Extending GeneTrajectory to a New Data Modality and Application

Abstract

Trajectory inference is a powerful framework for uncovering and understanding dynamic biological processes from single-cell omics data, but most methods struggle with datasets in which cells undergo multiple simultaneous biological processes. To address this, GeneTrajectory was recently developed to deconvolve gene programs associated with concurrent processes using optimal transport (OT) between gene distributions on a cell graph. In this work, we extend GeneTrajectory in two ways. First, we adapt the algorithm to single-cell ATAC-seq data, enabling the inference of accessibility peak trajectories. Second, we extend GeneTrajectory's OT-based algorithm to derive cell trajectories by modeling each cell as a distribution over genes and using the gene-gene OT distance matrix as the cost matrix. We benchmark both approaches against simpler alternatives—gene covariance and cell kNN graph distances—and show that OT-based trajectories better capture biologically meaningful structures in gene and cell space. Together, these results highlight the flexibility of OT-based trajectory inference across modalities and its potential to unify gene- and cell-level views of dynamic biological processes.

Problem Definition

Single-cell omics offer new opportunities to understand cellular dynamic processes. One way to study dynamic processes is through trajectory inference methods, which order cells in a dataset along a trajectory based on their similarities in gene expression.[1,2] Understanding cell trajectories is fundamental to biology and medicine - these trajectories reveal how stem cells differentiate into specialized cell types, how cells respond to disease and treatment, and how developmental processes unfold.

While there are many existing trajectory inference methods, they often struggle on datasets in which cells simultaneously undergo multiple processes, where a single latent variable cannot easily parameterize cell geometry. To address this, the authors of GeneTrajectory propose deconvolving independent processes in these cases by inferring gene trajectories, which are sequentially expressed gene programs, instead of cell trajectories.[3] GeneTrajectory would uncover a gene trajectory for every process. Specifically, GeneTrajectory learns gene trajectories by computing optimal transport (OT) distances between gene distributions across a cell KNN graph. If two genes are consecutively activated along a biological process, they are expected to overlap significantly in the cell graph and thus have a small OT distance from each other. Deconvolving simultaneously occurring processes is important because multiple processes usually occur in biology; for instance, cells mature (a linear process) while undergoing cell cycling (a cyclical process).[4]

While GeneTrajectory is a novel and creative method, the authors do not test whether GeneTrajectory can be applied to data modalities beyond single-cell RNA sequencing

(scRNA-seq), nor do they show how gene trajectories relate to cell-level trajectories usually obtained by trajectory inference methods. Therefore, we set out to extend GeneTrajectory to ATAC sequencing (scATAC-seq) data and explore if the same algorithm can be applied to yield cell trajectories.[5] We also benchmarked the trajectories we obtained using OT against simpler alternatives like k-nearest neighbors and covariance.

Methods (Algorithm & Implementation)

Data preprocessing

We want to extend Gene Trajectory to the new data modality of ATAC-seq. To do this, we identified a multimodal dataset from 10x Genomics that contains paired scRNA-seq and scATAC-seq data from peripheral blood mononuclear cells (PBMCs).[6] To process the raw data, we adapt the pipeline from the Muon tutorial that is specifically designed for the dataset.[7]

For the scRNA-seq component, we use Scanpy and Muon packages to build the preprocessing workflow.[7,8] It starts with quality control to ensure data integrity, in which it filters cells based on gene count thresholds (200-5,000 genes), total RNA content (under 15,000 counts), and mitochondrial percentage (below 20%). We also remove genes expressed in fewer than three cells. We then normalize the filtered data to 10,000 counts per cell and apply log-transformation, followed by identifying around 2,000 highly variable genes for downstream analysis.

Similarly, the scATAC-seq data starts with quality control filtering, removing cells with fewer than 2,000 or more than 15,000 peaks, and peaks detected in fewer than 10 cells. We then compute ATAC-specific quality metrics, including nucleosome signal and transcription start site (TSS) enrichment, to ensure the data exhibited expected biological patterns. Next, we normalize and log-transform the peak data similarly to the gene expression data to enable analysis between modalities.

