HelmholtzZentrum münchen
German Research Center for Environmental Health

Modeling dynamical biological processes through the lens of single-cell genomics

Marius Lange

May 2022

TECHNISCHE UNIVERSITÄT MÜNCHEN

TUM School of Computation, Information and Technology

Modeling dynamical biological processes through the lens of single-cell genomics

Marius Lange

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzende: Prof. Dr. Donna Ankerst

Prüfer*innen der Dissertation:

1. Prof. Dr. Dr. Fabian J. Theis
2. Prof. Dr. Dana Pe'er
3. Prof. Dr. Samantha Morris

Die Dissertation wurde am 02.05.2022 bei der Technischen Universität München eingereicht und durch die Fakultät für Mathematik am 02.08.2022 angenommen.

Acknowledgment

I want to thank everyone who helped me to accomplish writing this thesis. First of all, I would like to thank my supervisor Fabian Theis for teaching me how science works. You were a fantastic supervisor; I had a great time at the ICB! Further, thank you, Carsten Marr and Massimo Fornasier, for giving me valuable feedback in the thesis advisory committee meetings. I want to thank my thesis examination committe, Fabian Theis, Dana Pe'er and Samantha Morris, for carefully reading and evaluating my thesis, and Donna Ankerst for chairing the committe.

When I started in the lab, I was lucky to share an office with the best office members: Mo, Volker, and Subarna. I'm thankful for the time we spent together. I would like to thank all current and past members of the theislab for being such a friendly and fantastic crowd. In particular, thanks to my colleagues and friends Leander (for making every Wednesday unforgettable), Michal, Giovanni, Leon, Laura, Philipp, David, Sophie, Lisa, Niklas, Manuel, Adriana, Dominik, Maren, Malte, and Hananeh. I cannot thank the ICB office enough for helping me through the jungle of German academia; thank you, Sabine, Anna, Marianne, Elisabeth, and Dani (also for the cake!); you make every "Dienstreiseabrechnung" much easier to bear. Further, I would like to thank the QBM lecturers and office, Markus, Filiz, Julia, and Mara, for helping me to get started with this Ph.D. and teaching me the basics of biology. Thank you also, Helmholtz Mensa, for making an effort to be more veggie-friendly and for (sometimes) giving me a larger portion.

I was fortunate to go on two lab exchanges in this Ph.D.; one to the Dana Pe'er lab in New York and one to the Mor Nitzan lab in Jerusalem. Thank you, Dana and Mor, for making this possible and providing so much scientific guidance and supervision; I truly learned a lot from you. Tal, thank you for teaching me how to write beautiful papers! Further, thank you to all the Pe'er and Nitzan lab members for hosting me so nicely, including Ale, Joe, Manu, Sanjay, Cassandra, Doron, and Yubin from the Pe'er lab and Zoe, Noa, Yotam, and Matt from the Nitzan lab. Thank you to Souf, Raphael, and Daniel for introducing me to Jerusalem and Nati for feeding me. I would like to thank Zoe separately for helping me find a place to stay in Jerusalem and visiting our lab in Munich, and Ale for letting me sleep on his couch and for our wholesome Hell's kitchen experience.

Computational modeling in biology would not be fun without collaboration partners; thank you to everyone I collaborated with, including Herbert, Janine, Meshal, Bernhard, and Mostafa. I would further like to thank everyone who took their time to discuss science with me, in particular, Jens Timmer and my friend Georg Diez. Thank you to my graduate school QBM, the Munich Center for Machine Learning, the Joachim Herz Foundation, and

the Bayer Foundation for financially supporting my Ph.D. and lab exchanges.

I cannot possibly thank my family and friends enough for supporting me throughout this Ph.D. Mum, Dad, and Rosa, I could not have done it without you. Thank you, Charlotte, Hermann, Eitel-Friedrich, and Rosemarie, for supporting me throughout my studies leading up to this thesis. Sportsfreunde Maxvorstadt and Sendling (aka Sven), thanks for the great workouts. Thank you to all my other friends and flatmates who supported me throughout this process including Anja, Sofie, Marlene, Xiaoxiao, Lu, Hannah, Jonathan, Uli and Wolfgang. Lulu, thank you for being such a loving partner and always being with me throughout this Ph.D.; it means a lot to me.

Abstract

Cells dynamically change their molecular state in many situations, including development, the cell cycle, the circadian rhythm, and regeneration. Single-cell assays allow us to describe this state with unprecedented resolution; for example, single-cell RNA sequencing (scRNA-seq) quantifies the expression level of all genes. However, cells are destroyed upon sequencing, making it difficult to use these assays to study continuous processes that ideally require us to measure the same cell a few times. This fundamental difficulty has fueled the development of many mathematical methods that use ensembles of single cells at different internal states to reconstruct the average trajectory of a "typical cell," a concept known as *trajectory inference*. Recent experimental innovations, including RNA velocity [1] and lineage-tracing assays, provide new opportunities to improve trajectory inference; however, their harmonization with established modeling paradigms presents new mathematical challenges that need to be addressed.

The first innovation we considered in this thesis is RNA velocity [1]; a strategy to estimate the direction of expression changes based on the ratio of nascent versus mature transcripts. Essentially, RNA velocity approximates a high-dimensional vector field in the state manifold which points in a cell's future direction. We introduced the CellRank framework and showed how it combines RNA velocity with expression similarity into a Markov chain to robustly estimate initial and terminal states of cellular state changes as well as fate probabilities and driver genes. Applied to regeneration and reprogramming data examples, we showed how CellRank generalizes trajectory inference beyond normal development, a setting that most previous methods were limited to. The assumptions of the RNA velocity model do not hold in every biological system; in an effort to make CellRank widely applicable, we extended it towards other estimates of directed differentiation, thus transforming CellRank into a unified framework for single-cell fate mapping.

The second innovation we considered is lineage-traced scRNA-seq data which simultaneously contains molecular state and clonal history. We introduced `moslin`, an optimal-transport-based method to efficiently combine lineage with gene expression information to obtain more accurate couplings of cells across time points. On simulated data, we confirmed that this strategy recovers ground-truth couplings more accurately compared to methods that only use gene expression or lineage information and competing methods that combine both sources of information. Applied to *C. elegans* developmental data, we showed how a combination of `moslin` with CellRank recovered known decision driver genes. For time-course data with only gene expression information, we greatly accelerated previous approaches and made them applicable to much larger datasets. `moslin` is part of moscot, a new spatio-temporal framework for scalable optimal transport applications in single-cell

genomics.

CellRank and `moslin` extend trajectory inference towards new experimental approaches and massive datasets. They enable gaining a deeper understanding of dynamical processes in biology based on single-cell genomics assays.

Zusammenfassung

Zellen ändern ihren molekularen Zustand dynamisch in vielen Situationen, unter anderem in der Entwicklung, dem Zellzyklus und in Regenerationsprozessen. Einzelzell-Methoden erlauben uns, diesen molekularen Zustand detailgetreu zu messen; zum Beispiel kann die Einzelzell-Sequenzierung (scRNA-seq) die Expression jedes Gens messen. Allerdings werden Zellen in diesem Prozess zerstört, dies erschwert die Anwendung von Einzelzell-Methoden zur Untersuchung kontinuierlicher Prozesse. Hierzu würde man die gleiche Zelle gerne mehrere Male vermessen. Dieses fundamentale Problem hat zur Entwicklung vieler mathematischer Methoden beigetragen, welche aus Zellen in unterschiedlichen Zuständen eine "typische" Zell-Trajektorie rekonstruieren. Dieses mathematische Konzept wird als "Trajektorien-Rekonstruktion" bezeichnet. Neue experimentelle Errungenschaften erweitern die Möglichkeiten der Rekonstruktion, allerdings schafft die Integration von neuartigen Datenmodalitäten mit bestehenden Modelierungskonzepten mathematische Herausforderungen, für die Lösungen gefunden werden müssen.

Die erste experimentelle Neuerung, mit der sich diese Arbeit beschäftigt, ist “RNA velocity”; eine Strategie, mit welcher die Änderung der Genexpression aufgrund der Ratio von neuer zu alter mRNA geschätzt werden kann. Wir stellen CellRank vor und zeigen, wie mit dieser Methode ein Markov-Prozess aus RNA velocity und Ähnlichkeit in der Genexpression aufgestellt werden kann. Mithilfe dieses Markov-Prozesses schätzen wir die Anfangs- und Endzustände zellulärer Prozesse sowie die Entscheidungswahrscheinlichkeiten und Entscheidungsgene. Wir zeigen anhand von Regenerations- und Reprogrammierungsbeispielen, dass CellRank in Situationen jenseits normaler Entwicklung angewandt werden kann; dies war mit bisherigen Methoden größtenteils nicht möglich. Die Annahmen hinter RNA velocity gelten nicht in jedem biologischen System, um CellRank dennoch vielseitig einsetzen zu können erweitern wir die Methode mit anderen Strategien zur Abschätzung gerichteter Differenzierungsprozesse. Dadurch wird CellRank zu einer allgemeinen Methode um Entscheidungsprozesse in Einzelzellen zu untersuchen.

Die zweite experimentelle Neuerung, mit der sich diese Arbeit beschäftigt, sind Ansätze, welche Genexpression und Abstammung gleichzeitig in Einzelzellen messen. Wir stellen `moslin` vor und zeigen, wie diese Methode Optimalen Transport anpasst, um effektiv Abstammungs- mit Genexpressionsdaten miteinander zu kombinieren; dies ermöglicht uns, genauere Verknüpfungen von Zellen über experimentelle Zeitpunkte hinweg zu rekonstruieren. Wir zeigen auf simulierten Daten, dass unser mathematisches Modell besser funktioniert als alternative Strategien die entweder nur Abstammungs-, nur Genexpressionsdaten oder eine Mischung aus beidem verwenden. Auf echten C. elegans Daten demonstrieren wir, wie `moslin` und CellRank miteinander kombiniert werden können um Entschei-

dungsgene zu finden. Für Zeitreihen-Datensätze mit ausschließlich Genexpressionsinformationen stellen wir eine neue Implementierung bereit, welche deutlich schneller ist als vorhergehende Methoden und somit auf erheblich größere Datensätze angewandt werden kann. Wir integrieren `moslin` in moscot, unserem neuen Software Paket für skalierbare Anwendungen von Optimalem Transport in der Einzelzellgenomik.

CellRank und `moslin` erweitern die Möglichkeiten der Trajektorien-Rekonstruktion hin zu neuen experimentellen Ansätzen und größeren Datensätzen. Die Methoden ermöglichen tiefere Einblicke auf dynamische Prozesse in der Biologie, welche mithilfe von Einzelzellgenomik vermessen wurden.

Contents

Chapter 1

Introduction

The cell is the fundamental unit of life, and many processes in biology unfold at the level of individual cells. Thus, to gain new insights into how cells develop, make decisions in health and disease and react to external stimuli, we need to measure the molecular properties of individual cells. For a long time, technical limitations made this difficult: individual molecules like proteins or messenger RNAs (mRNAs) are small and usually available at low copy numbers only, complicating their detection in single cells. The situation started to change in 2009 when Tang et al. [2] introduced single-cell RNA-sequencing (scRNA-seq) and applied it to 9 cells. Since then, there has been a "single-cell revolution"; advanced molecular amplification and single-cell isolation techniques increased cellular throughput and sensitivity while lowering costs. Current studies sequence up to 4M single cells [3].

While initially focused on gene expression, single-cell technologies have been extended to the epigenetic, proteomic, and even multi-modal settings. These extensions have led to numerous biological insights, including a better understanding of SARS-CoV-2 [4, 5] and embryogenesis [6, 7]. Large international consortia, including the human cell atlas [8] (HCA) and the LifeTime initiative [9], make use of single-cell technologies to further our understanding of the cellular makeup of the human body and the dynamical processes underlying health and disease, respectively.

While single-cell technologies have advanced in several ways, they still share the common limitation that cells are destroyed upon sequencing. This is problematic as many processes in biology are continuous, for example, the development of hematopoietic stem cells towards differentiated immune cell types [10] or the regeneration of lung epithelial cells after injury [11]. Ideally, we would like to repeatedly measure the molecular state of the same cell while it undergoes such a process. This would enable us to link early molecular differences within a cell population to eventual fate outcomes and allow us to pinpoint the moment at which fate decisions are established. However, single-cell technologies yield static snapshots rather than trajectories of molecular state.

Since the early days of scRNA-seq, computational approaches have been developed to piece together static snapshots into trajectories; this problem is known as trajectory inference [12,

13] (TI). The basic idea is that most biological processes unfold unsynchronized across cells; thus, while we only have access to one observation per cell, these observations represent different positions along an underlying trajectory. Aligning static molecular snapshots of many different cells along the latent trajectory structure therefore allows us to reconstruct the dynamics of a typical cell. While many computational methods have been suggested for this task, most of them fall short of recovering the direction of the trajectory and thus remain limited to relatively simple scenarios where the direction is known a priori. This thesis introduces two new computational methods, CellRank [14] and `moslin` [15], which address this challenge by combining classical trajectory inference ideas with new sources of directionality and lineage information; in particular, CellRank includes RNA velocity [1, 16] and `moslin` includes lineage-tracing data [17].

1.1 Dynamics from cross-sectional single-cell data

Single-cell technologies yield measurements of cellular state, including gene expression, chromatin accessibility, protein availability or clonal identity, and sometimes even combinations of different molecular layers in multi-modal assays. Computational methods for trajectory inference compare these cellular states; if two cells A and B are similar in molecular state, they are placed close to each other on the reconstructed trajectory, reflecting the assumption that biological processes unfold gradually in small steps [18]. However, such similarity-based approaches do not reveal whether cell A goes to B or B goes to A. In well-studied systems, this is known and the trajectory can be manually directed by providing a root cell that signals the algorithm where the trajectory should start. In contrast, there exist many less-well studied clinically relevant systems such as reprogramming, regeneration, or cancer, where the direction is unknown and single-cell sequencing combined with classic TI provides limited insights only.

Alternative approaches exist to uncover directionality; these are based on cell-intrinsic properties or on time-series experiments. For example, the central dogma of molecular biology states that DNA is converted into RNA which in turn is converted into protein, with intermediate processing steps [19]. Thus, if molecular information for at least two stages in this sequence of processing steps is observed in the same cell, we can compare corresponding quantities to predict the future cellular state over a short time scale. RNA velocity is such an approach; it compares the amount of mRNA at two different processing stages [1, 16]. Alternatively, in time series experiments, it is reasonable to assume that cells from earlier time points should be placed before cells from later time points, thus introducing directionality [20]. We briefly review both approaches below and highlight

their current shortcomings.

1.1.1 RNA Velocity estimates the current state of gene regulation

In normal development, during the cell cycle and in disease, cells adapt their molecular state according to external signals or internal needs by up or down-regulating the expression of specific genes. scRNA-seq measures how many mRNA molecules corresponding to each gene exist in a given cell; however, it does not reveal whether a gene is currently up or down-regulated. RNA velocity compares corresponding mRNA levels in earlier and later processing stages to reveal the current state of gene regulation [1, 16]. In particular, it posits a simple biophysical model for immature (unspliced) molecules $u(t)$ and mature (spliced) molecules $s(t)$; the model is fitted to observed counts, and RNA velocity is defined as the time derivative of spliced molecules, i.e., $v^{(\mathrm{RNA})} = ds/dt$.

This elegant approach yields directional information at no additional experimental burden; however, the information is noisy, very high dimensional and reliable only over short time scales. It is currently an open question how to combine RNA velocity with gene expression similarity to robustly uncover directed, high-dimensional trajectories that reflect cellular fate choice's stochastic nature.

1.1.2 Single-cell lineage tracing recovers clonal relations and gene expression

Time-series experiments introduce directionality to cross-sectional data; on average, cells in earlier time points correspond to earlier biological process stages. To match cells from earlier to later time points, computational methods have been developed that successfully recover trajectories if expression distributions across adjacent time points are similar but are challenged by large differences [20]. Further, these methods are challenged by hidden variables, such as epigenetic fate priming, which manifests itself in measured gene expression profiles only with a time delay [10]. In contrast, single-cell lineage tracing (scLT) approaches label cells with heritable DNA "scars" which may be used to delineate fate relations over long time intervals [17]. Genetic scars are read out in a sequencing experiment, jointly with gene expression information. Computationally combining the two sources of information is currently an open question; solving it for destructive in-vivo time series experiments requires an approach that relates clonal information only within one time point while comparing gene expression across time points.

1.2 Research question and contributions

1.2.1 Research question

The central research question of this thesis is how to integrate RNA velocity and lineage-tracing information into trajectory inference to gain more accurate insights into cellular dynamics, especially in complex situations like regeneration and reprogramming. We break down this question into individual challenges as follows:

(i) RNA velocity is an elegant approach for recovering directional information at no additional experimental cost. However, it is a noisy, high-dimensional estimate, and it is currently unclear how it can be distilled into a robust, stochastic representation of cellular dynamics in high dimensions.

(ii) Given such a representation, an open question remains how it can be used to automatically detect a biological process's initial, intermediate and terminal states. Further, how can such a representation be used to model the gradual nature of fate establishment during which cells proceed from multi-potent to uni-potent states?

(iii) Besides RNA velocity, alternative approaches for estimating directionality for single-cell data have been suggested. Combining these into one unified framework would greatly accelerate progress in studying fate decisions with single-cell data.

(iv) Time series experiments provide a reasonable estimate for directionality; previous approaches have successfully applied optimal transport [21] to link cells from earlier to later time points. However, these approaches scale poorly in cell numbers and are thus challenged by the size of current datasets.

(v) Time series experiments can be supplemented with clonal information through scLT approaches; however, computational methods are missing that exploit both gene expression and clonal information to faithfully link cells across time points and derive robust trajectories.

1.2.2 My contributions

I address these challenges in my thesis. My contributions can be grouped into two categories:

(i) Markov chain-based modeling of directed cellular dynamics through one unified framework called CellRank which overcomes the limitations of classic TI when applied to complex systems with unclear directions. CellRank consists of kernels, which construct a Markov chain based on data modalities including RNA velocity and gene expression similarity, and estimators, which derive biological insights based on the Markov chain representation.

(ii) Optimal transport-based modeling of single-cell genomics data through one unified framework called moscot. moscot contains estimators for various mapping problems, including temporal and spatial problems which frequently arise. This thesis focuses on the temporal problem of mapping large single-cell datasets across time points, possibly including joint lineage readout.

For the first category, I make the following contributions:

(i) I introduce CellRank's `VelocityKernel`, a KNN-based approach to estimating a transition matrix given RNA velocity and gene expression similarity, propagating uncertainty. This contribution addresses challenge (i).

(ii) I adapt Markov State Models (MSM) to the single-cell context. I use them as a CellRank estimator to coarse-grain the Markov-chain into initial, intermediate, and terminal macrostates. Further, I introduce a new way to compute fate probabilities which scales to larger datasets than previous approaches. This contribution addresses challenge (ii).

(iii) I apply my proposed model to mouse embryonic fibroblast (MEF) reprogramming [22], pancreas development, [23] and lung regeneration past injury [11]. I predict a novel dedifferentiation trajectory; in collaboration with Janine Schniering and Herbert Schiller, we validate the existence of previously unknown intermediate states on the trajectory.

(iv) I extend CellRank by introducing the `PseudotimeKernel`, the `CytoTRACEKernel` and the `RealtimeKernel`, allowing the framework to use almost any source of prior directional information when setting up the Markov chain. This contribution addresses challenge (iii).

For the second category, I make the following contributions:

(i) I introduce `moscot-time`, an optimal transport-based model which scales much better

than previous approaches, thus enabling the application to current datasets containing large cell numbers. This contribution addresses challenge (iv).

(ii) I introduce `moslin`, a Fused Gromov-Wasserstein-based model that uses both inter-individual expression similarity and intra-individual lineage relationships when mapping cells across time. This contribution addresses challenge (v).

(iii) In collaboration with Zoe Piran and Michal Klein, I show how `moslin` outperforms competing approaches on simulated and real data. We apply `moslin` to *Caenorhabditis elegans* (*C. elegans*) embryogenesis [24], where it recovers known lineage drivers.

As parts of my contributions have already been published in peer-reviewed journals or are in the process of review or preparation, Chapter 3 and Chapter 4 of this thesis are to some extent identical to or correspond to the following publications:

(i) **Lange, M.**, Bergen, V., Klein, M., Setty, M., Reuter, B., Bakhti, M., Lickert, H., Ansari, M., Schniering, J., Schiller, H.B., Pe'er, D. and Theis, F.J., 2022. CellRank for directed single-cell fate mapping. *Nature Methods*, pp.1-12.

(ii) Weiler P.*, **Lange, M.***, Klein, M. and Theis, F.J., 2022. A unified framework to study single-cell fate decisions. *In preparation.*

(iii) **Lange, M.***, Piran, Z.*, Klein, M., Theis, F.J. and Nitzan, M., 2021. Mapping lineage-traced single-cells across time-points. *NeurIPS LMRL workshop contribution.*

(iv) **Lange, M.***, Piran, Z.*, Klein, M.*, Spanjaard, B.*, Junker, J.P., Theis, F.J. and Nitzan, M., 2022. Mapping lineage-traced single-cells across time-points. *In preparation.*

(v) Klein, D.*, Palla, G.*, **Lange, M.***, Klein, M.*, Piran, Z.*, Gander, M., Meng-Papaxanthos, L., Nitzan, M., Cuturi M., Theis F. J., Mapping cells through time and space with moscot. *In preparation.*

Note that "*" denotes an equal contribution; specifically, my contributions to these publications are as follows:

(i) I designed and developed the method, implemented a CellRank prototype, analyzed the data, and wrote the manuscript with contributions from co-authors. Further, I contributed to the pyGPCCA implementation.

(ii) I designed and developed the method, implemented prototypes, coordinated the project, and analyzed the data presented in this thesis in collaboration with Philipp Weiler and Michal Klein.

(iii) I designed and developed the method, wrote the manuscript in collaboration with Zoe Piran, and performed CellRank analysis on `moslin`'s couplings for *C. elegans*.

(iv) I designed and developed the method, coordinated the project, and interpreted results.

(v) I designed the overall framework, developed the temporal approach, coordinated the project, and interpreted results.

Furthermore, my doctoral research contributed to the following publications, which are not included in this thesis:

(i) Bergen, V., **Lange, M.**, Peidli, S., Wolf, F.A. and Theis, F.J., 2020. Generalizing RNA velocity to transient cell states through dynamical modeling. *Nature biotechnology*, 38(12), pp.1408-1414.

(ii) Strunz, M., Simon, L.M., Ansari, M., Kathiriya, J.J., Angelidis, I., Mayr, C.H., Tsidiridis, G., **Lange, M.**, Mattner, L.F., Yee, M., Ogar, P., Sengupta, A., Kukhtevich, I., Schneider, R., Zhao, Z., Voss, C., Stoeger T., Neumann, J.H.L., Hilgendorff, A., Behr, J., O'Reilly, M., Lehmann, M., Burgstaller, G., Königshoff M., Chapman, H.A., Theis, F.J. and Schiller H.B., 2020. Alveolar regeneration through a Krt8+ transitional stem cell state that persists in human lung fibrosis. *Nature communications*, 11(1), pp.1-20.

(iii) Tritschler, S., Büttner, M., Fischer, D.S., **Lange, M.**, Bergen, V., Lickert, H. and Theis, F.J., 2019. Concepts and limitations for learning developmental trajectories from single cell genomics. *Development*, 146(12), p.dev170506.

(iv) Erhard, F., Baptista, M.A., Krammer, T., Hennig, T., **Lange, M.**, Arampatzi, P., Jürges, C.S., Theis, F.J., Saliba, A.E. and Dölken, L., 2019. scSLAM-seq reveals core features of transcription dynamics in single cells. *Nature*, 571(7765), pp.419-423.

While these publications are not explicitly included in my thesis, the following connections exist:

(i) publication (i) generalizes the idea of RNA velocity to transient populations, a crucial step to make the CellRank model of Chapter 3 widely applicable. I contributed to

the development of validation metrics of the proposed scVelo model. In Section 2.5 of this thesis, I present the RNA velocity model in a new unified form.

(ii) publication (ii) serves as an application example of the CellRank method in Chapter 3. I contributed a velocity analysis of plasticity among club and alveolar-type cells in Fig. 6b. In the CellRank publication, I predict a dedifferentiation trajectory not described in the original publication.

(iii) publication (iii) gives an intuitive overview of the topic of trajectory inference; I contributed the section "The concepts of pseudotime and trajectory inference". In Section 2.4 of this thesis, I present an alternative introduction to the topic.

(iv) publication (iv) introduces a strategy for metabolic labeling of single cells, an alternative to RNA velocity. I contributed towards benchmarking RNA velocity with metabolic labeling in Fig. 2b,c. In Section 2.5 of this thesis, I describe metabolic labeling techniques.

1.3 Outline

In Chapter 2, we describe single-cell data modalities and recap mathematical background on Markov chains and optimal transport which is relevant for Chapter 3 and Chapter 4, respectively. We give an overview of previous approaches to trajectory inference, including RNA velocity.

In Chapter 3, we introduce the CellRank framework, in particular the `VelocityKernel`, which derives a Markov chain from RNA velocity, as well as the `GPCCAEstimator`, which coarse-grains the dynamics into interpretable macrostates. We apply the model to MEF reprogramming [22], pancreas development [23], and lung regeneration [11], where we predict and experimentally validate the existence of new intermediate cell states. We further show how to extend the CellRank framework to include other sources of directionality, including pseudotime [25], the CytoTRACE score [26], and real-time information [20].

In Chapter 4, we introduce the moscot framework, in particular, the `moscot-time` and `moslin` models with maps cells across time points for scRNA-seq and scLT data, respectively. `moslin` is a Fused Gromov-Wasserstein-based model that uses both clonal relations and gene expression; we show that it outperforms competing approaches on both simulated and real data. Applied to *C. elegans* embryogenesis [24], we demonstrate how `moslin` uncovers trajectories and putative decision driver genes. Finally, in Chapter 5, we summarize our contributions and discuss directions for future research.

Chapter 2

Background

We open this chapter by introducing single-cell assays and corresponding analysis approaches with a focus on biological questions and how they can be addressed using either data modality (Section 2.1). Further, we discuss Markov chains (Section 2.2) and optimal transport (Section 2.3) which provide mathematical background for Chapter 3 and Chapter 4, respectively. We combine the data (Section 2.1) with the methods (Section 2.2 and Section 2.3) when reviewing the field of trajectory inference (Section 2.4) where we highlight how Markov chains and optimal transport have previously been used to recover cellular dynamics. Finally, we present RNA velocity as a possibility to overcome the problem of uncertain directions in classic trajectory inference (Section 2.5).

Notation. We denote vectors by lower-case bold face symbols (e.g. $\boldsymbol{x}$), scalars by lower-case, non-bold symbols (e.g. x) and matrices by upper case, non-bold symbols (e.g. X). Vector elements appear non-bold, i.e. x_i.

2.1 Single-cell genomics

Single-cell assays open the door to study cellular heterogeneity which was masked in prior, population-based (bulk) assays [27]. The cell is the fundamental unit of life, and many biological processes can only be studied by probing individual cells for their molecular markup; for example: how cells make decisions when they change their state, from naive to differentiated [6, 7] (development), from normal to cancerous [28] (tumor evolution), from differentiated cell type A to B [29] (transdifferentiation), from differentiated back to pluripotent [20] (reprogramming) from injured back to healthy [11] (regeneration). These decisions are executed at the level of individual cells based on intrinsic properties of the cell [30] (e.g., gene expression, DNA accessibility), external stimuli [31] (e.g., signaling, tissue composition), and stochasticity [32] (e.g., fluctuations of molecular counts). Understanding cellular decision-making in health and disease enables designing therapies that intervene when normal mechanisms are perturbed, i.e., gene and cell therapies [9, 33]. Different single-cell technologies have been developed to study cellular states at differ-

ent molecular levels. We focus on the technologies relevant for later chapters: single-cell RNA-sequencing (scRNA-seq) [2, 34, 35], which probes gene expression, and single-cell lineage tracing technologies (scLT-seq) [17], which jointly probe lineage history and gene expression. We introduce both technologies along the following questions:

(i) molecular layer studied, related biological questions.

(ii) experimental basics, how does this technology function.

(iii) data specifics: dimensionality and distributions.

(iv) computational approaches: how is the data used to address biological questions?

Further modeling paradigms. This thesis is focused on studying dynamical biological processes through the lens of single-cell genomics; accordingly, we will exclude many single-cell modeling paradigms which are not immediately relevant. These include data integration and reference mapping [36–38], genetic and molecular perturbations [39–44], spatial technologies and related computational approaches [45–48], tissue, organ, patient and cohort variation [5, 49, 50], cell-cell communication [51–53] and gene regulatory network inference [54–56], among others.

Further data modalities. Besides scRNA-seq, further assays have been developed that probe different aspects of molecular makeup at single-cell resolution. These include proteomic assays like flow and mass cytometry (e.g. CyTOF [57, 58]) as well as epigenetic assays like Chip-Seq [59, 60] and CUT&Tag [61–63] for histone modifications and transcription factor occupancy, respectively, Bisulfite-Seq for DNA methylation [64–66] and single-cell assay for transposase accessible chromatin using sequencing (scATAC-seq) [67] for chromatin accessibility. Recently, uni-modal assays have been extended towards multimodal assays which probe more than one molecular layer in the same single cell; these include

- for P and T: CITE-seq [68] and REAP-seq [69].
- for T and C: 10x multiome, scM&T-seq [65], SHARE-seq [70], Paired-Seq [71], sci-CAR [72] and SNARE-seq [73]
- for C and P: ASAP-seq [74].
- for P, T and C: DOGMA-seq [74]

where P, T, and C denote aspects of the proteome, transcriptome, and epigenome, respectively. Multi-omic single-cell approaches have recently been reviewed by Stuart and Satija [75].

The methods we introduce in Chapter 3 and Chapter 4 are fundamentally based on *cell-cell distances*; for CellRank, these are used to build a KNN graph (Subsection 2.1.2) while for `moslin`, they are used to define a cost function (Section 2.3). Thus, while our application examples from Chapter 3 and Chapter 4 are based on transcriptomic scRNA-seq readout, the methods we introduce generalize to further data modalities by adapting the definition of the distance metric to the corresponding data modality. We make this explicit when deriving cell-cell distances from learned representations (Subsection 2.1.2).

2.1.1 Single-cell RNA-sequencing (scRNA-seq)

Molecular layer and biological questions. scRNA-seq probes gene expression in single cells, typical output is a *count matrix* $X^{(R)} \in \mathbb{N}_0^{N_c \times N_g}$, where N_c is the number of cells, N_g is the number of genes and $X_{ij}^{(R)}$ counts the number of mRNA molecules corresponding to gene j detected in cell i. Throughout this thesis, we use the symbol X to refer to unbiased measurements of cellular state; we denote by superscript the modality we refer to, e.g. (R) for scRNA-seq. scRNA-seq was the first single-cell technology to be employed at scale [77] and can be used to address a multitude of biological questions, ranging from unbiased descriptions of cellular state (e.g. detecting rare and transitional cell types [78], dissecting cellular heterogeneity [79]) via regulatory mechanisms in health and disease (e.g. gene regulation [55]) to the inference of cellular trajectories in continuous biological processes (e.g. trajectory inference, Section 2.4).

Experimental basics. scRNA-seq counts messenger RNA (mRNA) molecules; these are created from a DNA template by RNA polymerase in a process called *transcription*, transported to the ribosome in the cytoplasm, and in turn serve as a template to produce a protein in a process called *translation* (Figure 2.1a,b). Besides their function as an intermediate product between DNA and protein, RNA molecules also serve various regulatory processes in the cell [80]. Typically, mRNA counting is based on sequence complementary - mRNA molecules are extended at their 3' end with a stretch of repeated adenine ("A") nucleotides in a process called *polyadenylation*. By synthesizing a long stretch of repeated thymine ("T") nucleotides into a *poly(T) primer*, mRNA molecules can be bound ("A" binds to "T") and their information content can be read out in a sequencing machine.

Critically, cells must be equipped with unique *cell barcodes* before sequencing to achieve

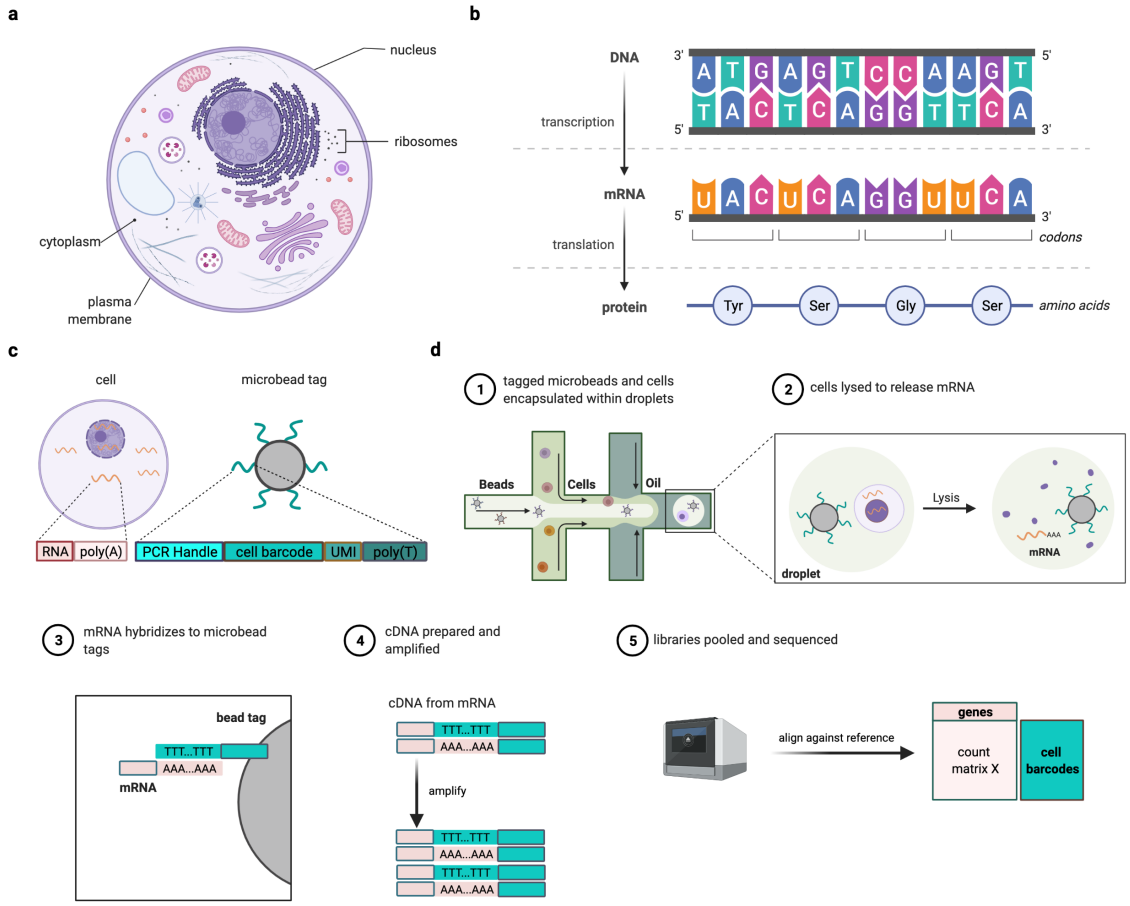

Figure 2.1: Experimental basics of droplet-based scRNA-sequencing a. Simplified overview of an animal cell. **b.** Central dogma of molecular biology [19]; colors denote the alphabet of DNA and RNA given by different nucleotides, opposite ends of DNA and mRNA are referred to by 3' and 5' ends. DNA is transcribed to mRNA in the nucleus (**a**), mRNA is transported to the ribosome (**a**) and translated into proteins. During translation, mRNA nucleotides are processed in groups of three referred to as *codons*; each codon corresponds to one of 20 *amino acids*, some examples are shown. **c.** Input to droplet-based sequencing protocols like Drop-seq [35] or 10x chromium [76]: cells and microbeads tags. **d.** Simplified workflow of droplet-based scRNA-seq. Adapted from the following templates: "Structural Overview of an Animal Cell", "Central Dogma", "CITE-seq Workflow", by BioRender.com (2022). Retrieved from `https://app.biorender.com/biorender-templates`.

single-cell resolution. Different experimental protocols exist for this purpose; early approaches isolated individual cells in microwell plates (plate-based technologies, e.g. Smart-seq2 [81], Smart-seq3 [82], MARS-Seq [83]), which have been extended to droplet based technoloiges (Figure 2.1c,d) (e.g. Drop-Seq [35], inDrop [34], 10x Chromium [76]), combinatorial indexing based technologies (e.g. sci-RNA-seq [84], SPLiT-seq [85]) and even

combinations of combinatorial indexing with droplet technologies (e.g. scifi [86]); these technologies differ in sensitivity and cellular throughput (see below). All datasets in this thesis have been generated using droplet-based technologies (Figure 2.1c,d).

Once individual cells have been barcoded, mRNA molecules are amplified in a process called *polymerase chain reduction* (PCR) [77]. Most approaches label individual mRNA molecules prior to this process with *unique molecular identifiers* (UMIs) [87] to be able to distinguish between biological and technical copies of mRNA molecules (amplification bias).

Once the sequences have been read by the sequencing machine, algorithms are employed to align the mRNA fragments against the reference genome (e.g. human or mouse) in order to associate each mRNA with the gene it has been derived from. After some additional error correction techniques, e.g. filtering empty droplets or droplets with more than one cell (doublets), the final cont matrix $X^{(R)}$ is constructed. We refer to Ziegenhain et al. [88], Zhang et al. [89], Svensson et al. [90], and Mereu et al. [91] for comparisons of different experimental approaches.

Cellular throughput and sensitivity. A typical scRNA-seq experiment yields a count matrix $X^{(R)} \in \mathbb{N}_0^{N_c \times N_g}$ where the number of genes N_g is about 20k - 30k depending on the organism. The number of cells N_c increases constantly from year to year, varies across experimental technologies, and is generally lower for plate-based technologies (approx. 1k - 50k cells) [81–83] and higher for droplet- and combinatorial indexing based technologies (approx. 10k - 4M cells) [3, 34, 35, 76, 92]. Another important consideration is the sensitivity of a technique, i.e. the probability of capturing a particular mRNA molecular present in the cell and converting it into a complementary DNA (cDNA) molecule present in the sequencing library [88]. In a benchmarking study of 13 commonly used scRNA-seq techniques, Mereu et al. [91] show that method sensitivity varies widely, with plate-based Quartz-seq2 [93] and Smart-seq 2 [81] as well as droplet-based 10x Chromium [76] performing best overall. All scRNA-seq techniques yield sparse count matrices $X^{(R)}$, albeit at varying, sensitivity-dependent levels.

Statistical distributions for scRNA-seq data. scRNA-seq data is subject to various sources of biological (e.g. transcriptional bursting [32], cell-to-cell variation [94]) and technical noise [95] (e.g. limited and variable detection sensitivity), thus appropriate statistical models must be employed when describing the data. In principle, single-cell gene expression is count data with empirically observed gene-specific overdispersion, thus,

a *Gamma-Poisson distribution*, also referred to as *Negative-Binominal (NB)* distribution seems appropriate,

$$\mathrm{NB}\left(k|\mu,\alpha\right) = \frac{\Gamma(k+\alpha)}{\Gamma(k+1)\Gamma(\alpha)}\left(\frac{\alpha}{\alpha+\mu}\right)^{\alpha}\left(\frac{\mu}{\mu+\alpha}\right)^{k}, \tag{2.1}$$

with mean and inverse-dispersion parameters $\mu \in \mathbb{R}$ and $\alpha \in \mathbb{R}_+$, respectively [96]. The NB distribution can be motivated mathematically by considering the biological process of gene expression [97–99].

In the early days of plate-based scRNA-seq, there has been concern about the fraction of zeros observed in single-cell data compared to bulk (population-based) assays; this concern exists to this day in parts of the community. Accordingly, computational models have been developed that reserve special treatment to excess zeros, also termed *dropouts*, including differential expression methods [100, 101] or dimensionality reduction methods [102, 103]. Further, *imputation techniques* [104, 105] were developed which replace zero values with other values that are deemed more likely and the zero-inflated negative binomial distribution [106] (ZINB) became a popular noise model for scRNA-seq data.

In contrast to this development, novel computational tools [107–110] demonstrated that for droplet-based scRNA-seq data, no zero inflation is observed and in fact, the simpler NB distribution describes the data just as well as the more complex ZINB distribution. While it was initially argued that plate-based scRNA-seq gives rise to zero inflation while droplet-based scRNA-seq does not [111], the difference was later attributed to the presence of UMIs [87]: while the distribution of read-counts (no UMIs) is zero-inflated, the distribution of UMI counts is not zero inflated [112, 113], possibly because UMIs deflate counts of genes with particularly high PCR amplification bias [112]. Note that major droplet-based protocols include UMIs. The community arrived at the conclusion that read-count scRNA-seq data (e.g. SMART-seq2 [81]) should thus be modeled using the ZINB distribution while UMI-count scRNA-seq data (e.g. 10x Chromium [76], Drop-seq [35], inDrop [34], Smart-seq3 [82]) should be modeled using the NB distribution. Modern latent-space models like scVI [96] or DCA [114] reflect this consensus by offering both distributions. All data considered in this thesis is UMI-count data, thus, when appropriate, we employ the NB distribution.

2.1.2 Computational approaches for scRNA-seq data

The field of computational models for scRNA-seq data is vast and constantly expanding, the `scrna-tools.org` database recently celebrated the addition of tool number 1,000 [129].

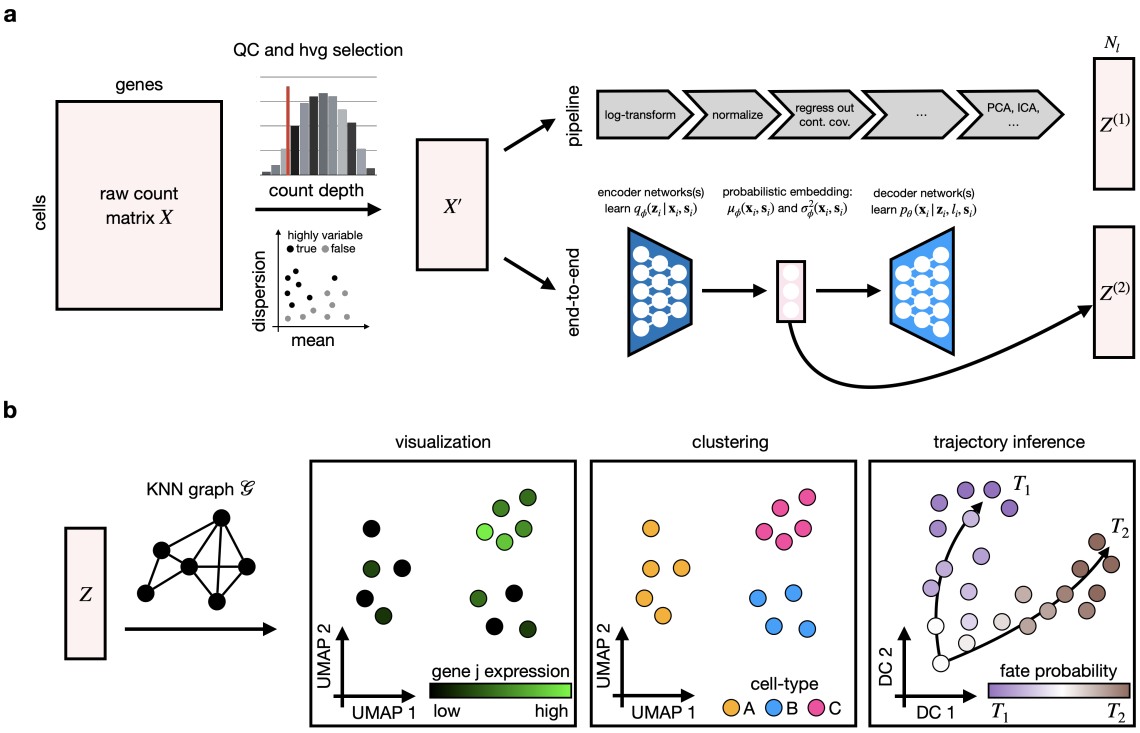

Figure 2.2: Representation learning and sample-centric analysis a. Starting from a cell-by-gene count matrix X, analysis usually starts with quality control (QC) where low-quality cells and certain genes (e.g. lowly detected genes, mitochondrial genes, ribosomal genes, cell-cycle genes, etc) are removed. In the next step, highly variable genes (HVGs) are selected to increase the signal-to-noise ratio. These first two steps yield a reduced count matrix X'. In pipeline approaches like SCANPY [115] or Seurat [116–119], sequential transformations are then applied to X' to arrive at an N_l-dimensional representation $Z^{(1)}$. In contrast, end-to-end approaches like DCA [114] or scVI [96] use autoencoders or variational autoencoders (VAEs) to learn $Z^{(2)}$ in a *bottleneck layer*. We illustrate the VAE approach here where an encoder network is trained to learn an amortized variational approximation q_ϕ of the posterior latent distribution while a decoder network learns the data likelihood p_θ given the latent representation [120]. This is a simplified illustration, largely omitting the size factors l_i and batch labels $\boldsymbol{s}_i$. **b.** Given an N_l-dimensional representation Z, we illustrate three common sample-centric downstream analysis techniques which often rely on an intermediate (KNN)-graph $\mathcal{G}$ of the data. Left: visualization in two-dimensions using non-linear dimensionality reduction like UMAP [121–123], t-SNE [124, 125] or diffusion maps [126–128], we overlay expression of gene j. Middle: clustering to detect cell types or states, we show cell types A, B, and C. Right: TI to uncover continuous trends in the data. We show two terminal states, T_1 and T_2, as well as fate probabilities towards them. DC, diffusion components [128].

