state are highlighted, while the right-hand UMAPs show the distribution of these cells in the clusters. For Goblet cell and Paneth cells, the branch probability plots appear discrete, reflecting the low number of fate-committed cells and the absence of a continuous trajectory. Conversely, Enteroendocrine cell and Brush cell are identified as weak terminal states, so their branch probabilities do not show substantial changes over pseudotime. Instead, as pseudotime advances, many cells reduce their probability of adopting Enteroendocrine or Brush fates and shift toward other terminal states. This is another evidence that most cells eventually differentiate into more defined lineages rather than committing to Enteroendocrine or Brush fates.

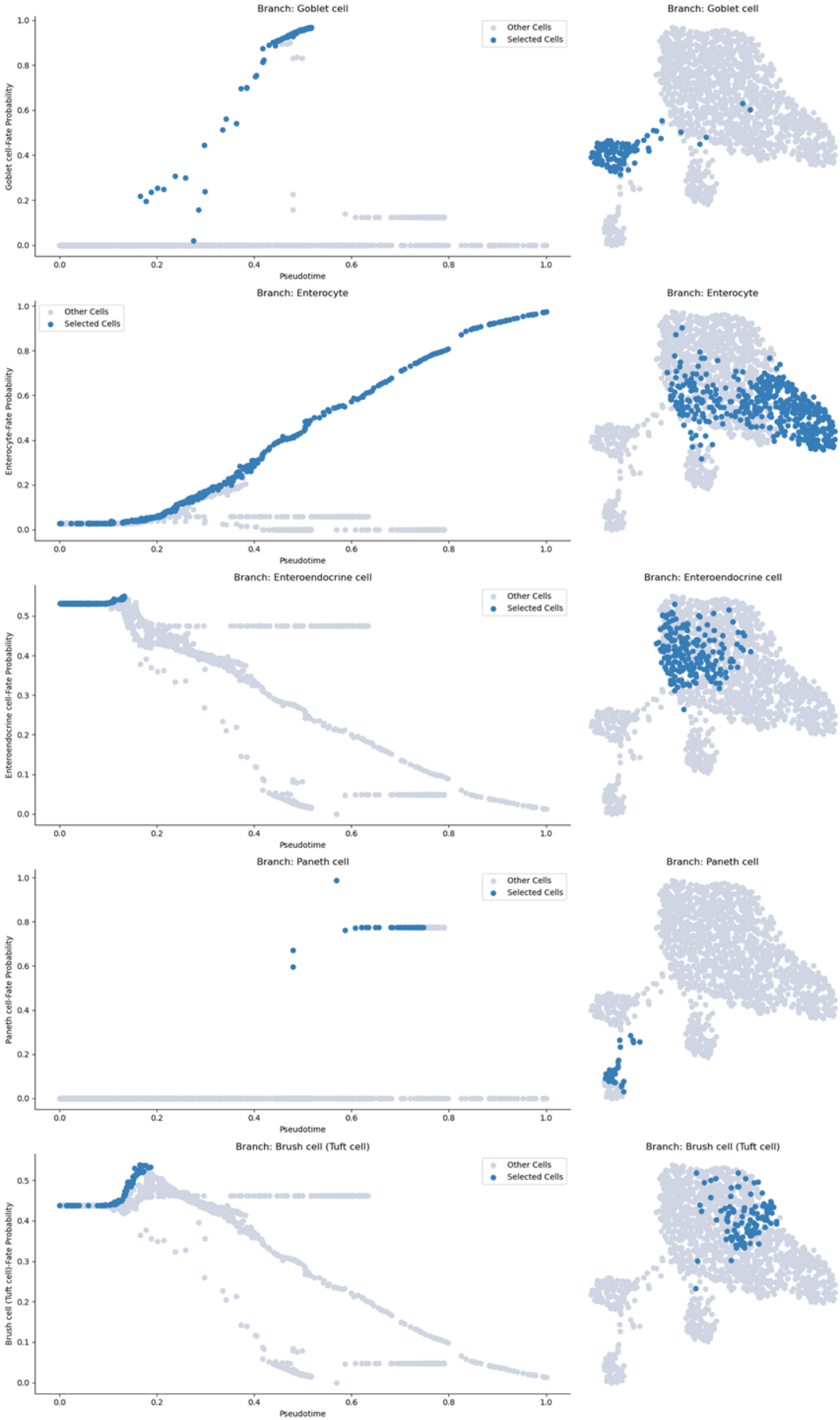


Figure 18 branch probabilities of each terminal state

6.3 Cell-cell Interactions Inferences

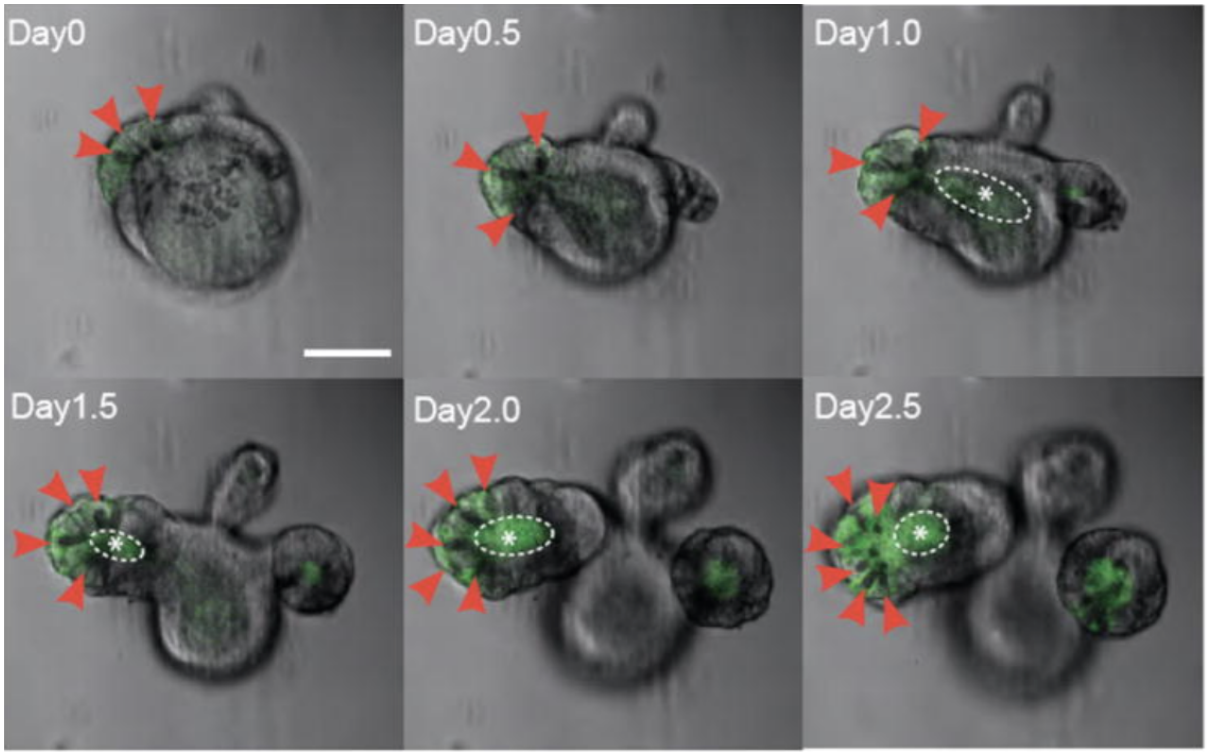


Figure 19 From Figure 1D of Sato *et al.* [13]

6.3.1 Paneth Cell Behaviours

Looking at both Figure 8 and Figure 9, we can see that Paneth cells are specialised niche broadcasters, which means they are located close to the stem cells and only send a few specific but strong signals. Figure 8 shows that the Paneth row and column are the coolest in the matrix (approximately -0.15 to -0.32). This is the result of averaging thousands of ligand-receptor pairs that the Paneth cells do not express in their gene profile, yet the single warmest square in that row is the Paneth $\rightarrow$ ISC entry. In addition to the narrow range of expressed ligands Paneth cells express which we have inferred, we can further infer from the Paneth $\rightarrow$ ISC that they are strong. So these handful of ligands that they express such as Wnt3, REG/EGF and DLL4 directed almost exclusively at ISC neighbours are very specific and strong. This selective pattern is apparent in the works of Sato et all [13]. Sato states that crypt-like organoids form in $<$1% of wells when stem cells are cultured alone but this figure climbs to 10% around the time of reaching a 1:1 Paneth-to-Stem-Cell ratio is attained. Live imaging seen in Figure 19 shows that every new crypt bud sprouts precisely where a Paneth cell touches the stem-cell cluster.