For both data types, we perform dimensionality reduction using Principal Component Analysis (PCA) followed by Uniform Manifold Approximation and Projection (UMAP) for visualization.[9] Interestingly, we see that the principal components from both modalities showed similar biological patterns, with the first component separating myeloid (monocytes) and lymphoid (T, B, NK) cells and the second component further distinguishing B cells from other lineages. Next, we construct cell neighborhood graphs for both modalities and apply Leiden clustering to identify cell population in each modality separately. After removing clusters that may contain doublets or low-quality cells, we assign cell type labels based on canonical marker genes for RNA data and accessibility at marker gene loci for ATAC data. Both modalities identified similar cell populations, which includes various T cell subsets (CD4+ naive, CD4+ memory, CD8+naive, etc), NK cells, B cell subsets (naive and memory), and monocytes (subsets (CD14+, CD16+ and intermediate).

Obtaining gene trajectories

The GeneTrajectory algorithm obtains gene trajectories in four main steps.

First, PCA by default is used to reduce the dimensionality of the input cell-by-gene matrix and a cell-cell kNN graph is constructed in which each cell is connected to its kNNs. Using this kNN graph, the algorithm calculates the graph distance between cells as the shortest path length.

Second, the algorithm computes pairwise graph-based Wasserstein distance between gene distributions on the cell graph. Each distribution is first normalized by its expression across cells to create a probability distribution. The Wasserstein distance is then computed between gene distributions, where the cost of transporting between genes is determined by the graph distance found in the previous step. In practice, GeneTrajectory improves computational efficiency through graph coarse-graining, which involves aggregation into meta-cells, and gene graph sparsification, which skips OT computations for genes far apart from each other. However, instead of approximating using Sinkhorn iterations, the GeneTrajectory algorithm computes the exact Wasserstein distance with the Python Optimal Transport library.[10]

Next, the algorithm converts the gene-gene Wasserstein distance matrix to an affinity matrix using a local-adaptive Gaussian kernel and uses a random walk with the affinities to generate low-dimensional embeddings of genes. The algorithm then identifies gene trajectories using a similar diffusion-based approach that is described in more detail in their paper.

Lastly, GeneTrajectory orders genes along each trajectory by recomputing the diffusion map on the Wasserstein distance matrix of the genes that are the members of each trajectory, and then uses the first non-trivial eigenvector (EV2) of the diffusion map to order the genes based on the ranking of the coordinates along EV2.

Obtaining peak trajectories

To extend GeneTrajectory to chromatin accessibility data, we adapt the original algorithm to work with scATAC-seq peaks rather than gene expression. Specifically, we only make necessary modifications to accommodate the unique characteristics of accessibility data while preserving the core principles of the algorithm.

For peak selection, we implement a two-step filtering process. We first filter peaks based on their accessibility, retaining only those detected in 1-50% of cells, as peaks with broader accessibility often represent housekeeping regions, while the rest may represent technical noise. Second, we identified highly variable peaks using a modified version of Scanpy's highly_variable_genes function with parameters tuned for ATAC-seq data to account for the greater sparsity and binary nature of accessibility data compared to gene expression.

When preparing the accessibility matrix for Wasserstein distance calculations, while gene expression data is typically log-normalized, peak accessibility data works better with binary or count-based representations that preserve the nature of chromatin accessibility as an "open" or "closed" state, We therefore use a counts layter for coarse-graining and distance calculations rather than log-normalized values.

For the calculation of peak-peak Wasserstein distances, we adapt the algorithm's function to handle the sparsity patterns typical of accessibility by extracting non-zero accessibility before the distance calculation. We then use the same diffusion mapping and trajectory extraction algorithms from GeneTrajectory to extract the peak trajectories using default parameters.

Obtaining cell trajectories

To obtain cell trajectories, we compute OT distances between cell distributions analogously to steps 2-4 of the GeneTrajectory method outlined above. Specifically, we model each cell as a distribution over the gene graph, normalizing for each cell's total distribution. However, instead of constructing a kNN graph from the dimensionality-reduced dataset and using the shortest path distances as the cost matrix, we use the gene-gene graph Wasserstein distance matrix as our cost matrix. After we obtained a cell-cell OT distance matrix, we applied the trajectory extraction steps of GeneTrajectory using default parameters.

Benchmark with gene covariance

We benchmark OT distance between genes against gene-gene covariances as an alternative approach. We first compute gene-gene covariances from the original cell-by-genes counts data using Numpy's cov function.[11] We then directly use the covariance matrix as the affinity matrix on which we apply GeneTrajectory's steps 3-4 to extract trajectories. Trajectories are plotted with default parameters but rotated for the best view. Gene-gene covariance captures the degree to which two genes are expressed in similar cell populations, so we expect two genes with high covariance to be involved in related biological processes and be placed close to each other in a gene trajectory.