We refer to Ziegenhain et al. [77], Luecken and Theis [130], and Vieth et al. [131] for comprehensive reviews and restrict ourselves here to introducing the main modeling paradigms to learn cellular representations which are relevant for later chapters: *pipeline approaches*

vs. *end-to-end approaches* (Figure 2.2a).

Both pipeline, as well as end-to-end approaches, yield N_l-dimensional cell representations $Z \in \mathbb{R}^{N_c \times N_l}$ for $N_l << N_g$ which are better suited for some downstream analysis tasks compared to the original gene expression matrix $X^{(R)}$ because they are much lower-dimensional (i.e. they suffer less from the *curse of dimensionality* [132]) and have been corrected for some sources of unwanted technical variation (e.g. batch effects, library size, etc.). The central idea is that distances measured in Z using e.g. euclidean distance are more meaningful compared to distances measured directly in $X^{(R)}$, this enables cell-centric downstream analysis tasks which rely on a robust definition of cell-cell distances including clustering, visualization and trajectory inference (Figure 2.2b). Using a cell-cell distance metric, many of these approaches approximate the phenotypic manifold of sampled cellular states with a graph $\mathcal{G}$ that connects each cell to its K nearest neighbors, known as a *K-nearest neighbors graph* (KNN graph).

In the following, we review pipeline and end-to-end approaches to compute low-dimensional representations Z, KNN graph $\mathcal{G}$ construction as well as clustering and visualization. We describe trajectory inference, a central theme of this thesis, later on (Section 2.4) once the mathematical concepts of Markov chains (Section 2.2) and optimal transport (Section 2.3) have been introduced.

Pipeline approaches for representation learning. Pipeline approaches like python-based SCANPY [115] or R-based Seurat [116–119] apply sequential transformations to the raw count matrix $X^{(R)}$ to arrive at an N_l-dimensional representation $Z \in \mathbb{R}^{N_c \times N_l}$; these transformations usually include (Figure 2.2a):

(i) quality-control filtering of cells and genes [130],

(ii) count normalization to correct for sequencing depth [95, 133–135],

(iii) log-transformation to stabilize the variance [130],

(iv) filtering to highly-variable genes [76, 116, 117],

(v) regressing out unwanted sources of continuous variation due to e.g. the cell cycle or the percentage of mitochondrial reads [116],

(vi) scaling to unit variance [130],

(vii) correcting for discrete technical variation termed *batch effects* [36, 117, 136–141],

(viii) computing a lower-dimensional representation Z (typically 30-100 dimensions) using principal component analysis (PCA), independent component analysis (ICA), or variants thereof [130].

Endless variations of this workflow exist [77, 130, 131], and individual steps have been the topic of intense debate, e.g. log transformation [142]. Rather than modeling raw counts using the negative binomial distribution from above, most pipeline approaches aim at transforming the data into a space where it looks more "normal" such that a method like PCA may be applied which assumes normally distributed data [132].

End-to-end approaches for representation learning. In contrast to pipeline approaches, end-to-end approaches act directly on the raw count matrix (filtered to highly variable genes) and output a latent representation Z of the data within one consistent modeling framework which makes them better suited to handle uncertainty (Figure 2.2a). Two poplar end-to-end methods, DCA [114] and scVI [96], are based on autoencoders and variational autoencoders [120, 143, 144], respectively. scVI formulates a generative model of the data,

$$p(\boldsymbol{x}_i, \boldsymbol{z}_i, l_i|\boldsymbol{s}_i) = p(\boldsymbol{x}_i|\boldsymbol{z}_i, l_i, \boldsymbol{s}_i)\; p(\boldsymbol{z}_i, l_i|\boldsymbol{s}_i)\,, \tag{2.2}$$

for gene expression vector $\boldsymbol{x}_i \in \mathbb{N}^{N_g}$, latent representation $\boldsymbol{z}_i \in \mathbb{R}^{N_l}$ ($N_l = 10$ by default [96]), cell-specific scaling factor $l_i \in \mathbb{R}_+$ and one-hot encoded batch covariate $\boldsymbol{s}_i \in \{0,1\}^B$ for B batches (e.g. datasets from different labs, individuals, experimental protocols, etc) . The likelihood is given by an NB distribution,

$$p(\boldsymbol{x}_i|\boldsymbol{z}_i, l_i, \boldsymbol{s}_i) = \Pi_{j=1}^{N_g} \mathrm{NB}\left(x_{ij}|\mu_\theta(\boldsymbol{z}_i, \boldsymbol{s}_i)_j\, l_i,\; \alpha_j\right)\,, \tag{2.3}$$

where $\mu_\theta : \mathbb{R}^{N_l} \times \{0,1\}^B \to \Delta_{N_g}$ is a neural network with parameters θ referred to as *decoder* or *generative* network, Δ_K is the probability simplex in K dimensions,

$$\Delta_K \coloneqq \left\{ a \in \mathbb{R}_+^K : \sum_{i=1}^{K} a_i = 1 \right\}\,, \tag{2.4}$$

and $\boldsymbol{\alpha} \in \mathbb{R}_+^{N_g}$ acts as a gene-specific inverse dispersion parameter of Equation (2.1). Using the probability simplex Δ_{N_g} ensures that $\mu_\theta(\boldsymbol{z}_i, \boldsymbol{s}_i)$ can be interpreded as gene frequencies per cell, i.e. relative contributions of a gene to a cell. The prior factorizes over $\boldsymbol{z}_i$ and l_i

and is given by

$$\begin{aligned} p(\boldsymbol{z}_i, l_i|\boldsymbol{s}_i) &= p(\boldsymbol{z}_i)\ p(l_i|\boldsymbol{s}_i) \\ &= \mathcal{N}\left(\boldsymbol{z}_i|0, I_L\right)\ \log \mathcal{N}\left(l_i|\boldsymbol{s}_i^\top \boldsymbol{l}^{(\mu)}, \boldsymbol{s}_i^\top \boldsymbol{l}^{(\sigma^2)}\right), \end{aligned} \tag{2.5}$$

for batch-specific $\boldsymbol{l}^{(\mu)}, \boldsymbol{l}^{(\sigma^2)} \in \mathbb{R}_+^B$ which represent the the library size mean and variance in log-space, respectively. Computing the posterior distribution,

$$p(\boldsymbol{z}_i, l_i|\boldsymbol{x}_i, \boldsymbol{s}_i) = \frac{p(\boldsymbol{x}_i, \boldsymbol{z}_i, l_i|\boldsymbol{s}_i)}{p(\boldsymbol{x}_i|\boldsymbol{s}_i)}, \tag{2.6}$$

is intractable and Lopez et al. [96] revert to the mean-field amortized variational approximation

$$q_\phi(\boldsymbol{z}_i, l_i|\boldsymbol{x}_i, \boldsymbol{s}_i) = q_\phi(\boldsymbol{z}_i|\boldsymbol{x}_i, \boldsymbol{s}_i)\ q_\phi(l_i|\boldsymbol{x}_i, \boldsymbol{s}_i), \tag{2.7}$$

with the two terms,

$$q_\phi(\boldsymbol{z}_i|\boldsymbol{x}_i, \boldsymbol{s}_i) = \mathcal{N}\left(\boldsymbol{z}_i|\mu_\phi(\boldsymbol{x}_i, \boldsymbol{s}_i), \mathrm{diag}\left(\sigma_\phi^2(\boldsymbol{x}_i, \boldsymbol{s}_i)\right)\right) \tag{2.8}$$

$$q_\phi(l_i|\boldsymbol{x}_i, \boldsymbol{s}_i) = \log \mathcal{N}\left(l_i|l_\phi^{(\mu)}(\boldsymbol{x}_i, \boldsymbol{s}_i), l_\phi^{(\sigma^2)}(\boldsymbol{x}_i, \boldsymbol{s}_i)\right), \tag{2.9}$$

for mean and variance neural networks, $\mu_\phi, \sigma_\phi^2 : \mathbb{N}^{N_g} \times \{0,1\}^B \to \mathbb{R}^{N_l}$, referred to as *inference* or *encoder* networks, and scaling factor mean and variance neural networks, $l_\phi^{(\mu)}, l_\phi^{(\sigma^2)} : \mathbb{N}^{N_g} \times \{0,1\}^B \to \mathbb{R}$, where ϕ denotes the set of all parameters these networks define. Note that in the formulation above, the encoder has access to the batch labels $\boldsymbol{s}_i$, which is by default not the case in the original scVI model, however, subsequent models like TotalVI [145] do include such a link which appears to increase data-integration performance empirically. The model parameters α, θ, ϕ are trained by maximizing the *evidence lower bound* (ELBO), which lower bounds the marginal likelihood $p(\boldsymbol{x}_i|\boldsymbol{s}_i)$,

$$\begin{aligned} \sum_i \log p(\boldsymbol{x}_i|\boldsymbol{s}_i) \geq &\sum_i \mathrm{E}_{q_\phi(\boldsymbol{z}_i, l_i|\boldsymbol{x}_i, \boldsymbol{s}_i)}\left[p(\boldsymbol{x}_i|\boldsymbol{z}_i, l_i, \boldsymbol{s}_i)\right] \\ &- \sum_i \left(\mathrm{KL}\left[q_\phi(\boldsymbol{z}_i|\boldsymbol{x}_i, \boldsymbol{s}_i)\,||\,p(\boldsymbol{z}_i)\right] + \mathrm{KL}\left[q_\phi(l_i|\boldsymbol{x}_i, \boldsymbol{s}_i)\,||\,p(l_i|\boldsymbol{s}_i)\right]\right), \end{aligned} \tag{2.10}$$

using variants of mini-batched *stochastic gradient descent* (SGD), i.e. Adam [146]. Many extensions of the VAE approach for scRNA-seq representation learning exist, e.g. to improve interpretability [147, 148], to include perturbations [39–41] and cell-type labels [149] or to transfer learned representations and labels across datasets [38].

Representation learning for other molecular layers. We stated in the introduction to this section that the methods we introduce in Chapter 3 (CellRank) and Chapter 4 (`moslin`) generalize beyond scRNA-seq data by adapting the definition of cell-cell distances; this translates directly into adapted representation learning techniques. Both pipeline, as well as end-to-end approaches, have been extended to other data modalities. For example, for the popular scATAC-seq [67] modality, Signac [150], EpiScanpy, [151] and ArchR [152] have been suggested as pipeline approaches and PeakVI [153] has been suggested as an end-to-end approach. An advantage of end-to-end approaches is that they can be adapted easily to other data modalities by changing the likelihood function $p(\boldsymbol{x}_i|\boldsymbol{z}_i, l_i, \boldsymbol{s}_i)$ (Equation (2.3)).

For the multi-modal setting, Seurat v4 [118] and Muon [154] have been suggested as pipeline approaches, and TotalVI [145] (proteins and RNA, e.g. CITE-seq [155] and REAP-seq [69]) as well as MultiVI [156] and Multigrate [157] (chromatin accessibility and RNA, e.g. SHARE-seq [70] and 10x multiome) have been suggested as end-to-end approaches. Thus, representation learning approaches are available that allow us to derive robust cell-cell distances beyond scRNA-seq data; these may be used to generalize CellRank and `moslin`.

Similarity graph construction from single-cell data. Given a low-dimensional representation Z obtained from either pipeline or end-to-end approaches, a common way to construct similarity graphs in the single-cell field is via weighted *K-nearest neighbor* (KNN) graphs $\mathcal{G} = (V, E)$ for vertices (or nodes) V given by the set of sequenced cells and edges in E connecting neighboring cells. Undirected edge weights are stored in the symmetric *adjacency matrix* $W \in \mathbb{R}_+^{N_c \times N_c}$. To compute W, the following steps are usually employed:

(i) for each cell i, compute distances to its K-nearest neighbors based on an initial distance metric, e.g. Euclidean or cosine similarity.

(ii) symmetrize the KNN relations such that cells i and j are nearest neighbors if either i is a nearest neighbor of j, or j is a nearest neighbor of i.

(iii) compute an adjacency matrix W based on the symmetrized distances containing similarity estimates between neighboring cells according to the manifold structure using e.g. a Gaussian kernel [128] or the UMAP method [121, 158].

KNN graphs can be computed efficiently for large cell numbers using recent approximation algorithms [159, 160]. Note that KNN graphs give rise to sparse adjacency matrices which enable downstream computations like clustering or visualization to scale very well with the number of cells.

Visualization. Cells can be visualized in two or three-dimensional scatter plots known as *embeddings* (Figure 2.2b). They provide a high-level summary of the data where each cell is visualized as a dot, colored according to clusters, gene expression, or other cell-level covariates. In principle, such embeddings can be computed with the pipeline or end-to-end approaches for representation learning by choosing $N_l = 2$ or 3; however, it has been found that these approaches do not preserve local neighborhood information well and are challenged by subtle differences in expression states. Thus, dedicated *dimensionality reduction* techniques have been suggested which provide informative visual summaries of the data [130]. The most popular techniques, t-SNE [124, 125], UMAP [121–123] and diffusion maps [126–128], are based on KNN graphs. While t-SNE and UMAP optimize an objective that encourages correspondence of low-and high-dimensional neighborhoods, diffusion maps are a spectral approach based on Laplacian eigenmaps (Section 2.2.5).

Despite their widespread use, two- or three-dimensional cell embeddings are intensely debated because they are prone to over-interpretation [161]. In particular, neighborhood-based methods like t-SNE and UMAP do not provide an accurate summary of global data topology [161–164]. For this reason, in Chapter 3, we argue against projecting high-dimensional velocity vector fields into two-dimensional embeddings for trajectory interpretation and present CellRank as an alternative approach that operates directly in high-dimensional space. In this thesis, we use two-dimensional embeddings to provide a visual summary of the data but avoid deriving biological hypotheses from them. We refer to Heiser and Lau [162] for a recent benchmark of dimensionality reduction methods in the single-cell field.

Clustering. While TI reveals continuous cell heterogeneity (Section 2.4), clustering reveals discrete cell heterogeneity including groups of cells related by type, disease state, cell-cycle phase or metabolic state (Figure 2.2b). Clustering approaches from an integral part of most single-cell analysis workflows and accordingly, a vast amount of clustering approaches have been suggested. These approaches range from classic K-means [78, 165] via hierarchical clustering [166, 167], mixture models [168], and ensemble models [169] to graph-based models [170–172]; see Kiselev, Andrews, and Hemberg [173] for a recent review.

Despite this variety, the overwhelming majority of published studies employ one of two graph-based community-detection approaches, Louvain [170, 172] or Leiden [171]. These approaches have been found to perform very well for single-cell data [130, 173], they scale to very large (greater than 10M) cell numbers [170] and they are accessible via major single-cell platforms including SCANPY [115] and Seurat [116–119]. Both approaches optimize

modularity, a measure that encourages intra-cluster edges and penalizes inter-cluster edges [170, 174, 175].

2.1.3 Single-cell lineage tracing (scLT)

The single-cell assays introduced above, including scRNA-seq, suffer from the fact cells are destroyed upon sequencing. Hence, when using these assays to study continuous biological processes in a time-series context, computational methods are frequently employed to reconstruct likely relationships between cells in early and late time points based on gene expression similarity (Section 2.4). To guide this challenging reconstruction problem, experimental techniques have been developed that record clonal relationship among cells. While early methods were labor-intensive, limited to transparent organisms, and relied on manual observation of individual cells in time-lapse microscopy [179], recent approaches are sequencing-based and make use of heritable genetic barcodes [180]. While a multitude of such approaches exists, we focus on those that fulfill the following two criteria which are essential to link molecular features to fate outcome in Chapter 4:

(i) methods that achieve single-cell resolution,

(ii) methods that give joint lineage and state (scRNA-seq) readout.

Whenever we use the term "single-cell lineage tracing" (scLT), we refer to methods that fulfill these two criteria. Experimental strategies for scLT have recently been reviewed in Wagner and Klein [17], Baron and Oudenaarden [180], Moreno-Ayala and Junker [181], and Olivares-Chauvet and Junker [182].

Molecular layer and biological questions. scLT approaches jointly record cellular state and lineage relations, typical output is a state matrix $X^{(R)}$, as well as a *barcode set* $B = \{b_1, ..., b_{N_c}\}$, where each b_i is given by a string that can be used to relate cells with respect to their clonal history. The nature of B and how it can be used depends on the lineage tracing technique and the experimental design. Lineage tracing techniques can be categorized into prospective and retrospective approaches, where the former makes use of engineered barcodes while the latter makes use of naturally occurring mutations (Figure 2.3a). Time-resolved experimental designs can be categorized into *clonal resampling* and *independent clonal evolution*, which assay cells from the same or different clones across several time points, respectively. (Figure 2.3b).

scLT approaches are frequently used to study fate decisions in complex biological systems

such as zebrafish [176–178, 183] and mouse [184, 185] embryonic development, regeneration [186], directed differentiation [187], neurogenesis [188, 189], or tumor evolution [190–192]. To give just one example, scLT approaches applied to tumor evolution revealed how epigenetic properties grant fitness advantages to certain clones under therapy-induced selective pressure [28].

Prospective scLT. In prospective scLT, heritable barcodes are introduced to the DNA such that they can be read out later on in a sequencing experiment (Figure 2.3a). Approaches differ in whether the introduced barcodes are static or dynamic over time, i.e. accumulate additional mutations.

Static barcoding approaches are mostly based on retroviral delivery and are limited in the clonal diversification they can capture. If static barcodes are only delivered at a single time-point (e.g. LARRY [10] or TREX [188]), then no clonal substructure can be inferred. Additional rounds of barcoding at later time points can increase sub-clonal resolution (e.g. CellTagging [22]).

Dynamic barcoding approaches resolve clonal substructure at high resolution through continuously evolving barcodes, either through CRISPR/Cas9 induced deletions and insertions (indels) or by using transposons. The first CRISPR-based methods demonstrating this principle were scGESTALT [176], LINNAEUS, [177] and ScarTrace [178]. While early applications focused on zebrafish embryonic development and regeneration [176–178, 186], the principle has meanwhile been adapted to mouse development [184, 185] and cancer progression in orthopedic [191, 192] and genetic mouse models [190]. The utility of transposon-based systems for dynamic scLT has been demonstrated in early zebrafish development using TracerSeq [183]. Dynamic barcoding approaches are particularly suited for in-vivo applications because they uncover high-resolution clonal substructures without the need for repeated labeling as in static approaches.

Retrospective scLT. Retrospective scLT approaches rely on naturally occurring somatic mutations in either nuclear or mitochondrial DNA and can therefore also be applied in humans, something that is not possible for their prospective counterparts (Figure 2.3a).

Nuclear DNA-based approaches aim to reconstruct lineage history mostly based on single nucleotide variants (SNVs), copy number variations (CNVs), LINE-1 retroelements, or microsatellite variations [180]. As nuclear somatic variations in healthy tissue are very rare, these approaches are mostly restricted to cancer applications where mutation rates

are much higher [193–195]. However, even in systems with elevated mutation rates, it remains difficult to couple nuclear approaches to unbiased state readout at high cellular throughput. One challenge is the length of nuclear DNA, making it difficult to detect mutations. Targeted approaches circumvent this problem by focusing on a few (1-3) known mutations (e.g. Genotyping of Transcripts, GOT [196]), however, recording such a low number of mutations is not sufficient to confidently reconstruct clonal relations.

Mitochondrial DNA based approaches have two key advantages over nuclear DNA based approaches: first, the mitochondrial DNA is much shorter (5 orders of magnitude), making it much easier to detect mutations, and second, mutations in mitochondrial DNA are much more common (by about a factor 10-100 [197–200]). While initial approaches were plate-based and costly [201, 202], the technique has meanwhile been extended to yield droplet-based ATAC (mtscATAC-seq [203]), RNA (MAESTER [204]), ATAC and protein (ASAP-seq [74]) and ATAC, RNA and surface protein (DOGMA-seq [74]) readout. Recent applications have focused on detecting pathological mitochondrial mutations [205] and identifying clonal substructure in chronic lymphocytic leukemia [28].

Clonal resampling. In clonal resampling, the aim is to observe the same clone (cells sharing the same barcode) across several time points, i.e. for a single phylogenetic tree, we aim to observe some ancestral nodes, besides the leaf nodes (Figure 2.3b). This setting is well suited to link the molecular state of early cells (e.g. CD34+ progenitor cells in hematopoiesis) to the eventual fate outcome of their sister cells (e.g. Monocytes or Neutrophils) [10]. As this approach relies on the repeated sampling of clonally related cells, it is mostly applicable to in-vitro settings, [10, 22, 187], in vivo transplantation settings [10] or in vivo regenerative systems like human PBMC and CD34+ samples [28, 206] or the zebrafish fin [178].

Independent clonal evolution. Beyond the transplantation and regenerative settings discussed above, applying time-series scLT in vivo requires independent clonal evolution, i.e. different individuals, sequenced at different time points with independent clonal evolution proceeding in each animal (Figure 2.3b). For example, Hu et al. [186] use the LINNAEUS system (prospective, CRISPR-based dynamic barcode generation) to study zebrafish heart generation. Barcoding is initiated during early development, heart injury is induced and fish are raised to adulthood. The barcoded heart cells are harvested at either 3, 7, or 30 days post-injury, thus creating a barcoded time series of independent clonal evolution.

Data specifics. The aim of the `moslin` model presented in Chapter 4 is to link cells across time points for in-vivo studies, thus, we focus on independent clonal evolution designs which give high-resolution clonal diversity without repeated cell sampling. The most promising experimental approaches to achieve this are prospective dynamic scLT based on CRISPR/Cas9 mutations and retrospective scLT based on mitochondrial mutations. With respect to the additional state readout these methods yield, the data-specific considerations for scRNA-seq from above hold.

In addition, barcode-specific sources of noise for the CRISPR/Cas9 approach include barcode homoplasy, i.e. unrelated cells acquiring the same barcode by chance, and barcode degradation, i.e. large deletions erasing prior indels. Further, it remains difficult to control the barcoding rate in both space (tissue/organ) and time (mutation rate/division rate). Mitochondrial DNA based approaches face some unique challenges, including heteroplasmy (related to the high copy number of the mitochondrial genome), relaxed replication (mitochondrial DNA is replicated all the time, not just in one defined cell-cycle phase), mitophagy (selective degradation of damaged mitochondria through a process called autophagy) as well as random assignment of mitochondria to daughter cells [28].

Computational approaches. Computational approaches for scLT data from independent clonal evolution include methods to reconstruct lineage trees from individual time points [207, 208], methods to infer unobserved ancestral states, [209] and one method, called LineageOT [210], to link cells across time points using lineage and gene expression information. Lineage OT, which is most relevant to our discussion in Chapter 4, has been developed as an extension of the Waddington OT [20] algorithm (Section 2.4) to take lineage relationships into account when mapping cells from earlier to later time-points. LineageOT takes as input time-series scRNA-seq data with phylogenetic trees reconstructed independently for each time point. When computing probabilistic couplings across pairs of time points, LineageOT corrects expression profiles in the later time point based on lineage similarity. This assumes a model in which cells of the same lineage randomly diffuse over short time scales and thus their earlier expression can be approximated by shrinking clonally related cells towards each other.

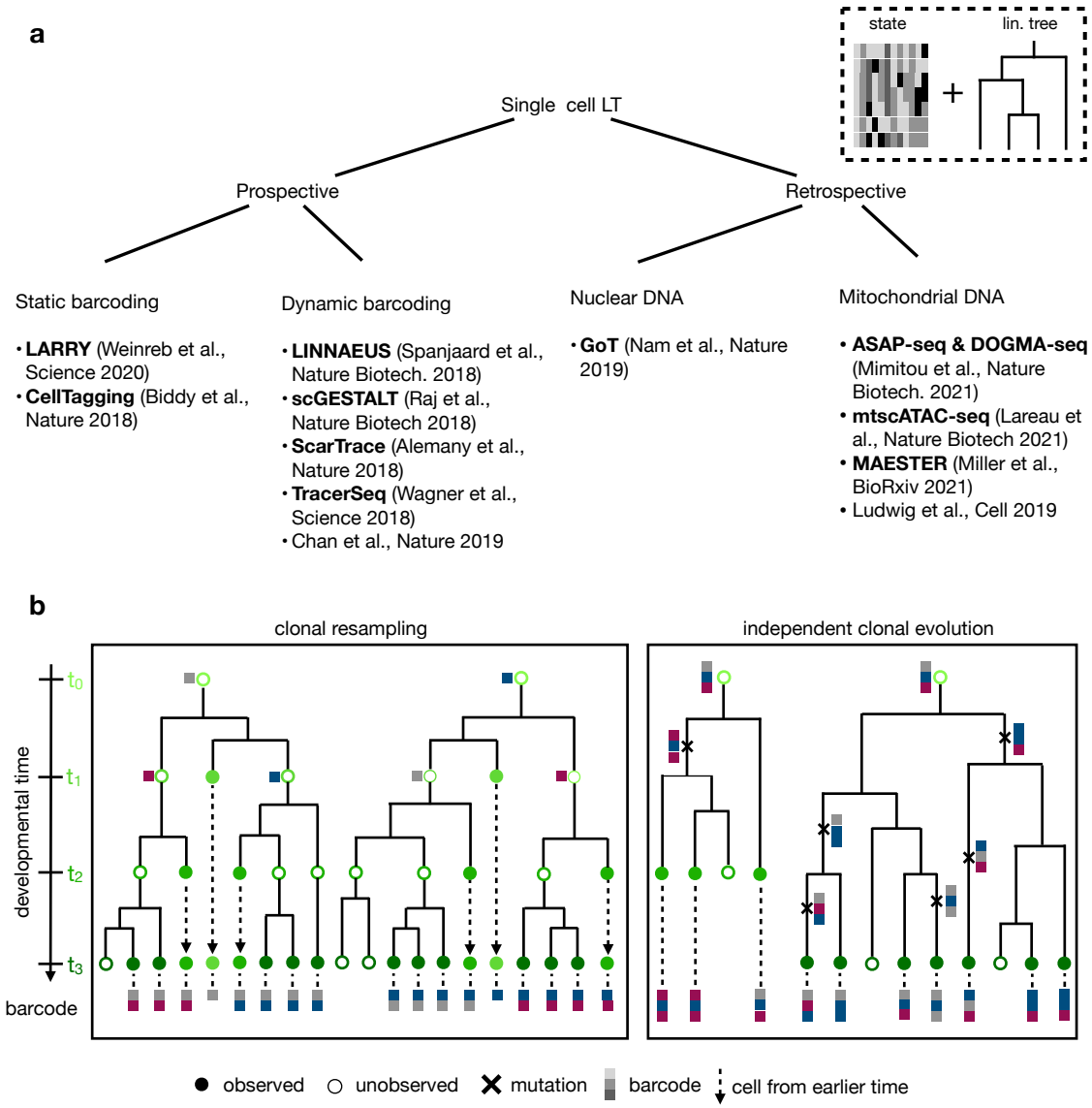

Figure 2.3: Overview of single-cell lineage tracing (scLT) techniques and experimental designs. a. Experimental techniques can be split into prospective and retrospective approaches; some examples are given. We focus on single-cell technologies which give joint state $X^{(R)}$ and lineage readout, highlighted in the inlet. **b.** Overview of common experimental designs; four time points are indicated on the horizontal axis. Dots (circles) denote observed (unobserved) cells, squares denote barcodes, and crosses denote mutations. Dots and circles are colored according to time. Left: clonal resampling with repeated static barcoding at time points t_0 and t_1 and sequencing at time points t_1, t_2 and t_3. Barcode combinations can be used to link cells from the same clone across time points. Cells can be unobserved either because they died (e.g. through apoptosis) or because they have not been sampled for sequencing. Right: independent clonal evolution for two cells dynamically barcoded (using e.g. a CRISPR editing system [176–178]) at time t_0 which are sequenced at t_2 and t_3, respectively. Mutations accumulate independently in the two trees. In both clonal resampling and independent clonal evolution, we omit the additional state readout.

2.2 Discrete-time Markov chains

While the previous section explained scRNA-seq and scLT data as well as common analysis approaches, the next two sections provide mathematical background necessary to develop the new methods of later chapters: discrete-time Markov chains (this section) for CellRank (Chapter 3) as well as optimal transport (Section 2.3) for `moslin` (Chapter 4). This section loosely follows Tolver [211] and Bressloff [212] in describing Markov chains in discrete time over discrete and finite state spaces. Concepts we explain here, such as random walks, absorption probabilities, and spectral graph theory, have been used in methods prior to CellRank (e.g. DPT [213], Palantir, [25] and PBA [214]) as we explain in our overview of trajectory inference approaches (Section 2.4).

2.2.1 Definition of a Markov chain

We consider the sequence $\{X_0, X_1, ...\}$ of discrete random variables, where each X_l can take one of the N values in the *statespace* $\mathcal{S} = \{0, 1, ..., N-1\}$. If this process satisfies the *Markov Property*,

$$\mathbb{P}\left[X_l = n | X_{l-1}, ..., X_0\right] = \mathbb{P}\left[X_l = n | X_{l-1}\right] , \tag{2.11}$$

for all $l \geq 1$ and $n \in \mathcal{S}$ then we call it a Markov chain [211] (MC). It this notation, $\mathbb{P}[A]$ defines the probability of event A.

Transition matrix. We describe the evolution of the MC using the *transition matrix* $T \in \mathbb{R}_+^{N \times N}$ which is defined as

$$T_{i,j} \coloneqq \mathbb{P}\left[X_{l+1} = j | X_l = i\right] \quad \forall (i,j) \in \mathcal{S}^2 . \tag{2.12}$$

We assume that the process is *time-homogeneous* which means that T is constant with respect to l. Note that T is a real, non-negative row-stochastic matrix, i.e. $T_{i,j} \geq 0 \; \forall i, j \in \mathcal{S}^2$ and $\sum_j T_{i,j} = 1 \; \forall i \in \mathcal{S}$. We can visualize T in a directed, weighted graph called the *state graph* $\mathcal{G}$ where nodes represent states and edges represent possible transitions among them, weighted by the corresponding transition probability in T (Subsection 2.2.5).

Initial distribution. To uniquely define the distribution of the MC for all $l \geq 0$, we need to specify the *initial distribution* $\boldsymbol{\phi} \in \mathbb{R}^N$,

$$\phi_i \coloneqq \mathbb{P}\left[X_0 = i\right] \quad \forall i \in \mathcal{S}\,. \tag{2.13}$$

The MC is uniquely defined by specifying the pair $(T, \boldsymbol{\phi})$.

l-step transition probabilities. One can show using the Chapman–Kolmogorov equations that the l-step transition probabilities are given by powers of the transition matrix T, i.e.

$$\mathbb{P}\left[X_{l_0+l} = j | X_{l_0} = i\right] = T^l_{i,j}\,. \tag{2.14}$$

Taken together, this allows us to find the probability vector of the process at an arbitrary time $l \geq 0$ as

$$\left(\mathbb{P}\left[X_l = 0\right], \mathbb{P}\left[X_l = 1\right], ..., \mathbb{P}\left[X_l = N-1\right]\right)) = \boldsymbol{\phi}^\top T^l\,. \tag{2.15}$$

2.2.2 Properties of Markov Chains

We introduce properties that are important to studying the long-term evolution of MCs (Subsection 2.2.3), this is relevant to identifying initial and terminal states in Chapter 3.

Communication classes and irreducibility. Define a *path* on an MC as a sequence of transitions between states which taken together has a probability greater than zero. We define state i to be *accessible* from state j iff there exists a path to get from j to i. Iff i is accessible from j and the converse holds as well, we say i and j *communicate* and we write $i \leftrightarrow j$. Communication defines an equivalence relation in the state space $\mathcal{S}$ [211]. We may use communication between states to define a disjoint partition of the state space,

$$\mathcal{S} = \bigcup_{k=0}^{K} \mathcal{C}_k\,. \tag{2.16}$$

We call the resulting partitions $\mathcal{C}_k$ *communication classes*. In the equivalent setting of directed graphs (Subsection 2.2.5), these objects are known as *strongly connected components* and can be identified efficiently in time $\mathcal{O}(N)$ using Tarjan's algorithm [215]. If an MC is composed of a single communication class, we call it *irreducible*, otherwise, we call it *reducible*. Many of the theoretical results we introduce below only hold for irreducible

MCs. In practice, when dealing with reducible MCs, it is therefore helpful to restrict the MC to one of its communication classes at a time such that it becomes irreducible.

Hitting and return time. Given any state $i \in \mathcal{S}$, define the *hitting time* of i as the first time visiting i. Mathematically, we define

$$H_i \coloneqq \inf\{l > 0 | X_l = i\}\,. \tag{2.17}$$

When we additionally condition on the process starting in sate i, we refer to H_i as the *return time*. Using the return time, we can differentiate between two different kinds of communication classes.

Recurrence and transience. A state $i \in \mathcal{S}$ is called *recurrent* iff the probability of returning to the state in finite time is 1, i.e. iff

$$\mathbb{P}\left[H_i < \infty | X_0 = i\right] = 1\,. \tag{2.18}$$

Otherwise, we call state i *transient*. Recurrent and transience are class properties, i.e. either all states $i \in \mathcal{C}_k$ are recurrent, or they are all transient (Thm. 11 in Tolver [211]). Therefore, we may speak of *recurrent classes* and *transient classes*. Intuitively, we never leave a recurrent class once we entered it. Denote a recurrent class by $\mathcal{R}_k$ and a transient class by $\mathcal{T}_k$. If a recurrent class consists of just one single state, we call this state an *absorbing state*.

Period of a state. For a given state $i \in \mathcal{S}$, define M to be the set of the lengths of all possible paths which start and end in i. Then, the *period* of state i is the largest non-negative integer that divides all elements of M. If a state has period 1, we call it *aperiodic*. Periodicity, like recurrence and transience, is a class property, so we may speak of the period of a class (Thm. 18 in Tolver [211]). We may further call a class aperiodic if it has period 1, and periodic otherwise.

Ergodicity. We call an MC *ergodic* if is irreducible and aperiodic. Given the definitions of irreducibility, recurrence, transience, periodicity, and ergodicity, we are in a position to study the long-term evolution of an MC.

2.2.3 Limiting behavior of a Markov chain

In many practical single-cell applications, we are interested in the long term evolution of a biological process described by a Markov chain, i.e. in the limit

$$\lim_{l\to\infty} \mathbb{P}\left[X_l = i\right] , \tag{2.19}$$

for a given state $i \in \mathcal{S}$. For a general MC, this limit does not exist. If the limit does exist, it is unique and we call the resulting probability distribution the *limiting distribution* of the MC. Intuitively, an MC for which there exists a limiting distribution will forget its initial conditions in the long-run regime, i.e. the probability of being in any state i for $l \to \infty$ will not depend on the initial distribution ϕ. Given that we can describe l-step transitions by T^l, this is equivalent to studying the behavior of $T^l_{i,j}$ as $l \to \infty$. In the following paragraph, we present a special case where a limiting distribution exists.

Limiting distribution for an ergodic Markov chain. If the MC is ergodic, i.e. irreducible and aperiodic, then the limit of Equation (2.19) exists and is given by the expression

$$\lim_{l\to\infty} \mathbb{P}\left[X_l = i\right] = \frac{1}{\mathrm{E}\left[H_i | X_0 = i\right]} , \tag{2.20}$$

where the term in the denominator on the right hand side (RHS) is the expected return time introduced in Equation (2.17) [211]. Given this result, the following questions remain:

(i) how do we compute the RHS of Equation (2.20)?

(ii) how does this extend to periodic MCs?

(iii) how does this extend to reducible MCc?

To address these questions, we introduce *invariant measures*.

Invariant measures. A non-negative vector $\boldsymbol{\pi} \in \mathbb{R}^N_+$ which satisfies the following system,

$$\pi_i = \sum_{j\in\mathcal{S}} \pi_j T_{j,i} \; \forall i \in \mathcal{S} , \tag{2.21}$$

is called an *invariant measure* of the transition matrix T. If $\boldsymbol{\pi}$ is a probability distribution (i.e. it sums to one), we call it an *invariant distribution*, often also a *stationary distribution*.

Computing invariant measures. For an irreducible MC, existence and uniqueness of an invariant measure can be shown via the Perron-Frobenius Theorem [216] (Section A.1 in Appendix A). From the definition given in Equation (2.21), it is clear that an invariant measure $\boldsymbol{\pi}$ should be a non-negative vector that has the property of being invariant under the linear transformation represented by T, i.e. we require

$$\boldsymbol{\pi}^\top T = 1\, \boldsymbol{\pi}^\top . \tag{2.22}$$

Therefore, an invariant measure for T associated with an irreducible MC is given by the non-negative left eigenvector with eigenvalue 1, which is unique up to multiplication.

Connecting invariant measures to the limiting distribution. We are now in a position to address question (i) from above by connecting the limiting distribution to the invariant distribution. For an ergodic MC and for any state $i \in \mathcal{S}$, it holds:

$$\lim_{l\to\infty} \mathbb{P}\left[X_l = i\right] = \pi_i = \frac{1}{\mathrm{E}\left[H_i | X_0 = i\right]}, \tag{2.23}$$

where $\boldsymbol{\pi}$ is the unique invariant distribution solving Equation (2.21), see Thm. 23 in Tolver [211] for a proof. In other words, for an ergodic MC, we simply compute the left eigenvector to eigenvalue 1 of T and ensure $\pi_i \geq 0$ and $\sum_i \pi_i = 1$.

Periodic Markov chains. We turn to irreducible but periodic MCs to address question (ii). For such MCs, we may still compute an invariant distribution but this will not represent the limiting distribution of the process. The intuitive reason is that in the long run, the process will converge to some cycling behavior where the position in the cycle depends on the initial distribution. In that sense, the chain does not forget its initial conditions. However, for an MC of period $d > 1$, we may define the average distribution over a period of length d via

$$\lim_{l\to\infty} \frac{\mathbb{P}\left[X_l = i\right] + \mathbb{P}\left[X_{l+1} = i\right] + ... + \mathbb{P}\left[X_{l+d-1} = i\right]}{d}, \tag{2.24}$$

for an arbitrary $i \in \mathcal{S}$ (Thm. 26 in Tolver [211]). Let π_i denote the value of the above limit for a given state i. If $\sum_i \pi_i = 1$, then $\boldsymbol{\pi}$ is the unique invariant distribution of the MC. In other words, in the case of an irreducible but periodic MC, we can still link the

long-run behaviour of the process to its invariant distribution, we only have to interpret $\boldsymbol{\pi}$ as the average distribution over one complete period in the limit $l \to \infty$. This addresses the second question.

Reducible Markov chains. In order to address question (iii), we need to partition the state space of the MC into its recurrent and transient classes, $\mathcal{T}_c$ and $\mathcal{R}_d$, respectively. Let $(X_l)_{l\geq 0}$ be a reducible MC with state space $\mathcal{S}$. In general, we may partition $\mathcal{S}$ as follows:

$$\mathcal{S} = \bigcup_{c=1}^{C} \mathcal{T}_c \quad \bigcup \quad \bigcup_{d=1}^{D} \mathcal{R}_d \,, \tag{2.25}$$

for $C \in \mathbb{N}_0$ transient and $D \in \mathbb{N}_0$ recurrent classes. For the number of states in each of these partitions, we define

$$N_c^{(T)} = |\mathcal{T}_c| \,, \quad N_d^{(R)} = |\mathcal{R}_d| \quad \forall (c,d) \in \{1,...,C\} \times \{1,...,D\}. \tag{2.26}$$

For any state i in any of the $\mathcal{T}_c$ transient classes, the following intuitive result holds:

$$\lim_{l\to\infty} \mathbb{P}\left[X_l = i\right] = 0 \,, \tag{2.27}$$

irrespective of the choice of initial distribution, see Thm. 25 in Tolver [211] for a proof. Therefore, we may in the following restrict our attention to recurrent states and classes.

Limiting behavior of recurrent states for reducible Markov chains. For a recurrent state j in any of the $\mathcal{R}_d$, the limiting behavior will necessarily depend on the initial distribution. However, if we restrict the MC to one recurrent class $\mathcal{R}_d$ at a time, it will be irreducible in that restriction, and all of the results for irreducible MCs hold. In particular, let $\boldsymbol{\pi}^{(d)} \in \mathbb{R}^{N_d^{(R)}}$ be the limiting distribution of the MC restricted to recurrent class $\mathcal{R}_d$. Define $\tilde{\boldsymbol{\pi}}^{(d)} \in \mathbb{R}^N$ as the vector filled with elements of $\boldsymbol{\pi}^{(d)}$ in the indices corresponding to $\mathcal{R}_d$ and zero elsewhere. Then $\tilde{\boldsymbol{\pi}}^{(d)}$ will be an invariant distribution to the unrestricted MC. In this way, we may define a set of vectors $\mathcal{V} = \{\tilde{\boldsymbol{\pi}}^{(1)}, ..., \tilde{\boldsymbol{\pi}}^{(D)}\}$. These vectors form a non-negative basis for the left 1-eigenspace of T. Note that any linear combination of the vectors in $\mathcal{V}$ will always be a left eigenvector to eigenvalue 1, and any convex combination of the vectors in $\mathcal{V}$ will be an invariant measure for the MC. We see from this that for reducible MCs, the invariant measure is no longer unique but can be made unique by focusing on individual recurrent classes.

2.2.4 Absorption probabilities for a Markov chain

In this subsection, we are interested in describing irreducible MCs with transient and recurrent classes. Once the MC enters one of the recurrent classes, it will never leave again, i.e. it's 'absorbed' by that class. Given that the MC is initialized in a transient state i, we would like to know how likely the MC is to be absorbed by each of the recurrent classes. This is relevant to computing *fate probabilities* of cells towards different terminal states in a biological process in Chapter 3. For this purpose, in Theorem 2.1, we reproduce the statement of Thm. 28 from Tolver [211].

Theorem 2.1 (Absorption probabilities)**.** *Consider an MC with transition matrix $T \in \mathbb{R}^{N\times N}$. We may rewrite T as follows:*

$$\begin{bmatrix} \tilde{T} & 0 \\ S & Q \end{bmatrix}, \tag{2.28}$$

where $\tilde{T}$ and Q are restrictions of T to recurrent and transient states, respectively, and S is the restriction of T to transitions from transient to recurrent states. The upper right zero is due to the fact that there can be no transitions back from recurrent to transient states. We define the matrix $M \in \mathbb{R}^{N\times N}$ via

$$M \coloneqq (I - Q)^{-1}. \tag{2.29}$$

Then, the ij-th entry of M contains the expected number of visits the process makes to state j before absorption, conditional on initialization in state i. M is usually referred to as the fundamental matrix of the MC. Further, the matrix

$$A \coloneqq (I - Q)^{-1}S, \tag{2.30}$$

in the ij-th entry contains the probability of j being the first recurrent state reached by the MC, given that it started in i.

Proof. See Thm. 28 in Tolver [211]. □

Absorption probabilities to recurrent classes. Note that we can apply Theorem 2.1 to situations where we are not interested in absorption probabilities to individual recurrent states but rather to recurrent classes by summing up absorption probabilities for all constituent recurrent states of a recurrent class.

2.2.5 Spectral graph theory

MCs can be analyzed in terms of their associated state graph, equally, certain graphs can be analyzed in terms of associated Markov chains, a principle that is used in many trajectory inference algorithms (Section 2.4). Given a similarity graph $\mathcal{G}$ of the data (Subsection 2.1.2), MC formulations and spectral properties can be used to derive distance metrics [213] and embeddings [126–128] of the data. For a general introduction to the topic, we refer to Von Luxburg [217] and Haghverdi [218] which we follow in this exposition.