Comparing these results to Figure 8, we see that Paneth cells look 'cold' in the overall matrix but are still essential. Since Paneth cells occur late in pseudotime, we only see activity in bins 8 and 9, where fully mature Paneth cells appear in clusters at the crypt base. Here, levels of Wnt3, EGF and DLL4 increase significantly and as a result, Paneth $\longleftrightarrow$ Paneth and Paneth $\rightarrow$ Goblets squares jump to the top of the scale. Paneth cells, although rare, use a small set of powerful ligands as they keep stem cells dividing and later stabilise secretory lineages. This pattern matches Sato's organoid experiment where adding just a few Paneth cells boosted stem-cell growth around tenfold.

Moreover, looking at Figure 10, we can see that as the Paneth cells mature in the gut crypt

near the stem cells, illustrated by bins 8 and 9, they turn out to express high raw interaction strengths.

6.3.2 Enteroendocrine Behaviours

Now turning our attention to the Enteroendocrine cells (EECs) in Figure 8, we notice that it is on average the darkest row. We interpret these darker squares as evidence for EECs expressing many ligand-receptor z-scores above the global average. This observation is readily explained by EEC biology as a single Enteroendocrine cell can express numerous peptide hormones therefore, it contains a larger ligand repertoire than the other cell types recorded. Enteroendocrine cells are predominantly present only in the earlier bins since a lot of these cells mature before a large portion of the TA cells have differentiated into terminal cells.

This computational pattern is supported by a recent paper by where we learn that without the EECs, the crypts acted like they were in a permanent 'fasting mode'[14]. In the EEC-deficient mice, stem-cell numbers increased drastically and the epithelial barrier weakened, and as panel J shows the animals' total energy expenditure shot up (red line in Figure 20), indicating that their entire metabolic system was running in overdrive.

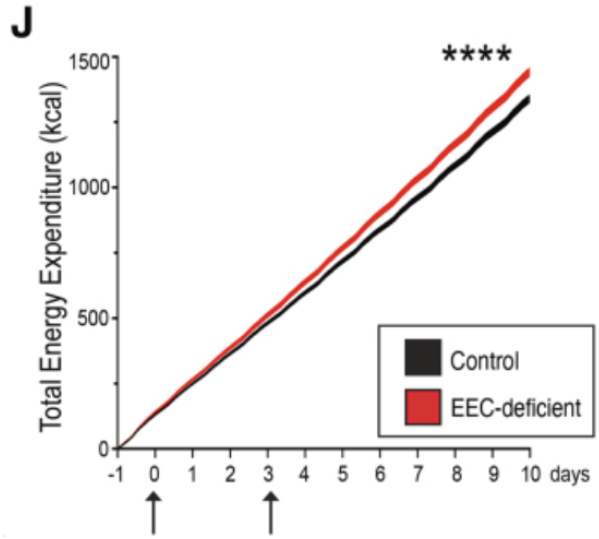


Figure 20 From Figure 1J of McCauley *et al.* [14]

The dark central square in Figure 8 showing strong EEC → EEC interaction in the heatmap could reflect autocrine feedback loops that allow individual EECs to self-regulate their hormone secretion. It could also reflect some paracrine signalling between neighbouring EECs to act out a more coordinated hormone release across the intestinal epithelium.

7 Conclusion

7.1 Discussion

In this paper, we made use of the scRNA-seq workflow to help us in understanding cell types within the intestine and their differentiation trajectories linking them. We also considered their interactions via ligand and receptors and looked into how they may affect cell fate decisions. By integrating in depth quality control, various trajectory-inference algorithms and finally the LIANA framework for ligand-receptor pair analysis, we managed to evaluate cell-cell interactions across pseudotime.

Our three main trajectory methods agree that there exists a main pathway that charges ahead to become the nutrient-absorbing enterocytes, while the other trajectories veer off into the secretory lineages (Paneth, Goblet, Tuft, Enteroendocrine). Moreover, we found that our global heatmap made Paneth cells look quieter on average, due to a lack of numerous unexpressed ligand-receptor pairs, but taking a closer look at the raw strengths in late-stage pseudotime bins revealed that there was high activity of both autocrine and paracrine Paneth interactions. Inspecting enteroendocrine cells provided us with interesting results on their importance in early pseudotime as shown in [14]. Since these cells harbour a large set of peptide signals, removing them would dismantle the interaction graph we constructed. This is due to the fact that over half of the high-confidence ligand-receptor pairs in our matrix originate from EECs and thus eliminating that source would erase those connections and leave the remaining cells with far fewer incoming signals.