Benchmark with naive cell trajectories

We compare the refined cell-cell distance matrix obtained using OT against the initial cell-cell kNN graph distance matrix. Specifically, we extract naive cell trajectories by applying steps 3-4 on the cell-cell kNN graph distance matrix. Trajectories are plotted with default parameters but rotated for the best view.

Results

Gene Trajectory can be robustly applied to a new scRNA-seq dataset

Before extending the algorithm to ATAC-seq, we validated it by applying it to the scRNA-seq part of the 10x PBMC multiome data.[6] Because the Gene Trajectory tutorial uses a myeloid dataset with four myeloid cell subtypes (Fig. 1a), we subsetted our dataset to the same cell types (Fig. 1b). As expected, the algorithm extracted three trajectories similar to those from the tutorial dataset (Fig. 1c and d).

To look more closely at how the gene trajectories are reflected in the cell graph, we visualized the four marker genes provided by the tutorial in our gene trajectories and cell UMAPs. We saw good correspondence between their positions in the trajectories (Fig. 1c and d) and their

expected regions of high expression in cell UMAP (Fig. 1e and f). From this we conclude that, as the authors of GeneTrajectory noted, each trajectory represents one aspect of the myeloid lineage differentiation process. Genes such as *CLEC5A, RETN, CCR2* and *SELL* are known to be associated with the initial CD14+ monocyte state and are grouped into Trajectory 1. Trajectory 2 includes genes such as *FCGR3A* and is associated with CD16+ monocyte differentiation. Trajectory 3 contains genes such as *CD1C* and *PKIB* which are highly expressed in CD14- myeloid type-2 dendritic cells.[12]

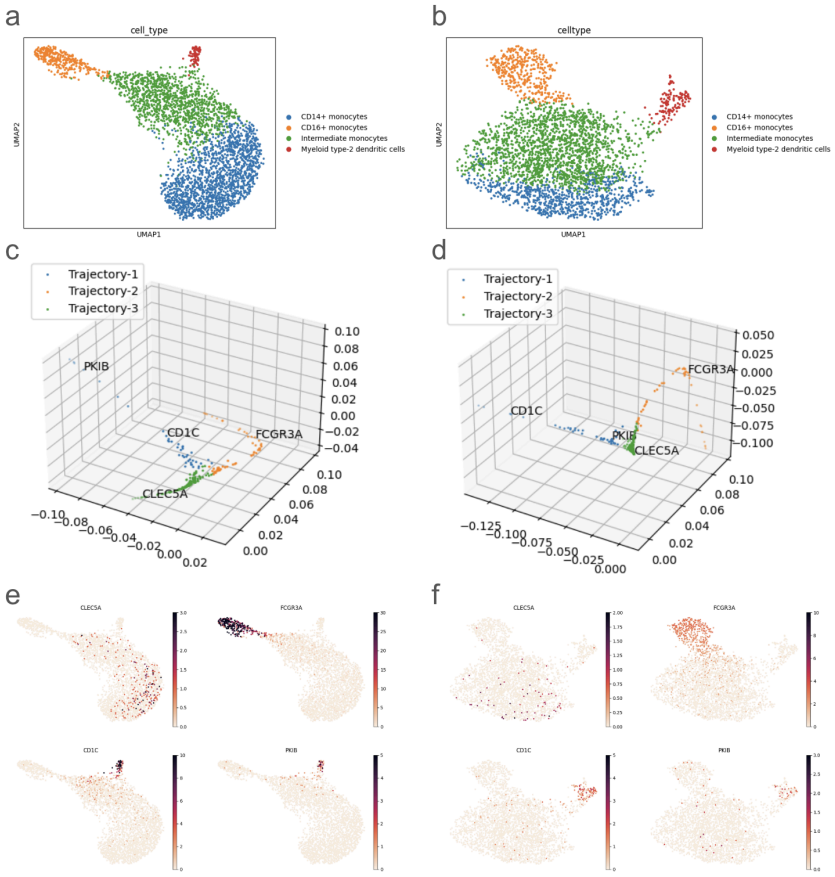

Fig. 1: Gene Trajectory can be robustly applied to a new scRNA-seq dataset

OT-based gene trajectories show improvement over covariance-based trajectories

The authors of Gene Trajectory propose computing OT between gene distributions on the cell graph, then using the gene-gene OT distance matrix to obtain gene trajectories. However, they do not benchmark their approach against simpler alternatives to obtain gene-gene distance matrices. To determine if OT offers advantages that are otherwise unattainable, we set out to perform this benchmark.