Graph laplacians. To study properties of the KNN similarity graph $\mathcal{G}$, we introduce the graph laplacian matrix $L \in \mathbb{R}^{N_c \times N_c}$ as well as its two normalized variants, $L^{(\mathrm{rw})}, L^{(\mathrm{sym})} \in \mathbb{R}^{N_c \times N_c}$ via

$$L = D - W\,, \tag{2.31}$$

$$L^{(\mathrm{rw})} = D^{-1}L = I - D^{-1}W\,, \tag{2.32}$$

$$L^{(\mathrm{sym})} = D^{-1/2}LD^{-1/2} = I - D^{-1/2}WD^{-1/2}\,, \tag{2.33}$$

for the diagonal *node-degree matrix* $D \in \mathbb{R}^{N_c \times N_c}$ with entries $D_{ii} = \sum_{j=1}^{N_c} W_{ij}\ \forall i$. The normalized laplacians carry the superscripts rw and sym because they are tightly connected with random walks and are symmetric, respectively.

Laplacian eigenmaps and diffusion maps. Consider the problem of embedding cells $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_{N_c}\}$ for $\boldsymbol{x}_i \in \mathbb{N}^{N_g}$ in N_l dimensions for $N_l << N_g$ such that their similarity captured by the adjacency matrix W is preserved. Denoting by $Y \in \mathbb{R}^{N_c \times N_l}$ the low-dimensional representation of the data, the problem may be written as

$$\min_Y \sum_{i,j=1}^{N_c} W_{ij}\, ||Y_{i,:} - Y_{j,:}||^2 = \min_Y \mathrm{trace}(Y^\top DY) \quad \text{s.t. } Y^\top DY = I\,, \tag{2.34}$$

where the constraint $Y^\top DY = I$ ensures existence of a unique solution. It can be shown [219] that the solution is given by a set of vectors $Y = [\boldsymbol{y}_1., ..., \boldsymbol{y}_{N_l}]$ which satisfy the following eigenvalue problem,

$$L\boldsymbol{y_i} = \lambda D\boldsymbol{y}_i\,, \tag{2.35}$$

or equivalently,

$$L^{(\mathrm{rw})}\boldsymbol{y}_i = \lambda \boldsymbol{y}_i \,. \tag{2.36}$$

In particular, for eigenvalues and right eigenvectors of $L^{(\mathrm{rw})}$, $0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_{N_c}$ and $\boldsymbol{\psi}_1, ..., \boldsymbol{\psi}_{N_c}$, respectively, the solution is given by the set of (appropriately normalized) right eigenvectors $\{\boldsymbol{\psi}_2, ..., \boldsymbol{\psi}_{N_l+1}\}$. The representation Y is called a *laplacian eigenmap*. Intuitively, it makes sense to exclude the first eigenvector $\boldsymbol{\psi}_1$ as it corresponds to the constant vector of ones,

$$L^{(\mathrm{rw})}\mathbf{1} = \left(I - D^{-1}W\right)\mathbf{1} = \mathbf{1} - \mathbf{1} = 0 \,, \tag{2.37}$$

for $\mathbf{1} = [1, ..., 1]^\top \in \mathbb{R}^{N_c}$. In the single-cell community, laplacian eigenmaps are widely used under the name *diffusion maps* [126, 128] where the adjacency matrix is constructed using a Gaussian kernel with density dependent scaling parameter and additional density correction [126] to account for non-uniform sampling of cells from the phenotypic landscape.

Diffusion distance. Random-walk based properties can also be used to define robust distance measures [126, 127, 218], we introduce the transition matrices corresponding to the graph laplacians from above,

$$T^{(\mathrm{rw})} = D^{-1}W \,, \tag{2.38}$$

$$T^{(\mathrm{sym})} = D^{-1/2}WD^{-1/2} \,. \tag{2.39}$$

To define a distance between sampled data points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, consider the probability of a random walk starting in $\boldsymbol{x}_i$ to reach $\boldsymbol{x}_j$ in exactly t steps, given by

$$T_{ij}^{(\mathrm{sym})^t} = \sum_{k=1}^{N_c} \gamma_k^t \phi_{ki}\phi_{kj} \,, \tag{2.40}$$

For eigenvalues and eigenvectors γ_k and $\boldsymbol{\phi}_k$ of $T^{(\mathrm{sym})}$, respectively, normalized such that $\boldsymbol{\phi}_i^\top \boldsymbol{\phi}_j = \delta_{ij}$. As $T^{(\mathrm{sym})}$ is symmetric, left and right eigenvectors are identical. Further, γ_k and $\boldsymbol{\phi}_k$ are related to the eigenvalues are eigenvectors of $L^{(\mathrm{rw})}$ via

$$\gamma_i = 1 - \lambda_i \,, \tag{2.41}$$

$$\boldsymbol{\phi}_i = D^{1/2}\boldsymbol{\psi}_i \,. \tag{2.42}$$

This can be seen easily by writing

$$T^{(\mathrm{sym})} D^{1/2} \boldsymbol{\psi}_i = D^{-1/2} W \boldsymbol{\psi}_i = D^{1/2} L^{(\mathrm{rw})} \boldsymbol{\psi}_i = (1 - \lambda_i) D^{1/2} \boldsymbol{\psi}_i \,. \tag{2.43}$$

Using the t-step transition probabilities as a feature representation, we define the diffusion distance [126] as

$$D^2_{\mathrm{diff},t}(i,j) \coloneqq ||T^{(\mathrm{sym})^t}_{i,:} - T^{(\mathrm{sym})^t}_{j,:}||^2 = \sum_{k=1}^{N_c} \left(T^{(\mathrm{sym})^t}_{i,k} - T^{(\mathrm{sym})^t}_{j,k} \right)^2 \tag{2.44}$$

which may be rewritten using Equation (2.40) as

$$D^2_{\mathrm{diff},t}(i,j) = \sum_{k=1}^{N_c} \gamma_k^{2t} \left(\phi_{ki} - \phi_{kj} \right)^2 \tag{2.45}$$

$$\tag{2.46}$$

In Section 2.4, we review *diffusion pseudotime* [213], an extension to diffusion distance that is frequently used in the single-cell field.

2.3 Optimal transport maps between distributions

Many problems in single-cell genomics involve mapping distributions over sampled cells from one space to another; this can be achieved in a probabilistic manner using optimal transport (OT). For example, in the temporal domain, we can only measure each cell once (Section 2.1), thus we obtain *cross-sectional* measurements at discrete time points which we have to link probabilistically to follow the evolution of cells from early to late stages [20]. In the spatial domain, experimental assays vary in the number of genes they can measure and the spatial resolution they achieve. Two extreme examples would be scRNA-seq, which measures all genes without any spatial resolution, and early microscopy approaches, which measure only around 10 genes, albeit at cellular (or even sub-cellular) spatial resolution. OT can be used to couple these two data modalities to infer expression values for the unobserved genes at cellular spatial resolution [220].

The original notion of OT was introduced by the French mathematician Gaspard Monge in 1781 who considered the problem of moving a pile of sand to a hole where the shape of the pile and the hole are prescribed and a certain cost is associated with moving each grain of sand from source to target locations. This problem is known as the *Monge problem*; it

leads to a constrained non-convex optimization problem that is not guaranteed to have a feasible solution in general [221]. Kantorovich [222] in 1942 famously proposed to relax the transport problem by allowing probability mass from a source destination to be split across several target destinations, thus evolving from deterministic *transport maps* to probabilistic *couplings*. OT is based on a rich body of mathematical theory and can be formulated for general (continuous) measures which must not be probability measures. In this section, however, we focus on the theory relevant to Chapter 4, i.e. discrete probability measures within the Kantorovich relaxation, and some extensions. We refer to Peyré and Cuturi [221] for an excellent overview of OT which we follow in this exposition.

2.3.1 The Kantorovich relaxation of Optimal Transport

For probability vectors (equivalently, *histograms*) $\boldsymbol{a} \in \Delta_N$ and $\boldsymbol{b} \in \Delta_M$, define the corresponding discrete measures α and β as

$$\alpha(\boldsymbol{x}) = \sum_{i=1}^{N} a_i \delta_{\boldsymbol{x}_i} \tag{2.47}$$

$$\beta(\boldsymbol{y}) = \sum_{j=1}^{M} b_i \delta_{\boldsymbol{y}_i}\,, \tag{2.48}$$

with respect to source and target locations, $(\boldsymbol{x}_i, \boldsymbol{y}_j) \in \mathcal{X} \times \mathcal{Y}$ for all $(i, j) \in \{1, ..., N\} \times \{1, ..., M\}$ where $\mathcal{X}, \mathcal{Y} \subseteq \mathbb{R}^D$ (Figure 2.4a). OT over discrete probability measures seeks to find a *coupling matrix* $P \in \mathbb{R}_+^{N \times M}$, transporting mass from α to β in a way that is optimal with respect to a cost function $c : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$ where $c(\boldsymbol{x}_i, \boldsymbol{y}_j)$ is the cost associated with moving a piece of unit mass from location $\boldsymbol{x}_i$ to $\boldsymbol{y}_j$. The set of feasible couplings, given by those matrices P that satisfy the marginal constraints imposed though the probability vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, may be written as

$$U(\boldsymbol{a}, \boldsymbol{b}) \coloneqq \left\{ P \in \mathbb{R}_+^{N \times M} : P\mathbf{1}_M = \boldsymbol{a},\ P^\top \mathbf{1}_N = \boldsymbol{b} \right\}\,, \tag{2.49}$$

for constant-one vectors $\mathbf{1}_N$ and $\mathbf{1}_M$ of lenghts N and M, respectively. With this at hand, we can state the Kantorovich relaxation of the OT problem as follows:

$$\mathcal{L}_c(\alpha, \beta) \coloneqq \min_{P \in U(\boldsymbol{a}, \boldsymbol{b})} \langle C, P \rangle\,, \tag{2.50}$$

for cost matrix $C \in R_+^{N \times M}$ with $C_{ij} = c(\boldsymbol{x}_i, \boldsymbol{y}_j)$ and $\langle C, P \rangle \coloneqq \sum_{ij} C_{ij} P_{ij}$. Following Peyré and Cuturi [221], we make the dependency of the OT problem $\mathcal{L}_c$ on the cost function c explicit. In case the cost function c is related to a proper distance metric d via

$c(\boldsymbol{x}_i, \boldsymbol{y}_j) = d^p(\boldsymbol{x}_i, \boldsymbol{y}_j)$, i.e. the cost is given by the p-th power of a distance metric, we define the *p-Wasserstein distance* as

$$W_p(\alpha, \beta) = \mathcal{L}_{d^p}(\alpha, \beta)^{1/p}\,, \tag{2.51}$$

where $\mathcal{L}_{d^p}$ is defined as in Equation (2.50). We refer to Peyré and Cuturi [221] for an overview and to Chen et al. [223] for a single-cell application of Wasserstein distances to describe patient-level variation in terms of single-cell gene expression.

The objective function defined in Equation (2.50) is linear with constraints given by the $N + M$ equality constraints imposed though $U(\boldsymbol{a}, \boldsymbol{b})$, thus it defines a convex linear program whose solution is in general non-unique. Various strategies have been suggested to solve Equation (2.50), among them network flow solvers and the auction algorithm [224], however, all of these remain limited to finding a solution in time $\mathcal{O}(N^3)$ for $M = N$, omitting logarithmic factors. This poses a scalability issue for applications in current single-cell genomics datasets which frequently contain hundreds of thousands of cells. Further, practical limitations include the difficulty to adapt these algorithms to run on GPUs and to be differentiable.

Entropic regularization. To overcome these practical limitations, consider the following entropically regularized [225] variant of the OT problem,

$$\mathcal{L}_c^\epsilon(\alpha, \beta) \coloneqq \min_{P \in U(\boldsymbol{a}, \boldsymbol{b})} \langle P, C \rangle - \epsilon H(P)\,, \tag{2.52}$$

for regularization strength $\epsilon > 0$ and entropy term

$$H(P) \coloneqq -\sum_{ij} P_{ij} \left(\log P_{ij} - 1\right)\,. \tag{2.53}$$

In contrast to the unregularized problem, Equation (2.52) is ϵ-strongly convex and thus possesses a unique global optimum. Further, for $\epsilon \to 0$ and $\epsilon \to \infty$, Peyré and Cuturi [221] and Cominetti and Martín [226] show the following asymptotic results:

$$\mathcal{L}_c^\epsilon(\alpha, \beta) \to \mathcal{L}_c \text{ for } \epsilon \to 0\,, \tag{2.54}$$

$$P_\epsilon \to \boldsymbol{a}\boldsymbol{b}^\top \text{ for } \epsilon \to \infty\,, \tag{2.55}$$

where P_ϵ refers to the solution of the regularized problem with regularization parameter ϵ. In particular, these results show that for small ϵ, the regularized problem approximates the unregularized problem while for large ϵ, the coupling converges to the outer product

of the two marginals, which is closely related to the *Maximum Mean Discrepancy* distance commonly used in generative models for distribution matching, see Lotfollahi et al. [40] for a single-cell application. To solve the regularized OT problem of Equation (2.52), we reproduce the following proposition from Peyré and Cuturi [221]:

Proposition 2.1 (Solution to the regularized OT problem)**.** *The unique solution to the regularized OT problem introduced in Equation* (2.52) *can be written as*

$$P_{ij} = u_i K_{ij} v_j \quad \forall (i,j) \in \{1,...,N\} \times \{1,...,M\}\,, \tag{2.56}$$

for the associated Gibbs kernel $K_{ij} = \exp\left(-C_{ij}/\epsilon\right)$ *and scaling variables* $(\boldsymbol{u}, \boldsymbol{v}) \in \mathbb{R}_+^N \times \mathbb{R}_+^M$ *to be inferred.*

Proof. The Lagrangian associated with the regularized OT problem of Equation (2.52) reads

$$\mathcal{E}(P, f, g) = \langle P, C\rangle - H(P) - \langle \boldsymbol{f}, P\mathbf{1}_M - \boldsymbol{a}\rangle - \left\langle \boldsymbol{g}, P^\top \mathbf{1}_N - \boldsymbol{b}\right\rangle\,, \tag{2.57}$$

for Lagrange multipliers $(\boldsymbol{f}, \boldsymbol{g}) \in \mathbb{R}^N \times \mathbb{R}^M$. Setting the first derivative with respect to the coupling matrix equal to zero (first order optimality conditions) yields

$$\frac{\partial \mathcal{E}}{\partial P_{ij}} = C_{ij} + \epsilon \log P_{ij} - f_i - g_j \stackrel{!}{=} 0\,, \tag{2.58}$$

which is equivalent to

$$P_{ij} = \mathrm{e}^{f_i/\epsilon} \mathrm{e}^{-C_{ij}/\epsilon} \mathrm{e}^{g_j/\epsilon} = \mathrm{e}^{f_i/\epsilon} K \mathrm{e}^{g_j/\epsilon}\,, \tag{2.59}$$

thus $u_i = \mathrm{e}^{f_i/\epsilon}$ and $v_j = e^{g_j/\epsilon}$. □

Sinkhorn algorithm. The form of the solution outlined in Proposition 2.1 can be used to construct an algorithm that iterates between scaling the rows and the columns of a candidate matrix. Impose therefore the marginal constraints of the feasible set $U(\boldsymbol{a}, \boldsymbol{b})$ on a solution in the form of Proposition 2.1,

$$\mathrm{diag}\left(\boldsymbol{u}\right) K \mathrm{diag}\left(\boldsymbol{v}\right) \mathbf{1}_M = \boldsymbol{a}\,, \tag{2.60}$$

$$\mathrm{diag}\left(\boldsymbol{v}\right) K^\top \mathrm{diag}\left(\boldsymbol{u}\right) \mathbf{1}_N = \boldsymbol{b}\,, \tag{2.61}$$

which can be simplified to give

$$\boldsymbol{u} \odot (K\boldsymbol{v}) = \boldsymbol{a}\,, \tag{2.62}$$

$$\boldsymbol{v} \odot (K^\top \boldsymbol{u}) = \boldsymbol{b}\,, \tag{2.63}$$

with $\odot$ denoting elements-wise multiplication. Iteratively solving these equations gives rise to *Sinkhorn's algorithm*,

$$\boldsymbol{u}^{(l+1)} := \frac{\boldsymbol{a}}{K\boldsymbol{v}^{(l)}}\,, \tag{2.64}$$

$$\boldsymbol{v}^{(l+1)} := \frac{\boldsymbol{b}}{K^\top \boldsymbol{u}^{(l+1)}}\,, \tag{2.65}$$

where the division is applied element-wise. Yule [227] originally suggested iterations of this form, Sinkhorn [228] proofed their convergence and Cuturi [225] suggested applying the algorithm to solve entropically regularized OT problems which gives rise to a differentiable solution. Note that Sinkhorn's algorithm is well suited to run on GPUs since it only relies on matrix vector products.

Cuturi et al. [229] recently used JAX [230] to implement Sinkhorn's algorithm in their Optimal Transport Tools (OTT) software package; OTT thus allows for just-in-time compilation, GPU acceleration, online cost function evaluation, and automatic differentiation. In contrast to the linear programming algorithms introduced above, Sinkhorn runs in time $\mathcal{O}(N^2)$ for $N = M$, omitting logarithmic factors. Thus, solving the regularized (Equation (2.52)) rather than the unregularized (Equation (2.50)) OT problem offers both practical and theoretical advantages. In Chapter 4, we introduce `moscot-time` and `moslin`, two methods which makes use of OTT's Sinkhorn implementation to link cells across time points. OTT's superior implementation translates into `moscot-time` outperforming previous methods that link cells across time both in terms of compute time and memory requirements by a large margin.

2.3.2 Extensions of optimal transport

Applications in single-cell genomics require two further extensions of the regularized OT problem, we follow Peyré and Cuturi [221] in their presentation. First, cells proliferate and die while they differentiate which should be reflected in the marginal distributions $\boldsymbol{a}$ and $\boldsymbol{b}$. However, as the rates of cellular growth and death are difficult to estimate based on scRNA-seq data, we relax the marginal constraints which leads to *unbalanced optimal transport*. Second, cells measured by different experimental technologies, such as spatial and non-spatial assays, reside in different metric spaces and cannot be compared directly

via a cost function c. Thus, we allow for pairwise comparisons only within the source and target spaces which leads to *Gromov-Wasserstein optimal transport* (GW).

Unbalanced optimal transport. To relax the marginal constraints, consider a generalization of the regularized OT problem by adding divergences D_ϕ between marginal constraints and row/columns sums of P to the objective function,

$$\mathcal{L}_c^{\epsilon,\tau}(\alpha,\beta) \coloneqq \min_{P\in\mathbb{R}_+^{N\times M}} \langle C,P\rangle + \tau_1 D_\phi(P\mathbf{1}_M|\boldsymbol{a}) + \tau_2 D_\phi(P^\top\mathbf{1}_N|\boldsymbol{b}) - \epsilon H(P)\,, \tag{2.66}$$

where the parameters $\tau_1,\tau_2>0$ control the weight given to the soft marginal constraints [221, 231]. This is a generalization of the original entropic OT problem of Equation (2.52) as in the limit $\tau_1,\tau_2\to\infty$, one recovers the original problem. A generalized version of the Sinkhorn algorithm may be applied to solve Equation (2.66); of particular importance in practical applications [20] has been the case $D_\phi(\cdot|\cdot) = \mathrm{KL}\left[\cdot\,||\,\cdot\right]$ for which the Sinkhorn updates read

$$\boldsymbol{u}^{(l+1)} \coloneqq \left(\frac{\boldsymbol{a}}{K\boldsymbol{v}^{(l)}}\right)^{\frac{\tau_1}{\tau_1+\epsilon}}\,, \tag{2.67}$$

$$\boldsymbol{v}^{(l+1)} \coloneqq \left(\frac{\boldsymbol{b}}{K^\top\boldsymbol{u}^{(l+1)}}\right)^{\frac{\tau_2}{\tau_2+\epsilon}}\,. \tag{2.68}$$

We refer to Liero, Mielke, and Savaré [231] for a treatment of the theory behind unbalanced OT and to Chizat et al. [232] for the derivation of practical algorithms. The generalized Sinkhorn algorithm provides an efficient solution to unbalanced OT problems which frequently arise in single-cell genomics [20].

Gromov-Wasserstein optimal transport (GW). In standard OT, we assume that point clouds $\{\boldsymbol{x}_1,...,\boldsymbol{x}_N\}$ and $\{\boldsymbol{y}_1,...,\boldsymbol{y}_M\}$ corresponding to bins in the histograms $\boldsymbol{a}$ and $\boldsymbol{b}$ may be compared using the cost function $c(\cdot,\cdot)$, i.e. $\mathcal{X}$ and $\mathcal{Y}$ correspond to the same metric space, giving rise to the cost matrix C. To relax this assumption, consider a situation where vectors $\{\boldsymbol{x}_i\}_{i=1}^N$ may be compared using the cost function $c^{\mathcal{X}}$ and vectors $\{\boldsymbol{y}_i\}_{i=1}^M$ may be compared using the cost function $c^{\mathcal{Y}}$, but no direct comparisons of vectors in $\mathcal{X}$ and $\mathcal{Y}$ are possible. Using these cost functions, we define the entropically regularized GW problem,

$$\mathcal{L}^\epsilon_{L,c^{\mathcal{X}},c^{\mathcal{Y}}}(\alpha,\beta) \coloneqq \min_{P\in U(\boldsymbol{a},\boldsymbol{b})} \sum_{ijkl} L\left(C^{\mathcal{X}}_{ij}, C^{\mathcal{Y}}_{kl}\right) P_{ik}P_{jl} - \epsilon H(P)\,, \tag{2.69}$$

for cost matrices $C^{\mathcal{X}} \in R_+^{N\times N}$, $C^{\mathcal{Y}} \in R_+^{M\times M}$ with $C^{\mathcal{X}}_{ij} = c^{\mathcal{X}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $C^{\mathcal{Y}}_{kl} = c^{\mathcal{Y}}(\boldsymbol{y}_k, \boldsymbol{y}_l)$ and distance metric L. The solution to (the unregularized version of) Equation (2.69) defines the *Gromov-Wasserstein distance* between two metric spaces, each equipped with a probability distribution. This distance has been introduced by Mémoli [233] as an extension to the Gromov-Hausdorff distance [234], combined with entropic regularization by Peyré, Cuturi, and Solomon [235] and Solomon et al. [236] and used in the single-cell field e.g. for data integration across modalities [237]. Equation (2.69) defines a non-convex, constrained, smooth optimization problem in P; before discussing its optimization, we introduce one further generalization.

Fused Gromov-Wasserstein optimal transport. In many practical applications, one encounters problems that possess characteristics of both OT (Equation (2.52)) and GW (Equation (2.69)); consider for example spatial imputation of gene expression data: given gene expression values in $\mathcal{X}$ and spatial coordinates in $\mathcal{Y}$, these cannot be compared directly and require a GW treatment. However, when expression values for a few genes are also known in the spatial domain (e.g. through a spatial transcriptomics assay [238]), the problem of mapping cells based on gene expression similarity between the two domains takes an OT form [220, 239]. Thus, problems of this kind require a combined objective function where some sampled features may be compared across spaces while others may only be compared within one space (Figure 2.4b,c). This kind of problem is known as *Fused Gromov-Wasserstein* (FGW), defined as

$$\mathcal{L}^{\epsilon,\alpha}_{L,c^{\mathcal{X}},c^{\mathcal{Y}},c}(\alpha,\beta) \coloneqq \min_{P\in U(\boldsymbol{a},\boldsymbol{b})} \alpha \sum_{ijkl} L\left(C^{\mathcal{X}}_{ij}, C^{\mathcal{Y}}_{kl}\right) P_{ik}P_{jl} + (1-\alpha)\sum_{ik} C_{ik}P_{ik} - \epsilon H(P)\,, \quad (2.70)$$

where $\alpha \in [0,1]$ controls the weight given to the OT versus GW terms, the within-space cost functions $c^{\mathcal{X}}$ and $c^{\mathcal{Y}}$ are defined for these features $\{\boldsymbol{x}_i\}_{i=1}^N$ and $\{\boldsymbol{y}_i\}_{i=1}^M$ which may not be compared across spaces and the across-space cost function c is defined for these features $(\boldsymbol{x}'_i, \boldsymbol{y}'_i)$ which may be compared across spaces [240]. For $\alpha = 1$, we recover the GW problem introduced above.

Introducing the 4-tensor [235],

$$\mathcal{T}(C^{\mathcal{X}}, C^{\mathcal{Y}})_{ijkl} \coloneqq L\left(C^{\mathcal{X}}_{ij}, C^{\mathcal{Y}}_{kl}\right)\,, \quad (2.71)$$

allows us to rewrite Equation (2.70) in shorter form,

$$\mathcal{L}^{\epsilon,\alpha}_{L,c^{\mathcal{X}},c^{\mathcal{Y}},c}(\alpha,\beta) \coloneqq \min_{P\in U(\boldsymbol{a},\boldsymbol{b})} \alpha \left\langle \mathcal{T}(C^{\mathcal{X}}, C^{\mathcal{Y}}) \otimes P, P\right\rangle + (1-\alpha)\left\langle C, P\right\rangle - \epsilon H(P) \quad (2.72)$$

with tensor multiplication defined via

$$(\mathcal{T} \otimes P)_{ij} \coloneqq \sum_{kl} \mathcal{T}_{ijkl} P_{kl} \,. \tag{2.73}$$

This tensor product may be computed in time $\mathcal{O}(N^3)$ for $M = N$ for a class of separable loss functions L, including l_2 loss and the KL divergence [235].

FGW optimization. Following Peyré, Cuturi, and Solomon [235], we use projected gradient descent with iterations

$$P^{(l+1)} = \mathrm{Proj}^{\mathrm{KL}}_{U(\boldsymbol{a},\boldsymbol{b})}\Big(P^{(l)} \odot e^{-\tau \nabla J(P)|_{P^{(l)}}}\Big)\,, \tag{2.74}$$

where $\mathrm{Proj}^{\mathrm{KL}}_{U(\boldsymbol{a},\boldsymbol{b})}(\tilde{P}) = \mathrm{argmin}_{P \in U(\boldsymbol{a},\boldsymbol{b})} \sum_{ij} P_{ij} \log\Big(P_{ij}/\tilde{P}_{ij}\Big)$ is a KL projection operator, τ is a step size, J is the FGW objective function defined in Equation (2.70) and $\odot$ denotes element-wise multiplication. The gradient of the objective function may be written as

$$\nabla J = (1-\alpha)C + \alpha \mathcal{T}\big(C^{\mathcal{X}}, C^{\mathcal{Y}}\big) \otimes P\,, \tag{2.75}$$

while the KL projection can be solved via an OT problem [241],

$$\mathrm{Proj}^{\mathrm{KL}}_{U(\boldsymbol{a},\boldsymbol{b})}(\tilde{P}) = \underset{P \in U(\boldsymbol{a},\boldsymbol{b})}{\mathrm{argmin}} \Big\langle -\epsilon \log \tilde{P}, P \Big\rangle - \epsilon H(P)\,. \tag{2.76}$$

Using Equation (2.75), Equation (2.76) and setting $\tau = 1/\epsilon$, we can re-write the update rule of Equation (2.74) as

$$P^{(l+1)} = \underset{P \in U(\boldsymbol{a},\boldsymbol{b})}{\mathrm{argmin}} \Big\langle (1-\alpha)C + \alpha \mathcal{T}\big(C^{\mathcal{X}}, C^{\mathcal{Y}}\big) \otimes P^{(l)}, P \Big\rangle - \epsilon H(P)\,, \tag{2.77}$$

which is the entropically regularized OT problem of Equation (2.52), solved efficiently at each iteration for an evolving cost matrix using the Sinkhorn algorithm [220, 225, 235]. The algorithm outlined here is applicable to both GW ($\alpha = 1$) and FGW $\alpha \in (0, 1)$) settings. The major computational bottleneck is the update of the tensor product of Equation (2.73) required at each update of Equation (2.77), which runs in time $\mathcal{O}(N^3)$ for $N = M$.

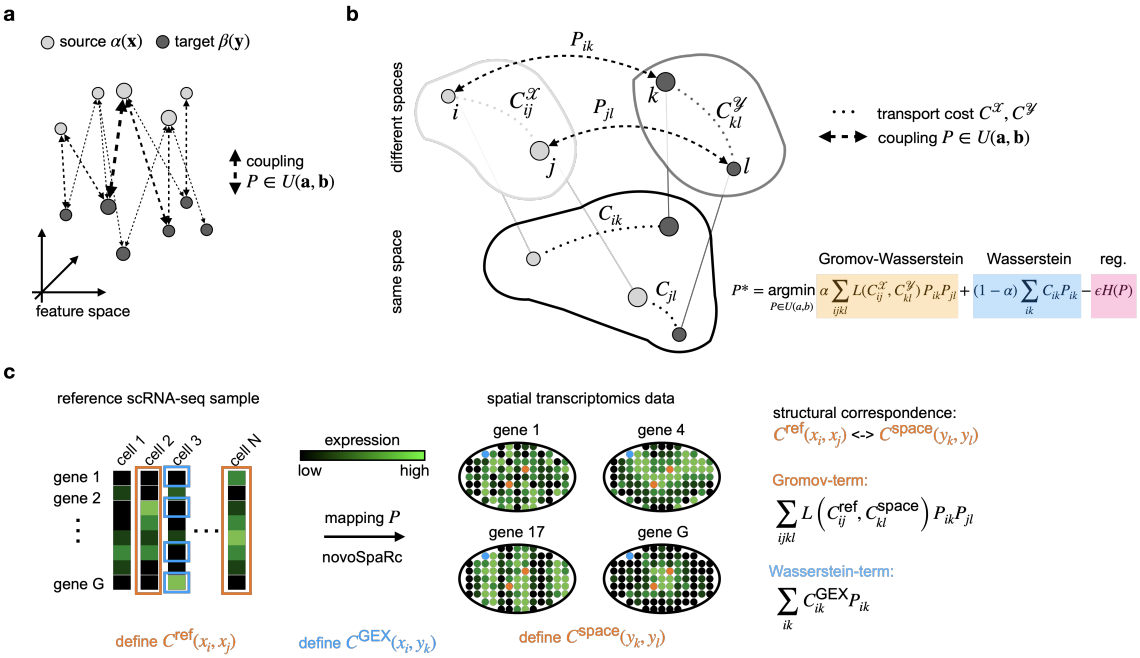

Figure 2.4: Optimal transport for single-cell genomics. a. Optimal transport problem between source $\alpha(\boldsymbol{x})$ and target $\beta(\boldsymbol{y})$ distributions [221]. Circle area denotes probability mass in the source $\boldsymbol{a}$ and target $\boldsymbol{b}$ histograms. The Kantorovich relaxation seeks to find a coupling matrix P from the set of feasible couplings $U(\boldsymbol{a}, \boldsymbol{b})$ which is optimal with respect to a cost function. **b.** Illustration of the objective function for a Fused Gromov-Wasserstein problem, defined in terms of both incomparable (top) and comparable (bottom) spaces. While direct sample-level comparisons are possible for features in the same space (expressed through cost matrix C), only within-space comparisons are possible for features in incomparable spaces (expressed through cost matrices $C^{\mathcal{X}}$ and $C^{\mathcal{Y}}$). **c.** Spatial mapping as formulated in novoSpaRc [220] provides a single-cell example for a Fused Gromov-Wasserstein problem. Cells in the scRNA-seq reference (left) are mapped to a tissue geometry (right) on the basis of a structural correspondence assumption between gene expression and physical distance (Gromov-term) as well as gene expression similarity (Wasserstein-term). Panel (**c**) inspired by Nitzan et al. [220]. reg., entropic regularization; GEX, gene expression; ref., reference.

2.4 Trajectory inference learns continuous representations from snapshot data

Single-cell assays like scRNA-seq provide unbiased measurements of cellular state. In an ideal scenario, one would be able to repeatedly apply these assays to the same cell to study the resulting trajectory. However, scRNA-seq and many related assays are destructive; each cell can only be measured once. Thus, to study continuous biological processes with these techniques, computational methods must be employed which reconstruct the dynamics of

a *typical cell* from snapshot measurements of many different cells, a problem known as *trajectory inference* (TI). TI problems have received considerable interest in the single-cell community early on and accordingly, many methods have been developed for this task. Here, we limit our attention to methods that are widely used and which are relevant for Chapter 3. We refer to Saelens et al. [12] and Deconinck et al. [242] for comprehensive method benchmarks and to Tritschler et al. [13] for an in-depth introduction to the concept of TI.

Basic idea of TI. TI methods make use of the fact that many biological processes unfold unsynchronized across cells, therefore, even when sequenced at the same wall-clock time, cells are in different stages internally and these stages can be aligned to reconstruct the process at hand. Different methods aim to reconstruct different types of biological processes - while early methods focused on one-dimensional, linear trajectories, later methods extended the focus to more complex, branching, and multifurcating trajectories via discrete and, more recently, probabilistic assignments (Figure 2.5a-c). The concepts introduced here can be used to study different biological processes including differentiation, reprogramming, regeneration, and cancer. However, they do not provide information about the *direction* of the biological process and thus remain limited to situations where the direction of the process has been established otherwise. In practice, this limits their applicability mostly to normal development. We extend upon these methods in Chapter 3 where we explicitly incorporate directionality.

2.4.1 Early methods focus on linear trajectory structure

Given a matrix of cell state measurements, i.e. $X^{(R)}$, sequenced at one or more experimental time-points, linear TI estimates a one-dimensional *pseudotime coordinate* $\tau \in [0, 1]$, where 0 and 1 correspond to the beginning and end of the biological process, respectively. Among the earliest methods to estimate pseudotime were Wanderlust [18] and Diffusion Pseudotime (DPT) [213]. Both methods compute pseudotime by constructing biologically meaningful distance measures using random walks on the phenotypic manifold. To approximate the phenotypic manifold, both methods start with KNN similarity graph construction (Subsection 2.1.2), however, they differ in how they measure distances along this graph. While Wanderlust uses iteratively refined shortest path distances from a set of sampled *waypoint cells*, DPT adapts the diffusion distance from Coifman and Lafon [127] (Subsection 2.2.5) to be *scale-free*, i.e. it averages over random walks of all possible lengths.

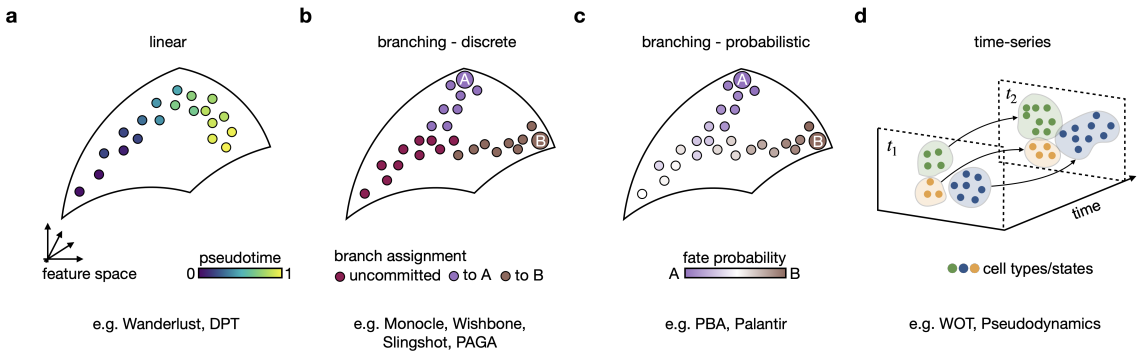

Figure 2.5: Trajectory inference in single-cell genomics. **a-c.** Coordinate system indicates (low-dimensional) feature representation, black outline denotes state manifold, dots denote cells colored according to pseudotime (**a**), discrete branch assignment (**b**) or probabilistic branch assignment (**c**). **a.** Pseudotime approaches infer a latent time-assignment; examples: Wanderlust [18] and DPT [213]. **b.** Discrete branching approaches infer a backbone trajectory that cells are assigned to; they assume the existence of a discrete "branching point". Cells before this point are labeled as "uncommitted". We show two hypothetical terminal states A and B as well as cells differentiating towards them. Examples: Monocle 1-3 [27, 92, 243], Wishbone [244], Slingshot [245] and PAGA [175]. **c.** Probabilistic branching approaches circumvent the assumption of a discrete branching point by computing a probabilistic branching probability, also called fate probability, per cell. Examples: PBA [214] and Palantir [25]. **d.** Temporal approaches relate cells measured at two or more experimental time points. Examples: WOT [20] and pseudodynamics [246].

Thus, we consider $T^{(\mathrm{sym})^t}$ for $t \to \infty$ and define the *accumulated transition matrix* M as

$$M = \sum_{t=1}^{\infty} \left(T^{(\mathrm{sym})} - \boldsymbol{\phi}_1 \boldsymbol{\phi}_1^\top \right)^t = \sum_{i=2}^{N_c} \left(\frac{\gamma_i}{1-\gamma_i} \right) \boldsymbol{\phi}_i \boldsymbol{\phi}_i^\top \,. \tag{2.78}$$

The geometric series in Equation (2.78) would not converge had we not removed the eigenspace corresponding to eigenvalue $\gamma_1 = 1 - \lambda_1 = 1 - 0 = 1$ [213]. Using the accumulated transition matrix M, DPT distance between cells i and j is defined via

$$D^2_{\mathrm{dpt}}(i,j) \coloneqq ||M_{i,:} - M_{j,:}||^2 = \sum_{k=2}^{N_c} \left(\frac{\gamma_k}{1-\gamma_k} \right)^2 (\phi_{ki} - \phi_{kj})^2 \,. \tag{2.79}$$

Pseudotime can now be defined for each cell i by computing DPT distance from a manually annotated root cell i_0, i.e.

$$\tau_i = D_{\mathrm{dpt}}(i_0, i) \; \forall i \,, \tag{2.80}$$

with appropriate normalization such that $\tau_i \in [0,1] \, \forall i$. In contrast to other scale-free

graph-based distance measures like the *commute time*, DPT has more favorable asymptotic properties as for $N_c \to \infty$, the commute time only conveys information about local density rather than data geometry [213, 247].

2.4.2 Discrete models of branching

Pseudotime inference allows cells to be placed along a one-dimensional trajectory, this limits the scope of such methods to "linear" biological processes. To model biological processes where cells can choose among one or more *fates* or *branches*, a multitude of extensions has been proposed; these usually follow a common pattern:

(i) project cells into a low dimensional space Z (Subsection 2.1.2),

(ii) construct a *backbone trajectory*, i.e. a graph that represents the average deterministic part of the observed dynamics,

(iii) align cells to this graph, either to the nodes, to the edges, or to a mixture of both.

We introduce the most common methods below in terms of these three major steps.

Cell-based: Monocle 1, Monocle 2, and Wishbone. The following methods construct the backbone trajectory (ii) directly on the level of sampled cells. Monocle [27] performs independent component analysis (ICA) to embed the data in a low dimensional space (i), constructs a minimum spanning tree (MST) in this space, and computes the longest path through the MST (MST diameter), which serves as the trajectory backbone (ii). To compute pseudotime, geodesic distance along the MST diameter is computed, making use of PQ trees [248] to handle uncertainty in the assignment (iii). Branches are detected by traversing the PQ tree for alternative paths. Monocle has been extended in version 2 [243] to use reversed graph embedding [249] to define a principal graph that defines the backbone trajectory (improves step ii). Wishbone [244] uses diffusion maps (Subsection 2.2.5) for dimensionality reduction (i), defines the branched trajectory backbone through normalized graph cut segmentation (ii), and uses Wanderlust to compute a pseudotime which is refined after branch assignment (iii).

Cluster-based: Slingshot, PAGA and Monocle 3. The following methods construct the backbone trajectory on the level of clusters that aggregate individual cells. Slingshot [245] can be coupled with any dimensionality reduction technique (i) and also with any

clustering method to group cells in Z space. An MST computed among the set of cell clusters (using a covariance-scaled euclidean distance measure among clusters means) serves as the trajectory backbone (ii). Going back from cluster- to cell-level, slingshot extends principal curves [250] to *simultaneous principal curves* which are fit to the cell sets corresponding to each branch in the MST. Pseudotime is assigned by orthogonal projection of cells onto the curves (iii). Simultaneous fitting of principal curves ensures that pseudotime values are consistent across principal curves prior to branching events.

Similar to slingshot, Probabilistic Approximate Graph Abstraction (PAGA) [175] allows any dimensionality reduction and clustering technique to be used (i). A KNN graph $\mathcal{G}$ (Subsection 2.1.2) is abstracted to a cluster-level graph $\mathcal{G}^*$ by comparing observed inter-cluster edges with expected inter-cluster edges under a model of random edge allocation. The abstracted graph $\mathcal{G}^*$ contains high-confidence edges among clusters and serves as the trajectory backbone (ii). The authors suggest computing pseudotime by adapting DPT to the case of disconnected graphs, essentially extending Equation (2.79) to the case of several eigenvectors ϕ_i corresponding to eigenvalue 1 [175]. In Monocle 3 [92], the authors build on these ideas by embedding cells in a low-dimensional UMAP space [121, 122] (i), clustering cells, and computing an abstracted PAGA graph (ii) which serves as a guide to principal graph construction using the SimplePPT algorithm [251]. To compute pseudotime, cells are projected onto the principal graph (iii).

2.4.3 Probabilistic models of branching

The TI methods introduced above generalize the simple pseudotime approaches we saw first to branched topologies, however, they model fate decisions as discrete events in time via hard assignments of cells to the branches. Before the decision point, usually referred to as *branching point*, cells are assigned to the same branch whereas after, they reside on different branches. In contrast, recent epigenetic studies [252–254] support a continuous interpretation of fate decisions: cells gradually transition from *multipotent*, i.e. naive stages, towards *unipotent*, i.e. differentiated stages. During this transition, various epigenetic modifications gradually restrict fate choice until cells reside on a path leading to a single fate outcome. Recent computational methods cater to this change in paradigm and model fate choice as a continuous process.

PBA connects the Fokker-Planck equation to spectral graph theory. Population Balance Analysis (PBA) [214] describes continuum cellular dynamics via a Fokker-Plack

equation,

$$\frac{\partial c}{\partial t} = \underbrace{\frac{1}{2} D \nabla^2 c}_{\text{diffusion}} + \underbrace{\nabla(c \nabla F)}_{\text{drift}} + \underbrace{Rc}_{\text{sources and sinks}}, \tag{2.81}$$

for cell state density $c(x,t) : \mathcal{X} \times [t_0, t_{\max}] \to \mathbb{R}_+$ with state-coordinate $x \in \mathcal{X}$ for gene expression space $\mathcal{X} \subset \mathbb{R}^{N_g}$ and time-coordinate $t \in [t_0, t_{\max}]$. Equation (2.81) describes changes in cell-state density $c(x,t)$ over an infinitesimal volume in gene expression space (left hand side) via three terms on the right hand side:

(i) diffusion term: stochastic fluctuations in gene expression which drive cells from high-density to low-density regions with diffusion strength $D \in \mathbb{R}_+$, assumed to be a constant.

(ii) drift term: directed motion of cells according to the potential function $F : \mathcal{X} \to \mathbb{R}$, which can be seen as a mathematical expression of Waddington's epigenetic landscape [255].

(iii) sources and sinks term: cellular growth (e.g. proliferation) and death (e.g. apoptosis) at state-dependent rate $R(x) : \mathcal{X} \to \mathbb{R}$.

To arrive at this description, PBA makes the following assumptions:

(i) cellular dynamics are Markovian, only the current state $x \in \mathcal{X}$ is informative for predicting future states (Section 2.2). This implies that x, i.e. the gene expression state, encodes all there is to know about the cell, in particular, there are no hidden variables.

(ii) there are no oscillatory gene expression dynamics, the directed part of the dynamics can thus be described via the gradient of the potential function F.

Violations to both assumptions are frequently observed in practice, i.e. epigenetic marks encode cellular memory (i) and the cell cycle or the circadian rhythm give rise to oscillatory gene expression dynamics (ii). However, given only gene expression samples from the cell-state density $c(x,t)$, avoiding these assumptions renders the problem of dynamical inference ambiguous - the same observed data can support several dynamical models [214].

In scRNA-seq experiments consisting of a single time point, we do not have access to the rate of change of cell state density $\partial c / \partial t$, thus, the authors make a third simplifying assumption:

(iii) The biological system is in steady-state, i.e. $\partial c/\partial t = 0$.

Despite these assumptions, solving Equation (2.81) exactly is impossible using current numerical solvers for data dimensionalities encountered in scRNA-seq data. Instead, the authors make use of a recent result from Ting, Huang, and Jordan [256] which relates Equation (2.81) to a Markov chain constructed on samples of $c(x, t)$. In particular, the authors build a KNN graph $\mathcal{G}$ and compute the graph laplacian matrix L^{rw} (Subsection 2.2.5). They define a potential $V = 1/2L^{(\text{rw})^+}R$, where $L^{(\text{rw})^+}$ denotes the pseudoinverse of $L^{(\text{rw})}$, and a Markov chain transition matrix $T \in R_+^{N_c \times N_c}$ with elements

$$T_{ij} = \mathrm{e}^{(V_i - V_j)/D}\,, \tag{2.82}$$

for all cells i and j which are neighbors in $\mathcal{G}$ and zero otherwise. Ting, Huang, and Jordan [256] show that in the limit $N_c \to \infty$, this Markov chain converges to the continuous process described in Equation (2.81). Equipped with Equation (2.82), the authors model gradual fate decisions by fixing a set of terminal cell states and computing absorption probabilities (Subsection 2.2.4) towards these. Thus, each cell is assigned a set of *fate probabilities* to transition towards any terminal state. This naturally allows cells to gradually transition from initial states (fate probabilities towards several terminal states) to terminal states (fate probability towards a single terminal state). PBA's main limitations are

(i) both R and D are unknown and difficult to estimate in practice.