7.2 Limitations

There are some limitations and assumptions in our approach that may influence the validity of our conclusions. In particular, since we could not directly measure protein levels or ligand secretion, we used gene expression counts from the scRNA-seq in their place while assuming that a high gene expression implies a high production of the corresponding receptor or ligand protein. Furthermore, some limitations in scRNA-seq data can affect the accuracy of these measurements. For example, gene dropout is where the copies of expressed genes are undetected thereby making it appear inactive and may be due to a low capture efficiency. Another limitation is gene length bias, which arises due to longer genes being more likely to be captured and detected than shorter ones. These factors could clearly distort the cell interaction signals.

Another limitation is that we did not have spliced and unspliced RNA counts, which are typically used to calculate RNA velocity. As a result, we had to apply a pseudo-time kernel by modifying the transition matrix to respect pseudo-time order. Pseudotime orders cells along a progression path based on their gene expression profiles, but it does not provide information about the speed or strength of those transitions. So it's a static ranking based on gene expression, rather than using instantaneous and dynamic velocities in continuous space. Therefore, using a pseudo-time kernel in CellRank to build a transition matrix is at most an estimation through a non-equilibrium Markov Process.

Lastly, our data does not contain any spatial information, so we cannot determine or infer what types of interactions are taking place, such as paracrine, autocrine, and juxtacrine signalling. This means we can only infer general interactions between cell types, without knowing how or where they occur.

7.3 Future Work

As demonstrated in the results section, we were able to find interesting inferences for Paneth cells and Enteroendocrine cells, and were able to closely explain their behaviour in the many diagrams presented in section 5. A direct follow-up of this would be to also examine goblet cells in more depth. In particular, we can silence a dominant mucus peptide and then re-measure LIANA scores and check whether these fluctuate. A clear decrease would indicate that mucus-related peptides are driving Goblet Cell interactions with other cell-types.

Moreover, one could train a machine learning model that predicts each cell's terminal fate using just its LIANA incoming signal profile. If the model has a high accuracy, we get strong evidence that cell-cell signalling is sufficient to predict lineage commitment. Conversely, a low accuracy would reveal that additional layers are required and cell-cell interactions are not solely responsible for determining cell fates.

Future work should test our top predicted interactions through wet lab experiments as these are currently only supported by current academic literature.

This forms a direction kNN graph where an edge $i \to j$ exists if $j \in \mathcal{N}_k(i)$

Acknowledgments

We performed our single-cell data preprocessing and analysis using the Scanpy Python package. The official Scanpy documentation can be found here at https://scanpy.readthedocs.io/. We performed trajectory inference with the Palantir Python package; its official documentation is available at https://palantir.readthedocs.io/. Fate-probability and absorption analyses were carried out using CellRank (https://cellrank.readthedocs.io/). Ligand–receptor inference and network construction were done with LIANA (https://liana-py.readthedocs.io/).

All analysis scripts, notebooks, and environment files are openly available in the project repository at https://github.com/ssohamsud/M2R.

We would like to thank Dr Omer Karin for his supervision and critical discussion throughout this project.

References

[1] Fabian A Wolf, Philipp Angerer, and Fabian J Theis. Scanpy: large-scale single-cell gene expression data analysis. *Genome Biology*, 19(1):15, 2018. doi: 10.1186/s13059-017-1382-0. URL https://doi.org/10.1186/s13059-017-1382-0.

[2] Bassem Khalil, Eric J. Miller, and Sarah L. Lappin. *Physiology, Cellular Receptors*. StatPearls Publishing, Treasure Island, FL, 2024. URL https://www.ncbi.nlm.nih.gov/books/NBK554403/. Last Update: September 19, 2024.

[3] Noah F Shroyer Taeko K Noah, Bridgitte Donahue. Intestinal development and differentiation. *Experimental Cell Research*, 317(19):2702–2710, 2011. doi: 10.1016/j.yexcr.2011.09.006. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC3210330/.