An intuitive alternative is to compute a gene-gene covariance matrix, on which we can then apply the later steps of GeneTrajectory to extract trajectories. Gene-gene covariance captures

how often two genes tend to be up- or down-regulated together in their expression across cells, so we hypothesized it could serve as a proxy for what OT can capture without the intensive computation. We performed this benchmark on myeloid scRNA-seq data from both the Gene Trajectory tutorial and the 10x paired dataset, obtaining similar results. We show the results on the 10x paired dataset below.

As we can see in Fig. 2a and b, the trajectories we obtain using gene covariance lack the structure of the trajectories obtained using OT. The marker genes are also generally collapsed onto each other, except for *FCGR3A* (Fig. 2b). To further analyze the results of using gene covariance, we identified groups of genes that are near each other in the unstructured trajectories and visualized the expression of those genes in the original cell UMAP (Fig. 2c and d). In agreement with our observations in Fig. 2b, genes highly expressed in CD16+ monocytes were identified through computing covariance and mapped closely to each other (Fig. 2c). Another group of extracted neighbor genes seem to mark CD14+ and intermediate monocytes (Fig. 2d). In contrast, genes for myeloid type-2 dendritic cell differentiation were lost in the trajectory extraction process. Overall, trajectories computed with OT show significant advantages over the trajectories computed with gene covariance.

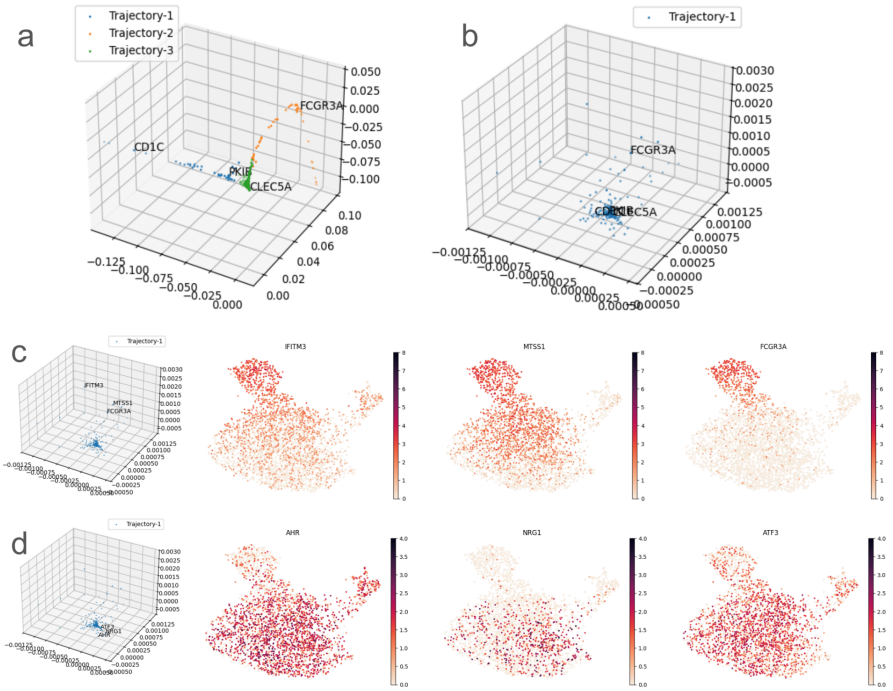

Fig. 2: OT-based gene trajectories show improvement over covariance-based trajectories

Extending the algorithm to ATACseq data yields peak trajectories

After validating the paired data by applying the scRNA-seq component to extract the gene trajectories, we then applied the modified pipeline to extract peak trajectories from the

scATAC-seq data. As shown in Figure 3, Fig. 3a displays the gene trajectories extracted from the scRNA-seq data, while Fig. 3b shows the extracted peak trajectories from the corresponding scATAC-seq data. Unfortunately, we did not observe the same distinct trajectories in the peak trajectories as we did in the gene trajectories – we suspect this is due to the noisy nature of scATAC-seq data. However, they do exhibit a similar pattern, starting from the same region and then branching out into two main trajectories.