(ii) many biological systems of interest are not in a steady state.

Palantir uses pseudotime for graph construction. Palantir [25] heuristically overcomes these limitations with an adjusted Markov chain construction based on precomputed KNN graph $\mathcal{G}$ with adjacency matrix W and pseudotime τ. In particular, Palantir biases graph edges in $\mathcal{G}$ to point in the direction of increasing pseudotime,

$$T_{ij} = \begin{cases} W_{ij} & \text{for } \tau_j \geq \tau_i - \sigma_i \\ 0 & \text{otherwise}\,, \end{cases} \tag{2.83}$$

for all cells i and j which are neighbors in $\mathcal{G}$ and zero otherwise. Palantir uses the local scaling parameter σ_i to reflect uncertainty in pseudotime inference by retaining some graph edges that point into the pseudotime past. Terminal states are identified via outliers in the invariant distribution $\boldsymbol{\pi}$ of T (Subsection 2.2.3). As in PBA, fate probabilities towards them are computed via absorption probabilities (Subsection 2.2.4). Palantir's main limitations

are

(i) the choice of pseudotime is restricted to Palantir's own pseudotime, limiting the method's application to situations where other pseudotime methods work better.

(ii) for pseudotime construction, a root cell must be manually provided by the user, limiting the application to datasets with unknown initial states.

(iii) Palantir's implementation of absorption probabilities does not scale well to large cell numbers.

(iv) Palantir ignores information provided by real experimental time points.

We show in Chapter 3 how limitations (i-iii) can be overcome and we outline below how other methods overcome limitation (iv).

2.4.4 Including temporal information

While some biological systems, including adult hematopoiesis [10] and mouse adult neurogenesis [257], are in a dynamical steady state, many others, including embryonic development [6, 258] and regeneration [11], are not, thus necessitating to sequence cells at different experimental time points in a *time-series experiment* (Figure 2.5d). The methods introduced above are not well adapted to this setting; while some are only applicable to steady-state systems [214], the majority simply ignore the extra information provided by experimental time labels [25, 175, 213].

To explicitly make use of this information, Schiebinger et al. [20] introduced Waddington Optimal Transport (WOT), a method which uses unbalanced OT (Subsection 2.3.2) to compute coupling matrices between pairs of time points. Let t_1 and t_2 represent such a pair of time points containing N and M cells, respectively. For cost matrix $C \in \mathbb{R}^{N\times M}$ containing l_2 distances in a local PCA representation computed for just the $N+M$ cells in the two time points (ignoring potential further time points), WOT solves

$$\mathcal{L}_c^{\epsilon,\tau}(\alpha,\beta) \coloneqq \min_{P\in\mathbb{R}_+^{N\times M}} \langle C,P\rangle + \tau_1 \mathrm{KL}\left[P\mathbf{1}_M \,||\, \boldsymbol{a}\right] + \tau_2 \mathrm{KL}\left[P^\top \mathbf{1}_N \,||\, \boldsymbol{b}\right] - \epsilon H(P)\,, \qquad (2.84)$$

using the generalized Sinkhorn algorithm of Subsection 2.3.2 with τ_1, τ_2, ϵ and H defined as in Section 2.3. While the right marginal $\boldsymbol{b}$ is chosen to be uniform, $b_j = 1/M\,\forall j \in \{1,...,M\}$, WOT adjusts the left marginal to accommodate for cellular growth and death

between t_1 and t_2,

$$a_i = \frac{g(\boldsymbol{x}_i)^{t_1 - t_2}}{\sum_{j=1}^{N} g(\boldsymbol{x}_j)^{t_1 - t_2}} \; \forall i \in \{1, ..., N\}\,, \tag{2.85}$$

where $g : \mathbb{N}^{N_g} \to \mathbb{R}$ is modeled as the expected value of a birth-death process with proliferation at rate $\beta(\boldsymbol{x})$ and death at rate $\delta(\boldsymbol{x})$, thus $g(\boldsymbol{x}) = \mathrm{e}^{\beta(\boldsymbol{x}) - \delta(\boldsymbol{x})}$ for $\beta(\boldsymbol{x})$ and $\delta(\boldsymbol{x})$ estimated from curated marker gene sets for proliferation and apoptosis, respectively. The unbalanced OT framework accounts for uncertainty in the estimation of g (Section 2.3); by default, τ_1 is chosen small ($\tau_1 \approx 1$) to allow variation from the adjusted left marginal $\boldsymbol{a}$ while τ_2 is chosen large ($\tau_2 \approx 50$) to strictly enforce the uniform right marginal $\boldsymbol{b}$. A sequence of time points can be coupled by computing pairwise coupling matrices and matrix-multiplying these to yield long-range couplings.

The WOT model has meanwhile been extended to incorporate prior information from RNA velocity [1, 16] (Section 2.5) via dynamical OT [259] and continuous normalizing flows [260] in TrajectoryNet [261]. Applying WOT in practice is challenging if there is a large number of cells (approximately greater than 10-20k) per time point due to both time and memory scaling quadratically in cell numbers. Further, while the (generalized) Sinkhorn algorithm can be efficiently executed on GPUs, WOT relies on a custom implementation that only runs on CPUs. In Chapter 4, we introduce `moscot-time`, an adaptation of the WOT model which overcomes these scalability limitations through both engineering-type innovations (e.g. GPU support, online cost-function evaluation) as well as recent theoretical innovations (low-rank factorizations [262, 263]), cumulating in linear time and memory complexity and the applicability to truly large datasets. Further, we extend the model to take into account experimental barcoding information [176–178] with `moslin`.

In Fischer et al. [246], the authors propose an alternative view of including temporal information with the *pseudodynamics* model; an approach that uses a convection-diffusion equation to describe the evolution of cell density in one-dimensional pseudotime space along experimental time including terms for cell growth and death. While this method is not concerned with mapping individual cells from earlier to later time points, it provides insights into deterministic versus stochastic aspects of T-cell maturation.

2.5 RNA velocity

The approaches to trajectory inference introduced in the previous section cover a wide range of biological use cases, from linear to bi- and multifurcating trajectories with either

discrete or probabilistic fate assignments. However, they fall short of assigning directions to the recovered trajectories - this is easy to illustrate in the case of DPT, where we had to manually provide the root cell, but it holds much more generally for all *similarity-based approaches* to TI. That is because the similarity between two cells, be it on the gene expression level, the chromatin accessibility level, or any other molecular level, does not reveal which cell is likely to transition into the other.

The assumption we make in similarity-based TI is that cellular state changes proceed gradually with many intermediate steps, thus we can use the ensemble of sampled snapshots to reconstruct the underlying continuum of gene expression changes by connecting and ordering cells that are similar. However, the user needs to define where this ordering should start, i.e. they have to provide the set of *initial states*. Further, in situations where similarity does not imply an actual transition, these TI methods will output false predictions. Thus, similarity-based TI is mostly limited to well-studied systems in normal development where initial states are known and prior knowledge can be used to prevent false predictions by guiding the analysis (Figure 2.6a).

For de-novo prediction of the direction of cellular state changes, Manno et al. [1] introduced RNA velocity based on a single-cell model of the mRNA life cycle. Genetic information on the DNA is structured into genes, each gene codes for one protein (ignoring the extra diversity achieved through post-transcriptional and post-translational modifications). Each gene is further divided into two kinds of genetic regions: *exons*, which are translated into the actual protein, and *introns*, which serve regulatory functions. Introns are removed from transcribed mRNA molecules in a process called *splicing*, thus, each mRNA exists in either the *unspliced* or *spliced* state (Figure 2.6b). Manno et al. [1] showed that all major scRNA-seq techniques capture both spliced and unspliced molecules, the exact ratio depends on the technology but spliced counts are much more frequent. Relating these two internal states to one another reveals the direction of gene regulation: high (low) unspliced to spliced count ratio is indicative of up- (down) regulation.

To formalize the notion introduced above, consider the following model of splicing kinetics:

$$\varnothing \underset{\alpha^{\text{on/off}}}{\longrightarrow} u \underset{\beta}{\longrightarrow} s \underset{\gamma}{\longrightarrow} \varnothing\,, \tag{2.86}$$

where unspliced molecules $u(t)$ are created with transcription rate $\alpha^{\text{on/off}}$ and converted into spliced molecules $s(t)$ with splicing rate β, which in turn are degraded with degradation rate γ. We make the assumption that the degradation of unspliced mRNA molecules $u(t)$ in the nucleus is negligible compared to the other reactions [264]. Let $\boldsymbol{\theta} = [\alpha^{\text{on/off}}, \beta, \gamma]^\top$ denote a vector of these parameters. While this model is gene-specific, we omit the corre-

sponding subscript j everywhere in this section and focus on just a single gene. Following Li [265], let $P_{mn}(t)$ be the probability to have m unspliced and n spliced molecules at time t, i.e.

$$P_{mn}(t) \coloneqq \mathbb{P}\left[(u(t), s(t)) = (m, n) \in \mathbb{N}^2\right] . \tag{2.87}$$

With this definition, we can describe the process illustrated in Equation (2.86) as a continuous time Markov process on the discrete state space $\mathcal{S} = \mathbb{N}^2$ with chemical master equation (CME) (Section A.2 in Appendix A) given by

$$\begin{aligned} \frac{dP_{mn}}{dt} &= \alpha^{\text{on/off}}(P_{m-1,n} - P_{m,n}) \\ &\quad + \beta\left[(m+1)P_{m+1,n-1} - mP_{mm}\right] \\ &\quad + \gamma\left[(n+1)P_{m,n+1} - nP_{mn}\right] . \end{aligned} \tag{2.88}$$

The CME is given by a set of infinitely many coupled ordinary differential equations (ODEs), each describing the probability evolution for one combination of spliced and unspliced molecules (m, n). In the following, we make use of this model to offer a unifying perspective on various approaches that have been introduced for RNA velocity analysis. We structure this into *forward models*, describing how observations (u, s) are generated given parameters, and *inference schemes*, describing how parameters $\boldsymbol{\theta}$ are computed given observations. For the remainder of this section, let $\hat{\cdot}$ denote a measured quantity and $\tilde{\cdot}$ an estimated parameter.

2.5.1 Forwards models

Two of the models we describe in the following require aggregate descriptions of the fully stoachstic CME dynamics, we therefore derive an equation for arbitrary uncentered moments of the system state $\langle m^l n^k \rangle$ by multiplying Equation (2.88) with $m^l n^k$ and summing over m and n,

$$\begin{aligned} \frac{d\langle m^l n^k \rangle}{dt} &= \sum_{m,n} \alpha^{\text{on/off}} \left((m+1)^l n^k - m^l n^k\right) P_{mn} \\ &\quad + \beta \left(m(m-1)^l (n+1)^k - m^{l+1} n^k\right) P_{mn} \\ &\quad + \gamma \left(n(n-1)^k m^l - m^l n^{k+1}\right) P_{mn} , \end{aligned} \tag{2.89}$$

which can be rearranged [16] to read

$$\begin{aligned}\frac{d\langle m^l n^k\rangle}{dt} &= \alpha^{\text{on/off}}\Big\langle (m+1)^l n^k - m^l n^k \Big\rangle \\ &+ \beta\Big\langle m\left((m-1)^l(n+1)^k - m^l n^k\right)\Big\rangle \\ &+ \gamma\Big\langle n\left((n-1)^k m^l - m^l n^k\right)\Big\rangle .\end{aligned} \tag{2.90}$$

We make use of this result in the following paragraphs for different values of l and k.

Deterministic - first-order moments. By using Equation (2.90) for first order moments with $m + l = 1$, we obtain,

$$\frac{d\langle m\rangle}{dt} = \alpha^{\text{on/off}} - \beta\langle m\rangle \quad \text{and} \quad \frac{d\langle n\rangle}{dt} = \beta\langle m\rangle - \gamma\langle n\rangle \ , \tag{2.91}$$

i.e. a deterministic version of the RNA velocity model we introduced in Equation (2.86) which may be used to define RNA velocity as follows,

$$v^{(\text{RNA})} = \frac{d\langle n\rangle}{dt} = \beta\langle m\rangle - \gamma\langle n\rangle \, . \tag{2.92}$$

The solution to the first order moment equations for $\beta, \gamma > 0$ and $\beta \neq \gamma$, subject to $u(0) = u_0, s(0) = s_0$, is given by

$$\begin{aligned}\langle u(t)\rangle &= u_0 \mathrm{e}^{-\beta t} + \frac{\alpha^{\text{on/off}}}{\beta}(1 - \mathrm{e}^{-\beta t}) \, , \\ \langle s(t)\rangle &= s_0 \mathrm{e}^{-\gamma t} + \frac{\alpha^{\text{on/off}}}{\gamma}\left(1 - \mathrm{e}^{-\gamma t}\right) + \frac{\alpha^{\text{on/off}} - \beta u_0}{\gamma - \beta}\left(\mathrm{e}^{-\gamma t} - \mathrm{e}^{-\beta t}\right) \, ,\end{aligned} \tag{2.93}$$

see Section A.3 in Appendix A for a derivation.

Including covariance - second-order moments. Including second order moments is a simple way to introduce some stochasticity to the system while ensuring the equations remain solvable in closed form. By using Equation (2.90) for second order moments with

$m + l = 2$, we obtain,

$$\frac{d\langle m^2 \rangle}{dt} = \alpha^{\text{on/off}} \langle 2m + 1 \rangle + \beta \langle m - 2m^2 \rangle \,, \tag{2.94}$$

$$\frac{d\langle mn \rangle}{dt} = \alpha^{\text{on/off}} \langle n \rangle + \beta \langle m^2 - mn - m \rangle - \gamma \langle mn \rangle \,, \tag{2.95}$$

$$\frac{d\langle n^2 \rangle}{dt} = \beta \langle 2mn + m \rangle + \gamma \langle n - 2n^2 \rangle \,. \tag{2.96}$$

These equations can be combined with first-order moment equations and solved in closed form, following the general procedure described in Section A.3 in Appendix A.

Fully stochastic - solving the CME. Rather than solving simplifying, aggregate descriptions of the system given by moment equations, the CME itself can be solved in closed form (Section A.4 in Appendix A). This yields a fully stochastic description of the system, however, it assumes absolute molecule numbers which we do not have access to in practice as scRNA-seq assays only sample a small fraction of available mRNA molecules in the cell (Section 2.1). Thus, one either has to introduce an additional sampling or scaling step to a CME-based model or revert to aggregate descriptions, given e.g. by the moment equations. We proceed with the latter option in the following description of parameter inference.

2.5.2 Parameter inference

In practice, we observe unspliced and spliced counts, $\hat{\boldsymbol{m}}, \hat{\boldsymbol{n}} \in \mathbb{N}^{N_c}$, for a particular gene j with no temporal resolution, i.e. we do not know which time t an observation $(\hat{m}_i, \hat{n}_i)$ corresponds to. Moments of different orders $\langle \hat{m}_i^l \hat{n}_i^k \rangle$ for fixed cell i are approximated by considering cells which are neighbors of i in a KNN graph $\mathcal{G}$. Let $\langle \hat{\boldsymbol{m}}^l \hat{\boldsymbol{n}}^k \rangle \in \mathbb{R}^{N_c}$ denote a vector of such empirical, graph-based moments. Empirical RNA velocity for cell i becomes

$$v_i^{(\text{RNA})} = \beta \langle \hat{m}_i \rangle - \gamma \langle \hat{n}_i \rangle \,. \tag{2.97}$$

Two approaches to fit parameters for the RNA velocity models have been explored, these are

(i) *steady-state approaches*, in which we suppose a (sub-) population of sampled cells is in steady-state. Model parameters can be inferred by fitting a simplified model with no temporal dependency on this population.

(ii) *dynamical approaches*, which simultaneously aim to infer the latent temporal assignment as well as model parameters, thus considering all cells, not just the ones in

steady-state.

Many biological systems involve either none or very few steady-state cells; even if they do involve these, it is difficult in practice to decide whether a given cell is in steady-state or not. Thus, while steady-state approaches are computationally easier to fit, they remain limited to fewer biological settings. In the following, we present two steady-state [1, 16] as well as one dynamical [16] approach to RNA velocity parameter inference based on moment equations.

Steady-state approximation for first-order moments. The simplest approach to solve the first order moment equations for the unknown model parameters, β and γ, is to nondimensionalize by dividing trough β and defining $\gamma' = \gamma/\beta$, $v'^{(\mathrm{RNA})} = v^{(\mathrm{RNA})}/\beta$. The steady state approximation yields

$$0 = v'^{(\mathrm{RNA})} = \langle m \rangle - \gamma' \langle n \rangle \,. \tag{2.98}$$

The parameter γ' can be estimated from empirical data through maximum likelihood estimation (MLE) with statistical model

$$p(\langle \hat{m}_i \rangle | \gamma', \langle \hat{n}_i \rangle) = \mathcal{N}\left(\langle \hat{m}_i \rangle | \gamma' \langle \hat{n}_i \rangle, \sigma^2\right) \,, \tag{2.99}$$

for constant variance σ^2. The associated negative log-likelihood for the entire dataset reads

$$\mathcal{L}(\gamma') \coloneqq -2 \log \Pi_{i=1}^{N_c} \mathcal{N}\left(\langle \hat{m}_i \rangle | \gamma' \langle \hat{n}_i \rangle, \sigma^2\right) = \sum_{i=1}^{N_c} \left(\langle \hat{m}_i \rangle - \gamma' \langle \hat{n}_i \rangle\right)^2 + \text{ const.} \tag{2.100}$$

Minimizing $\mathcal{L}$ with respect to γ' gives the solution

$$\tilde{\gamma}' = \frac{\sum_i \langle \hat{m}_i \rangle \langle \hat{n}_i \rangle}{\sum_i \langle \hat{n}_i^2 \rangle} = \frac{\langle \hat{\boldsymbol{m}} \rangle^\top \langle \hat{\boldsymbol{n}} \rangle}{\langle \hat{\boldsymbol{n}} \rangle^\top \langle \hat{\boldsymbol{n}} \rangle} \,. \tag{2.101}$$

Manno et al. [1] suggested this approach in their original RNA velocity publication and implemented it in the Velocyto software package.

Steady-state approximation for second-order moments. The second-order moment equation for $d\langle n^2 \rangle / dt$ in steady-state can be used as additional information when

estimating γ' [16], leading to the system

$$\langle m \rangle = \gamma' \langle n \rangle \,, \tag{2.102}$$

$$2\langle mn \rangle + \langle m \rangle = \gamma' \left(2\langle n^2 \rangle - \langle n \rangle \right) \tag{2.103}$$

which may be re-written as $\boldsymbol{y} = \gamma' \boldsymbol{x}$ for

$$\boldsymbol{y} \coloneqq \begin{bmatrix} \langle m \rangle \\ 2\langle mn \rangle + \langle m \rangle \end{bmatrix}, \qquad \boldsymbol{x} \coloneqq \begin{bmatrix} \langle n \rangle \\ 2\langle n^2 \rangle - \langle n \rangle \end{bmatrix} . \tag{2.104}$$

To estimate γ' from empirical data $\{\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{y}}_i\}_{i=1}^{N_c}$, consider the following model,

$$p(\hat{\boldsymbol{y}}_i | \gamma', \hat{\boldsymbol{x}}_i) = \mathcal{N}\left(\hat{\boldsymbol{y}}_i | \gamma' \hat{\boldsymbol{x}}_i, \Sigma_i\right) \,, \tag{2.105}$$

where we included the covariance matrix Σ_i to account for correlation among the components of $\boldsymbol{y}$. The negative log-likelihood for the entire dataset becomes

$$\mathcal{L}(\gamma') \coloneqq -2 \log \Pi_{i=1}^{N_c} \mathcal{N}\left(\hat{\boldsymbol{y}}_i | \gamma' \hat{\boldsymbol{x}}_i, \Sigma_i\right) = \sum_{i=1}^{N_c} (\hat{\boldsymbol{y}}_i - \gamma' \hat{\boldsymbol{x}}_i)^\top \Sigma_i^{-1} (\hat{\boldsymbol{y}}_i - \gamma' \hat{\boldsymbol{x}}_i) + \text{ const.} \tag{2.106}$$

where we treat Σ_i as constant because it is estimated prior to the fit from the residuals of an ordinary least squares (OLS) fit. Minimizing $\mathcal{L}$ with respect to γ' gives

$$\tilde{\gamma}' = \frac{\sum_i \hat{\boldsymbol{x}}_i^\top \Sigma_i^{-1} \hat{\boldsymbol{y}}_i}{\sum_i \hat{\boldsymbol{x}}_i^\top \Sigma_i^{-1} \hat{\boldsymbol{x}}_i} \,. \tag{2.107}$$

We suggested this approach in Bergen et al. [16] and implemented it in the scVelo software package.

EM inference for first-order moments. The steady-state approaches yield velocity estimates $v'^{(\mathrm{RNA})}$ which are not comparable across genes as the implicit scaling factor β varies among them. Further, they provide no information on the actual model parameters $\boldsymbol{\theta}$ but only on the ratio $\gamma' = \gamma/\beta$ and they remain limited to biological systems with large steady-state populations. Thus, in Bergen et al. [16], we suggest a dynamical model which fits the time-dependent solution to first order moment equations (Equation (2.91)) directly to empirical moments $\{\langle \hat{\boldsymbol{m}}_i \rangle, \langle \hat{\boldsymbol{n}}_i \rangle\}_{i=1}^{N_c}$ in an EM-framework [266] which iterates between optimizing model parameters $\boldsymbol{\theta}$ and latent time and state assignments t_i and k_i, respectively (Figure 2.6c). The states k_i denote the different phases of splicing kinetics, i.e. on-state, off-stage or steady-state.

In the M-step, given inferred latent time $\tilde{t}_i$ and state $\tilde{k}_i$ assignments, a normally-distributed likelihood given by

$$p(\hat{\boldsymbol{x}}_i|\boldsymbol{\theta}, \sigma^2, \tilde{t}_i, \tilde{k}_i) = \mathcal{N}\left(\hat{\boldsymbol{x}}_i|\boldsymbol{x}_{\boldsymbol{\theta},\tilde{k}_i}(\tilde{t}_i), \sigma^2 I\right), \tag{2.108}$$

is optimized for the model parameters $\boldsymbol{\theta}$, where $\hat{\boldsymbol{x}}_i \coloneqq [\langle \hat{m}_i \rangle, \langle \hat{n}_i \rangle]^\top$ denotes empirical moments, $\boldsymbol{x}_{\boldsymbol{\theta},\tilde{k}_i}(\tilde{t}_i) \coloneqq [\langle m_{\boldsymbol{\theta},\tilde{k}_i}(\tilde{t}_i)\rangle, \langle n_{\boldsymbol{\theta},\tilde{k}_i}(\tilde{t}_i)\rangle]^\top$ is the state $\tilde{k}_i$ and parameter $\boldsymbol{\theta}$ dependent solution to the first order moment equations (Equation (2.91)) at time point $\tilde{t}_i$ and σ is a gene specific variance term.

In the E-step, given parameter estimates $\tilde{\boldsymbol{\theta}}$, time t_i and state k_i assignments are computed by minimizing the distance between each observed sample $\hat{\boldsymbol{x}}_i$ and the phase-trajectory given by $\boldsymbol{x}_{\tilde{\boldsymbol{\theta}},k_i}(t_i)$.

Gene-specific time assignments are aggregated towards one *latent time* value per cell which serves as a pseudotime (Section 2.4). With RNA velocity defined as before (Equation (2.92)), estimating model parameters with the dynamical model ensures that velocities are comparable across genes by fixing the time scale of aggregated latent time. We suggested the dynamical model in Bergen et al. [16] and implemented it in the scVelo software package.

2.5.3 Downstream usage of velocities

RNA velocity yields an estimate of transcriptional regulation $v_{ij}^{(\mathrm{RNA})} = d\langle n_{ij}\rangle/dt$ for each gene j in every cell i which can be used to linearly extrapolate a cells current state into the future.

Visual inspection of projected velocity vectors. The original Velocyto [1] method projects velocities into two-dimensional (2D) embeddings where they are visualized as arrows that point towards a cell's future state. The projected vector fields are frequently used for visual trajectory inference, i.e. to determine what the eventual fate of a cell may be. This is problematic as 2D embeddings frequently obscure biological variation from the original space [161–164], projected vector fields are overly smooth, and do not permit uncertainty quantification and aggregation of local velocity information into global fate decisions by visual inspection is error-prone. The single-cell genomics field has reached a consensus that clustering cells in 2D or 3D representations must be avoided [130], and similarly, we argue that two or three-dimensional velocity projections must not be used to address detailed questions of trajectory inference.

Quantitative approaches to interpreting velocity vector fields. To find the initial and terminal states of the biological system, Velocyto defines a velocity-based Markov process which is iterated until convergence, an equivalent procedure to computing the invariant distribution $\boldsymbol{\pi}$ (Section 2.2). This approach yields a distribution where cells frequently visited by the Markov process are expected to have larger entries; however, it is unclear how individual initial and terminal states can be computed in such a way. Alternative approaches for quantitative analysis of velocity vectors have been suggested including Dynamo [267], VeloDyn, [268] and CellPath [269]. Dynamo learns a functional form of the velocity field using a sparse approximation to regularized vector field learning [270]. Methods from dynamical systems are applied to the reconstructed vector field to find initial and terminal states as well as fate probabilities, largely ignoring the stochastic nature of fate decisions and the uncertainty in velocity vectors. VeloDyn, which also uses dynamical systems approaches, takes velocity uncertainty into account via bootstrap sampling, however, it is limited to 2D PCA embeddings and cannot compute fate probabilities. CellPath computes trajectories in high-dimensional space via a sampling strategy that involves meta-cell aggregation, greedy trajectory selection, and custom pseudotime assignment. The algorithm is heuristic with no theoretical basis and ignores velocity uncertainty. In Chapter 3, we present CellRank, a method that systematically aggregates velocities into long-range fate predictions and computes individual initial and terminal states without relying on 2D embeddings.

2.5.4 Extensions and alternatives

RNA velocity is a proxy for the current state of transcriptional regulation which is fundamentally based on recovering both spliced and unspliced counts from the same cell. Recovering a sufficient amount of unspliced transcripts across many genes is difficult in practice as the processes of polyadenylation and splicing happen mostly simultaneously [271] (Section 2.1). As a consequence, the majority of unspliced counts are due to *internal priming* events where the poly(T) primer binds to poly(A) stretches in intronic regions of the unspliced transcript [1, 272, 273], rather than to the poly(A) tail as is the case for spliced transcripts. This means that the expression level of unspliced counts for a certain gene could depend on a number of factors, including

(i) the length of the gene [272],

(ii) the number and length of intronic regions within the gene,

(iii) the amount of poly(A) stretches within introns,

(iv) the relative position of poly(A) stretches within an intron,

all of which result in biases that are difficult to diagnose and impossible to control externally. If genes that are important for a certain biological process lack sufficient unspliced counts for any of the reasons outlined above, their velocity estimate will be wrong and the overall velocity of the cells in this system will be dominated by potentially uninformative genes with large numbers of unspliced counts. To circumvent this limitation, RNA velocity type models have been formulated for other molecular layers, and metabolic labeling has been adapted to the single-cell setting as an experimental alternative to RNA velocity.

Further limitations to RNA-velocity type models have been reviewed and assessed recently by Bergen et al. [274], Gorin et al. [275], and Marot-Lassauzaie et al. [276].

Extension to other molecular layers. The RNA velocity model has been extended to include protein [277] and ATAC [278] information by adding extra reactions to the model of Equation (2.86). Alternative formulations have been suggested for just chromatin accessibility [279] or multimodal RNA and ATAC data [70].

Metabolic labeling as an experimental alternative. Where RNA velocity relies on unspliced and spliced counts as a proxy for new and old mRNAs, metabolic labeling directly labels mRNA molecules produced after a certain time point t_0, thus creating an experimental label for new versus old mRNAs. The approach has been coupled to high-throughput sequencing in bulk [280–282] and computational models have been developed which correct for experimental measurement noise when quantifying the proportion of new versus old transcript counts per gene from sequenced reads [280, 283]. Further, metabolic labeling has been adapted for plate-based (scSLAM-seq [284], NASC-seq [285] and scEU-seq [286]), combinatorial indexing (sci-fate [287]) and droplet-based (scNT-seq [288]) single-cell technologies and used in RNA-velocity type models to estimate the state of gene regulation [267]. Currently, the main limitation of these technologies is that they are difficult to set up and limited to in-vitro systems. In contrast, RNA velocity requires no additional experimental work and can be applied to in-vivo systems. For a review of single cell metabolic labeling techniques, we refer to Olivares-Chauvet and Junker [182].

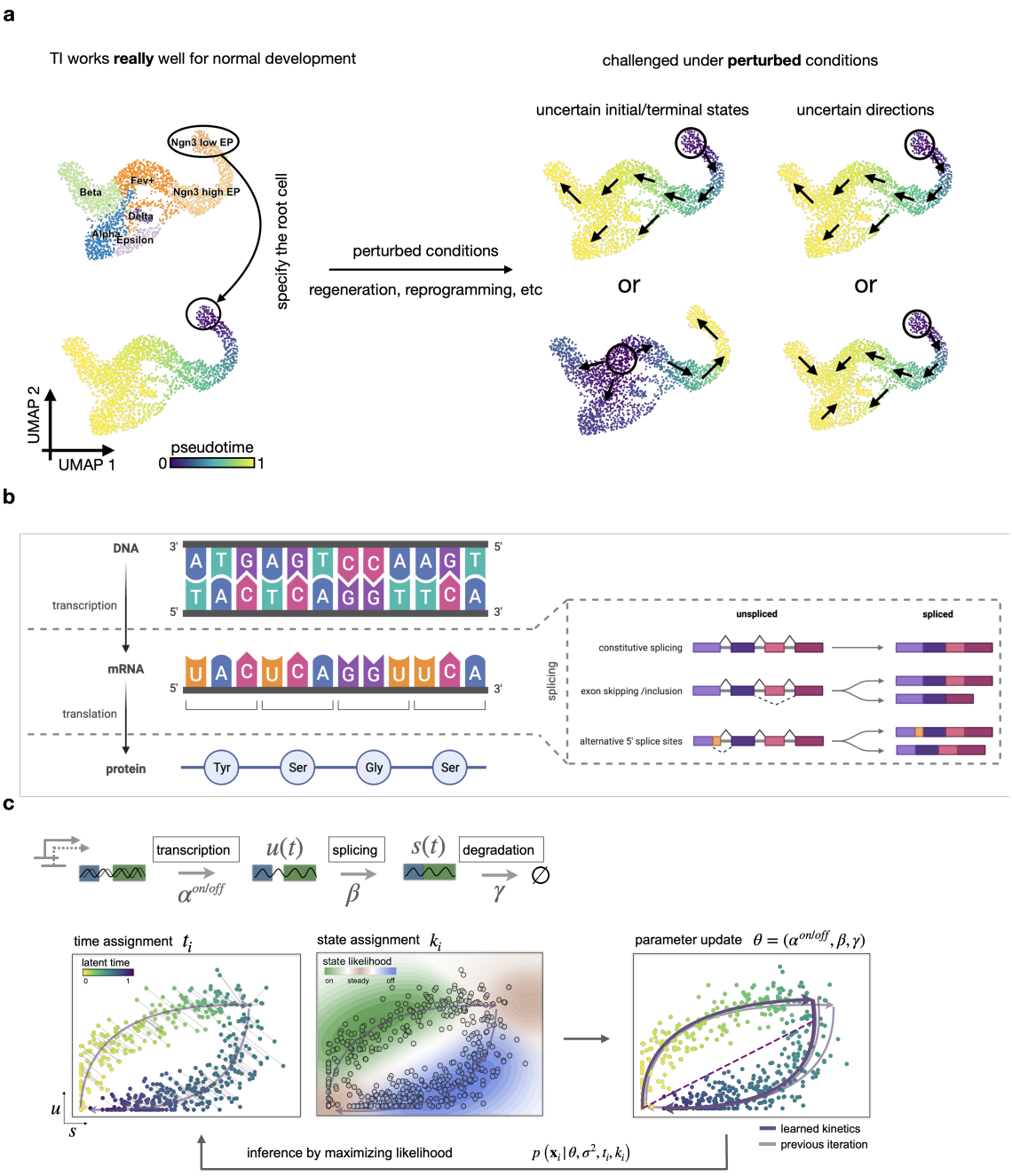

Figure 2.6: RNA velocity from spliced and unspliced counts. a. Illustrating the pitfalls of classic TI using pancreatic endocrinogenesis [23] as an example; each dot denotes a cell in UMAP embedding, colored according to either cluster (top left) or pseudotime (rest). **b.** Central dogma of molecular biology of Figure 2.1; dashed box highlights an intermediate processing step of mRNA called *splicing* during which introns (black triangles) are removed and exons (colored boxes) remain. Alternative splicing can lead to different outcomes, three examples are shown. **c.** RNA velocity makes use of the time delay between spliced and unspliced stages of mRNA captured in standard protocols; scVelo's dynamical model of splicing kinetics uses the first-order moment approximation to the CME and an EM-scheme for parameter inference. Panel (**b**) adapted from the following templates: "Central Dogma" and "mRNA Splicing Types" by BioRender.com (2022). Retrieved from `https://app.biorender.com/biorender-templates`. Panel (**c**) adapted from Bergen et al. [16].

Chapter 3

CellRank generalizes trajectory inference

Cells undergoing dynamical state transitions in biological processes including development, regeneration, reprogramming, and cancer, typically do so in a highly asynchronous manner [18]. scRNA-seq successfully captures the resulting heterogeneity, but it loses lineage relationships because each cell can be measured only once (Section 2.1). This prompted the development of computational approaches to reconstruct pseudotime trajectories [12, 18, 25, 27, 213, 244, 245] which build on the observation that developmentally related cells tend to be similar in their gene expression profiles (Section 2.4). Pseudotime approaches have been used extensively for ordering cells along differentiation trajectories and for studying cell-fate decisions.

However, computational trajectory inference typically demands prior biological knowledge to determine the directionality of cell state changes, often by specifying an initial cell [214], thereby limiting its applicability to normal developmental scenarios with known cell-fate hierarchies. RNA velocity [1] has been shown recently to alleviate this problem by reconstructing trajectory direction based on the spliced-to-unspliced mRNA ratio (Section 2.5). This promising approach has been generalized to include transient cell populations and further molecular modalities [16, 277, 278]; however, estimated velocity vectors are noisy and their interpretation has been limited mostly to low-dimensional projections. These do not easily reveal long-range probabilistic fates or allow quantitative interpretation (Section 2.5).

In this chapter, we present CellRank, a method that combines the robustness of similarity-based trajectory inference (Section 2.4) with directional information from RNA velocity (Section 2.5) to learn directed, probabilistic state-change trajectories for normal or perturbed conditions. In particular, we demonstrate how CellRank overcomes the challenges outlined in Section 1.2:

- we address challenge (i), the need for robust representations of cellular dynamics that originate from noisy RNA velocity estimates in Section 3.2 where we introduce

the `VelocityKernel`, a method to compute a Markov transition matrix from RNA velocity and gene expression similarity.

- we address challenge (ii), the identification of initial and terminal states and the description of fate establishment in Section 3.3 where we adapt ideas from Markov state modeling to coarse-grain the Markov chain and present a salable approach to compute fate probabilities with the `GPCCAEstimator`.
- we address challenge (iii), the need for a unifying framework to model cellular fate decisions that incorporates various estimates of directionality in Section 3.1 where we introduce CellRank's modular design as well as in Section 3.5 where we extend this modular design with the `PseudotimeKernel`, the `CytoTRACEKernel`, and the `RealtimeKernel`.

We demonstrate and benchmark our contributions in practical applications to MEF reprogramming [22], pancreas development [23], and lung regeneration [11] in Section 3.4. This chapter corresponds to, and is in part identical, with the following publications:

(i) **Lange, M.**, Bergen, V., Klein, M., Setty, M., Reuter, B., Bakhti, M., Lickert, H., Ansari, M., Schniering, J., Schiller, H.B., Pe'er, D. and Theis, F.J., 2022. CellRank for directed single-cell fate mapping. *Nature Methods*, pp.1-12.

(ii) Weiler P.*, **Lange, M.***, Klein, M. and Theis, F.J., 2022. A unified framework to study single-cell fate decisions. *in preparation.*

Note that "*" denotes an equal contribution.

3.1 The CellRank modeling framework

With CellRank, we aim to automatically detect the initial, terminal and intermediate states of a biological system and to define a global fate map that probabilistically assigns each cell to these states. The fate map enables us to compute trajectory-specific gene expression trends which we visualize in several ways. We make the following assumptions:

- On the gene expression level, daughter cells are generally similar to their mother cells. State transitions proceed gradually along a low-dimensional phenotypic manifold from initial via intermediate to terminal states.

- cellular sampling covers all intermediate states such that the entire state change trajectory is covered with no "gaps".
- while an individual cell's memory is stored in epigenetic modifications, we describe averaged cellular dynamics that occur without memory.
- we have access to an estimate of the direction of cell state changes, for example, from RNA velocity.

On the basis of these assumptions, we model cellular state transitions using a discrete-time Markov chain $(X_t)_{t\geq 0}$ where each state is given by an observed cellular profile (Section 2.2). To define the Markov chain, we compute a transition matrix $T \in \mathbb{R}^{N_c \times N_c}$ which describes how likely each cell is to transition into another; we describe this process for RNA velocity in Section 3.2 and for other modalities in Section 3.5.

Identifying initial, terminal and intermediate states. In order to define an initial (terminal) state, consider an ensemble of molecular profiles which, when taken together, characterize the starting (end) point of one particular state-change trajectory. Intermediate states are defined analogously via an ensemble of molecular profiles which characterizes a point in between initial and terminal states on the state-change trajectory. To identify initial, intermediate and terminal states, we coarse-grain the transition matrix T into macrostates and an associated coarse-grained transition matrix $\tilde{T}$ (Section 3.3).

Fate probabilities towards macrostates. We define the fate probability of cell i to reach macrostate $j \in \{1, ..., N_m\}$ for N_m, the number of macrostates, in biological terms as the probability that cell i executes a series of regulatory programs which adapt its phenotype to match the phenotype of cells that reside in macrostate j. We are typically interested in fate probabilities towards macrostates which are either terminal or intermediate.

Mathematically, we translate this to the probability that a random walk initialized in cell i reaches a cell from macrostate j before reaching any cell from another macrostate. In CellRank, we efficiently compute these probabilities in closed form using absorption probabilities (Section 3.3).

3.1.1 Kernels and estimators

As outlined above, there are three main steps to the CellRank workflow:

(i) Compute transition probabilities among observed cells (Figure 3.1a-d). These probabilities quantify how likely a cell in a given state is to adapt its gene expression profile to that of a target cell. We aggregate the transition probabilities in the transition matrix T and use it to model cell-state transitions as a Markov chain.

(ii) Coarse-grain the Markov chain into macrostates of cellular dynamics (Figure 3.1e) and aggregate *coarse-grained transition probabilities* in $\tilde{T}$. Using this matrix, we classify macrostates into initial, intermediate, and terminal states.

(iii) Compute fate probabilities towards a subset of the macrostates (Figure 3.1f). We compute the probability of each cell transitioning into each of the selected macrostates; these values are returned in a fate matrix F.

We designed a modular interface around these three main steps, structuring CellRank into *kernels* and *estimators* (Figure 3.2). While kernels compute a transition matrix based on various input data modalities (step i), estimators compute initial and terminal states, fate probabilities, and possibly other Markov chain derived quantities that can be used to generate biological insights (steps ii and iii). This design choice is crucial to CellRank - it allows any kernel to be combined with any estimator, thus enabling a vast amount of applications in a flexible manner. Moreover, the modular interface makes it easy to extend CellRank in two principal directions:

(i) extension towards new single-cell data modalities by including new kernels.

(ii) extension towards new trajectory descriptions that generate different hypotheses by including new estimators.

We introduce our kernels in Section 3.2 (`VelocityKernel` and `ConnectivityKernel`) as well as in Section 3.5 (`PseudotimeKernel`, `CytoTRACEKernel` and `RealtimeKernel`). We further introduce our `GPCCAEstimator` in Section 3.3 and we showcase applications of kernels and estimators in Section 3.4.

Kernel arithmetics. A single kernel may not be able to capture all that is necessary to describe the biology in a given dataset. We thus provide a convenient way to combine different kernels, each capturing one aspect of cellular dynamics, into one joint dynamical representation given by an aggregate transition matrix. For any two kernels k_1, k_2, CellRank implements their global combination via

$$k = ak_1 + (1 - a)k_2\,, \tag{3.1}$$

for a weight parameter $a \in [0, 1]$. Under the hood, the kernel API computes the corresponding normalized linear combination of the transition matrices stored in each kernel.

Exploiting sparsity. Besides modularity, achieved through kernels and estimators, a second key design principle of the CellRank framework is sparsity. All kernels presented in this thesis yield sparse transition matrices T, either because they are KNN graph-based (`ConnectivityKernel`, `VelocityKernel`, `CytoTRACEKernel`, `PseudotimeKernel`), or because they employ an adaptive thresholding strategy (`RealtimeKernel`). Further, the `GPCCAEstimator` exploits sparsity in all major computations (Section 3.3). The efficient use of sparsity allows CellRank to scale to very large datasets; we show in Section 3.4 how the `GPCCAEstimator` computes macrostates and fate probabilities on a 100k cell dataset in a few seconds.

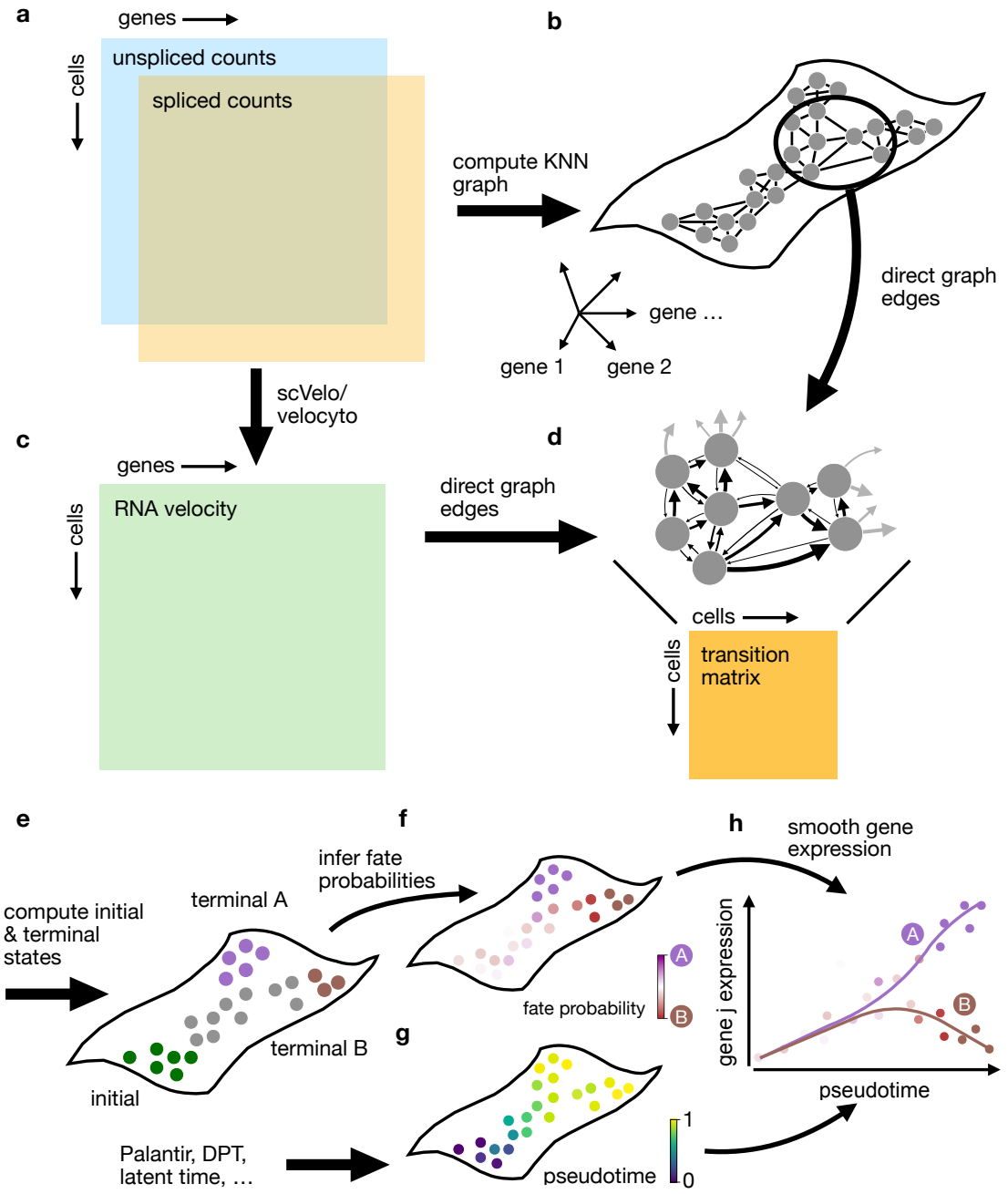

Figure 3.1: Main steps of the CellRank workflow, illustrated for RNA velocity data. a-c. Based on spliced and unspliced molecular counts (**a**), a KNN graph (**b**, Section 2.1.2) as well as RNA velocity (**c.**, Section 2.5) are computed. **d.** Both sources of information are combined into a transition matrix T (Section 3.2). **e.** The transition matrix T is coarse grained into a set of macrostates. The associated coarse-grained transition matrix $\tilde{T}$ classifies macrostates into initial, intermediate and terminal states (Section 3.3). **f.** Fate probabilities denote how likely each cell is to reach each terminal state (Section 3.3). **g-h.** By combining any precomputed pseudotime (**g**) with fate probabilities, trajectory-specific gene expression trends are computed (Section 3.3). Figure reproduced from Lange et al. [14].