[4] Harry Beghtel Maaike M.W. van den Born Elena Sancho Gerwin Huls Jan Meeldijk Jennifer Robertson Marc van de Wetering Tony Pawson Hans Clevers Eduard Batlle, Jeffrey T. Henderson. ß-catenin and tcf mediate cell positioning in the intestinal epithelium by

controlling the expression of ephb/ephrinb. *Cell*, 111(2):251–263, 2002. doi: 10.1016/S0092-8674(02)01015-2. URL https://doi.org/10.1016/S0092-8674(02)01015-2.

[5] Leonard H. Augenlicht Jiahn Choi. Intestinal stem cells: guardians of homeostasis in health and aging amid environmental challenges. *Experimental Molecular Medicine*, 56: 495–500, 2024. doi: 10.1038/s12276-024-01179-1. URL https://doi.org/10.1038/s12276-024-01179-1.

[6] C. Cui, F. Wang, Y. Zheng, H. Wei, and J. Peng. From birth to death: The hardworking life of paneth cell in the small intestine. *Frontiers in Immunology*, 14, 2023. doi: 10.3389/fimmu.2023.1122258. URL https://doi.org/10.3389/fimmu.2023.1122258.

[7] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, oct 2008. doi: 10.1088/1742-5468/2008/10/P10008. URL https://dx.doi.org/10.1088/1742-5468/2008/10/P10008.

[8] Volker Bergen, Marius Lange, Stefan Peidli, F. Alexander Wolf, and Fabian J. Theis. Generalizing rna velocity to transient cell states through dynamical modeling. *Nature Biotechnology*, 38(12):1408–1414, 2020. doi: 10.1038/s41587-020-0591-3. URL https://doi.org/10.1038/s41587-020-0591-3.

[9] Philipp Weiler, Koen Van den Berge, Kelly Street, and Simone Tiberi. A guide to trajectory inference and rna velocity. In Raffaele A. Calogero and Vladimir Benes, editors, *Single Cell Transcriptomics: Methods and Protocols*, volume 2584 of *Methods in Molecular Biology*, pages 269–292. Humana, Springer, 2022. doi: 10.1007/978-1-0716-2756-3_14. URL https://doi.org/10.1007/978-1-0716-2756-3_14.

[10] Krzysztof Setty, Vytautas Kiseliovas, Jacob Levine, Adam Gayoso, Linas Mazutis, and Dana Pe'er. Palantir characterizes cell fate continuities in human hematopoiesis. *Nature Biotechnology*, 37(4):451–460, 2019. doi: 10.1038/s41587-019-0068-4. URL https://www.nature.com/articles/s41587-019-0068-4.

[11] Maximilian Lange, Volker Bergen, Marion Klein, Manu Setty, Benjamin Reuter, Mahdi Bakhti, Heiko Lickert, Mumin Ansari, Julian Schniering, Christian Schütt, Dominik S. Fischer, and Fabian J. Theis. Cellrank for directed single-cell fate mapping. *Nature Methods*, 19(2):159–170, 2022. doi: 10.1038/s41592-021-01346-6. URL https://doi.org/10.1038/s41592-021-01346-6.

[12] Laleh Haghverdi, Max Büttner, Filip J Wolf, Florian Buettner, and Fabian J Theis. Diffusion pseudotime robustly reconstructs lineage branching. *Nature Methods*, 13(10):845–848, 2016. doi: 10.1038/nmeth.3971. URL https://www.nature.com/articles/nmeth.3971.

[13] Hugo J. Snippert Daniel E. Stange Robert G. Vries Maaike van den Born Nick Barker Noah F. Shroyer Marc van de Wetering Hans Clevers Toshiro Sato, Johan H. van Es. Paneth cells constitute the niche for lgr5 stem cells in intestinal crypts. *Nature*, 469(7330):415–418, 2011. doi: 10.1038/nature09637. URL https://doi.org/10.1038/nature09637.

[14] Enriquez JR Zhang X Watanabe-Chailland M Sanchez JG Kechele DO Paul EF Riley K Burger C Lang RA Wells JM. McCauley HA, Riedman AM. Enteroendocrine cells protect the stem cell niche by regulating crypt metabolism in response to nutrients. *Cellular and Molecular Gastroenterology and Hepatology*, 15(6):1293–1310, 2023. doi: 10.1016/j.jcmgh.2022.12.016. URL https://doi.org/10.1016/j.jcmgh.2022.12.016.