To further investigate whether the peak trajectories truly extracted the correct trajectories and to facilitate easier comparison with gene trajectories, we chose three peaks, each at the end of the trajectories, and highlighted the cells that have accessible peaks on the celltype UMAP (Fig. 3d). Similarly, for the gene trajectories, we selected three genes at the end of each trajectory and highlighted the cells that express the genes on the cell UMAP (Fig. 3c). In Fig. 3c, we see that the highlighted cells are closely distributed in the region where the cell types are located, indicating that these genes correspond to certain cell types and that the gene trajectory successfully captured the sequential changes in genes – for example, CKB is mostly activated in CD16 monocytes, and FCER1A is mostly activated in myeloid type-2 dendritic cells.

The highlighted cells for each peak, as shown in Fig. 3d, are more diffuse and less conclusive compared to Fig. 3c. However, we do see that the peaks at the end of the trajectories are mostly detected in a certain cell type. For example, peaks ch3:143294126-143294795 that are at the end of Trajectory 1 are mostly detected in CD16+ monocytes, and most of the cells that have peak chr16:56121044-56121502 detected are myeloid type-2 dendritic cells, indicating the trajectory indeed orders the peaks according to cell differentiation. Nevertheless, since the chosen peaks are detected in fewer cells, the overall distribution appears more spread out, and further analysis needs to be done before concluding that the peak trajectories have captured the sequential changes in the accessibility of regulatory elements that regulate cell differentiation.

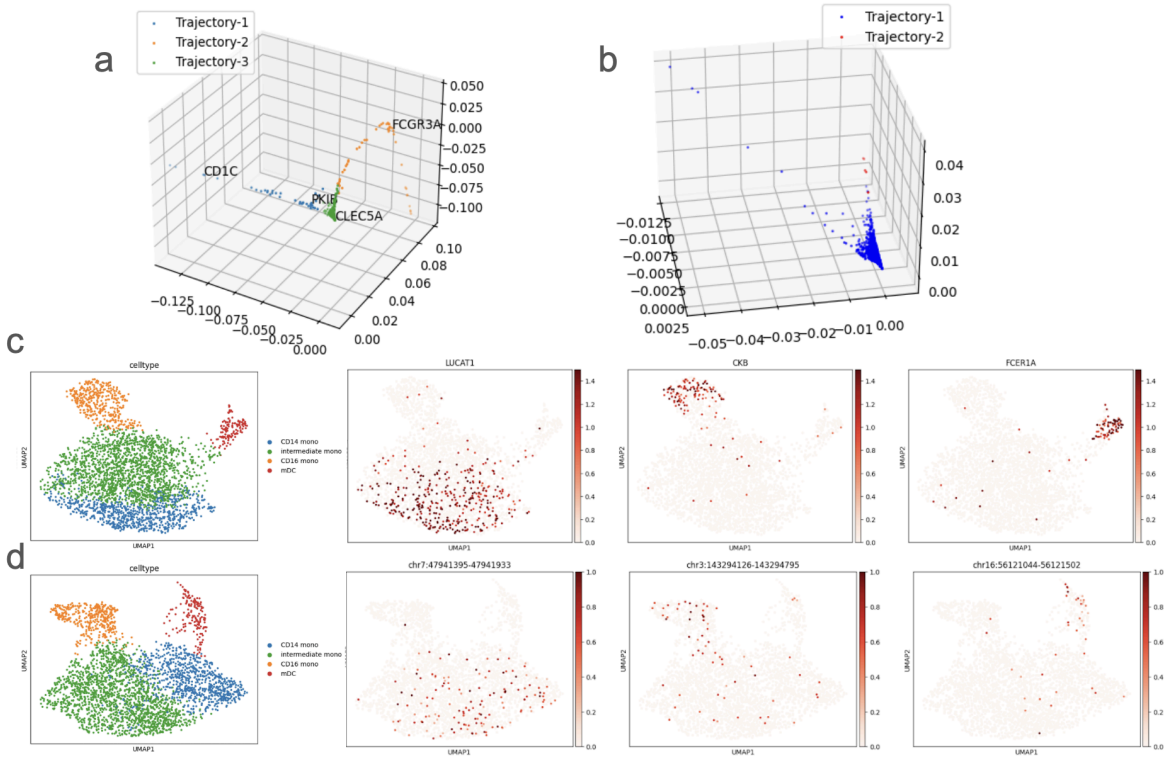

Fig. 3: Extending the algorithm to ATACseq data yields peak trajectories

Cell trajectories can be obtained from gene trajectories

Cell trajectory inference and gene trajectory inference complement each other to tackle different types of biological questions. While gene trajectories reveal the sequence of genes associated with a biological process, at times we might want to understand how individual cells in a dataset relate to each other and understand biological processes in terms of lineages of cells.