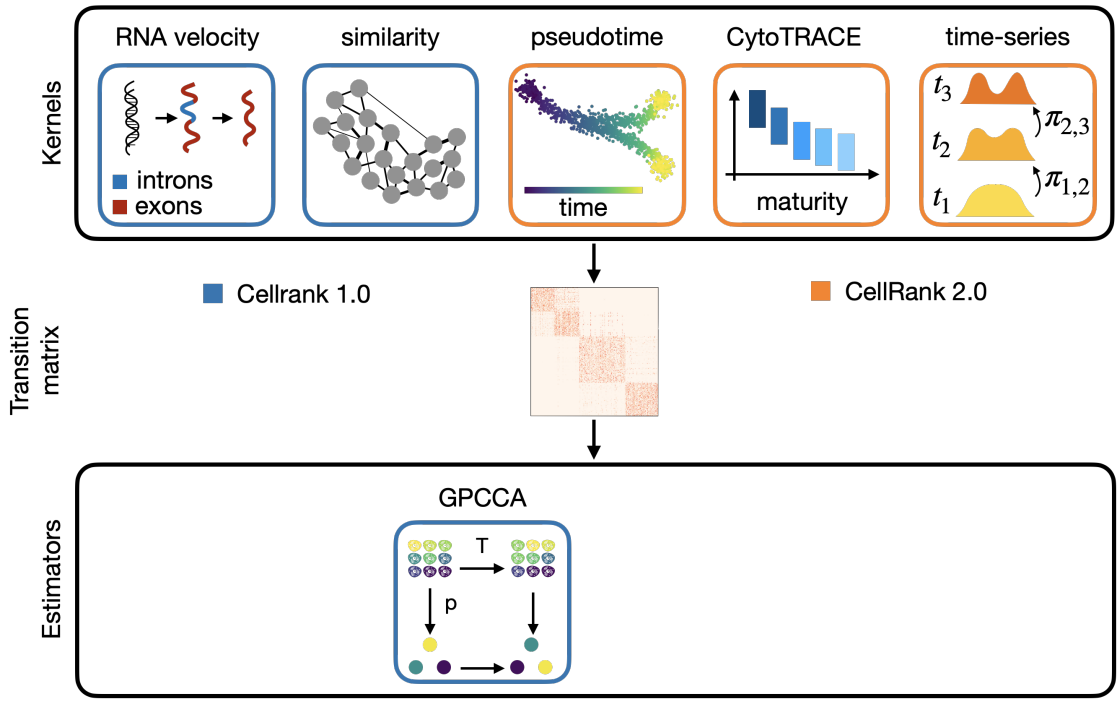

Figure 3.2: Modular design of the CellRank framework. We differentiate between features that were present in the first version of CellRank [14] (CellRank 1.0, blue) and recent extensions (CellRank 2.0, orange).

3.2 CellRank's `ConnectivityKernel` and `VelocityKernel`

To compute transition probabilities among cells, we make use of gene expression similarity and of RNA velocity; the former defines the global topology of the phenotypic manifold while the latter directs local movement on this manifold. We start by motivating our treatment of velocity and gene expression data, proceed to introduce the `ConnectivityKernel` (Section 3.2.1) and `VelocityKernel` (Section 3.2.2) and conclude with a strategy to propagate uncertainty in the velocity vectors (Section 3.2.3).

Transition matrix construction in the presence of noise. scRNA-seq yields noisy gene expression profiles (Section 2.1); as RNA velocity is computed on the basis of these noisy profiles, it represents a substantially noisy quantity itself (Section 2.5). In particular, the unspliced transcripts required to estimate velocity vectors are very sparse and their abundance varies depending on the gene structure (Section 2.5). In addition, choices made in the preprocessing pipeline, e.g. for alignment, heavily impact the final velocity estimate [289]. We adopt four strategies to cope with uncertain velocity estimates:

- in the `VelocityKernel`, we restrict the set of allowed transitions to those consistent with the KNN-graph-defined topology of the phenotypic manifold (Section 3.2.2).
- we use a stochastic formulation based on Markov chains to describe cell-state transitions.
- we combine RNA velocity information with trancriptomic similarity as captured by the `ConnectivityKernel` (Section 3.2.1).
- in the `VelocityKernel`, we propagate uncertainty in $\boldsymbol{v}_i$ into the transition matrix (Subsection 3.2.3).

Combining velocity with transcriptomic similarity. For applications to scRNA-seq data with RNA velocity information, including the applications of Section 3.4, we globally combine the `ConnectivityKernel` with the `VelocityKernel` using the strategy outlined in the previous section where we give a weight of 0.2 to the `ConnectivityKernel` which we have found to increase robustness to noisy velocity vectors. We show in Section 3.4 that CellRank's results are robust with respect to the exact weight parameter used.

In Li [265], the authors provide mathematical motivation for this kernel combination. They describe cellular dynamics as a stochastic differential equation (SDE) using a chemical

Langevin equation [290] where the drift term is given by a velocity vector field. Upon decomposing the vector field into equilibrium and non-equilibrium parts, they interpret the kernel combination as weighting between these two contributions.

3.2.1 The `ConnectivityKernel`

The transition matrix $T^{(\mathrm{co})}$ computed by the `ConnectivityKernel` is equivalent to the transition matrix $T^{(\mathrm{sym})}$ described in Section 2.2.5 in the context of the Diffusion distance, i.e. it represents symmetric, gene-expression distance-based transition probabilities and is computed based on a KNN graph $\mathcal{G}$ computed in some latent representation Z.

3.2.2 The `VelocityKernel`

The computation of a transition matrix in the `VelocityKernel` is based on the same KNN graph $\mathcal{G}$ used by the `ConnectivityKernel` above. However, in the `VelocityKernel`, RNA velocity information is used to direct edges in the graph.

Directing the KNN graph based on RNA Velocity. We direct the edges in $\mathcal{G}$ using RNA velocity information; neighboring cells whose displacement is better aligned with the direction prescribed by the velocity vector get higher probability. Specifically, for cell i with gene expression $\boldsymbol{x}_i \in \mathbb{N}^{N_g}$ and velocity vector $\boldsymbol{v}_i \in \mathbb{N}^{N_g}$, consider its neighbors $\mathcal{N}_i = \{1, 2, ..., K_i\}$ with gene expression profiles $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_{K_i}\}$. Note that the graph construction of Subsection 2.1.2 leads to a symmetric KNN graph where K_i is not constant across cells but $K_i \geq K \, \forall i \in \{1, ..., N_c\}$. For each neighboring cell $j \in \mathcal{N}_i$, compute the corresponding state-change vector with cell i, $\boldsymbol{s}_{ij} = \boldsymbol{x}_j - \boldsymbol{x}_i \in \mathbb{N}^{N_g}$. Next, we compute Pearson correlations $\boldsymbol{c}_i \in [-1, 1]^K$ of $\boldsymbol{v}_i$ with all state change vectors via

$$c_{ij} = \frac{(\boldsymbol{s}_{ij} - \mathbf{1}\bar{s}_{ij})^\top (\boldsymbol{v}_i - \mathbf{1}\bar{v}_i)}{\|\boldsymbol{s}_{ij} - \mathbf{1}\bar{s}_{ij}\| \, \|\boldsymbol{v}_i - \mathbf{1}\bar{v}_i\|} \in [-1, 1]^{K_i}, \tag{3.2}$$

where $\bar{s}_{ik}$ and $\bar{v}_i$ are averages over the state change vector and the velocity vector, respectively (Figure 3.3a,b). A value of 1 means perfect (positive) correlation between the observed displacement between the reference cell and a nearest neighbor and the gene expression change predicted by the local velocity vector.

The `VelocityKernel` supports alternative ways of computing similarity between state change and velocity vectors including the cosine and dot product schemes suggested in Li [265]. In the same work, the authors study the convergence of the discrete velocity schemes towards their continuum limit in the infinite sample setting; in this limit, the evolution of

probability density in state space can be described by a Fokker-Planck equation [265].

Transforming similarities into transition probabilities. To use the vector $\boldsymbol{c}_i$ as a set of transition probabilities to neighboring cells, we need to make sure it sums to one and it is positive. For a given cell i, define the set of transition probabilities $\boldsymbol{p}_i \in \Delta_{K_i}$ via

$$p_{ij} = \frac{\exp(\sigma c_{ij})}{\sum_{k=1}^{K_i} \exp(\sigma c_{ik})}, \tag{3.3}$$

which is known as a *softmax function* where $\sigma > 0$ is a scalar constant that controls how centered the categorical distribution is around its most likely value, i.e. around the state-change transition with maximum correlation. We repeat this for all (i, j) which are nearest neighbors in $\mathcal{G}$ to compute the transition matrix $T^{(\mathrm{v})} \in \mathbb{R}^{N_c \times N_c}$.

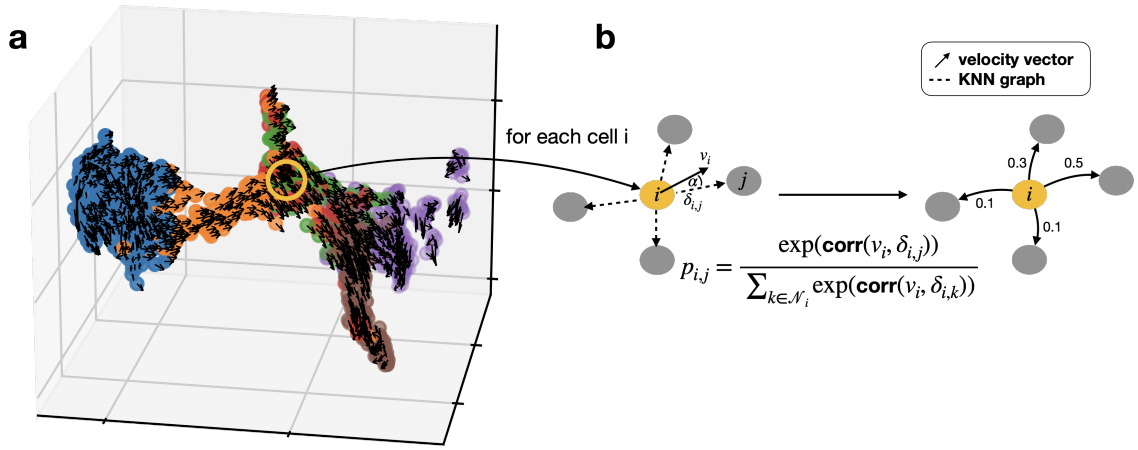

Figure 3.3: Main steps of the `VelocityKernel`. a. Synthetic example; each dot represents a cell in high dimensions colored by cell type, black arrows denote RNA velocity. **b.** One particular cell i is compared with its neighbors by computing the Pearson correlation between the velocity vector $\boldsymbol{v}_i$ and the state change vectors δ_{ij}. Correlations are normalized to yield transition probabilities. Figure adapted from Lange et al. [14].

Automatically determine σ. We reasoned that the value of σ should vary with Pearson correlations observed between velocity vectors and state change vectors; thus, we use the following heuristic choice:

$$\sigma = \frac{1}{\mathrm{median}(\{|c_{ij}| \; \forall i, j\})}. \tag{3.4}$$

Accordingly, if the median absolute Pearson correlation observed in the data is large (small), we use a small (large) value for σ. For sparsely sampled datasets where velocity vectors only roughly point in the direction of neighboring cells, we upscale all correlations. Typical values for σ we compute this way range from 1.5 for the lung example [11] to 3.8 for the pancreas example [23] of Section 3.4.

3.2.3 Propagating velocity uncertainty

Thus far, we treated individual velocity vectors as deterministic quantities, i.e. we assigned no measurement error to them. However, this is problematic as RNA velocity is estimated on the basis of spliced and unspliced gene counts, which are noisy quantities. Hence, the velocity vectors $\boldsymbol{v}_i$ themselves should be treated as random variables which follow a certain distribution and this should be taken into account when estimating transition probabilities. A possible solution to this would be to employ Monte Carlo (MC) sampling; however, this would incur large computational costs through repeated computations. To get around this problem, we construct an analytical approximation to the MC scheme. The analytical approximation only has to be evaluated once and we can omit the sampling. We show in Section 3.4 that the analytical approximation gives very similar results to the MC scheme and improves over a deterministic approach by a large margin.

Modeling the distribution over velocity vectors. To propagate uncertainty, we need to model the velocity vector distribution i.e. we need to quantify the uncertainty present in velocity vectors estimated by scVelo [16] or velocyto [1]. Preferentially, these packages would model uncertainty in the raw spliced and unspliced counts to propagate it into a distribution over velocity vectors. However, that is currently not the case; alternative approaches include using Fischer-information [132] or the profile-likelihood [291] to estimate uncertainty in the estimated model parameters $(\alpha^{\text{on/off}}, \beta, \gamma)$; this can easily be propagated into velocity estimates using standard error propagation.

CellRank is based on scVelo which currently uses the derivative-free Nelder–Mead algorithm [292] for optimization; therefore, we choose a different path and model uncertainty directly on the level of velocity vectors. We make an assumption about the velocity-vector distribution and set expectation and variance based on neighboring velocity vectors. To ease notation and to illustrate the core ideas, we drop the subscript i in this section and focus on a fixed cell and its velocity vector $\boldsymbol{v}$. Suppose that $\boldsymbol{v}$ follows a multivariate normal

(MVN) distribution,

$$\boldsymbol{v} \sim \mathcal{N}\left(\boldsymbol{\mu}, \Sigma_{\boldsymbol{v}}\right) , \tag{3.5}$$

with mean vector $\boldsymbol{\mu} \in \mathbb{R}^{N_g}$ and covariance matrix $\Sigma_{\boldsymbol{v}} \in \mathbb{R}^{N_g \times N_g}$. The MVN is a reasonable choice here; velocities can be both positive and negative and they are approximately symmetric around their expectation. Further, we assume a diagonal covariance matrix corresponding to gene-wise velocity independence. This is a reasonable assumption as gene-wise velocities in both velocyto [1] and scVelo [16] are computed independently. To compute values for $\boldsymbol{\mu}$ and $\Sigma_{\boldsymbol{v}}$, consider the current cell and its K nearest neighbors. To estimate $\boldsymbol{\mu}$ and the diagonal elements of $\Sigma_{\boldsymbol{v}}$, we compute first and second-order moments for the velocity vectors of these neighboring cells.

Approximating the expected transition matrix. We compute the expected value of transition matrix entries given the distribution over velocity vectors (Figure 3.4a,b). For a particular draw $\boldsymbol{v}$ from the distribution in Equation (3.5) and a set of state-change vectors $\boldsymbol{s}_k$, we compute a vector of probabilities $\boldsymbol{p} \in \Delta_K$ as outlined above. We denote the mapping from $\boldsymbol{v}$ to $\boldsymbol{p}$ by h,

$$\begin{aligned} h : \mathbb{R}^{N_g} &\to \Delta_K \,, \\ \boldsymbol{v} &\mapsto h(\boldsymbol{v}) = \boldsymbol{p} \,. \end{aligned} \tag{3.6}$$

We may formulate our problem as finding the expectation of h when applied to $\boldsymbol{v}$, i.e.

$$\mathrm{E}\left[h(\boldsymbol{v})\right]_{\boldsymbol{v} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma_{\boldsymbol{v}})} \,. \tag{3.7}$$

In order to derive an approximation, expand the i-th component of h in a Taylor-series around $\boldsymbol{\mu}$,

$$h_i(\boldsymbol{v}) = h_i(\boldsymbol{\mu}) + \nabla_{\boldsymbol{v}}^{\top} h_i(\boldsymbol{v})|_{\boldsymbol{\mu}} (\boldsymbol{v} - \boldsymbol{\mu}) + \frac{1}{2} (\boldsymbol{v} - \boldsymbol{\mu})^{\top} \nabla_{\boldsymbol{v}}^2 h_i(\boldsymbol{v})|_{\boldsymbol{\mu}} (\boldsymbol{v} - \boldsymbol{\mu}) + \mathcal{O}(\boldsymbol{v} - \boldsymbol{\mu})^3 \,. \tag{3.8}$$

Define the Hessian matrix of h_i at $\boldsymbol{v} = \boldsymbol{\mu}$ as

$$H^{(i)} = \nabla_{\boldsymbol{v}}^2 h_i(\boldsymbol{v})|_{\boldsymbol{\mu}} \,. \tag{3.9}$$

Taking the expectation of h_i and using the Taylor-expansion,

$$\mathrm{E}\left[h_i(\boldsymbol{v})\right] \approx h_i(\boldsymbol{\mu}) + \frac{1}{2} \mathrm{E}\left[(\boldsymbol{v} - \boldsymbol{\mu})^{\top} H^{(i)} (\boldsymbol{v} - \boldsymbol{\mu}) \right] \,. \tag{3.10}$$

The first order term cancels as $\mathrm{E}\left[\boldsymbol{v}-\boldsymbol{\mu}\right]=0$. We simplify the second-order term by explicitly writing out the matrix multiplication,

$$\mathrm{E}\left[(\boldsymbol{v}-\boldsymbol{\mu})^{\top}H^{(i)}(\boldsymbol{v}-\boldsymbol{\mu})\right]=\sum_{j,k=1}^{N_g}H^{(i)}_{j,k}\,\mathrm{E}\left[(\boldsymbol{v}-\boldsymbol{\mu})_j(\boldsymbol{v}-\boldsymbol{\mu})_k\right]\,, \tag{3.11}$$

where we moved the expectation inside the sum and the matrix element outside the expectation as it does not involve $\boldsymbol{v}$. For $j\neq i$, the two terms inside the expectation involving $\boldsymbol{v}$ are independent given our distributional assumptions on $\boldsymbol{v}$ and the expectation can be taken separately. Using again the fact that $\mathrm{E}\left[\boldsymbol{v}-\boldsymbol{\mu}\right]=0$, the sum equals zero for $j\neq i$. It follows

$$\sum_{j,k=1}^{N_g}H^{(i)}_{j,k}\,\mathrm{E}\left[(\boldsymbol{v}-\boldsymbol{\mu})_j(\boldsymbol{v}-\boldsymbol{\mu})_k\right]=\sum_{j=1}^{N_g}H^{(i)}_{j,j}\mathrm{E}\left[(v_j-\mu_j)^2\right]=\sum_{j=1}^{N_g}H^{(i)}_{j,j}\,\mathrm{Var}\left[v_j\right]\,. \tag{3.12}$$

In summary, our second-order approximation to the transition probabilities, given the distribution over $\boldsymbol{v}$, reads

$$\mathrm{E}\left[h_i(\boldsymbol{v})\right]\approx h_i(\boldsymbol{\mu})+\frac{1}{2}\sum_{j=1}^{N_g}H^{(i)}_{j,j}\,\mathrm{Var}\left[v_j\right]\,. \tag{3.13}$$

We repeat the above procedure for all components i and for all cells to obtain the second-order approximation to the expected transition matrix given the distribution over velocity vectors. To compute the Hessian matrices $H^{(i)}$ of Equation (3.13), we use automatic differentiation as implemented in JAX [230] rather than hard-coding the derivatives. This ensures our approach is independent of the function used to compute transition probabilities given velocity vectors - one can use Pearson correlations as suggested above or any other differentiable similarity measure.

Approximating the expected final quantities. We arrive at our final quantities of interest, i.e. macrostate assignments and fate probabilities (Section 3.3), through using the expected transition matrix and proceeding as in the deterministic case (Figure 3.4c). We check that this approximation scheme gives very similar results to a fully stochastic approach based on MC sampling (Section 3.4). The MC approach is also available through the `VelocityKernel` interface by setting `mode='sampling'` in the method call to compute the transition matrix. Thus, the user may choose conveniently between two options: (i) a fast approximate method given by our analytical approximation and (ii) a slower, but asymptotically exact method given by MC sampling.

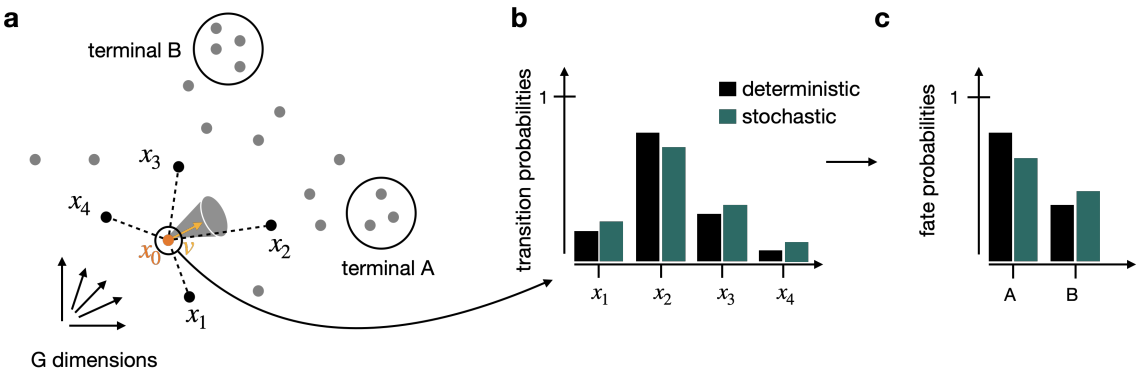

Figure 3.4: Uncertainty propagation in the `VelocityKernel`. a. Each dot represents a cell in high-dimensional gene expression space, two terminal states (A and B) are circled. A reference cell $\boldsymbol{x}_0$ with noisy velocity vector $\boldsymbol{v}$ as well as nearest neighbors $\{\boldsymbol{x}_i\}_{i=1}^4$ are indicated. **b.** Propagating the distribution in $\boldsymbol{v}$ changes transition probabilities to nearest neighbors. **c.** Adapted transition probabilities have an effect on downstream quantities like fate probabilities. Figure adapted from Lange et al. [14].

3.3 The `GPCCAEstimator`

Combining the `VelocityKernel` with the `ConnectivityKernel` as outlined in the previous section yields a robust description of cellular dynamics from a scRNA-seq dataset with RNA velocity information in form of a transition matrix T. In this section, we introduce various methods to interpret this matrix within the context of Markov chains. We start by coarse-graining T into macrostates of cellular dynamics (Section 3.3.1), proceed with computing fate probabilities (Section 3.3.2) and conclude by presenting various biologically motivated use cases of fate probabilities (Section 3.3.3). We packaged these methods into the `GPCCAEstimator`; they may be applied to any transition matrix, including the ones from Section 3.2 and Section 3.5. For Markov transition matrices computed outside the CellRank framework, we enable applying the `GPCCAEstimator` through the `PrecomputedKernel` which accepts any external transition matrix as an input.

3.3.1 Coarse-graining the Markov chain

The transition matrix T defines a Markov chain among the set of observed cells; however, it is difficult to directly interpret T in terms of biological trajectories because T is a fine-grained, noisy representation of cell state transitions. Therefore, we seek to reduce T to its essence: macrostates representing key biological states and their transition probabilities among each other. We accomplish this using *Generalized Perron Cluster Cluster Analysis* (GPCCA) [293–296], a method originally developed to study the dynamics of protein

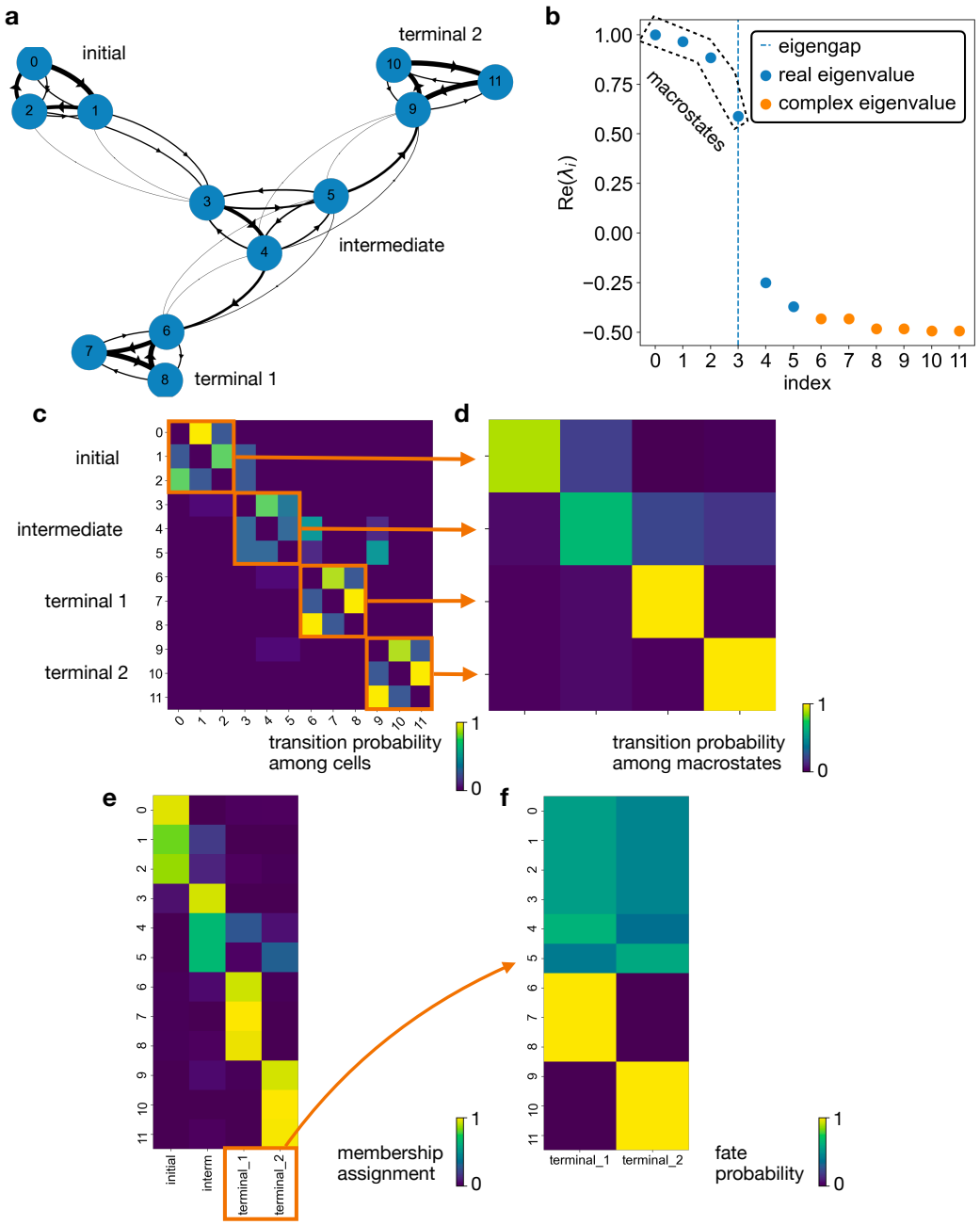

Figure 3.5: Main steps of the `GPCCAEstimator`. **a.** A synthetic example consisting of initial, intermediate and terminal states. Arrow thickness indicates transition probability. **b.** Spectrum of the transition matrix T. Eigengap after four eigenvalues suggest using $N_m = 4$. **c.** Heatmap of T. **d.** Coarse-grained transition matrix $\tilde{T}$ allows identifying initial, indermediate and terminal states. **e.** Membership matrix χ. **e.** Heatmap of the fate matrix F, containint fate probabilities towards the two terminal states. Figure reproduced from Lange et al. [14].

folding. We adapt this method to single-cell genomics data and utilize it to project the large transition matrix T onto a much smaller coarse-grained transition matrix $\tilde{T}$ that describes transitions among the set of macrostates. Macrostates are defined through a membership matrix $\chi \in \mathbb{R}_+^{N_c \times N_m}$ where N_m denotes the number of macrostates. Rows $\chi_{i,:} \in \Delta_{N_m}$ contain the soft assignment of each cell to the set of macrostates.

Generalized Perron Cluster Cluster Analysis (GPCCA). The aim of the GPCCA method is to project the large transition matrix T onto a much smaller coarse-grained transition matrix $\tilde{T}$, which describes transitions between macrostates of the biological system [293, 294] (Figure 3.5a-d). For the projected dynamical process to be Markov, we require the projection to be based on an invariant subspace of T, i.e. a subspace W for which

$$T^\top x \in W \ \forall x \in W \,. \tag{3.14}$$

In the case of a reversible T, real invariant subspaces are spanned by the eigenvectors of T [297]. However, many of the transition matrices constructed by CellRank kernels are in general irreversible, this holds in particular for the `VelocityKernel`. The eigenvectors corresponding to irreversible matrices are in general complex; since the GPCCA method can not cope with complex vectors, we revert to the real Schur decomposition to define real invariant subspaces [293, 294, 298]. The real Schur decomposition of T is given by

$$T = QRQ^\top \,, \tag{3.15}$$

where columns of the matrix $Q \in \mathbb{R}^{N_c \times N_c}$ represent the Schur vectors while the Schur form $R \in \mathbb{R}^{N_c \times N_c}$ is quasi-upper triangular [299]. The matrix R has 1-by-1 or 2-by-2 blocks on the diagonal; 1-by-1 blocks correspond to real eigenvalues and 2-by-2 blocks are associated with pairs of complex conjugate eigenvalues.

Invariant subspaces of the transition matrix. Columns of the matrix Q corresponding to real eigenvalues span real invariant subspaces. Columns of the matrix Q corresponding to pairs of complex conjugate eigenvalues span real invariant subspaces only when considered jointly, but not if they are separated. Particularly, for columns $\boldsymbol{q}_j$ and $\boldsymbol{q}_k$ of Q belonging to a pair of complex conjugate eigenvalues, the space $W_0 = \text{span}(\boldsymbol{q}_j, \boldsymbol{q}_k)$ is invariant under T, but the individual $\boldsymbol{q}_j$ and $\boldsymbol{q}_k$ are not [300]. Different dynamical properties of T can be projected onto $\tilde{T}$ depending on the constructed subspace. For Schur vectors associated with real eigenvalues close to 1, metastabilities are recovered; for Schur

vectors associated with complex eigenvalues close to the unit circle, cyclic dynamics are recovered [293, 294]. Both options are available through the `GPCCAEstimator`, defaulting to the recovery of metastabilities.

Projecting the transition matrix. Let $\tilde{Q} \in \mathbb{R}^{N_c \times N_m}$ be the matrix formed by selecting N_m columns from Q according to a given criterion (cyclicity or metastability). Let $\chi \in \mathbb{R}^{N_c \times N_m}$ be a matrix obtained via linear combinations of the columns in $\tilde{Q}$, i.e.

$$\chi = \tilde{Q}A\,, \tag{3.16}$$

for invertible rotation matrix $A \in \mathbb{R}^{N_m \times N_m}$. Rows of the matrix χ define macrostate membership; we describe both χ and A in more detail below. In order to obtain the projected transition matrix, we use an invariant subspace projection [293, 294],

$$\tilde{T} = (\chi^\top D\chi)^{-1}(\chi^\top DP\chi)\,, \tag{3.17}$$

for D, the diagonal matrix of a weighted scalar product. We require the Schur vectors in $\tilde{Q}$ to be orthogonal with respect to this scalar product, i.e. $\tilde{Q}^\top D\tilde{Q} = I$ with the N_m-dimensional unit matrix I, to yield the required projection based on an invariant subspace. Note that the diagonal elements of D are in principle arbitrary; they may be chosen to represent some distribution over cellular states, e.g. the stationary distribution (Section 2.2). If we choose the uniform distribution, which is the default in the `GPCCAEstimator`, we ensure an indiscriminate handling of cellular states.

Properties of the invariant subspace projection. We define the coarse-grained transition probabilities among macrostates via an invariant subspace projection (see Equation (3.17)) of the original, large transition matrix onto the set of macrostates. More precisely, the transition matrix T is projected onto a low-dimensional invariant subspace defined by the membership vectors in χ which are linear transformations of the Schur vectors (Equation (3.16)). A consequence of the invariant subspace projection is that the projection error vanishes; the propagation operation commutes with the coarse-graining operation [293, 301].

To put it differently, for an initial density over cell states, the following two procedures yield the same result: (1) propagating the distribution over cell states using the original transition matrix T and projecting the propagated distribution onto our set of macrostates, (2) projecting the initial cell distribution onto the macrostate level and propagating it using

the coarse-grained transition matrix $\tilde{T}$. From this, it follows that the projected, coarse-grained Markov chain preserves the slow time scales of the process, i.e. the transitions between metastable subsets of the phenotypic manifold [293].

Computing the membership vectors. In principle, it is possible to use any invertible rotation matrix A in Equation (3.16). However, we aim to interpret the columns of χ as membership vectors that define assignment weights for the N_c cells to the N_m macrostates. Therefore, we seek a rotation matrix A that minimizes the overlap between membership vectors in χ, i.e. a rotation matrix A that minimizes off-diagonal entries in $\chi^\top D\chi$. For matrix $S \in \mathbb{R}^{N_m \times N_m}$ defined via

$$S = (\tilde{D}^{-1}\chi^\top D\chi)\,, \tag{3.18}$$

this is equivalent to maximizing trace(S) where $\tilde{D} \in \mathbb{R}^{N_m \times N_m}$ is a diagonal matrix which row-normalizes the expression with

$$D_{ii} = \sum_{j=1}^{N_m} \left(\chi^\top D\chi\right)_{ij}\,. \tag{3.19}$$

When aiming to recover metastability by selecting Schur vectors with real eigenvalues close to one, maximizing trace(S) can be interpreted as maximizing the metastability of the macrostates in the system. In practice, we minimize the objective function given by

$$f_{N_m}(A) = N_m - \text{trace}(S)\,. \tag{3.20}$$

Note that S is a function of A, this can bee seen by substituting Equation (3.16) into Equation (3.18). The objective function defined in this way is convex on the feasible set and bounded below by zero [297]. We must minimize f_{N_m} with respect to the constraints

$$\chi_{ij} \geq 0\ \forall (i,j) \in \{1,...,N_c\} \times \{1,...,N_m\}, \qquad \text{(positivity)}\,, \tag{3.21}$$

$$\sum_j \chi_{ij} = 1\ \forall i \in \{1,...,N_c\}, \qquad \text{(partition of unity)}\,. \tag{3.22}$$

We can re-express the conditions of Equation (3.21) and Equation (3.22) using Equation (3.16) and a result from Weber [302] in terms of A, the invertible rotation matrix, and

$\tilde{Q}$, the matrix of selected Schur vectors, via

$$A(1,j) = -\min_{l\in\{1,...,N_c\}} \sum_{i=2}^{N_m} \tilde{Q}_{li}\, A_{ij}\ \forall j \in \{1,...,N_m\}, \qquad \text{(positivity)}\,, \tag{3.23}$$

$$A(i,1) = \delta_{i,1} - \sum_{j=2}^{N_m} A_{ij}\ \forall i \in \{1,...,N_m\}, \qquad \text{(partition of unity)}\,. \tag{3.24}$$

Optimizing Equation (3.20) subject to the constraints of Equation (3.23) and Equation (3.24) is tricky; we perform unconstrained optimization on $A_{2:N_m,2:N_m}$ by imposing the constraints after each iteration step. This transforms the unfeasible solution into a feasible solution [297]. However, as this approach is non-differentiable, we use the derivative-free Nelder-Mead method [292] as implemented in the Scipy routine `scipy.optimize.fmin` [303] for the optimization.

Positivity of the projected transition matrix. The projected transition matrix $\tilde{T}$ can have negative elements if macrostates largely overlap. A suboptimal number of macrostates N_m is typically the cause of this issue; adapting N_m resolves the problem. We may interpret $\tilde{T}$ as the Markov-transition matrix among the set of macrostates as long as it is non-negative withing numerical precision [293].

Tuning the number of macrostates. The number of macrostates N_m can be chosen using a number of different approaches including the eigengap heuristic or the crispness of the solution $\xi = \text{trace}(S)/N_m$ [297] (Figure 3.5b); the larger ξ, the less macrostates overlap and the better the solution. These options are available trough the `GPCCAEstimator`.

Scalable Python implementation of GPCCA. Following the original MATLAB implementation [296], we designed GPCCA as a general algorithm in Python and created an independent package for it called `pyGPCCA` [304]. While `pyGPCCA` serves as the backbone for the `GPCCAEstimator`, we anticipate it to be used outside the single-cell community as well, e.g. in the study of protein conformational dynamics. A bottleneck in the implementation of `pyGPCCA` was the scalability of the Schur decomposition. A full Schur decomposition has time complexity $\mathcal{O}(N_c^3)$ and would be infeasible to compute for modern single-cell datasets. We overcome this challenge by computing only those Schur vectors which are required for $\tilde{Q}$ through an iterative, Krylov-subscpace-based algorithm implemented in the `SLEPSc` [305, 306] toolbox. The algorithm optimally exploits sparsity in T induced by the KNN graph $\mathcal{G}$ which forms the basis of many CellRank kernels, including the `VelocityKernel`. Over-

all, exploiting sparsity in this way reduces computational complexity to be approximately linear in the number of cells (Section 3.4). This allows us to apply the `GPCCAEstimator` to very large cell numbers.

$\tilde{T}$ identifies terminal states. To identify terminal states, we search for the most stable macrostates according to the coarse-grained transition matrix $\tilde{T}$ (Figure 3.5d). Define the *stability index* SI_i of a macrostate $i \in \{1, ..., N_m\}$ through its corresponding diagonal element in $\tilde{T}$,

$$\mathrm{SI}_i \coloneqq \tilde{T}_{ii}\,, \tag{3.25}$$

and classify macrostates as terminal for which $\mathrm{SI}_i \geq \epsilon_{\mathrm{SI}}$ with $\epsilon_{\mathrm{SI}} = 0.96$ by default; this is a method parameter that can be adjusted by the user. This mechanism is motivated by the intuition that cells in terminal populations are unlikely to transition to cells which reside in other populations; they distribute the vast majority of their outgoing probability mass to cells from their own terminal population.

$\tilde{T}$ identifies initial states. To identify initial states, we introduce the *coarse-grained stationary distribution* $\tilde{\boldsymbol{\pi}} \in \Delta_{N_m}$ given by

$$\tilde{\boldsymbol{\pi}} = \chi^\top \boldsymbol{\pi} \tag{3.26}$$

where $\boldsymbol{\pi} \in \Delta_{N_c}$ is the stationary distribution of the original transition matrix T (Section 2.2). The coarse-grained stationary distribution $\tilde{\boldsymbol{\pi}}$ describes the long-term evolution of the Markov chain given by $\tilde{P}$. It assigns large (small) values to macrostates that the process spends a large (little) amount of time in when run for an infinite amount of time. As such, we may use it to identify initial states by looking for macrostates that are assigned the smallest values in $\tilde{\boldsymbol{\pi}}$. This is motivated by the intuition that initial states are states that the process is unlikely to visit again once it left them. The number of initial states is a method parameter which we set to one by default.

$\tilde{T}$ identifies intermediate states. We classify the remaining macrostates as intermediate; these have neither been detected as initial nor terminal. Biologically, these states correspond to intermediate, transient cell populations on the trajectory of state change.

Handling reducible Markov chains. For most CellRank kernels, transition matrix construction ensures that, as long as the underlying KNN graph $\mathcal{G}$ is connected, the resulting Markov chain is irreducible (Section 2.2). For the `VelocityKernel`, this is a consequence of allowing transitions against the direction dictated by the local RNA velocity vector with a small probability. However, for non-connected $\mathcal{G}$, the resulting Markov chain will be reducible. Applying the `GPCCAEstimator` to reducible Markov chains is unproblematic; communication classes can be identified from $\tilde{T}$ and the computation of $\boldsymbol{\pi}$ and $\tilde{\boldsymbol{\pi}}$ can be restricted to one communication class at a time.

3.3.2 Computing fate probabilities

On the basis of the soft assignment of cells to macrostates by χ and the identification of terminal states through $\tilde{T}$, we compute how likely each cell is to transition towards these terminal states. Let N_t be the number of identified terminal states. While the below computations apply to all macrostates, let us assume for the sake of clarity that we are only interested in fate probabilities towards terminal states.

Problem setup. For each terminal state $t \in \{1, ..., N_t\}$, we choose f cells which are strongly assigned to t according to χ, i.e. we extract the corresponding column from χ and we calculate the terminal index set $\mathcal{R}_t$ of cells which have the largest values in this column of χ. If cell i is assigned to the terminal index set $\mathcal{R}_t$, we assume it is well-suited to characterize the terminal macrostate t. We store indices corresponding to remaining cells in the transient index set $\mathcal{T}$. The index sets $\{\mathcal{R}_t \,|\, t \in \{1, ..., N_t\}\}$ and $\mathcal{T}$ form a disjoint partition of the state space $\mathcal{S} = \{1, ..., N_c\}$ (Section 2.2). For each cell i in $\mathcal{T}$, we would like to compute a vector of probabilities $\boldsymbol{f}_i \in \Delta_{N_t}$ which specifies how likely this cell is to transition into any of the terminal states characterized trough $\{\mathcal{R}_t\}_t$. We accumulate the $\boldsymbol{f}_i$ column-wise in the fate matrix $F \in R^{N_c \times N_t}$; rows corresponding to cells in any of the $\{\mathcal{R}_t\}_t$ are assigned the corresponding indicator vector.

Fate probabilities through absorption probabilities. We define fate probabilities in our context as absorption probabilities on the Markov chain (Section 2.2), i.e. the fate probability of cell i to reach terminal state t_0 is computed as the absorption probability of a random walk initialized in state i to reach the terminal index set $\mathcal{R}_{t_0}$ before reaching any other terminal index set $\mathcal{R}_t$. In order to compute absorption probabilities, we approximate the terminal index sets as recurrent classes, i.e. we remove any outgoing edges found in these sets. We then apply Theorem 2.1 of Section 2.2, which, for each cell $i \in \mathcal{T}$, yields absorption probabilities towards each of the f cells in each of the N_t recurrent index

sets. We aggregate these to yield absorption probabilities towards the recurrent index sets themselves; this is achieved by summing up absorption probabilities towards individual cells in these sets (Figure 3.5e,f).

Computing absorption probabilities efficiently. A naive implementation of absorption probabilities has time complexity $\mathcal{O}(N_c^3)$ due to the matrix inversion in Theorem 2.1. This inevitably fails for large cell numbers. We alleviate this by re-writing Equation (2.30) of Theorem 2.1 as a linear problem,

$$(I - Q)A = S\,. \tag{3.27}$$

Note that Q is very sparse for all CellRank kernels as it describes transitions between nearest neighbors. Per row, Q has approximately K entries where K is the number of neighbors used in KNN graph construction. To exploit this level of sparsity, iterative solvers are appealing as their per-iteration cost applied to this problem is linear in N_c and K. However, to apply an iterative solver, we must rewrite Equation (3.27) such that the right-hand side is vector-valued,

$$(I - Q)a_1 = s_1,\ ...,\ (I - Q)a_{fn_t} = s_{fN_t}\,, \tag{3.28}$$

where fN_t is the total number of cells that are assigned to approximately recurrent classes. We use the iterative GMRES [307] algorithm to solve these individual problems because it efficiently makes use of sparsity structure. We use the PETSc implementation [306] which makes use of efficient message passing, among other practical performance enhancements. Lastly, we parallelize solving the fN_t linear problems. In combination, the tricks introduced here allow us to rapidly compute absorption probabilities even for very large cell numbers (Section 3.4).