With this motivation, we set out to explore whether we could extract cell trajectories using a similar OT-based approach. In particular, we model cells as distributions over genes by normalizing each cell by its total expression. We then use as the cost matrix the gene-gene graph Wasserstein distance matrix obtained with GeneTrajectory and compute OT again between cell distributions. We then use GeneTrajectory's diffusion map function to extract trajectories from the cell-cell OT distance matrix. This approach assumes that cells from adjacent pseudotime states will have a small OT distance from each other. This is biologically feasible because cells of similar or adjacent states should have similar distributions over the gene graph, indicating that they express the same genes or genes consecutively involved in the same processes.

As Fig. 4a shows, the extracted cell trajectories exhibit an overall branching structure that corresponds to the bifurcation of CD14+ monocytes to differentiate into either CD16+ monocytes or myeloid type-2 dendritic cells. We can further obtain an ordering of cells along any given trajectory by computing a diffusion map embedding of cells in that trajectory and taking

the first nontrivial eigenvector. This ordering indicates where each cell is in the differentiation process. We respectively visualized the orderings for myeloid type-2 dendritic cells (Fig. 4b) and CD16+ monocytes (Fig. 4c) on the cell UMAP.

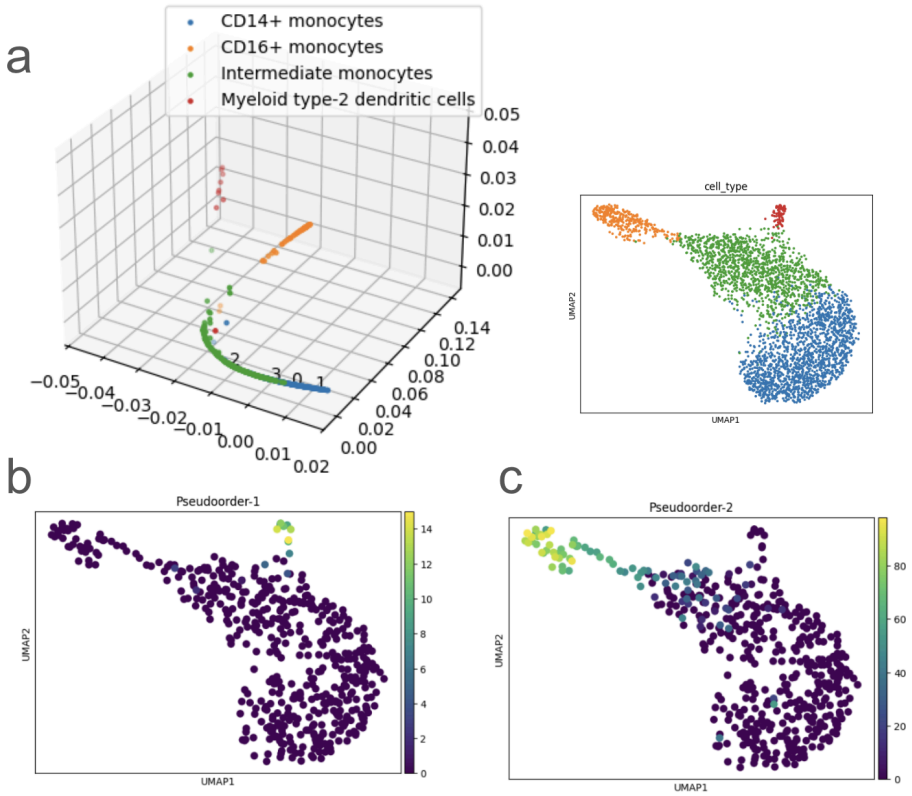

Fig. 4: Cell trajectories can be obtained from gene trajectories

OT-based cell trajectories show advantages over kNN-based trajectories

Theoretically, we could have extract cell trajectories from the initial cell-cell kNN graph distance matrices that were used as input to the GeneTrajectory algorithm. Thus, we benchmarked our OT-based cell trajectories against these “naive” trajectories. While modern trajectory inference algorithms are more sophisticated, many involve kNN graph construction as a first step, making our benchmarking experiment relevant.[1,13]