3.3.3 Biological use cases of fate probabilities

Once fate probabilities have been computed, they can be used to answer a number of biological questions; we present four use cases in this section: (1) visualization of the phenotypic manifold in a 2D circular embedding, guided by each cell's probability of reaching each terminal state, (2) the quantification of multi-lineage potential, (3) the identification of genes which may be crucial for a certain lineage decision and (4) the visualization of smooth, trajectory specific gene expression trends.

Visualizing fate probabilities through circular embeddings. This presentation follows work by Velten et al. [308] which in turn is based on *circular a posteriori projections* [83]. Let $F \in \mathbb{R}^{N_c \times N_t}$ by the matrix of fate probabilities for N_c cells and N_t terminal states such that $F_{i,:} \in \Delta_{N_t}$ represents fate probabilities for cell i. We aim to find a two dimensional arrangement of cells that reflects their fate probabilities; therefore, we arrange the terminal states evenly spaced around the unit circle and assign each terminal state an angle α_t. We transform each cell's fate probability vector $F_{i,:}$ into a 2D representation (x_i, y_i) by using

$$x_i = \sum_t f_{i,t} \cos \alpha_t \tag{3.29}$$

$$y_i = \sum_t f_{i,t} \sin \alpha_t \,. \tag{3.30}$$

The final representation depends on the order in which we arrange terminal states around the unit circle. To find a good ordering, we compute pairwise similarities among fate probabilities $F_{:,t}$ and we choose the arrangement that maximizes the set of pairwise similarities. This ensures that similar terminal states are placed next to each other. We use cosine correlation to quantify similarity by default.

Quantifying multi-lineage potential. During development, cells gradually transition from multi-potent (naive) towards uni-potent (differentiated) states; potency can be quantified in CellRank to assess each cell's position on the state-change trajectory. We provide two ways of quantifying multi-lineage potential on the basis of computed fate probabilities:

- through $H(F_{i,:})$, the entropy over fate probabilities (called 'diffusion potential' in Palantir [25])
- through $\mathrm{KL}\left[F_{i,:} \,||\, \bar{\boldsymbol{f}}\right]$, the KL divergence between fate probabilities $F_{i,:}$ and the mean per-lineage fate probability across cells $\bar{f}_j = 1/N_c \sum_i F_{ij}$ (called 'priming degree' in STEMNET [308])

Intuitively, $H(F_{i,:})$ quantifies how far from uniform the distribution $F_{i,:}$ is and $\mathrm{KL}\left[F_{i,:} \,||\, \bar{\boldsymbol{f}}\right]$ quantifies how far from the average fate distribution $F_{i,:}$ is. The higher $H(F_{i,:})$ and the lower $\mathrm{KL}\left[F_{i,:} \,||\, \bar{\boldsymbol{f}}\right]$, the less committed a cell is. If initial cells already have a dominant direction of fate bias, we suggest using the KL divergence; it will increase monotonically as cells move from initial to terminal states while the entropy will reach its maximum at some point in between initial and terminal states which comes closest to uniform.

Uncovering putative decision-driver genes For finding putative fate-determining genes, we compute Pearson correlations between expression levels of a set of genes and fate probabilities. We use correlation values to sort genes and consider high-scoring genes as potential driver genes. By default, we include all genes which have passed pre-processing gene filtering thresholds. The computation of correlation values can be restricted to a set of pre-defined clusters if one is interested in driver genes acting in a specific region of the phenotypic manifold. We implement two options for computing p-values for the correlations, using either a Fisher transformation or a permutation test.

Visualizing trajectory-specific gene expression trends. Combining fate probabilities with any pseudotemporal measure like DPT [213] or Palantir's pseudotime [25] (Section 2.4) allows us to infer trajectory-specific gene expression trends. CellRank does not compute a pseudotime itself but it can guide pseudotime algorithms by providing the initial state. To visualize trends, we fit *Generalized additive models* (GAMs) to gene expression values which have been imputed by borrowing information from neighboring cells via a KNN graph. Using GAMs allows us to flexibly model many different kinds of gene trends in a robust and scalable manner. We fit the expression trend for trajectory t (associated with terminal state $t \in \{1, ..., N_t\}$) in gene j via

$$x_{ij} = \beta_0 + f(\tau_i) \; \forall i : F_{it} > 0 \,, \tag{3.31}$$

where x_{ij} denotes expression of gene j in cell i, τ_i is the pseudotemporal value of cell i and F is the fate matrix. We use cubic splines for the smoothing functions f by default; these have been shown to be effective in capturing non-linear relationships [309].

For visualization of the smoothed trend, we select 200 equally spaced points along pseudotime and predict gene expression using the fitted model of Equation (3.31) on this test set. To estimate gene trend uncertainty, we use the standard deviation of the residuals of the fit [310]. For the fitting of Equation (3.31), we provide interfaces to both the R package `mgcv` [311, 312] as well as the Python package `pyGAM` [313]. We parallelize gene fitting to scale well in gene numbers.

3.4 Validation, application, and benchmarking

In this section, we apply the machinery developed to real scRNA-seq data examples, in particular, we use the `ConnectivityKernel` and the `VelocityKernel` to compute a joint

transition matrix T and we identify initial and terminal states and fate probabilities using the `GPCCAEstimator`. We start by validating our framework on lineage-traced in-vitro MEF reprogramming data [22] (Section 3.4.1), proceed with an application to in-vivo pancreas development [23] (Section 3.4.2), benchmark the framework on the same data (Section 3.4.3) and conclude with an application to in-vivo lung regeneration [11] (Section 3.4.4).

Data pre-processing. All datasets considered in this section were preprocessed following standard SCANPY and scVelo workflows (Section 2.1); we filtered to genes that have at least 20 counts in both spliced and unspliced modalities, we normalized the total counts to be the same across cells, we log-transformed the data and computed a PCA representation in the space of the top 2000 highly variable genes (we kept the PCA from the original publication for the lung example [11]). Using the top $N_l = 30$ principal components, we computed a $K = 30$ ($K = 50$ for the lung) nearest neighbor graph $\mathcal{G}$, used throughout CellRank's kernels. To compute velocities, we run scVelo's dynamical model of splicing kinetics, i.e. the EM algorithm applied to first-order moment equations to recover both modal parameters as well as latent time assignments (Section 2.5).

Cell type classification and low-dimensional visual embeddings. For all data examples, we kept the cell-type labels that were supplied with the original publications. Two-dimensional data representations for purely visual purposes (not supplied to CellRank kernels) were obtained as follows: for the MEF example [22], we kept the original t-SNE embedding, for the pancreas example [23], we computed a PAGA-initialized UMAP [121, 175] while for the lung [11], we computed a BBKNN [141] batch-corrected UMAP (Section 2.1).

CellRank parameters. For all data examples, we computed a joint transition matrix T by combining the `ConnectivityKernel` with the `VelocityKernel` with weights 0.2 and 0.8, respectively. The kernels were supplied with a KNN graph $\mathcal{G}$ and velocity vectors computed as described above. The `VelocityKernel` was run with automatic kernel-width parameter σ identification and analytical noise propagation (Section 3.2). The `GPCCAEstimator` was run for a custom number of macrostates (indicated below for each data example), guided by the eigengap heuristic. The fate probabilities towards a subset of macrostates were computed using the fast linear-solve approach (Section 3.3). We illustrate in the pancreas example that CellRank's results are robust with respect to small changes in the key preprocessing parameters (Section 3.4.2).

3.4.1 Validation on a lineage-traced MEF reprogramming timecourse

To validate our proposed method, we applied CellRank to an in-vitro study of 48,515 MEFs reprogramming towards induced endoderm progenitors [314] (iEPs) across six time points [22]. We expect only around 1% of cells to successfully reprogram (marked by *Apoa1*) and the other cells to enter a "dead-end" state (marked by *Col1a2*) [22] (Figure 3.6a). This dataset is equipped with CellTagging lineage tracing labels that can be used to infer clonal relationships among cells, thus providing ground truth on the ultimate fate (successful versus dead-end) of early cells [22]. We were interested to see how well CellRank's fate probabilities recovered ground truth reprogramming outcome in this challenging setting.

CellRank recovers successful and dead-end states. We computed velocities using scVelo [16] and projected them on the original t-SNE embedding of Biddy et al. [22] (Figure 3.6b). Projected velocities were uninformative of a path towards the successful state, most likely because the reprogramming signal is too subtle to be picked up in a two dimensional representation. CellRank's macrostates, in contrast, included both a dead-end and the rare successful state (Figure 3.6c,d).

CellRank predicts reprogramming outcome. When we compared fate probabilities towards these states with lineage-tracing derived labels (Figure 3.6e), we found that fate probabilities were highly predictive of reprogramming outcome. As expected, predictive accuracy decreased for earlier days in the time course (Figure 3.6f).

Fate probabilities were compared to CellTag derived ground-truth labels from the original publication [22] via a classification task. The ground-truth labels were binary (successful versus dead-end) and available for a subset of the cells. We restricted the comparison to days 12, 15, and 21 where ground-truth labels were available for 374, 582, and 1,312 cells, respectively. There were more ground-truth labels available for dead-end cells than for successful cells which can give rise to misleading classification accuracy. To make make proportions even, we subsampled dead-end cells.

For the classification task, we randomly assigned 60% of labeled cells per day into a training set and the remaining cells into a test set. The final cell sets contained 208 (124 training / 84 testing), 308 (184 training /124 testing) and 652 (391 training / 261 testing) cells for days 12, 15 and 21, respectively. We trained logistic regression classifiers for each day independently to predict the ground-truth success/dead-end labels based on CellRank's fate probabilities on the training set using the `scikit-learn` implementation [316].

To quantify predictive performance, we visualized receiver operating characteristic (ROC) curves for each day on the test set. In short, ROC curves are computed by iterating over the decision threshold which is used to classify points as successful/dead-end, recording and plotting the true positive rate (TPR) against the false positive rate (FPR) for each decision threshold [317]. For each day, we also compute the area under the ROC curve (AUC); a measure between zero and one which summarized the entire ROC curve into a threshold-independent value. An AUC of 1 corresponds to perfect classification and 0.5 corresponds to random guessing (an uninformative classifier).

3.4.2 Application to pancreas development

Moving from in-vitro to in-vivo settings, we applied CellRank to a scRNA-seq dataset of E15.5 murine pancreatic development [23]. A UMAP [122] representation with original cluster annotations and scVelo-projected velocities recapitulated the main developmental trends [16] (Figure 3.7a); cells traverse trajectories from an initial cluster of endocrine progenitor cells (EPs) expressing the transcription factor neurogenin 3 (*Neurog3* or *Ngn3*) at low levels towards alpha, beta, epsilon and delta cell fates.

We computed CellRank's transition matrix T, coarse-grained it into 12 macrostates (Figure 3.7b) and computed the associated coarse-grained transition matrix $\tilde{T}$ (Figure 3.7c). Macrostates, annotated according to how much they overlap with orthogonal gene expression clusters, comprised all developmental stages in this dataset, from an initial Ngn3^{low} EP state, to intermediate $\text{Ngn3}^{\text{high}}$ EP and Fev+ states, to hormone-producing terminal alpha, beta, epsilon, and delta cell states.

The `GPCCAEstimator` identifies initial and terminal states. According to the coarse-grained transition matrix $\tilde{T}$, the three most stable states were the alpha (SI = 0.97), beta (SI = 1.00) and epsilon (SI = 0.98) macrostates which were accordingly labeled as terminal by the `GPCCAEstimator`, consistent with known biology (Figure 3.7d). Additionally, we recovered one relatively stable (SI = 0.84) macrostate which largely overlapped with delta cells. We identified the Ngn3^{low} EP_1 state as initial because it was assigned the smallest value in $\tilde{\boldsymbol{\pi}}$ (2×10^{-6}). Well-known marker genes confirm our automatic identification of initial and terminal states, including *Ins1* and *Ins2* for beta, *Gcg* for alpha, *Sst* for delta, and *Ghrl* for epsilon cells and ductal cell markers *Anxa2*, *Sox9*, and *Bicc1* for the initial state [23, 318] (Figure B.1 in Appendix B).

Fate probabilities recover expression trends of driver genes. We computed fate probabilities and visualized them in a *fate map*, a scatter plot in which each cell is colored according to its most likely fate with color intensity reflecting the degree of lineage priming (Figure 3.7e). This analysis correctly identified the beta-cell fate as dominant within the $\text{Ngn3}^{\text{high}}$ EP cluster at E15.5, consistent with known biology [23] (Figure 3.7e, inset). Using a cell within the Ngn3^{low} EP_1 macrostate as the starting state for Palantir [25], we ordered cells in pseudotime and overlaid the expression of master regulators *Arx* [318] (alpha), *Pdx1* [319] (beta) and *Hhex* [320] (delta), and the lineage-associated gene *Irs4* [321] (epsilon) (Figure 3.7f) to chart trends based on CellRank's fate probabilities. We found all of these genes to be upregulated correctly when approaching their corresponding terminal populations.

Robustness analysis. We wanted to evaluate the robustness of CellRank's results with respect to the following key pre-processing parameters:

- the weight given to the `ConnectivityKernel` versus the `VelocityKernel` (Section 3.1).
- the number of neighbors K used for KNN graph construction (Subsection 2.1.2) .
- scVelo's gene-filtering parameter `min_shared_counts` which determines how many counts we require for a gene in both spliced and unspliced layers.
- scVelo's gene filtering parameter `n_top_genes` which determines how many highly variable genes are included in the velocity computation.
- the number of principal components N_l used for KNN graph construction (Subsection 2.1.2)

We varied one parameter at a time and computed macrostates as well as fate probabilities. We then compared fate probabilities for different values of the parameter by computing pairwise Pearson correlation among all possible pairs, separately for each lineage. We always computed sufficient macrostates so that the alpha, beta, epsilon, and delta states were included. The precise location of the terminal states changed slightly across parameter combinations; for this reason, the correlation values we report reflect the robustness of the entire CellRank workflow, including the computation of terminal states and fate probabilities. In addition to the 5 pre-processing parameters, we were interested to see how much fate probabilities change when we randomly subsample to 90% of cells. We subsampled 20 times, computed macrostates and fate probabilities, and compared pairwise,

as above. We found that all components of CellRank were extremely robust to parameter variation and random subsampling of cells (Figure B.2 in Appendix B).

Uncertainty propagation increases robustness to noise. We used the pancreas dataset to illustrate the effects of uncertainty propagation (Figure 3.8a). We selected two cells, one from a low noise region where velocity vectors of neighboring cells often agree and one from a high noise region. To compute transition probabilities towards nearest neighbors with the `VelocityKernel`, we used a deterministic approach that does not propagate uncertainty, our analytical approximation and MC sampling (Section 3.2.2). Differences between deterministic and stochastic transition probabilities were greatest in the high noise region; this highlights how uncertainty propagation automatically down-weights transitions towards cells in noisy areas where individual velocity vectors are less trustworthy (Figure B.2b). We confirmed that our analytical approximation gives very similar results to the exact sampling scheme (Figure B.2c,d).

We further checked whether propagating uncertainty increased robustness with respect to key pre-processing parameters, using as an example the number of neighbors K. We fixed the terminal state assignment and computed pairwise correlations with and without uncertainty propagation in the `VelocityKernel`. Next, we performed a one-sided Wilcoxon signed-rank test separately for each lineage using the scipy [303] implementation with an exact distribution for the test statistic. This test assumes independently distributed paired data. In our case, pairs are given by correlations of fate probabilities for two different numbers of neighbors K, computed with and without uncertainty propagation. We assume these to be paired as the same number of neighbors yields similar correlation values with and without uncertainty propagation. We found that propagating uncertainty leads to increased robustness of fate probabilities with respect to K; we show similar results for other pre-processing parameters in the original publication [14] (Figure B.3 in Appendix B).

3.4.3 Benchmarking against other methods

We evaluated the impact of including velocity information by benchmarking CellRank with similarity-based methods that provide cell-fate probabilities (Palantir [25], STEMNET [308] and FateID [322]) and a velocity-based method that computes initial/terminal states (velocyto [1]) on the pancreas data. Only CellRank correctly identified both initial and terminal states (Figure 3.9a). Palantir requires user-provided initial states and only identified 2 out of 4 terminal states, and STEMNET and FateID cannot determine either initial or terminal states. Velocyto cannot identify individual initial or terminal states, but outputs distributions for initial and terminal states which only overlap with beta and

$Ngn3^{low}$ EP cells, respectively (Section 2.5). Next, we supplied all methods with CellRank's terminal states and tested cell fate probabilities, finding that only CellRank and Palantir correctly identified beta as the dominant fate among $Ngn3^{high}$ EP cells (Figure 3.9b). Velocyto does not provide fate probabilities. For lineage-specific gene expression, CellRank and Palantir correctly predicted trends for key lineage drivers, whereas FateID failed to predict upregulation of *Pdx1* along the beta lineage; we show similar results for more genes in the original publication [14]. STEMNET and velocyto cannot compute gene expression trends (Figure 3.9c).

We also compared runtime and memory usage on the MEF reprogramming dataset [22] from Section 3.4.1 because it contained more cells (full dataset: 100k cells) than the pancreas dataset. (Figure 3.9d). CellRank took about 33 sec to compute macrostates on this large dataset. For fate probabilities, the (generalized) linear model STEMNET was fastest, as expected, taking only 1 min, while CellRank took about 2 min and Palantir took 1h 12 min. FateID on 90k cells took even longer and failed on 100k cells due to memory constraints while velocyto was the slowest, exceeding our time budget of 10k seconds for cell numbers exceeding 40k. Results for memory usage were similar, with CellRank requiring 3 and 5 times less peak memory than Palantir and FateID, respectively, to compute fate probabilities on 100k cells (Figure 3.9e). Only STEMNET required even less memory. The most memory-hungry method was velocyto, requiring more memory on 40k cells than any other method on 100k cells.

3.4.4 Application to lung regeneration

In regenerative settings, the typical assumption of unidirectional transitions to more differentiated states does not hold; we applied CellRank to murine lung regeneration in response to acute injury [11] to demonstrate its ability in this challenging context. The scRNA-seq example comprised 24,882 lung alveolar and airway epithelial cells, sequenced at 13 time points spanning days 2-15 past bleomycin injury (Figure B.4a,b in Appendix B) with Drop-seq [35], a lower resolution single-cell platform (Section 2.1). High plasticity between epithelial cell types has been observed when homeostasis is perturbed and the tissue environment changes, including injury-induced reprogramming of differentiated cell types to bona fide long-lived stem cells in the lung [325] and other organs [326]. In our current model of airway cell lineage hierarchy, multipotent basal cells give rise to club cells, which in turn can give rise to secretory goblet and ciliated cells [327]. Interestingly, Tata et al. [325] have shown that ablation of basal stem cells can cause luminal secretory cells to dedifferentiate into fully functional basal stem cells. Here, we applied CellRank for unbiased discovery of unexpected regeneration trajectories of airway cells.

Marcostates and fate probabilities for airway epithelial cells. We computed scVelo velocities, applied CellRank and identified 9 macrostates that were used to compute fate probabilities (Figure 3.10a,b). Fate probabilities assigned high multilineage potential to MHC-II+ club cells; this is in agreement with previous results [11] (Figure 3.10c). Focusing our analysis on airway cells, we identified three macrostates in ciliated cells, one in basal cells, and one in goblet cells. In agreement with lineage tracing experiments [328], we observed a high probability for club cells to give rise to ciliated cells (Figure 3.10c). The goblet cell macrostate was distinguished from club cells by the expression of specific mucin genes, including *Muc5b* and *Muc5ac*, as well as secreted proteins involved in innate immunity, including *Bpifb1* (Figure B.4c). Analysis of fate probabilities towards basal and goblet states revealed that, surprisingly, goblet cells are likely to dedifferentiate towards Krt5+/Trp63+ basal cells (Figure 3.10c,d and Figure B.5 in Appendix B).

CellRank predicts goblet to basal dedifferentiation. We computed a diffusion map restricted to goblet and basal cells to study the trajectory at higher resolution (Figure B.6a in Appendix B). We confirmed that the proportion of basal cells increases over time and that gene-wise velocities support the dedifferentiation hypothesis (Figure B.6b,c). Using the `GPCCAEstimator` and the coarse-grained stationary distribution $\tilde{\boldsymbol{\pi}}$, we identified early cells in the transition which we used to compute a pseudotime with Palantir (Figure B.7 in Appendix B). We combined pseudotime with basal-fate probability to define stages within the dedifferentiation trajectory in the data subset (Figure 3.10e), splitting cells that had at least 66% probability of reaching the basal state into three equally sized pseudotime bins. Stage 1 consists of goblet cells characterized by high expression of the goblet marker *Bpifb1*. Stage 2 comprises an intermediate set of cells that express both *Bpifb1* and the basal marker *Krt5*. Stage 3 consists of terminal basal cells, characterized by the basal markers *Krt5* and *Trp63*, and no expression of *Bpifb1* (Figure 3.10e).

Immunofluorescence confirms novel intermediate cell states. Our novel model of goblet cell dedifferentiation predicts that after injury, the proportion of cells in stage 2 should increase as these represent intermediate cells in the dedifferentiation bridge towards basal cells. To validate this prediction, we assessed *Bpifb1*, *Krt5* and *Trp63* expression by immunofluorescence of mouse airway epithelial cells on days 10 and 21 post-bleomycin treatment, as well as in untreated animals (Figure 3.10f). We found cells from stage 1 (goblet) and stage 3 (basal) in both control and treated mice. However, we found intermediate stage 2 cells only in 10-day post-treatment mice (Figure 3.11g). Moreover, we also found triplet positive cells which only appeared after injury (see the original publication

[14]). Goblet cell hyperplasia, an increase in the number of mucous secreting cells in the airways, is a prominent feature in several chronic inflammatory conditions [329]. The novel dedifferentiation trajectory to basal stem cells that CellRank analysis predicted is unexpected; it suggest a route for generating multipotent stem cells in the resolution phase of the regenerative injury response.

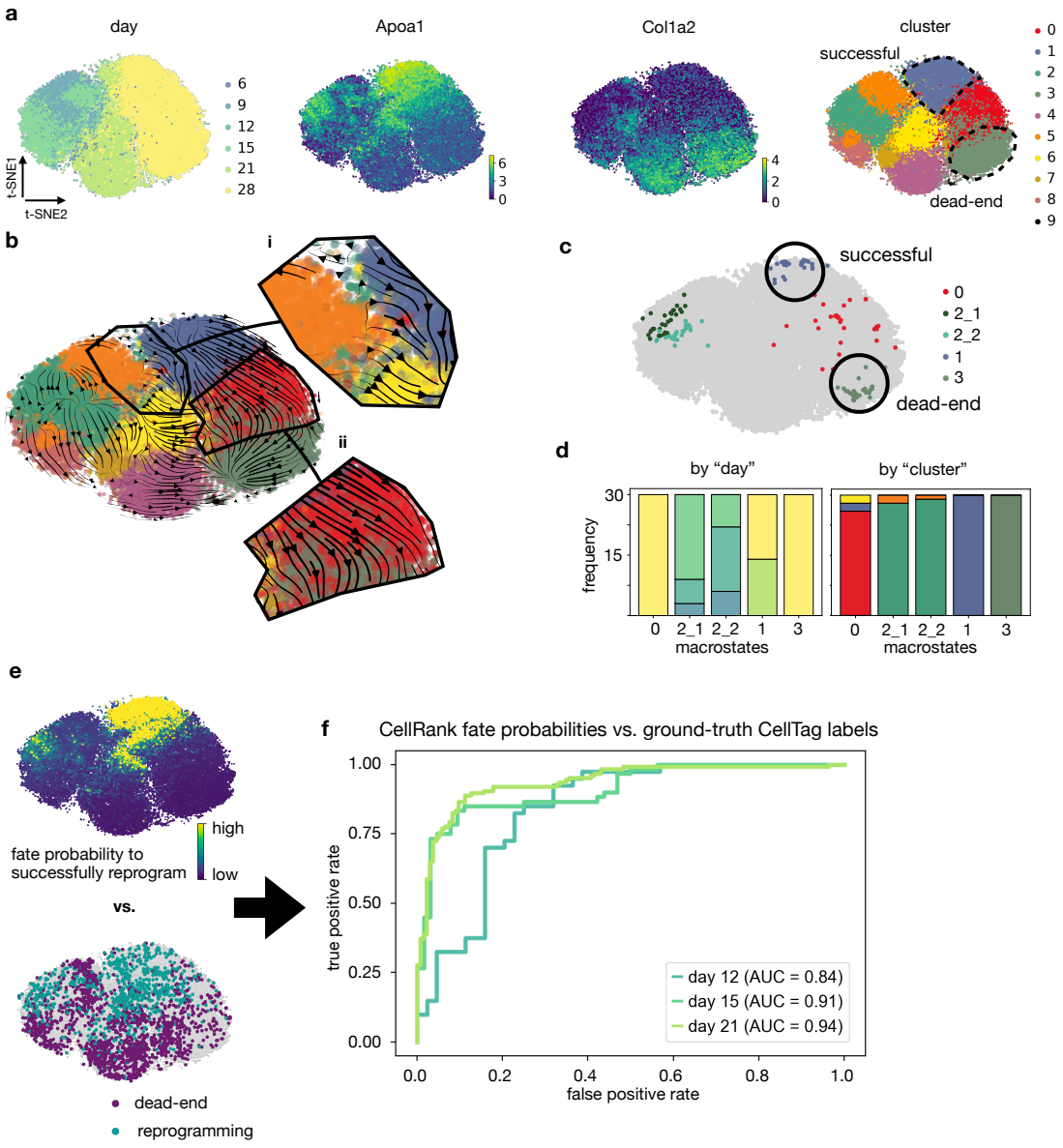

Figure 3.6: CellRank predicts reprogramming outcome. **a**. t-SNE [124, 125] embeddings of 48,515 MEFs reprogramming to induced endoderm progenitors [314, 315] (iEPs). Each cell is represented by a dot, colored by days since reprogramming induction; expression of the iEP marker *Apoa1* (indicative of reprogramming success); expression of the MEF marker *Col1a2* (indicative of reprogramming failure); or cluster annotations [22]. Clusters 1 and 3 (dashed lines) represent successful and dead-end states, respectively [22]. **b.** scVelo [16] velocities shown as streamlines. Velocities do not reveal a route towards successful reprogramming (**i**) and falsely show a transition from successful to dead-end states (**ii**). **c.** CellRank-computed macrostates, colored by cluster from (**a**) that they mostly overlap with. **d.** Distribution over reprogramming day (left) and cluster (right), colored as in (**a**). Macrostates 1 and 3 only contain late-stage cells (days 21 and 28) from clusters 1 and 3, respectively; thus, they represent the successful and dead-end states. **e.** CellRank's fate probabilities towards the successful macrostate 1 (top) and ground-truth labels from CellTagging [22] lineage tracing (bottom). **f.** ROC curves of CellRank fate probabilities at days 12, 15, and 21, based on classifiers trained to predict reprogramming outcome using CellTag labels (**e**) as ground truth. Figure reproduced from Lange et al. [14].

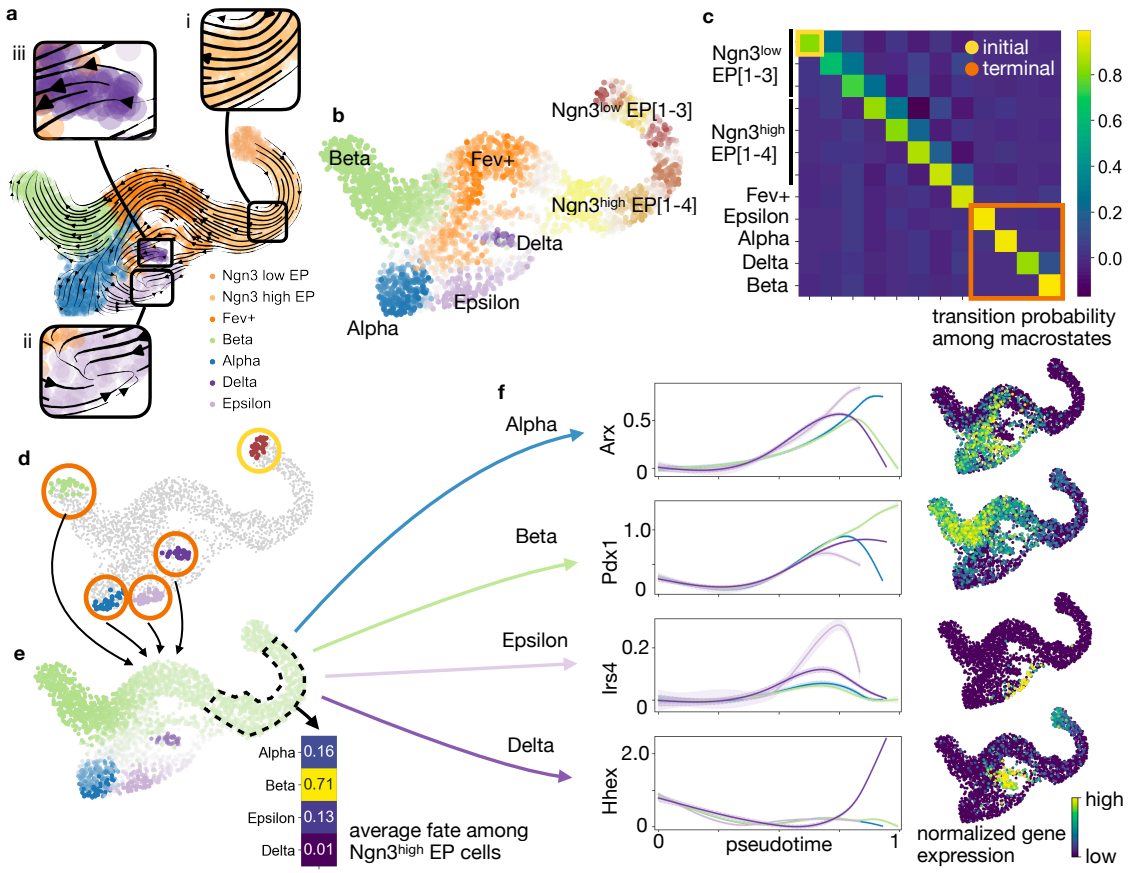

Figure 3.7: Delineating fate choice in pancreatic development. **a.** UMAP of E15.5 mouse pancreatic development with scVelo projected streamline velocities. Colors correspond to published cluster annotations [23]. CellRank provides additional insights regarding (**i**) the fate of early cells, (**ii**) the identification of terminal states and (**iii**) likely progenitors of terminal fates (boxed insets). **b.** Soft assignment of cells to macrostates. Cells are colored by the most likely macrostate; color intensity reflects the degree of confidence, and grey cells reside between multiple macrostates. **c.** Coarse-grained transition probabilities between macrostates. Terminal macrostates are circled in red and the initial Ngn3^{low} EP$_1$ macrostate is circled in yellow. **d.** Highlight of the 30 most confidently assigned cells for each initial and terminal macrostate, colored according to (**b**). **e.** UMAP displaying probabilities for reaching alpha, beta, epsilon, and delta terminal fates. Fates are colored as in (**b**), with darker color indicating elevated probability. Inset shows mean fate probabilities of cells in the Ngn3^{high} EP cluster marked with a dashed line. **f.** Smoothed pseudotime gene expression trends; each colored trend is weighed by `GPCCAEstimator`-computed fate probabilities as indicated for the lineage determinants *Arx* [318] (alpha), *Pdx1* [319] (beta), and *Hhex* [320] (delta) as well as the lineage associated gene *Irs4* [321] (epsilon). We show for each gene and trajectory the trend leading up to the indicated terminal population. Right column, expression values for the corresponding gene in the UMAP. Figure reproduced from Lange et al. [14].

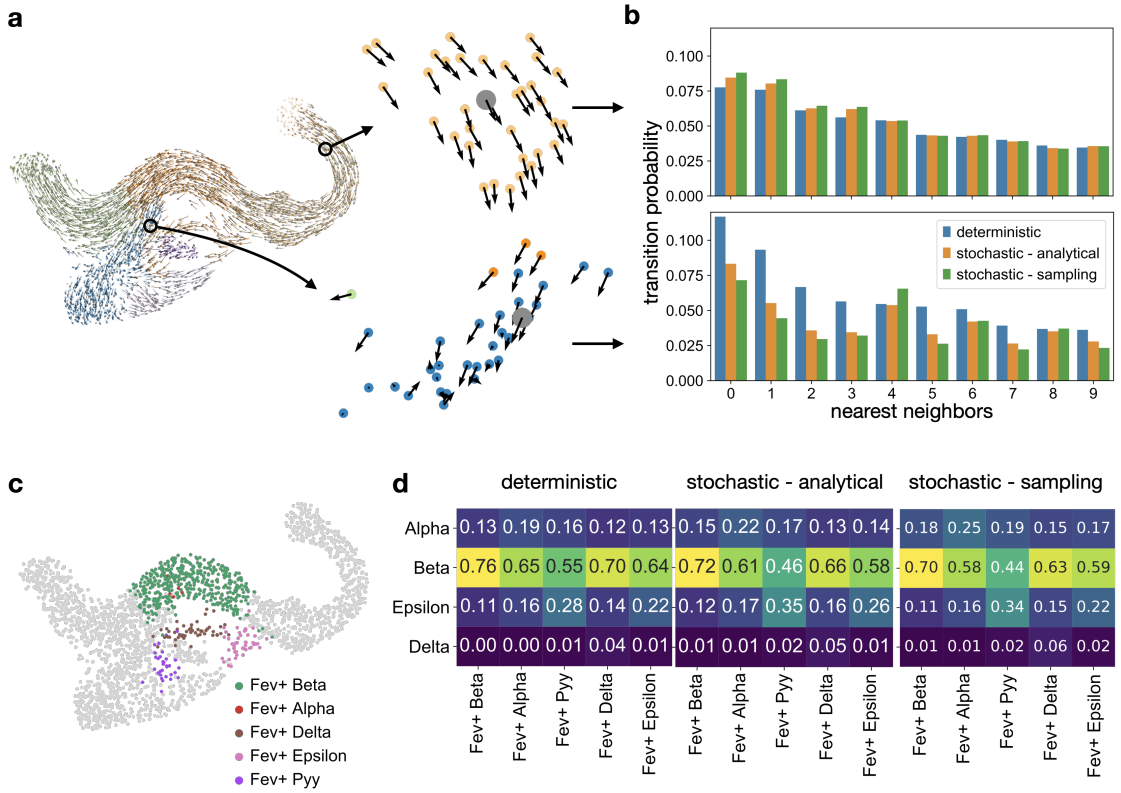

Figure 3.8: Uncertainty propagation adjusts for noise in RNA velocity vectors. a. Effect of noise propagation in the `VelocityKernel`, illustrated on the pancreas data. We highlight one cell from a low noise region, where velocity vectors from neighboring cells tend to point in the same direction (top), and one from a high noise region, where vectors from neighboring cells point in different directions (bottom). **b.** Transition probabilities from the reference cell to its 10 nearest neighbors using a deterministic or stochastic (analytical approximation or Monte Carlo sampling-based) approach, for both low and high-noise cells. Corrections applied by stochastic approaches are larger in the high noise region. **c.** Subclustering of cells in the Fev+ cluster [23]. *d.* Comparing mean fate probabilities per subcluster. We obtained these from not propagating ("deterministic") or propagating ("stochastic - analytical" and "stochastic-sampling") velocity uncertainty. Both stochastic approaches agree on down-weighting probability towards the dominant beta fate and up-weighting probability towards the alpha, delta, and epsilon fates. Figure reproduced from Lange et al. [14].

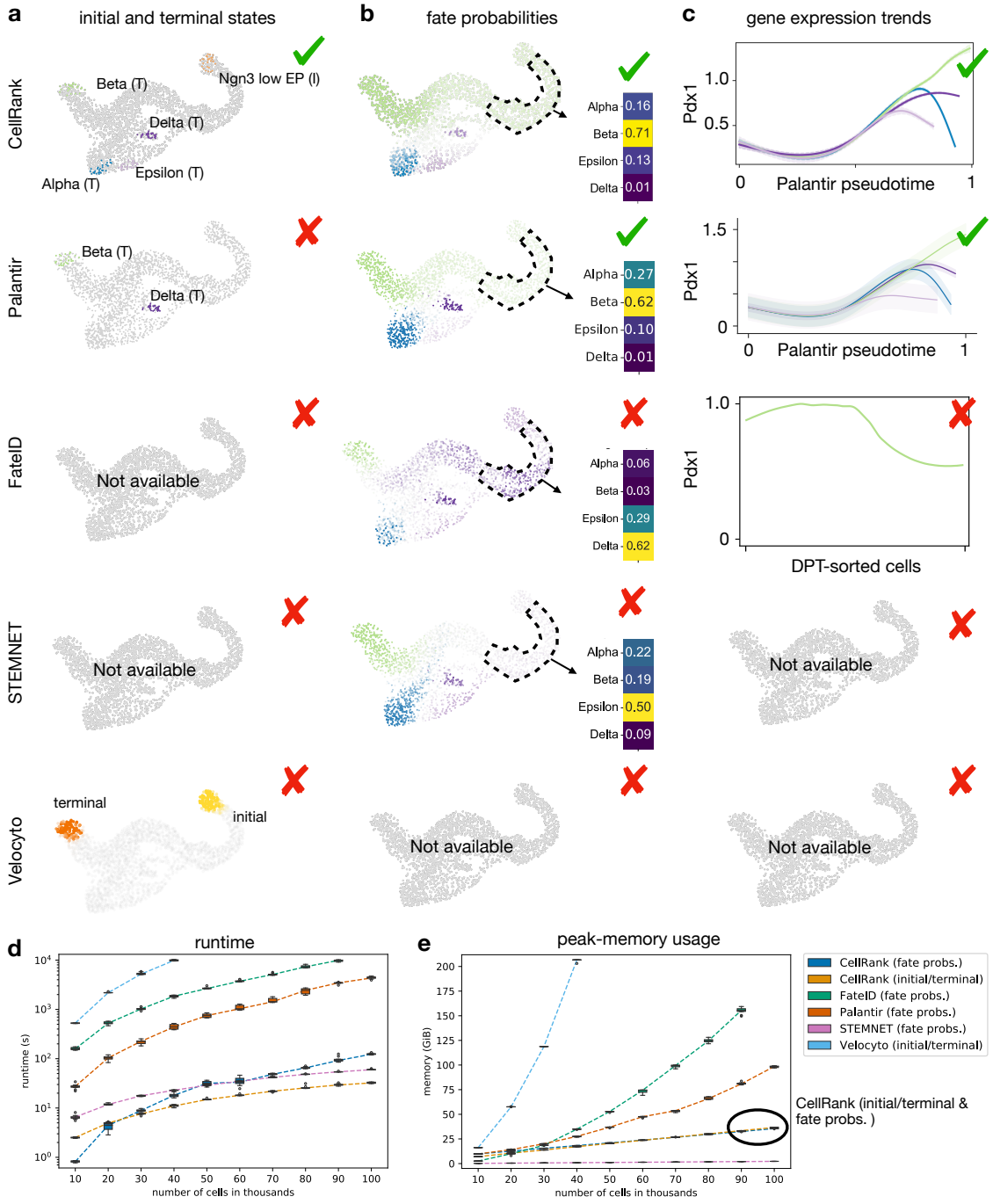

Figure 3.9: CellRank outperforms other cell-fate inference methods. (a–c) **a.** CellRank automatically identified the terminal alpha, beta and epsilon states as well as the initial Ngn3^{low} EP state. Palantir [25] identified terminal beta and delta states. **b.** Only CellRank and Palantir correctly predict the dominant beta fate among Ngn3^{high} EP cells. **c.** Gene expression trends for the beta-regulator *Pdx1* [319, 323, 324]. On the x-axis is pseudotime used by the corresponding method, and on the y-axis is gene expression. We show one smoothed trend per trajectory for CellRank and Palantir and a smoothed trend along just the beta trajectory for FateID. CellRank and Palantir correctly identify *Pdx1* upregulation along the beta lineage. FateID fails to do so while STEMNET [308] and velocyto do not offer options to visualize lineage-specific gene expression trends. **d,e.** Boxplots comparing computational runtime (**d**) and peak-memory usage (**e**) on the MEF reprogramming dataset [22] for different methods. We split the dataset into 10 subsets of increasing size and run each method on each subset 10 times. Boxes cover 25 to 75% quantiles, the line indicates median, whiskers extend to 1.5x the interquartile range and dots represent outliers. Figure reproduced from Lange et al. [14].

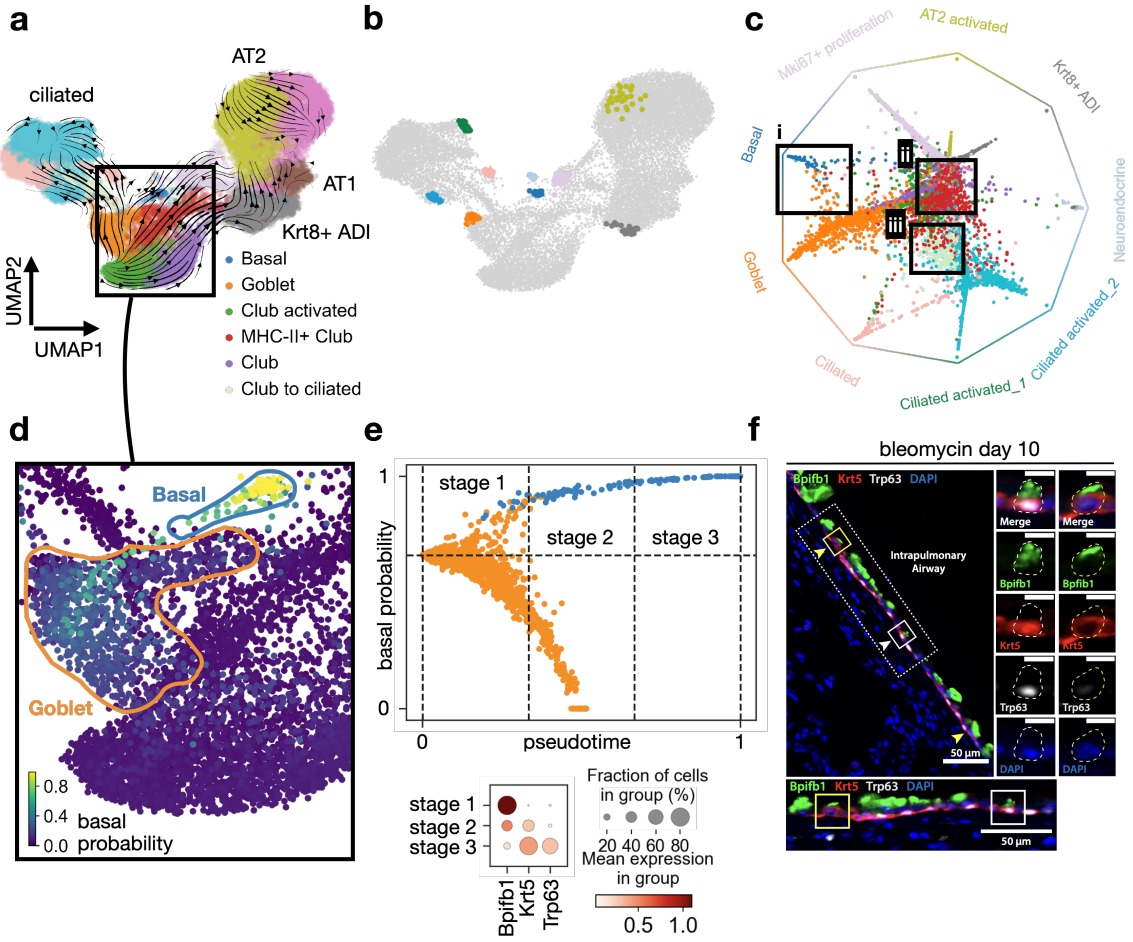

Figure 3.10: CellRank predicts a novel dedifferentiation trajectory in murine lung regeneration. **a.** UMAP of 24,882 epithelial cells from 13 time points colored according to cluster annotations [11]. Streamlines show projected scVelo velocities and the box highlights a subset of airway cells. **b.** CellRank computed macrostates. **c.** Circular projection [83, 308] of cells according to fate probabilities towards the macrostates shown in (**b**). Boxes highlight goblet cells likely to reach the basal terminal state (**i**), MHC-II+ club cells with high multi-lineage potential (**ii**) and club cells likely to transition to ciliated cells (**iii**). **d.** Cells in UMAP colored by CellRank-computed fate probabilities towards the basal cell macrostate, showing a route from goblet to basal cells **e.** The dedifferentiation stages are characterized by the expression of *Bpifb1* (goblet), *Krt5* (early basal), and *Trp63* (late basal); stage 1 corresponds to goblet, stage 2 to intermediate, and stage 3 to basal cells (bottom). **f.** Immunofluorescence stainings for *Bpifb1* (green), *Krt5* (red), *Trp63* (white), and DAPI (blue) in mouse lung tissue sections 10 days past bleomycin injury. We detect cells from the intermediate stage 2 (*Bpfib1+/Krt5+/Trp63-*) in bleomycin-injured lungs (yellow squares and arrowheads). Scale bars represent 50μm, 10μm for zoom-in images. In each panel, dotted boxes are magnified at the bottom, and solid-boxed cells are magnified at the right, showing individual and merged channels. Representative images are derived from two independent biological replicates. Figure adapted from Lange et al. [14].