When comparing the naive trajectories (Fig. 5a) side-by-side with the OT-based trajectories (Fig. 5b), we can see that the OT-based approach produces trajectories that better separate cells from different differentiation stages. The difference arises because the OT-based approach penalizes transitions between cells that activate unrelated gene programs, even if their overall expression levels are similar. In contrast, the naive approach relies solely on local neighborhood structure and is sensitive to the cell type proportions of the dataset. For instance, CD14+ and intermediate monocytes are not well separated in the naive trajectory (Fig. 5a) because the

large number of cells of those cell types likely causes the shortest distance between those cell types to be reduced in the kNN graph.

To further compare the two approaches for trajectory inference, we visualized the final diffusion map embedding distance between an anchor in the dataset with all other cells in the dataset on the cell UMAP. Results for one CD16+ monocyte anchor cell are shown in Fig. 5c and d. We see that the kNN-based results show the anchor cell being close in distance to even some CD14+ dendritic cells, again likely due to the high number of CD14+ and intermediate monocytes (Fig. 5c).

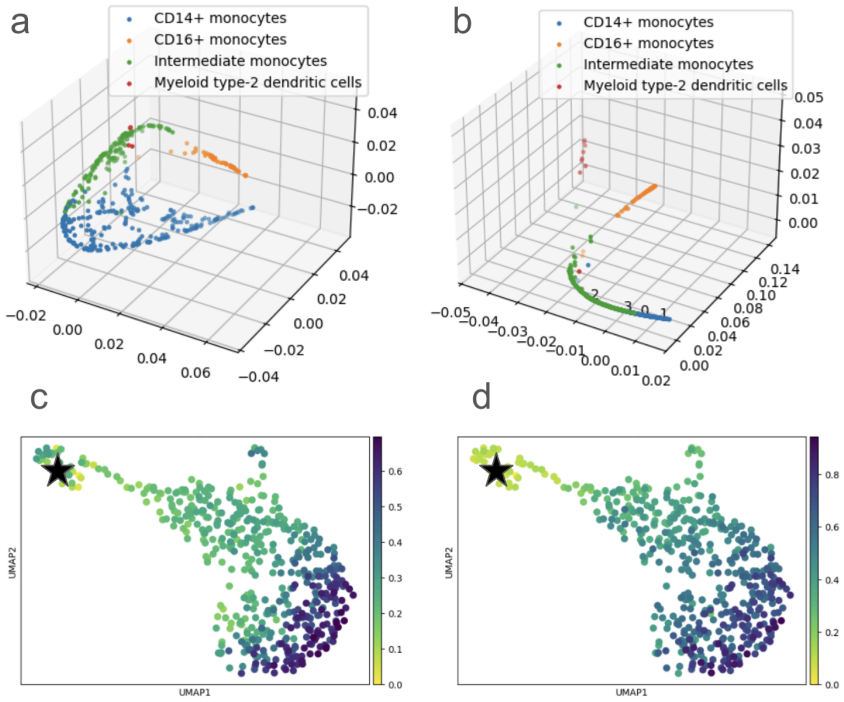

Fig. 5: OT-based cell trajectories show advantages over kNN-based trajectories

Discussion

In this work, we demonstrate that GeneTrajectory is applicable to data modalities beyond scRNA-seq and can be extended to infer cell trajectories. Additionally, we benchmarked the trajectories inferred by our extended GeneTrajectory against simpler approaches, such as k-nearest neighbors and covariance-based methods.

We adapted the GeneTrajectory pipeline to scATAC-seq data and extracted the peak trajectories, with the hope of identifying sequential changes in regulatory element accessibility that drive cell differentiation, since chromatin accessibility changes often precede gene expression changes. We made necessary modifications to the GeneTrajectory codebase and applied it to the 10X PBMC multiome dataset that has paired scRNA-seq and scATAC-seq data, extracting both gene trajectories and peak trajectories for the new data. While the gene

trajectories were highly similar to those from the paper, the peak trajectories were not, although both shared certain patterns. It is worth noting that further analysis needs to be done before concluding that the peak trajectories have correctly identified the sequential chromatin accessibility that drives cell differentiation in this data, as the number of cells that have detected for the peaks is substantially less than the number of cells that express certain genes.