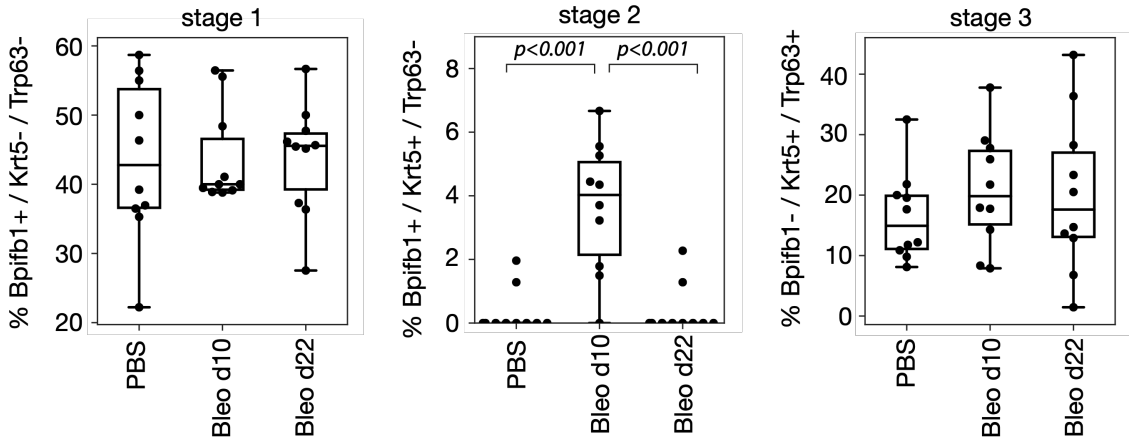

Figure 3.11: Quantification of stage abundance post-injury. **a.** Quantification of stage-dependent cell abundance in wild type (PBS), 10 days post bleomycin injury (bleo d10), and 22 days post-injury (bleo d22) mice. We quantified ten independent pulmonary airway regions per condition over 2 biologically independent experiments. Bleo d10 is significantly enriched for stage 2 cells (Nested One-Way ANOVA with Tukey's multiple comparison test, $P < 10^{-3}$). Figure adapted from Lange et al. [14].

3.5 Extensions of the CellRank framework

The previous sections introduced CellRank, a modular framework to study cellular state transitions, and showcased its performance in a number of applications. To derive the transition matrix T, we made use of gene expression similarity and RNA velocity through the `ConnectivityKernel` and the `VelocityKernel`, respectively. While this combination successfully captured known biology in the MEF [22] and pancreas [23] examples and predicted new biology in the lung example [11], there are challenges to velocity-type models which limit the applicability of this approach in practice.

Limitations of RNA velocity. We touched upon the main limitation in Section 2.5; RNA velocity depends on unspliced count abundance which is biased by gene structure. Other limitations include its dependency on the time scales of splicing kinetics which cannot be controlled externally, batch effects which currently cannot be corrected for in velocity data, and high noise levels which in practice only allow for moment-based modeling. Besides these experimental challenges, computational challenges include a lack of velocity models which estimate time- and state-dependent transcriptional parameters $\alpha^{\text{on/off}}, \beta, \gamma$ to account for phenomena such as transcriptional bursting [274, 330], or models which include gene-gene interactions rather than fitting each gene individually.

CellRank, through the combination of `VelocityKernel` and `ConnectivityKernel`, alleviates some of these challenges by using a KNN graph to regularize velocity vectors, propagating noise, and using a stochastic formulation. Nevertheless, we had very little success applying this approach to systems like hematopoiesis where velocity vectors are systematically biased. The reason for this bias is currently under debate; in developmental settings, transcriptional bursting has been identified as a possible cause [330]. For example, on a dataset of CD34+ human bone marrow cells [25], RNA velocity vectors point in the opposite direction the known ground truth in the system despite sufficient overall capture of unspliced transcripts (33%) (Figure 3.12a,b). We speculated this may be caused by a cluster of common lymphoid progenitor cells (CLPs) which form an outlier in phase portraits and heavily bias scVelo's parameter fits (Section 2.5 and Figure 3.12c). However, upon removing CLPs and re-running the model, velocities remained largely inconsistent with ground truth (Figure 3.12d), possibly due to a number of top-likelihood genes which require time- and state-dependent velocity parameters currently not supported by scVelo's model of splicing kinetics (Figure 3.12e) [274, 330].

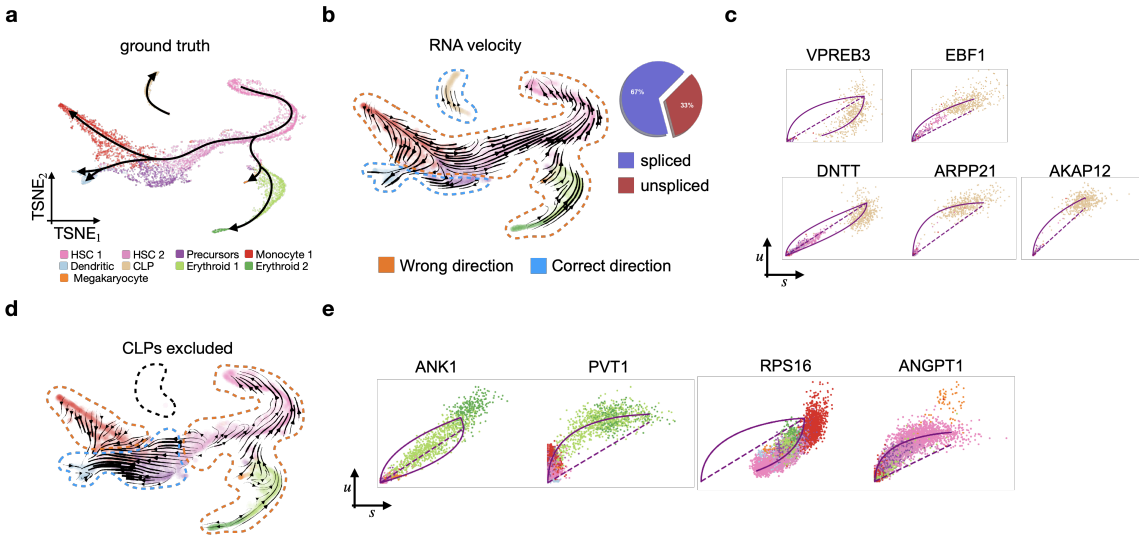

Figure 3.12: RNA velocity is systematically biased for hematopoiesis. a. t-SNE embedding of CD34+ human bone marrow cells, each dot denotes a cell, colored according to original cluster annotations [25]. Arrows denote the ground-truth direction of differentiation which is known for this well-studied system. (HSC: hematopoietic stem cell, MK: megakaryocyte, CLP: common lymphoid progenitors) **b.** RNA velocity, displayed as streamlines in the embedding of (**a**). Orange (blue) outlines highlight incorrect (correct) velocity flow. The pie chart in the top-right corner displays the proportion of spliced and unspliced transcripts. **c.** Top-likelihood genes according to scVelo's likelihood-based dynamical model [16] (Section 2.5); x-axis (y-axis) shows spliced (unspliced) counts, each dot denotes a cell, colored according to clusters as in (**a**). CLP cells form an outlier in all fits and bias inference. **d.** Upon removing CLPs, projected velocities remain largely inconsistent with ground truth. **e.** Top likelihood genes as in (**c**) upon removing CLPs; *ANK1* and *PVT1* show different dynamics between Erythroid clusters and the rest, *RPS16* has the inverse time assignment, possibly due to bursty kinetics [274, 330] and *ANGPT1* shows different dynamics between HSC clusters and the rest. Phase portraits computed by Philipp Weiler.

CellRank's modular design overcomes the limitations of RNA velocity. CellRank overcomes the limitations of RNA velocity by implementing kernels that estimate the direction of cellular state changes in the absence of velocity information. We introduce three of these additions here:

- the `PseudotimeKernel` draws on the vast amount of methods developed for pseudotime inference (Section 2.4); it uses their output (the pseudotime) to bias graph edges to point in the direction of cellular state changes (Section 3.5.2).
- the `CytoTRACEKernel` extends the `PseudotimeKernel` to situations where no pseudotime can be computed, possibly because the root cell is unknown. As a proxy for pseudotime, it computes the CytoTRACE score, an unsupervised measure of

developmental potential [26] (Section 3.5.3).

- the `RealtimeKernel` makes use of real-time information where it is available to direct cellular state changes; it interfaces with an optimal transport-based method [20] to compute couplings P across time-points (Section 3.5.4).

Besides these new kernels, which allow applications to new settings, we also introduce new shared kernel methods, i.e. functionalities that are implemented in a base kernel class and which are inherited by every derived kernel. These functionalities include random walk simulations as well as embedding projections of the transition matrix T.

Downstream of transition-matrix computation, the `GPCCAEstimator` is agnostic with respect to kernel choice; it works for any (sparse) cell-cell transition matrix T. Thus, the full functionality demonstrated in the context of the velocity applications can be transferred to the new setting. Moreover, the kernel arithmetics introduced in Section 3.1 extend to the new kernels, making it possible to combine diverse sources of directionality within the unified CellRank framework.

3.5.1 Methods available in every CellRank kernel

We introduce two visual approaches to qualitative transition matrix interpretation which are available through every CellRank kernel.

Random walk simulation. Random walks can be initialized in random cells or in a predefined cluster of "early" cells, they can be stopped after a certain number of steps or when they reach a predefined set of clusters. Random walks can be plotted in any 2D embedding and provide a qualitative first check of the dynamics captured by T.

Projection of the transition matrix into a 2D embedding. Embedding projections have been very popular for the visual interpretation of high dimensional RNA velocity vector fields [1, 16] and can be extended to any KNN-graph-based CellRank kernel (applies to all kernels but the `RealtimeKernel`). An embedding projection of T is given by a 2D vector field, visualized in any low-dimensional embedding. While we argued against these projections as the sole method to analyze velocity data, we advertise them as part of an analysis workflow that starts with a more intuitive interpretation and proceeds to quantitative analysis through the `GPCCAEstimator`.

To compute the projected vector $\boldsymbol{v}_i \in \mathbb{R}^2$ for embedded cell $\boldsymbol{z}_i \in \mathbb{R}^2$ according to T,

consider its neighbors $\mathcal{N}_i = \{1, 2, ..., K_i\}$ in the KNN graph $\mathcal{G}$. Suppose these neighbors have embedded profiles $\{\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_{K_i}\}$. The 2D embedding coordinates can be computed using any dimensionality reduction technique, including t-SNE [124, 125], UMAP [121, 122] and PCA. We define the projected vector via

$$\boldsymbol{v}_i = \sum_{j \in \mathcal{N}_i} \left(T_{ij} - \frac{1}{K_i} \right) \frac{\boldsymbol{z}_j - \boldsymbol{z}_i}{\|\boldsymbol{z}_j - \boldsymbol{z}_i\|}, \tag{3.32}$$

where substracting $1/K_i$ ensures that the projection is zero for uniform transition probabilities to nearest neighbors. CellRank can visualize the vector field itself or smoothed versions thereof, using e.g. a Gaussian kernel over a grid [1], or scVelo's popular streamplots [16]. In the following, we show examples for kernel projections into low-dimensional embeddings for the `PseudotimeKernel` (Figure 3.14b) as well as the `CytoTRACEKernel` (Figure 3.16d).

3.5.2 The `PseudotimeKernel`

The motivation for developing the `PseudotimeKernel` was to make use of the vast amount of methods available to compute pseudotemporal orderings of cells [12] and to integrate their output into CellRank. Pseudotime can be computed robustly for systems with a single known initial state with unidirectional transitions out of this state towards a set of terminal states. Examples of this setting are given by adult hematopoiesis [10, 25] and many developmental systems [258]. Albeit the initial state being known, the `GPCCAEstimator` can be applied to such settings to identify terminal states and to compute fate probabilities.

Computing the transition matrix. Given a KNN graph $\mathcal{G}$ and any precomputed vector of pseudotimes $\boldsymbol{\tau} \in [0, 1]^{N_c}$, the `PseudotimeKernel` biases edges in $\mathcal{G}$ to point into the direction of increasing pseudotime. This is similar to the Palantir [25] model; however, while Palantir discarded edges that point into the "pseudotime past", the `PseudotimeKernel` employs an adaptive scheme that gradually down-weights graph edges in the pseudotime past (Figure 3.13).

In particular, consider reference cell $\boldsymbol{x}_i \in \mathbb{R}^{N_g}$ with nearest neighbors $\mathcal{N}_i = \{1, 2, ..., K_i\}$ according to KNN graph $\mathcal{G}$. Suppose $\mathcal{G}$ is weighted with (sparse) adjacency matrix $W \in \mathbb{R}_+^{N_c \times N_c}$ reflecting cell-cell similarity on any molecular layer (Section 2.2.5). Using the precomputed vector of pseudotimes $\boldsymbol{\tau} \in [0, 1]^{N_c}$, define the pseudotime displacements $\Delta\tau_{ij} \coloneqq (\tau_i - \tau_j) \, \forall j \in \mathcal{N}_i$. We compute a directed version W' of W by adaptively down-weighting graph edges that point into the pseudotime past, $W'_{ij} = f(\Delta\tau_{ij}) W_{ij}$ for weighting

function $f : \mathbb{R} \to \mathbb{R}_+$ given by

$$f(\Delta\tau_{ij}) \coloneqq \begin{cases} 1 & \Delta\tau_{ij} \geq 0\,, \\ \frac{2}{\sqrt[\nu]{1+\mathrm{e}^{b\Delta\tau_{ij}}}} & \Delta\tau_{ij} < 0\,, \end{cases} \tag{3.33}$$

where we use default values of $b = 10$ and $\nu = 0.5$. The trajectory inference method VIA [331] used a similar scheme to adaptively weight graph edges in $\mathcal{G}$. Transition probabilities T are computed from directed similarities W' by softmax normalization as in the `VelocityKernel` (Section 3.2.2). The resulting transition matrix T is sparse and allows for rapid application of the `GPCCAEstimator`.

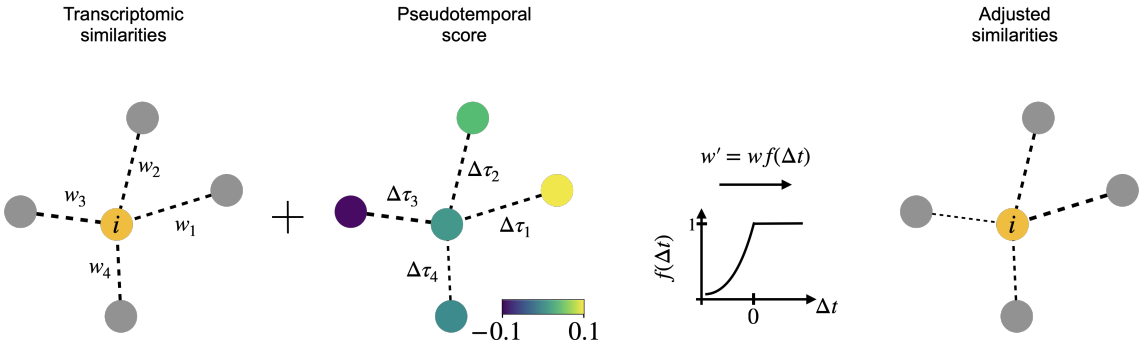

Figure 3.13: The `PseudotimeKernel` directs KNN graph edges. Schematic of the `PseudotimeKernel`; cell-cell similarities from the adjacency matrix W of a KNN graph are biased into the direction of increasing pseudotime τ by means of a weighting function f.

Application to adult hematopoiesis. Applications of RNA velocity to the hematopoietic system have not been successful to date; velocity vectors often point in opposite directions to the expected, known developmental hierarchy. To demonstrate this, we computed RNA velocity for a dataset of steady-state adult hematopoiesis [25] and projected the velocity vectors into the original t-SNE embedding (Figure 3.14a). As expected, the projected vectors consistently pointed opposite to the ground-truth direction.

For developmental hematopoiesis, the problem of biased velocity estimates has been attributed to "bursty" expression kinetics which are not captured by the current model of the mRNA lifecycle [330] (Section 2.5). For the example of adult (steady-state) hematopoiesis we discussed above (Figure 3.12), we speculated this system might require state- and time dependent kinetic rate parameters. Further, there may be a mismatch of the time scales of hematopoiesis and splicing kinetics or informative genes for the process might not give rise to sufficient unspliced counts as a result of their gene structure (Section 2.5).

To overcome the challenge of biased velocity vectors, we applied the `PseudotimeKernel`, using as input a cell-cell similarity KNN graph $\mathcal{G}$ as well as Palantir's pseudotime which has been shown to capture the biology of this system well [25] (Figure 3.14b). We projected the resulting transition matrix T onto the original t-SNE embedding using the generic kernel projection method presented above. The projected transition matrix, visualized via streamlines in the embedding, captures the gradual commitment of hematopoietic stem cells (HSC) towards the various endpoints in this system, including Monocytes and Erythroids. When we further applied the `GPCCAEstimator` for a quantitative assessment of the dynamics, we successfully recovered all initial and terminal states in the system (Figure 3.14b).

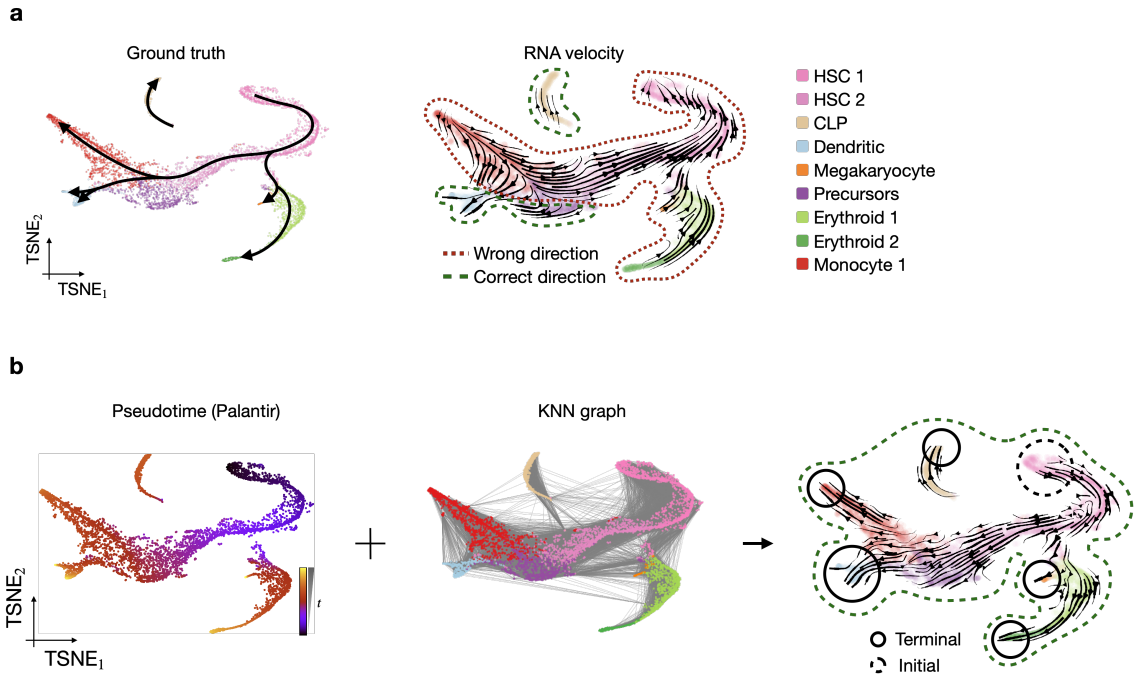

Figure 3.14: The `PseudotimeKernel` captures adult hematopoiesis a. Left: t-SNE embedding of cells from the adult hematopoietic system; embedding coordinates and cluster annotations as in the original publication [25]. Arrows indicate the known developmental hierarchy. Right: scVelo-computed RNA-velocity estimates projected into the t-SNE embedding and shown as streamlines. **b.** Palantir's pseudotime is combined with the KNN graph in the `PseudotimeKernel` to compute transition probabilities; an embedding projection of T is consistent with ground truth and the `GPCCAEstimator` finds the correct initial and terminal states. Visualizations in (**a**) and (**b**) created jointly with Philipp Weiler.

3.5.3 The `CytoTRACEKernel`

Following the success of the `PseudotimeKernel`, we wanted to enable a similar transition matrix construction in situations where no precomputed pseudotime $\boldsymbol{\tau}$ is available, possibly because the initial state is unknown or because multiple initial states exist. An alternative approach to pseudotime construction which requires no prior knowledge and which can handle several initial states is given by the CytoTRACE score [26], a heuristic measure of developmental potential between zero and one where naive (differentiated) cells are assigned large (small) values. For CytoTRACE score given by $\boldsymbol{\tau}' \in [0,1]^{N_c}$, we use $\boldsymbol{\tau} \coloneqq \mathbf{1} - \boldsymbol{\tau}'$ as a pseudotime and proceed with transition matrix construction as for the `PseudotimeKernel`.

The CytoTRACE score. CytoTRACE is based on the assumption that on average, stem-like cells express more genes (at low levels) than differentiated cells because their chromatin is regulated less tightly [26]. While this assumption is likely to hold in many developmental systems, it is incorrect in most perturbed or steady-state systems. Within developmental systems, cycling populations challenge this assumption as they express more genes than their quiescent counterparts, without necessarily being more stem-like. Thus, we recommend applying the `CytoTRACEKernel` to developmental systems and to look out for cycling populations to make sure they are classified correctly. Based on the assumption that stem-like cells express more genes, the method performs the following steps:

(i) For each cell $\boldsymbol{x}_i \in \mathbb{N}^{N_g}$, count the number of expressed genes $n_i \coloneqq \sum_{j=1}^{N_g} \mathbb{1}\left(x_{ij} > 0\right)$ for indicator function $\mathbb{1}\left(\cdot\right)$. Let $\boldsymbol{n} \in \mathbb{N}^{N_c}$ be a vector containing these integers.

(ii) For each gene j, compute Pearson correlation of its expression across all cells, $X^{(R)}{:}, j \in \mathbb{N}^{N_c}$, with the number of genes expressed per cell $\boldsymbol{n}$. Let the set J contain the indices j corresponding to the top M genes with the highest positive correlation.

(iii) Let $X' \in \mathbb{R}^{N_c \times N_g}$ represent a matrix of imputed gene expression values, computed using a method like MAGIC [104] or scVelo's moment function [16]. Compute the average expression level of the genes in J according to X' via $\zeta_i \coloneqq 1/M \sum_{j \in J} X'_{i,j}$. The final CytoTRACE score $\boldsymbol{\tau}'$ is a normalized version of $\boldsymbol{\zeta}$, i.e. $\boldsymbol{\tau}' \coloneqq \boldsymbol{\zeta} / \max(\boldsymbol{\zeta}) \in [0,1]^{N_c}$; we call $\boldsymbol{\tau} \coloneqq (1 - \boldsymbol{\tau}')$ the CytoTRACE pseudotime.

The raw signal is contained in the number of genes expressed per cell $\boldsymbol{n}$, steps (ii) and (iii) can be understood as post-processing/smoothing steps which the authors of the original publication have empirically shown to enhance performance across a wide set of examples [26]. The number of top genes to be included is set to $M = 200$ by default.

Adaptation of the CytoTRACE score. In order to use the CytoTRACE score within the `CytoTRACEKernel` to direct KNN graph edges, we had to adapt it because (i) the original method was written in R, while CellRank is written in python, and (ii) the original implementation relied on an imputation step which was inefficient and did not scale well with cell number. In our adaptation, we used scVelo's moment function [16] for the imputation step, which is a simple KNN-based neighborhood smoothing that scales extremely well with cell numbers. We validated that our adapted implementation performed at least as well as the original implementation in a number of benchmark examples where a notion of ground truth was available (Figure 3.15a-c).

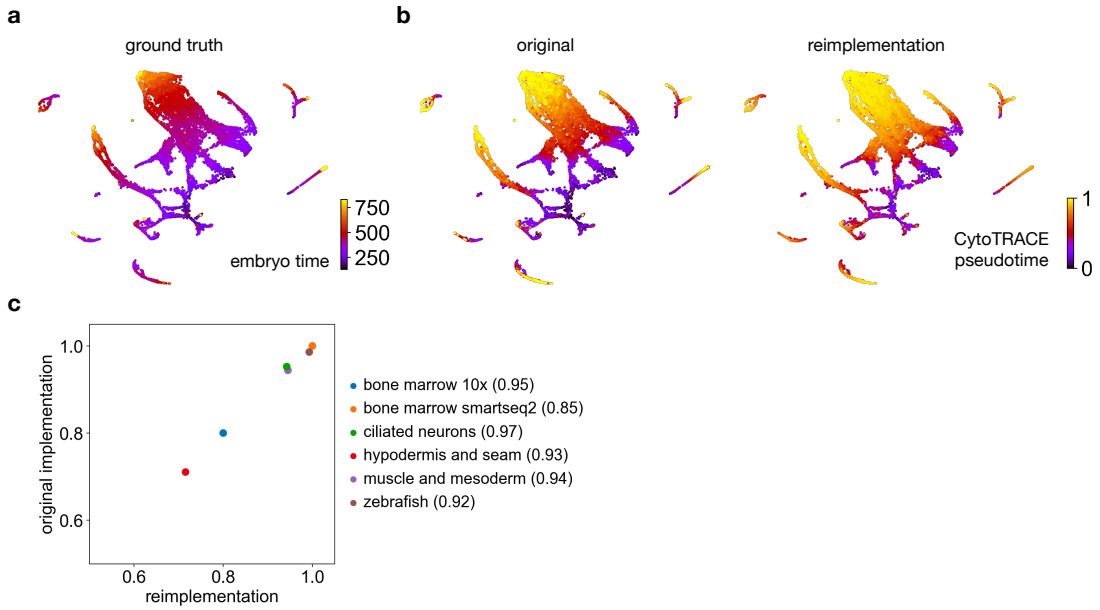

Figure 3.15: Faithful adaptation of the CytoTRACE score. a. Original t-SNE embedding of 22,370 muscle and mesoderm cells from *C.elegans* embryos, colored according to developmental stage which serves as a proxy for ground-truth cellular ordering [24]. **b.** Same t-SNE embedding, colored according to the original CytoTRACE pseudotime (left) and our adaptation (right). **c.** Systematic comparison of CytoTRACE implementations across 6 datasets, each shown as one dot. This includes two bone marrow datasets [332], three datasets of different aspects of *C.elegans* embryogenesis [24] (ciliated neurons, hypodermis and seam and muscle and mesoderm) as well as one developmental zebrafish dataset [333]. On the x- and y-axis, we measure Spearman rank correlation between average CytoTRACE score and ground truth developmental staus per dataset-defined stage for the adapted and the original implementation, respectively. Additionally, we report the Pearson correlation between the two scores for each dataset in parenthesis in the legend, validating that (i) both scores achieve a similar agreement with ground truth and (ii) the two scores are similar.

Application to early zebrafish development. We applied the `CytoTRACEKernel` to a Drop-seq [35] dataset of zebrafish embryogenesis [333] containing 694 embryos harvested across 12 stages of early development. When we projected scVelo-computed velocities on the original force-directed embedding, they displayed very noisy patterns, largely pointing opposite to the known direction given by stage progression; this was particularly pronounced for the axial mesoderm lineage (Figure 3.16a). An explanation for these noisy velocities could be given by the low fraction of unspliced reads which was only at 3% (Figure 3.16b); this value is low even for Drop-seq [35], which usually results in somewhat lower unspliced read fraction (about 10-20%).

Focusing on the axial-mesoderm lineage (Figure 3.16c), we applied the `CytoTRACEKernel`; the resulting CytoTRACE pseudotime largely followed the ordering prescribed by experimental stages. Next, we used this score to direct graph edges and computed a transition matrix T. When we projected T into the original embedding using the method described above and visualized transitions with streamlines, we visually observed a good correspondence with stage progression (Figure 3.16d). To interpret these results in a quantitative fashion, we applied the `GPCCAEstimator` to T and compared with cluster labels from the original publication (Figure 3.16e). Macrostates contained the two terminal states (Prechordal Plate and Notochord) as well as the initial state (Early Blastomeres). When we computed fate probabilities towards the two terminal states, these revealed gradual lineage commitment in agreement with the force-directed embedding (Figure 3.16f).

3.5.4 The `RealtimeKernel`

Many scRNA-seq datasets contain samples taken at different (experimental) timepoints; these provide additional information on the direction of cellular state changes which we ignored in the kernels presented so far. We discussed the Waddington Optimal Transport [20] (WOT) method in Section 2.4 which uses optimal transport (OT) to link cells across timepoints; this approach provides an explicit strategy to include timepoint information. In this section, we introduce the `RealtimeKernel` which wraps around WOT and allows the resulting coupling to be interpreted as a Markov transition matrix within the CellRank framework. We build on this work in Chapter 4 where we greatly accelerate WOT in `moscot-time` and include lineage-tracing information in `moslin`.

From couplings to transition matrices. Given two timepoints t_i and t_{i+1} with N and M cells, respectively, application of WOT yields a coupling matrix $P \in \mathbb{R}_+^{N \times M}$ which probabilistically relates cells at the two timepoints. For uniform left marginal $\boldsymbol{a} \in \Delta_N$ with $a_i = 1/N \, \forall i$, the coupling P may be re-normalized to yield a row-stochastic transition

matrix $T \in \mathbb{R}_{+}^{N \times M}$. For non-uniform left marginal $\boldsymbol{a}$, re-normalization ignores information contained in $\boldsymbol{a}$ such as cellular growth- and death-rates - a current limitation of the `RealtimeKernel` (Section 2.4).

Combining within- timepoint with across timepoint transitions. Given a time series scRNA-seq dataset with timepoints t_i for $i \in \{1, ..., I\}$, WOT computes pairwise coupling matrices $P^{(t_i, t_{i+1})}$ which are translated to pairwise transition matrices $T^{(t_i, t_{i+1})}$ by the `RealtimeKernel`. To combine these into one large transition matrix T spanning all time points, we place the individual $T^{(t_i, t_{i+1})}$ on the first super-diagonal of the large T (Figure 3.17a). This yields a transition matrix T in which every cell from the final timepoint t_I represents an absorbing state in the Markov chain; every random walk terminates as soon as the final time point is reached (Section 2.2). To allow random walks to diffuse within the final time point, we apply the `ConnectivityKernel` to the final time point and place the resulting transition matrix in the corresponding diagonal spot on the large T (Figure 3.17b). Similarly, to allow random walks to transition within each timepoint rather than moving directly to the next, we apply the `ConnectivityKernel` to each remaining earlier timepoint and include the resulting transition matrices on the diagonal (Figure 3.17c). The final transition matrix T explicitly includes timepoint information and allows transitions both within- as well as across (subsequent) timepoints.

Sparsifying coupling matrices to accelerate computations. The large transition matrix T constructed as above contains dense blocks on the super-diagonal resulting from re-normalized coupling matrices $P^{(t_i, t_{i_1})}$; these are dense as they have been computed using entropically regularized OT in WOT [20] (Section 2.3). However, this is problematic as the `GPCCAEstimator` is designed for sparse matrices; the computations of both macrostates, as well as fate probabilities, exploit sparsity through iterative algorithms which apply matrix-vectors products. For large dense matrices, application of the `GPCCAEstimator` becomes prohibitively expensive from a computational standpoint. In order to overcome this challenge, we implemented an adaptive thresholding scheme that excludes entries in the individual couplings $P^{(t_i, t_{i+1})}$ if they fall below a certain threshold, thus sparsifying the couplings before converting them to transition matrices $P^{(t_i, t_{i+1})}$. We used an in-vitro reprogramming dataset [20] to validate that fate probabilities towards fixed macrostates were extremely similar with and without the thresholding scheme. In particular, for each one of the four terminal states considered (Neural, iPSC, Stromal, and Trophoblast), the correlation between fate probabilities computed with and without the thresholding scheme was above 0.99. However, the computation time for both macrostates and fate probabilities

was reduced by about one order of magnitude when we used the adaptive thresholding scheme.

Application to MEFs reprogramming towards iPSCs. We applied our transition matrix construction from above to a timecourse dataset of MEFs reprogramming towards induced pluripotent stem cells (iPSCs) and a few other endpoints including Neural, Trophoblast, and Stomal fates [20]. The dataset contained 18 days with dense temporal resolution (sequenced at least every 12h, Figure 3.18a). When we computed a transition matrix T using the `RealtimeKernel` and visualized it in a force-directed embedding using random walks of 200 steps, we found most random walks to terminate in the expected final cell states (Figure 3.18).

To gain further insights into the reprogramming process, we applied the `GPCCAEstimator` and computed 6 macrostates which contained mostly cells from later days and overlapped with the expected terminal states (Figure 3.18c). Focusing on the Neural_1, IPS, Trophoblast, and Stomal macrostates, we computed fate probabilities and found that only a small, distinct set of cells has the potential to successfully reprogram towards IPS cells, as shown in the original publication [20] (Figure 3.18).

To further validate fate probabilities, following the original publication [20], we computed log-odds ratios to transition towards IPS versus any other state and visualized these across timepoints. When we overlaid both *Obox6* expression, a TF known to be involved in the reprogramming process [20], as well as cell type annotations, we found that cells in the mesenchymal-to-epithelial transition (MET) preferentially reprogrammed towards IPSCs and expressed *Obox6*, a finding consistent with the original publication [20] (Figure 3.18e). Finally, we correlated both Trophoblast as well as Neural fate probabilities with gene expression and arranged genes in the plane according to their correlation values (Figure 3.18f). We found that correlation values recapitulated gene importance for Trophoblast and Neural fates shown in the original publication. As expected, genes implicated in Stromal fate establishment correlated negatively with both Trophoblast and Neural identities.

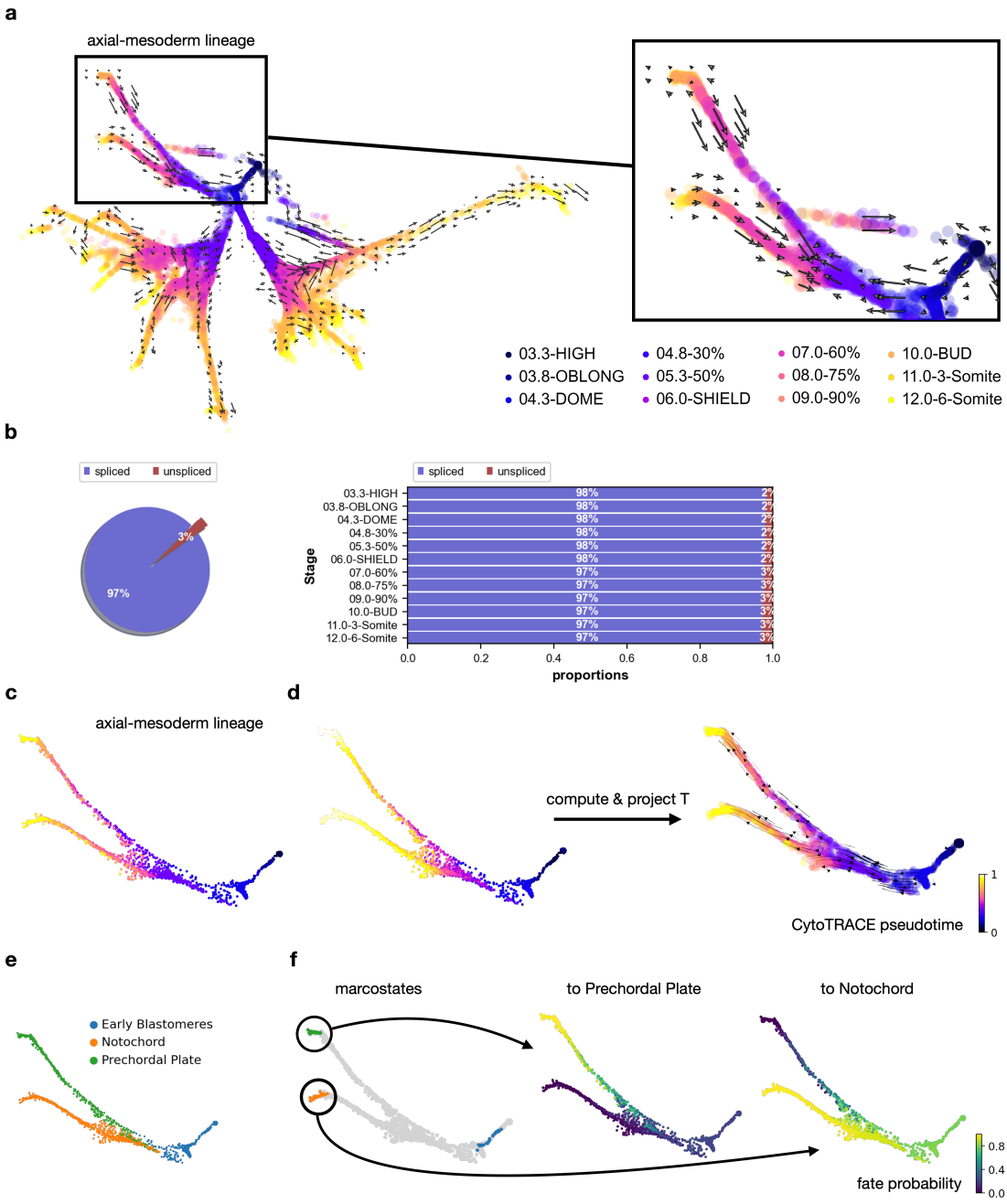

Figure 3.16: The `CytoTRACEKernel` captures early zebrafish development. a. Force-directed graph drawing from the original publication of 33,519 cells of early zebrafish development, colored according to 12 developmental stages [333]. Arrows correspond to projected scVelo-velocities, smoothed on a regular Gaussian grid [16]. Zoom-in highlights the axial-mesoderm lineage where arrows point opposite to ordering given by stages. **b.** Low fraction of unspliced reads could explain the problems with velocity arrows. **c.** Zoom-in to the axial mesoderm lineage of (**a**), colored by stage. **d.** Application of the `CytoTRACEKernel`; embedding colored according to CytoTRACE pseudotime (left) and projected transition matrix in a stream plot (right) **e.** Axial-mesoderm lineage colored by original cluster annotations [333] **f.** Application of the `GPCCAEstimator`; showing for three macrostates the top 30 cells assigned to each (left), fate probabilities towards the Prechordal Plate (middle) and towards Notochord (right).

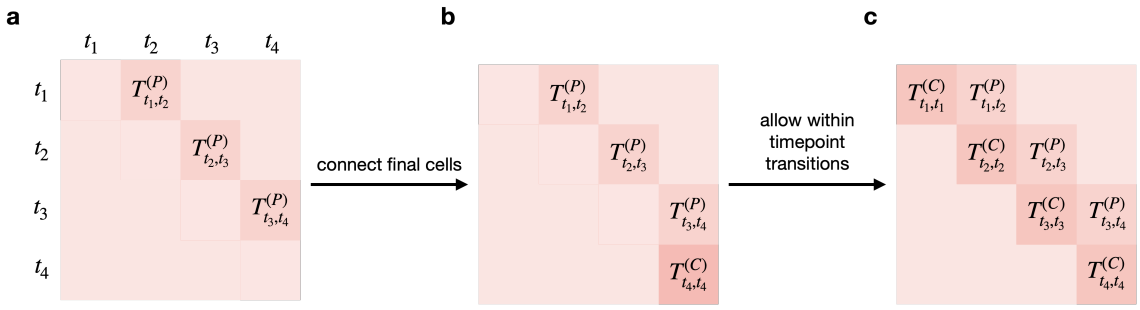

Figure 3.17: The `RealtimeKernel` combines within- with across-timepoint transitions **a.** Schematic heatmap of a joint transition matrix T including four timepoints. Sequential transitions between subsequent timepoints are given by application of WOT [20], resulting in pairwise transition matrices $T^{(P)}_{t_i,t_{i+1}}$ on the super-diagonal. **b.** Application of the `ConnectivityKernel` to the final timepoint t_4 yields transition matrix $T^{(C)}_{t_4,t_4}$; including this matrix on the diagonal allows transitions between final cells. **c.** Similarly, application of the `ConnectivityKernel` to earlier timepoints yields transition matrices $T^{(C)}_{t_i,t_i}$ which we include on the diagonal. The final transition matrix combines within time-point with across time-point transitions. Visualizations (**a.**-**c.**) created in collaboration with Philipp Weiler.

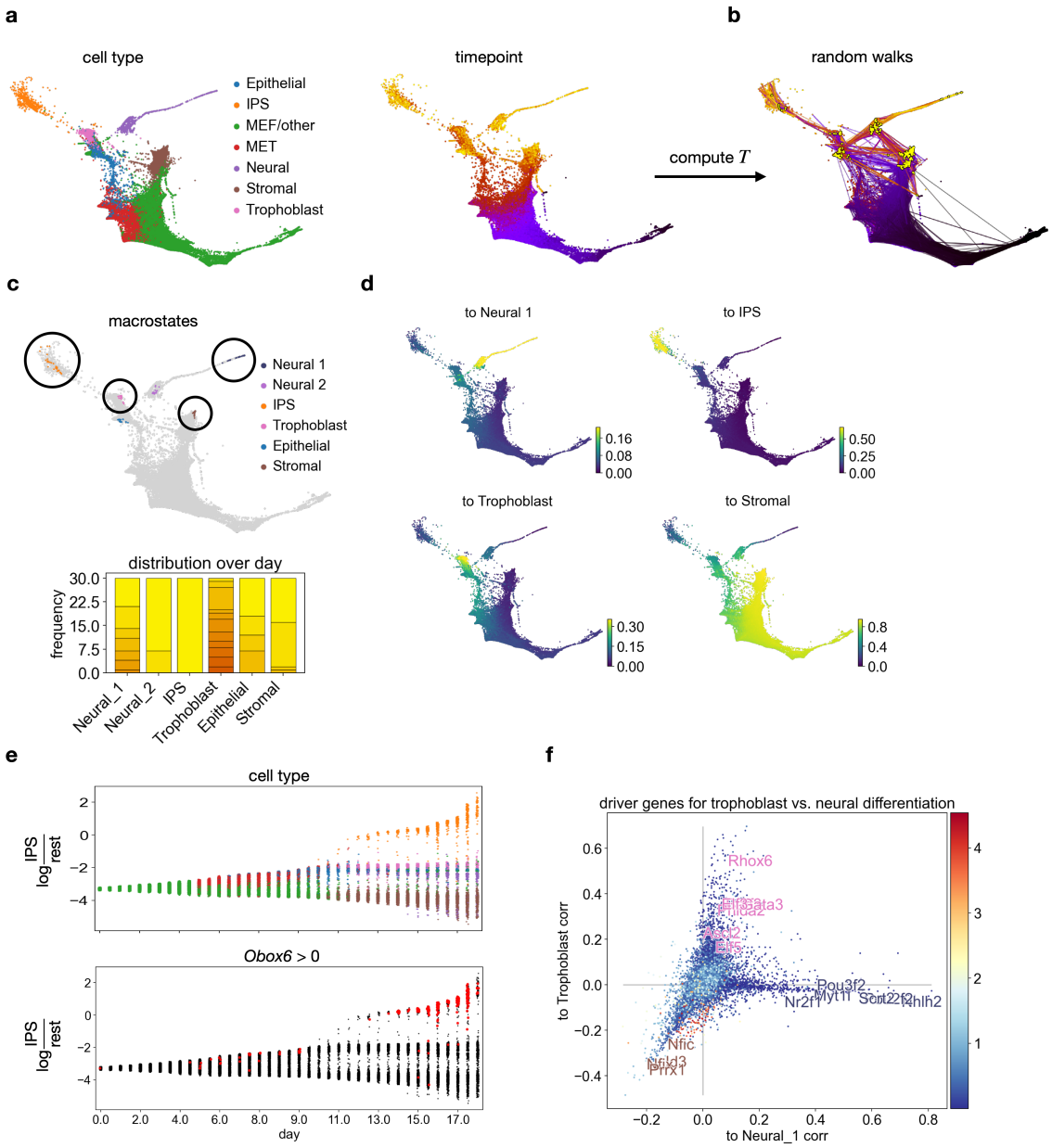

Figure 3.18: Combining the `RealtimeKernel` with the `GPCCAEstimator` a. Original force-directed embedding of 41,473 MEFs reprogramming towards iPSCs across 39 timepoints spanning days 0-18 in a serum condition [20], colored according to coarse cell types (left) and time points on a black (early) to yellow (late) color scale (right). **b.** Black lines indicate random walks, simulated using the `RealtimeKernel` transition matrix T. Black (yellow) dots denote random walk start (end) points. **c.** Showing for each of 6 computed macrostates the 30 most confidently assigned cells (top) as well as macrostate composition over timepoints (bottom). **d.** Fate probabilities towards the 4 macrostates indicated with circles in (**c**). **e.** Log-odds ratio of fate probabilities towards IPS vs. other across timepoint; each dot denotes a cell, colored according to cell type (top) or the binarized expression of *Obox6* (bottom). **f.** Scatter plot of Trophoblast vs. Neural 1 fate probabilities; each dot denotes a gene, colored according to mean expression level. We annotated genes that have been implicated in the reprogramming process in the original publication [20] and colored them by the corresponding cell type [20].