While gene trajectories can elucidate the gene programs underlying biological processes, cell trajectories are useful for applications where we want to see how cells in a sample relate to one another. Thus, we explored extending the algorithm to compute cell trajectories on top of gene-gene OT distances. Our results suggest that OT-based cell trajectory inference is more effective than the naive kNN-based approach. We believe OT-based cell trajectories are more biologically meaningful because they utilize precomputed gene-gene OT distances, which capture gene-gene relationships. With this approach, two cells expressing genes from unrelated processes will have a greater OT distance from each other than two cells expressing consecutive genes from the same process, despite the cells potentially differing in expression by the same number of genes in the two cases.

While modern trajectory inference tools like PAGA and Slingshot are more sophisticated than the kNN benchmark we perform, they similarly treat genes as independent features before dimensionality reduction.[1,13] This means that all genes are considered to be equally distant, regardless of the underlying gene-gene relationships and whether genes are functionally or temporally related in biological processes.

Our work demonstrates the flexibility of OT for extracting relationships between genes, peaks, or cells in single-cell omics datasets. There are many opportunities for future work. One direction we began to explore but did not focus on is taking an iterative approach to refining and cell and gene graphs, where the OT distance matrix of one is fed into the GeneTrajectory algorithm to obtain a matrix of the other, which could potentially give us more accurate gene trajectories. Another way to potentially improve the inferred gene trajectories is to incorporate biological priors, such as gene regulatory networks or known transcription factor-enhancer pairs, when we construct the cost matrix to bias the model towards learning known gene-gene interactions.

Moreover, while we demonstrated that peak trajectories extracted from scATAC-seq data share similar patterns with gene trajectories from scRNA-seq, a more integrated approach could align the two modalities directly – for example, by computing OT distances between genes and peaks or between cells across modalities, enabling joint trajectory inference. Additionally, more principled feature selection strategies beyond basic filtering could potentially improve the robustness of the algorithm, especially in sparse modalities like scATAC-seq, where signal-to-noise ratios are lower. Finally, we could achieve a richer biological interpretation of the trajectory space by explicitly linking accessibility pears to their putative target genes, allowing us to trace the temporal dynamics of expression and accessibility separately with ease.

Contributions

E.C. and S.Y. conceived the project and coded the implementation. S.Y. performed preprocessing on the scATAC-seq dataset and obtained results for the cross-modality adaptation. E.C. benchmarked the algorithm against alternative approaches and obtained results for the cell trajectory inference extension. E.C. and S.Y. wrote the report.

References

1. Street, K. et al. Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. *BMC Genomics* 19, 477 (2018).
2. Van den Berge, K. et al. Trajectory-based differential expression analysis for single-cell sequencing data. *Nat. Commun.* 11, 1201 (2020).
3. Qu, R., Cheng, X., Sefik, E. et al. Gene trajectory inference for single-cell data by optimal transport metrics. *Nat Biotechnol* 43, 258–268 (2025).
4. Ruijtenberg, S. & van den Heuvel, S. Coordinating cell proliferation and differentiation: antagonism between cell cycle regulators and cell type-specific gene expression. *Cell Cycle* 15, 196–212 (2016).
5. Pott, S. & Lieb, J. D. Single-cell ATAC–seq: strength in numbers. *Genome Biol.* 16, 172 (2015).
6. 10x Genomics, Single Cell Multiome ATAC + Gene Exp. Dataset by Cell Ranger ARC 1.0.0. (2020)
7. Danila Bredikhin, Kats, I. & Stegle, O. MUON: multimodal omics analysis framework. *Genome biology* 23, (2022).
8. Wolf, F. A., Angerer, P. & Theis, F. J. SCANPY: large-scale single-cell gene expression data analysis. *Genome biology* 19, (2018).
9. McInnes, L., Healy, J., Saul, N. & Lukas Großberger. UMAP: Uniform Manifold Approximation and Projection. *The Journal of Open Source Software* 3, 861–861 (2018).
10. Rémi Flamary *et al.* POT: Python Optimal Transport. *Journal of Machine Learning Research* 22, 1–8 (2021).
11. Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, et al. Array programming with NumPy. *Nature*. 585:357–62. (2020)
12. Patel, A. A. et al. The fate and lifespan of human monocyte subsets in steady state and systemic inflammation. *J. Exp. Med.* 214, 1913–1923 (2017).
13. Wolf, F. A. et al. PAGA: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. *Genome Biol.* 20, 59 (2019).