3.6 Summary and discussion

In this chapter, we introduced CellRank, a flexible framework for Markov chain-based modeling of single-cell data. In particular, we introduced the `VelocityKernel` and the `ConnectivityKernel` which may be combined to yield a robust representation of noisy cellular dynamics as captured by RNA velocity [1, 16] (challenge i and contribution i). Further, we showed how the `GPCCAEstimator` builds on such a representation to identify initial and terminal states and to estimate fate probabilities which enable a number of downstream applications (challenge ii and contribution ii). We showed how the combination of `VelocityKernel`, `ConnectivityKernel`, and `GPCCAEstimator` recovers known biology for MEFs reprogramming towards iEPs [22] as well as for pancreatic development [23] and how it predicts new biology for lung regeneration [11] (contribution iii). Besides RNA velocity, the `PseudotimeKernel`, the `CytoTRACEKernel`, and the `RealtimeKernel` enable the application of the CellRank framework to other data modalities which we showcased on adult hematopoiesis [25], zebrafish development [333] and MEFs reprogramming towards iPSCs [20] (challenge iii and contribution iv).

CellRank's key design principles. The implementation of the CellRank framework followed two key design principles: modularity and sparsity. First, modularity is achieved by structuring the framework into kernels and estimators; it allowed us to easily extend the framework towards new estimates of cellular state changes with the kernels of Section 3.5. Further, it makes it easy for others to contribute to the CellRank framework; for example, Zhang et al. [334] interface from their StationaryOT method to CellRank through an *external kernel*, our mechanism for including community contributions. We actively encourage and support such contributions through *contribution guidelines* and *contribution tutorials*. Second, sparsity is baked into all KNN-graph-based kernels (all kernels but the `RealtimeKernel`) and is achieved through thresholding in the remaining kernels (only the `RealtimeKernel`). The `GPCCAEstimator` makes use of sparsity throughout all computations and scales to large datasets, as we have shown in our comprehensive benchmark (Section 3.4.3).

CellRank generalizes trajectory inference. Similarity-based trajectory approaches have been mainly limited to studying biological processes in which the starting cell and direction are clear (Section 2.4). In contrast, CellRank generalizes beyond normal development; we showed how the combination of `ConnectivityKernel` and `VelocityKernel` successfully recovered lineage-derived ground truth during in-vitro MEF reprogramming

towards iEPs [22] and predicted a novel goblet-to-basal cell dedifferentiation trajectory upon lung injury [11]. We experimentally validated the existence of a novel intermediate state between goblet and basal cells; however, the direction of the proposed trajectory still needs to be confirmed with lineage tracing.

Current limitations and outlook. CellRank currently has a few limitations on the kernels side; for example, the `RealtimeKernel` requires the left marginal to be uniform, otherwise, information about cellular growth and death rates is lost. We anticipate overcoming this limitation with an adjusted Markov chain construction. A promising direction to overcome RNA velocity limitations in some biological settings is to make use of metabolic labeling data (Section 2.5); this has recently been pioneered by the dynamo method [267] and can be included in CellRank via a new kernel. Further, multi-modal data, in particular, shared RNA and ATAC readout, can be used to reason about the direction of cellular state changes; this has recently been suggested by Ma et al. [70] and Li et al. [278] and could likewise be included via a new kernel.

In terms of estimators, the current `GPCCAEstimator` assigns fate probabilities to each cell on the basis of its probability to reach a certain macrostate, however, it does not consider the path the cell takes towards this macrostate. Situations in which this is important to describe and understand biology include transcriptional convergence, i.e. several paths leading to the same cell state [7]. To model transcriptional convergence, we envisage constructing a new estimator based on transition path theory [335] (TPT).

Once fate probabilities have been computed, CellRank currently supports a simplistic way to identify putative lineage drivers via correlation with gene expression. However, to identify genes that show a specific activation pattern, e.g. periodic, towards a particular terminal state, we envisage building an interface to methods that support more advanced differential expression tests in pseudotime, in particular TradeSeq [336]. TradeSeq could benefit from CellRank's fate probabilities to define different trajectories leading up to terminal states.

Chapter 4

Mapping lineage-traced cells across time

Many biological processes do not unfold in a dynamical steady-state and thus require time-series experimental designs to capture the entire state change trajectory. Experimental time points provide a good proxy for directionality; on average, cells captured at earlier time points correspond to earlier states on the trajectory. As cells are destroyed upon sequencing, computational methods like WOT [20] (Section 2.4) have been developed to reconstruct couplings from earlier to later cells. In the previous chapter, we introduced CellRank's `RealtimeKernel` which builds on WOT to allow both within time point as well as across time point transitions and we showed how such an approach successfully recovered macrostates and fate probabilities (Section 3.5).

However, there exist limitations to the WOT approach: first, both compute time and memory scale quadratically in cell number, and second, couplings become ambiguous when distributions between gene expression states are too different [20] or when hidden variables dominate the state change trajectory [10] (e.g. epigenetic fate priming not observed in scRNA-seq data). The first limitation is of practical nature; as scRNA-seq datasets are constantly increasing in cell number, it is important for computational methods to keep up. However, especially the quadratic memory complexity means that in practice, modern datasets simply will not fit into memory. The second limitation is more fundamental; many biological processes require additional information on the cell level such that populations can be matched reliably across time points. Such information is currently not taken into account in purely gene expression-based methods like WOT.

In this chapter, we present **m**ulti-**o**mic **s**ingle-**c**ell **o**ptimal transport **t**ools, (moscot), a framework that unlocks optimal transport (OT) for large-scale applications in single-cell genomics. moscot can be applied to mapping problems in both time and space; we focus on the temporal applications in this thesis. In particular, we demonstrate how moscot overcomes the challenges outlined in Section 1.2:

- we address challenge (iv), the scalability of OT-based matching of cells across time points in Section 4.2 where we introduce `moscot-time`, an adaptation of the WOT approach which achieves linear time and memory complexity in cell number.

- we address challenge (v), the need for computational methods which exploit both clonal information as well as gene expression to faithfully match cells in Section 4.3 where we introduce `moslin`. Our approach is based on a Fused Gromov-Wasserstein formulation which combines within time point lineage similarity with across time point gene expression similarity.

We demonstrate and benchmark the proposed `moslin` model in practical applications to simulated data and *C. elegans* embryogenesis [24] in Section 4.4. This chapter corresponds to, and is in part identical, to the following publications:

(i) **Lange, M.***, Piran, Z.*, Klein, M., Theis, F.J. and Nitzan, M., 2021. Mapping lineage-traced single-cells across time-points. *NeurIPS LMRL workshop contribution.*

(ii) **Lange, M.***, Piran, Z.*, Klein, M.*, Spanjaard, B.*, Junker, J.P., Theis, F.J. and Nitzan, M., 2022. Mapping lineage-traced single-cells across time-points. *In preparation.*

(iii) Klein, D.*, Palla, G.*, **Lange, M.***, Klein, M.*, Piran, Z.*, Gander, M., Meng-Papaxanthos, L., Nitzan, M., Cuturi M., Theis F. J., Mapping cells through time and space with moscot. *In preparation.*

Note that "*" denotes an equal contribution.

4.1 The moscot modeling framework

OT has found numerous applications in single-cell genomics, including mapping cells across time points [20, 210, 261, 334, 337, 338], mapping cells from molecular to physical space [220, 339], aligning spatial transcriptomics samples [340], integrating data across molecular modalities [237, 341], learning patient manifolds [223, 342] or mapping cells across different experimental perturbations [343] (Figure 4.1a). These applications use different variants of OT, including classic OT [20, 210, 223, 334, 337, 341] (Subsection 2.3.1), Gromov- and Fused Gromov-Wasserstein [220, 237, 339, 340] (GW and FGW, Subsection 2.3.2), barycenters [340], neural OT [343], surrogate OT [261, 338] and convolutional OT [342], and are implemented using various backends to solve the final OT problem including the python-based optimal transport toolbox [210, 220, 237, 334, 337, 339, 340, 344] (POT), pyKeOps [334, 345], pyTorch [338, 341, 342] as well as custom python implementations

[20, 342] and the R-based transport [223] package. Despite the obvious success of OT-based solutions to problems in single-cell genomics, their community-wide adaptation is currently hindered by a fractured tools landscape with implementations split across various backends, most of which are not compatible with the SCANPY [115] ecosystem and do no scale well to large datsets.

A unified, scalable framework for OT-based analysis To overcome these challenges, we propose the moscot framework. moscot consists of problem-specific estimators which interact with the wider SCANPY ecosystem to set up an OT problem for a biological problem at hand; the OT problem is passed to an external OT backend where a solution is computed and returned to the estimator object where further analysis is enabled through analysis mixins (Figure 4.1b).

This modular design offers unified access to OT-based solutions for single-cell genomics problems and overcomes the challenge of a fractured tools landscape with incompatible APIs. For the OT backend, we interface with the optimal transport tools (OTT) package which is implemented in JAX [230], a python framework that allows for GPU acceleration, just-in-time compilation (jitting), and automatic differentiation. These engineering-type advantages, jointly with theoretical improvements in OTT we describe below, allow moscot to overcome the scalability problems of most OT implementations. While moscot currently implements both spatial as well as temporal estimators, we focus on the temporal domain in this thesis and restrict all further expositions to this setting.

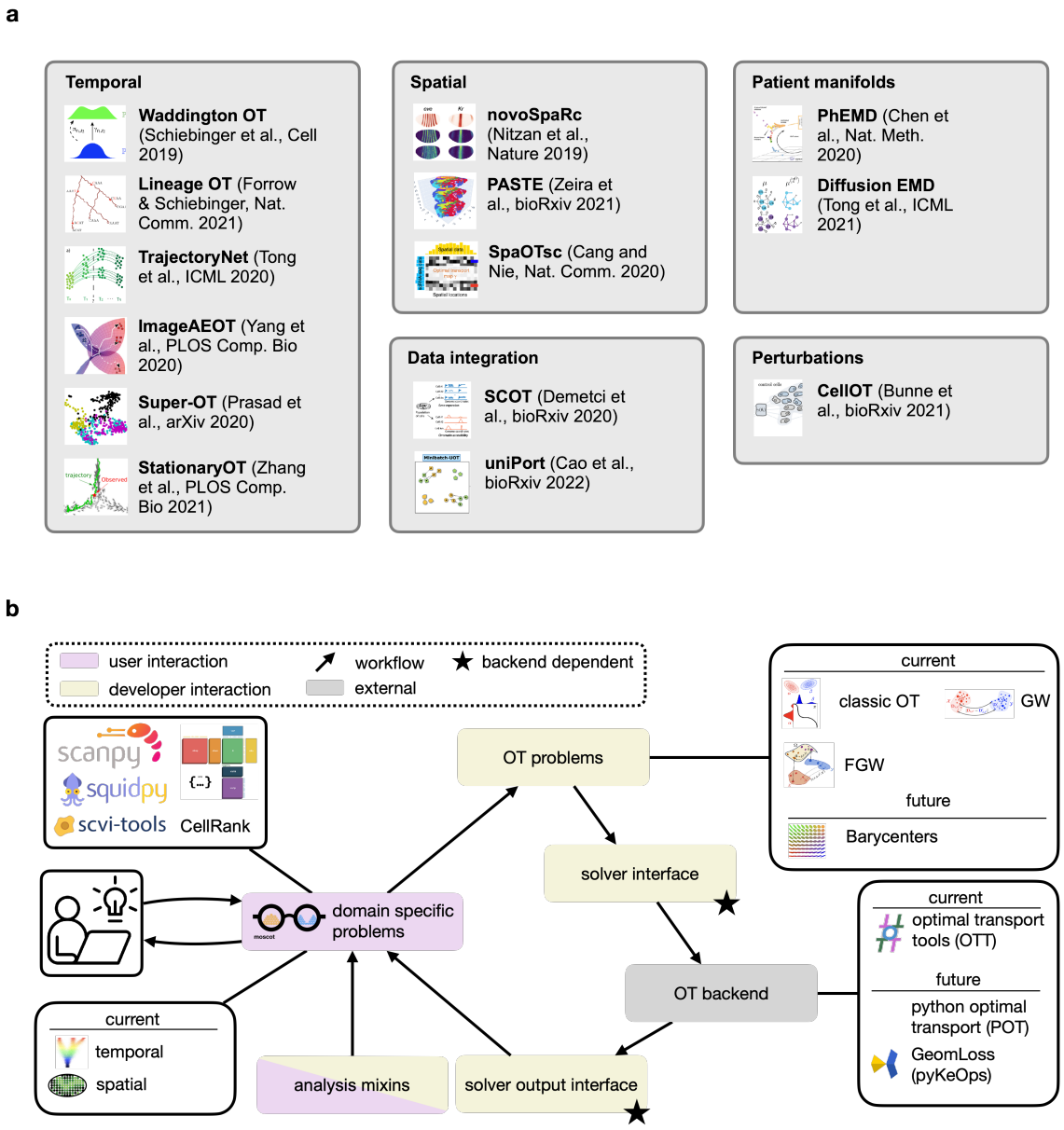

Figure 4.1: OT in single-cell genomics and moscot. a. OT has found numerous applications to problems arising in single-cell genomics. Icons correspond to figures reproduced from the original publications. **b.** Overview of the moscot framework. Users interact with domain-specific problems (currently temporal and spatial), these are translated into OT problems by the `OT problems` class and solved by an `OT backend` (currently Optimal transport tools, OTT). The solution is passed back to the domain-specific problems classes via a `solver output interface`. Downstream analysis is enabled through `analysis mixins`, these can be (partially) shared across domain-specific problems. To set up the domain-specific problems and to offer downstream analysis, we interface with the wider SCANPY [115] ecosystem including squidpy [45], scvi-tools [346], CellRank[14] and AnnData [347]. Visualization in (**b**) created in collaboration with Dominik Klein.

4.2 Scaling up WOT with `moscot-time`

To couple cells across time points, `moscot-time` solves the same unbalanced OT problem as WOT [20] (i.e. Equation (2.84) in Section 2.3). We follow WOT in their definition of the adjusted left marginal distribution $\boldsymbol{a}$ to accommodate cellular growth and death; however, we deviate in the definition of the cost function c between adjacent time points. Where WOT measures distances using an l_2 norm in a local PCA space, computed just for cells in the two time points (Section 2.3), `moscot-time` measures distances using an l_2 norm in a global scVI [96] latent space, using all cells in the time-series experiment. Thus, by using a non-linear latent representation, `moscot-time` is better positioned to capture non-linear dynamics in the data and to account for potential batch effects between replicates within one time point.

The bottleneck in any method like WOT [20] that uses OT to link cells across time points are the matrix-vector products $K\boldsymbol{v}$ and $K^\top \boldsymbol{u}$ in the (generalized) Sinkhorn iterations [221] for scaling vectors $\boldsymbol{u} \in \mathbb{R}^N, \boldsymbol{v} \in \mathbb{R}^M$ (Section 2.3). For simplicity, suppose the number of cells in both time points is the same, i.e. $N = M$. For precomputed Gibbs kernel $K = \exp(-C/\epsilon)$ with cost matrix $C \in \mathbb{R}_+^{N \times N}$, this results in both memory and compute time time scaling quadratically in N. In practice, compute time may be reduced by running computations on GPUs; however, memory becomes the bottleneck as GPUs typically have much less memory compared to CPUs. Moreover, WOT is based on a custom OT implementation that can only be run on CPUs. In moscot, we tackle the scalability issue from two complementary angles:

- we use engineering-type improvements to run computations on GPU with linear memory complexity.

- we exploit theoretical advances to restrict the rank of coupling matrices [262, 263, 348]; this results in linear time and memory complexity.

These improvements have been implemented in OTT separately from our work on moscot; we interface with OTT in the backend (Figure 4.1b). While the first set of improvements leads to the exact same solution of the convex OT problem, the second set of improvements is an approximation that will impact the quality of the obtained solution; we quantify this tradeoff in numerical experiments.

4.2.1 Engineering-type improvements for large-scale GPU application

In principle, all moscot models can be run on GPU as OTT is implemented in JAX [230] which offers GPU acceleration throughout. In practice, however, the quadratic memory complexity of storing the kernel K leads to out-of-memory problems even on modern GPUs for moderately-sized datasets (see below). OTT employs a trick to circumvent this issue; rather than storing the entire kernel K, matrix-vector products with K are computed element-wise by evaluating the cost function c on the fly. Consider computing $[K\boldsymbol{v}]_i = K_{i,:}\boldsymbol{v} = \exp(-c(\boldsymbol{x_i}, Y)/\epsilon)\,\boldsymbol{v}$; this only requires evaluating the cost to transport mass from $\boldsymbol{x}_i$ to any sample $\boldsymbol{y}_j$ which has linear memory complexity in cell number (Figure 4.2a). OTT calls this mode *online evaluation*; we call the alternative of pre-computing the entire C and K matrices *offline evaluation* to clearly separate between the two.

4.2.2 Low-rank factorizations yield linear time and memory complexity

While the engineering improvements introduced above allow the application to large datasets through GPU acceleration with linear memory complexity, they still suffer from quadratic time complexity. To overcome this limitation, various authors have suggested approximations to the Sinkhorn iterations that yield linear time complexity. Altschuler et al. [349] suggest computing a low-rank approximation to the kernel matrix K using the Nystrom method [350]; their approach remains limited to squared euclidean cost functions c, is non-differentiable and only works for large regularization strength ϵ where inner iterations remain positive. Scetbon and Cuturi [351] suggest an alternative way of computing low-rank approximations to K via random positive feature projections; while their approach is differentiable and works for a larger range of ϵ values, it remains limited kernels of a certain form.

Forrow et al. [348] suggest a different route that imposes low-rank constraints on the feasible set of couplings $U(\boldsymbol{a}, \boldsymbol{b})$ rather than on the kernel matrix K. Their approach leads to an elegant solution via a barycenter problem; however, it remains limited to squared euclidean cost functions c. Scetbon, Cuturi, and Peyré [263] generalize this approach to arbitrary cost functions c; their proposed solution is differentiable and applicable for a wide range of ϵ values, including no entropic regularization ($\epsilon = 0$). This approach is implemented in OTT and available through moscot; we refer to it as *low-rank Sinkhorn*. It has meanwhile been extended from the classic OT to a (F)GW setting [262] which is also implemented in OTT and available to FGW-based moscot models like `moslin`.

For the low-rank Sinkhorn approach, following Scetbon, Cuturi, and Peyré [263] define the

nonnegative rank of a coupling matrix $P \in R_+^{N\times M}$ to be

$$\mathrm{rk}_+(P) \coloneqq \min\left\{ q \,\middle|\, P = \sum_{i=1}^{q} R_i \,, \mathrm{rk}(R_i) = 1,\, R_i \geq 0 \right\}, \tag{4.1}$$

for rank rk. For $r \geq 1$, we make use of this to define the set of rank-r couplings via

$$U(\boldsymbol{a}, \boldsymbol{b}, r) \coloneqq \{P \in U(\boldsymbol{a}, \boldsymbol{b}) \,|\, \mathrm{rk}_+(P) \leq r\}\,, \tag{4.2}$$

where $U(\boldsymbol{a}, \boldsymbol{b})$ is the set of feasible couplings defined in Section 2.3. The rank-constrained feasible set $U(\boldsymbol{a}, \boldsymbol{b}, r)$ allows us to formulate the low-rank OT problem via

$$\mathcal{L}_c^{\epsilon,r}(\alpha, \beta) \coloneqq \min_{P\in U(\boldsymbol{a},\boldsymbol{b},r)} \langle P, C\rangle - \epsilon H(P)\,. \tag{4.3}$$

An explicit characterisation of couplings P in $U(\boldsymbol{a}, \boldsymbol{b}, r)$ is given by

$$P = Q\,\mathrm{diag}\,(1/\boldsymbol{g})\,R^\top \text{ for } \boldsymbol{g} \in \Delta_r^*,\; Q \in U(\boldsymbol{a}, \boldsymbol{g}),\; R \in U(\boldsymbol{b}, \boldsymbol{g})\,, \tag{4.4}$$

where Δ_r^* denotes the r-simplex with strictly positive elements. Using this factorization, Scetbon, Cuturi, and Peyré [263] derive a mirror descent optimization scheme for the low-rank OT problem of Equation (4.3); the time- and memory bottleneck in this algorithm is given by matrix-matrix multiplications of the form CR and $C^\top Q$ for $Q \in \mathbb{R}^{N\times r}$ and $R \in \mathbb{R}^{M\times r}$. Thus, without any assumptions on the cost matrix C, the low rank approach remains at memory complexity $\mathcal{O}(MN)$ and time complexity $\mathcal{O}(NMr)$.

To improve upon this complexity, assume that C itself admits a low-rank factorization of the form

$$C = AB^\top \text{ for } A \in \mathbb{R}^{N\times D},\; B \in \mathbb{R}^{M\times D}\,, \tag{4.5}$$

such that matrix-matrix multiplications $CR = A(B^\top R)$ and $C^\top Q = B(A^\top Q)$ can be evaluated in memory $\mathcal{O}\left((D + r)(M + N) + Dr\right)$ and time $\mathcal{O}\left(rD(N + M)\right)$, i.e. both linear in the total cell number $N + M$. In particular, such a factorization can be obtained if the cost results from the application of a squared euclidean cost function, i.e. $C = c(X, Y) = ||X - Y||_2^2$. In such a case, C may be written as

$$C = \boldsymbol{p}\mathbf{1}_M^\top + \mathbf{1}_N\boldsymbol{q}^\top - 2X^\top Y\,, \tag{4.6}$$

for $\boldsymbol{p} \coloneqq [||\boldsymbol{x}_1||_2^2, ..., ||\boldsymbol{x}_N||_2^2]$ and $\boldsymbol{q} \coloneqq [||\boldsymbol{y}_1||_2^2, ..., ||\boldsymbol{y}_M||_2^2]$. The desired factorization is obtained by defining $A \coloneqq [\boldsymbol{p}, \mathbf{1}_N, -2X^\top] \in \mathbb{R}^{N\times(N_l+2)}$, $B \coloneqq [\mathbf{1}_M, \boldsymbol{q}, Y^\top] \in \mathbb{R}^{M\times(N_l+2)}$ for

cells $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ embedded in some latent space of dimension N_l. In general, low-rank factorizations of cost matrices C can be computed in linear time using randomized algorithms as long as the cost function c is given by a proper distance metric [263, 352, 353].

Different flavors of `moscot-time` in practice. We compared peak memory consumption (on CPU) and compute time (on GPU) for different flavors of `moscot-time` (offline versus online as well as full rank vs. rank constrained) with WOT [20] on data simulated using TedSim [354], a simulation tool for temporal single-cell data (Figure 4.2b). Memory was benchmarked on CPU for simplicity; GPU memory is expected to behave very similarly. WOT was run on CPU throughout as it does not support GPU acceleration. Our dataset contained 164k cells in total; we subsampled to various fractions to study scalability with respect to increasing cell numbers. All computations were run by Dominik Klein.

For peak CPU memory consumption on the full dataset, WOT and moscot (offline) required 229 and 129 GiB of memory, respectively (Figure 4.2b). While moscot (offline) performed much better, 129 GiB still exceeds available memory on almost all modern GPUs. As expected, the online mode overcame this challenge, requiring less than 2 GiB on the full dataset. Low-rank approaches performed similarity, requiring between 6 GiB ($r = 1000$) and 1 GiB ($r = 50$). Thus, both the online mode of moscot as well as low-rank approximations require an order of magnitude less memory compared to the original WOT [20] approach enabling the application to datasets containing millions of cells.

For GPU compute time on the full dataset, WOT took 52 minutes while moscot (offline) failed due to memory limitations, as outlined above (Figure 4.2b). Note that WOT was run on CPU as the implementation does not support GPU acceleration. moscot's online mode resolved the memory problem encountered in offline mode and finished in under 1 minute. Moreover, all low-rank approaches finished in under 1 minute.

As low-rank approaches solve an approximation of the original OT problem, we were interested in comparing the accuracy of the obtained coupling across different ranks r (Equation (4.3)). We employed a benchmarking task termed *geodesic interpolation* in the original WOT publication [20]: given three time points t_i, t_{i+1} and t_{i+2} in a scRNA-seq time series dataset, hold out the middle one t_{i+1}, compute a coupling between t_i and t_{i+2} and use this coupling to interpolate the cell-state distribution at the middle time point t_{i+1}. The distance between the interpolated and the real, held-out distribution at t_{i+1} is measured in terms of Wasserstein-1 (W_1) distance in gene expression space (Section 2.3), the lower this distance the better the computed coupling. Following the WOT approach, we

considered two baselines: the W_1 distance between two experimental replicates sequenced at t_{i+1} ("Baseline", represents approximate best-case distance) as well as the distance obtained by considering an uninformative (random) coupling between t_i and t_{i+2} ("Null 1", represents an approximate worst-case distance).

The outer time-points t_i and t_{i+2} we considered contained 3,678 and 3,799 cells, respectively. As expected, accuracy was at the level of random guessing for extremely low ranks, $r = 1$, and approached full-rank performance for higher ranks, $r \approx 100$ (Figure 4.2c). The method was robust to the exact rank r used as long as it was high enough; for $r > 10$, variations in accuracy became very small. The highest accuracy, corresponding to the lowest W_1 distance between distributions, was reached for $r = 100$, the second-highest rank considered. In that case, the distance obtained for low-rank was slightly lower than the distance obtained for the full-rank approach. This highlights the fact that low-rank regularization can lead to better performance in practice due to better statistical properties, i.e. it is less prone to overfitting [263, 348].

The scalability improvements we introduced in this section are important to take advantage of the increased resolution offered by ever-increasing cell numbers in current single-cell experiments; in particular, they enable the `moslin` model of the following section to be applicable to large datasets.

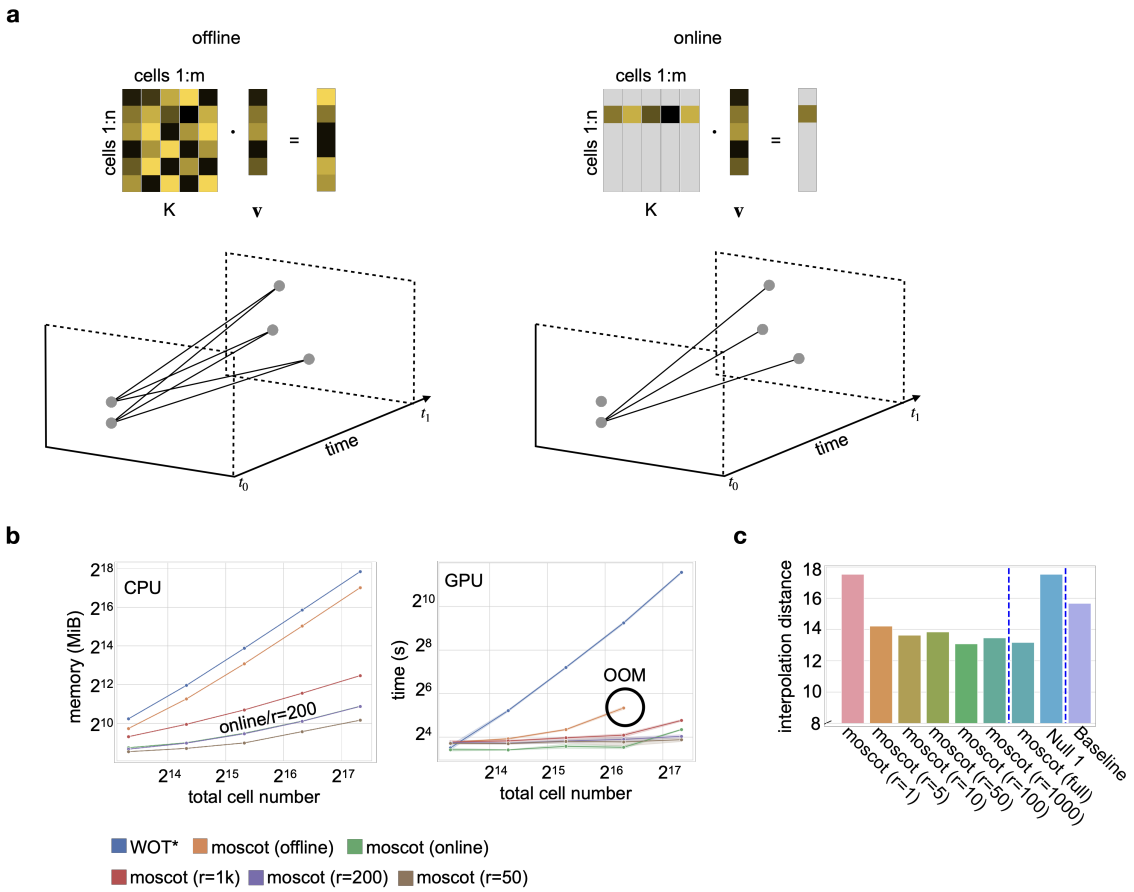

Figure 4.2: `moscot-time` **scales up the temporal mapping problem. a.** Comparison of offline (left) versus online (right) cost function evaluation. On the top row, heatmaps visualize the matrix-vectors products $K\boldsymbol{v}$ employed in Sinkhorn iterations, evaluated batch-wise (left) or row-wise (right). On the bottom row, consider two time-points t_0 and t_1 with just 2 and 3 cells, respectively. While in offline mode, we pre-compute the cost of transporting mass from any sample at t_0 to any sample at t_1 (left), these cost-evaluations are iterated over for samples at t_0 in online mode (right). **b.** Comparison of peak memory consumption on CPU (left) and compute time on GPU (right) across WOT and different flavors of moscot on TedSim [354] simulated data for a total of 164k cells. Methods denoted by * cannot be run on GPU by design; OOM denotes an out of memory error. **c.** Accuracy comparison for low-rank approaches on MEF reprogramming data in terms of geodesic interpolation [20]. Bar height denotes the W_1 distance between interpolated and held-out distributions. "Null 1" denotes a random coupling that satisfies the marginal constraints, and "Baseline" refers to the distance between two experimental replicates at the same time point. Comparisons in (**b**) and (**c**) were run by Dominik Klein.

4.3 moslin for scLT data

Coupling cells across time points purely on the basis of gene expression similarity is challenging when state distributions are very different or when hidden variables like epigenetic fate priming are present [10, 20]; scLT provides additional information which can guide the coupling process [17]. For destructive in-vivo experimental designs, each time point corresponds to a different replicate/individual. High-resolution clonal relationships are best captured with prospective dynamic barcoding approaches [176–178] or with retrospective mitochondrial lineage-tracing approaches [74, 201, 204, 206] (Figure 4.3a and Subsection 2.1.3), giving rise to the independent clonal evolution setting: while gene expression can be compared across time points, lineage relationships are only valid within one time point.

With `moslin` we combine both sources of information into a joint model; our inputs are given by gene expression profiles, captured lineage barcodes and marginal distributions over cell states at two time points t_1 and t_2. The model outputs a coupling matrix P relating cells at the two time points which may be used for further analysis in CellRank through the `RealtimeKernel` (Section 3.5 and Figure 4.3b,c). To compute P, we formulate a FGW problem; let $C^{(1)} \in \mathbb{R}_+^{N\times N}$ and $C^{(2)} \in \mathbb{R}_+^{M\times M}$ capture lineage distance at t_1 and t_2 for N and M cells, respectively. We compute $C^{(i)}$ by reconstructing a lineage tree at t_i on the basis of sequenced barcodes using Cassiopeia [207]; lineage distance is captured by the shortest path distance among cells in the tree. Further, let $C \in \mathbb{R}^{N\times M}$ define a cost matrix computed on the basis of gene expression distance between cells in t_1 and t_2; by default, we use l_2 distance in an scVI [96] latent space. With these definitions, the objective function reads

$$\mathcal{L}^{\epsilon,\alpha}_{L,c^{\mathcal{X}},c^{\mathcal{Y}},c}(\alpha,\beta) \coloneqq \min_{P\in U(\boldsymbol{a},\boldsymbol{b})} \alpha \sum_{ijkl} L\left(C^{(1)}_{ij}, C^{(2)}_{kl}\right) P_{ik}P_{jl} + (1-\alpha)\sum_{ik} C_{ik}P_{ik} - \epsilon H(P)\,, \tag{4.7}$$

where α is a tunable parameter that controls the weight given to lineage similarity within a time point versus gene expression similarity across time points and ϵ represents the degree of entropic regularization applied (Figure 4.3d). For optimization, we employ the mirror descent scheme of Section 2.3 which we access though OTT. The `moslin` model is implemented in moscot and is accessible through our unified API. Further, it takes advantage of the scalabilty improvements outlined above (Section 4.2).

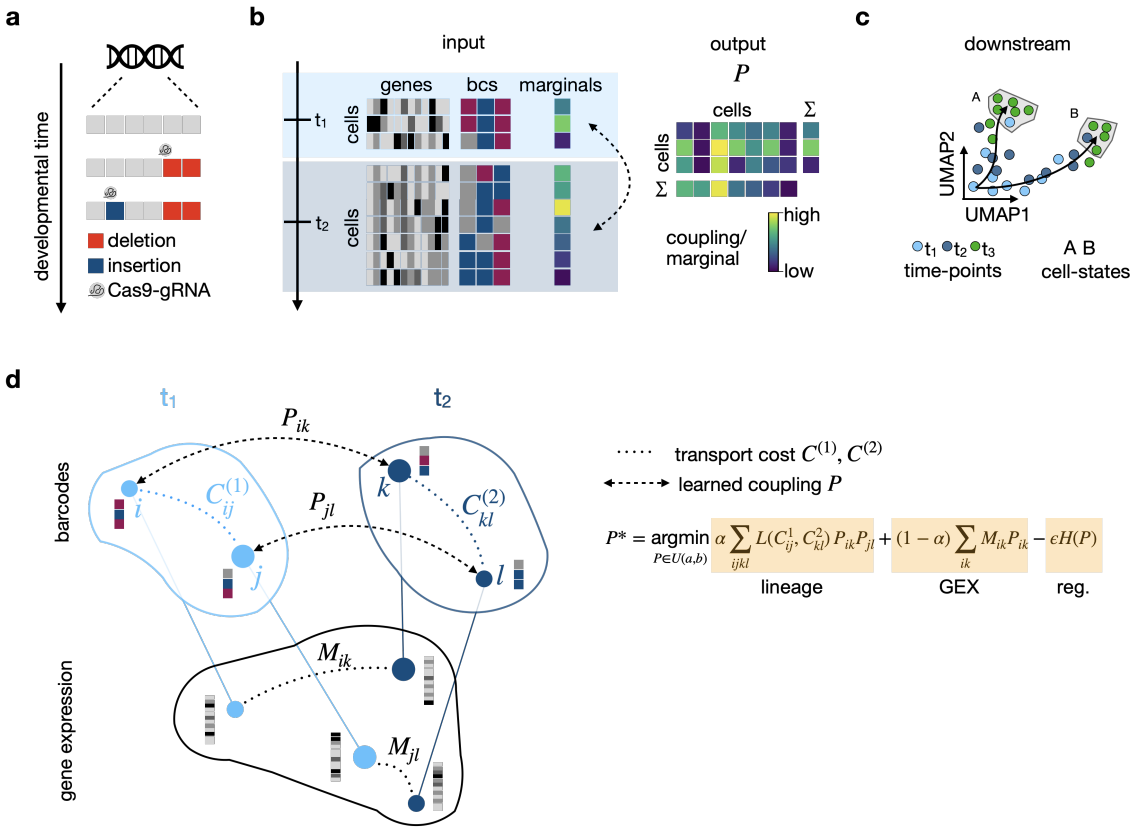

Figure 4.3: `moslin` **maps lineage-traced single-cells across time points. a.** Prospective dynamic barcoding approaches based on CRISPR/Cas9 use random insertions and deletions to record complex lineage relationships. **b.** Schematic of `moslin` inputs (left) and outputs (right). Inputs consist of gene expression matrices, barcode arrays (bcs) and marginal distribution at t_1 and t_2. The output is a coupling matrix P satisfying the marginal constraints, shown here as a heatmap. **c.** The coupling matrix P can be used for downstream analysis e.g. in CellRank to follow cell-state trajectories across time points. **d.** Visualization of the `moslin` objective function (left) as well the objective function itself (right). On the left, dots denote cells, colored according to time point as in (**b**). Dot size corresponds to marginal distribution weight at t_1 and t_2, respectively. Cells are observed in barcode space (top) and in gene expression space (bottom). While gene expression similarity, captured in M, can be compared directly across time points, lineage similarity, captured by $C^{(1)}$ and $C^{(2)}$ at t_1 and t_2, respectively, can only be compared pairwise. The objective function on the right includes terms for lineage similarity within a time point, gene expression (GEX) similarity across time points as well as entropic regularisation. Figure adapted from Lange et al. [15].

4.4 Benchmarks and applications of the `moslin` model

Our `moslin` model makes use of both gene expression as well as lineage information when mapping cells across time points; we compared it to models that just use gene expres-

sion (OT objective function) or lineage (GW objective function) information on simulated data (Subsection 4.4.1). In addition, we compared `moslin` with LineageOT [210], the only competitor for this data type (Subsection 2.1.3), on both simulated and real data (Subsection 4.4.2).

4.4.1 Benchmarks on simulated data

We turned to a simple simulation setup introduced in the original LineageOT publication containing four different topologies of time series datasets containing just two time points each [210] (Figure 4.4a). The ground truth coupling between early and late cells in known for this data; we compared it to the couplings inferred by either method via the mean between ancestor and descendant errors ("mean error"). The ancestor (descendant) error is computed by using the inferred coupling to compute a W_1 distance between inferred and ground-truth ancestors (descendants) for the observed late (early) cells.

GW and `moslin` perform well for ground-truth lineage information. In the first experiment, methods were supplied with lineage distances computed for the ground truth tree obtained from the simulation (Figure 4.4b). In this setup, OT had the highest mean error, followed by LineageOT. Both `moslin` as well as GW consistently outperformed LineageOT; GW for partial convergent as well as mismatched cluster topologies even outperformed `moslin`.

`moslin` outperforms competing methods for noisy lineage information. In reality, we do not have access to the ground truth tree; therefore, we investigated in a second experiment how methods performed when supplied with lineage distances computed along a lineage tree inferred on the basis of simulated lineage barcodes (Figure 4.4b). As expected, this had no effect on the performance of OT as it does not make use of lineage information. GW in this setup performed much worse and had the largest mean error in all but the mismatched cluster topology. This is due to the fact that GW does not have access to gene expression information and must compute a coupling purely based on noisy lineage distances. In contrast, both LineageOT and `moslin` were much less affected by the noisy lineage information since they use gene expression for regularization. `moslin` consistently had the smallest mean error, outperforming LineageOT on all topologies.

TedSim for realistic scLT data. While our first experiments were conducted on simple simulated data containing only two genes, we next turned to a more realistic simulation

setup provided by TedSim [354], a simulation tool dedicated to scLT data. The three key parameters in TedSim are given by

- the *asymmetric division probability* p_a: the probability that a mother cell gives rise to two different daughter cells, one of which resembles itself while the other one is more advanced along the differentiation trajectory.
- the *step size*: the distance the "advancing" daughter cell in an asymmetric division event travels along the differentiation trajectory, i.e. a measure of differentiation pace.
- the *stochastic silencing rate*: the amount of noise injected into simulated lineage barcodes.

In our experiments, we fixed the asymmetric division rate $p_a = 0.4$ and varied the remaining two parameters. As TedSim's simulations involve a ground-truth lineage tree, we used the mean error introduced above to measure discrepancies between inferred couplings and the ground-truth coupling given by the simulated lineage tree.

`moslin` outperforms LinegeOT on TedSim simulated data. In the first experiment, we kept the stochastic silencing rate fixed at 0.05 and varied the step size between 0.2 and 1.0. Across this range, `moslin` consistently outperformed LineageOT in terms of lower mean error (Figure 4.4c). Further, `moslin` was less affected by the step size compared to LinegeOT. In a second experiment, we kept the step size fixed at 0.6 and varied the stochastic silencing rate between 0 (no noise in barcodes) and 0.2 (noisy barcodes). Across the range, `moslin` achieved lower mean error compared to LineageOT (Figure 4.4c). As expected, the mean error of both methods increased as we injected more noise into simulated lineage barcodes.

4.4.2 Application to *C. elegans* embryogenesis

Going beyond simulation studies, we applied `moslin` to a scRNA-seq time course of *C. elegans* embryonic development [24]. Following the approach suggested in Forrow and Schiebinger [210], the known lineage tree [179] of *C. elegans* provided lineage distances required for the GW term in `moslin` and was used to compute the mean error as in the simulation studies above. In Packer et al. [24], the authors mapped gene expression profiles of individual cells to the known lineage tree. Thus, while the tree is known, there is uncertainty in the mapping and the relationship between lineage nodes and expression profiles is noisy.

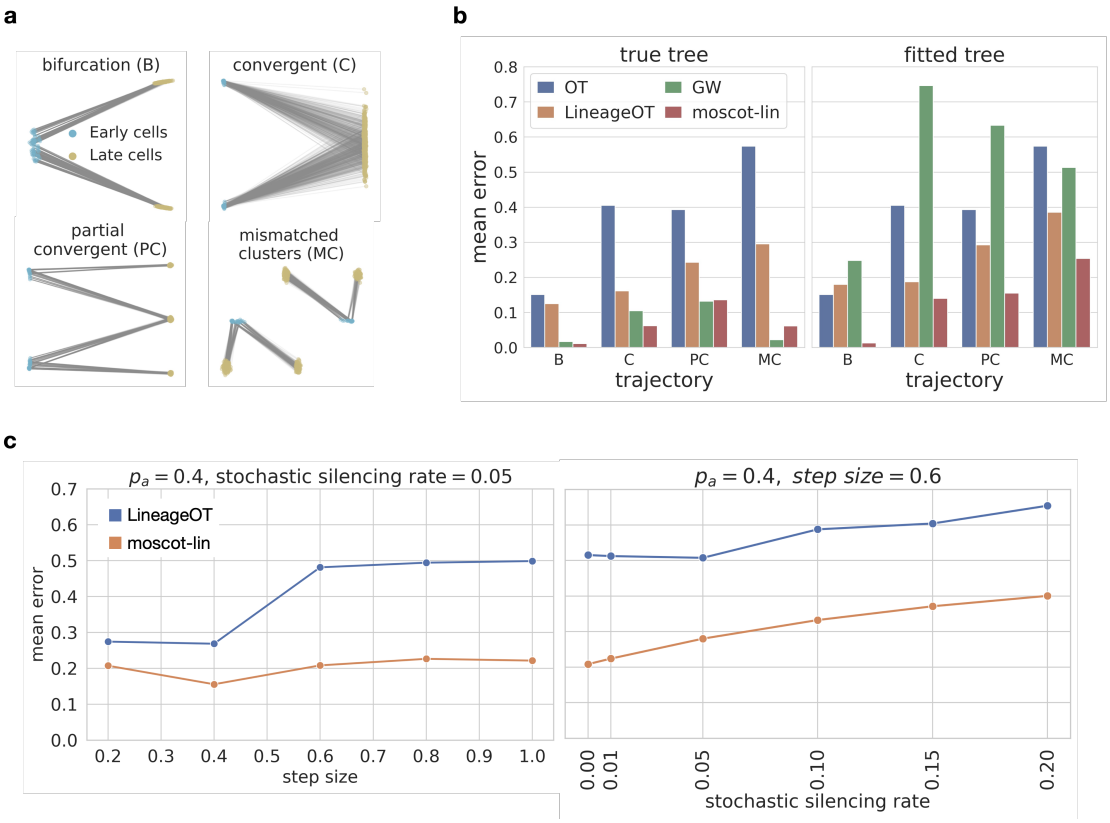

Figure 4.4: `moslin` outperforms LineageOT on simulated data. **a.** Four simulated time series datasets [210]; each dot denotes a cell in 2D gene expression space (only 2 genes simulated), colored according to early or late time point. Grey lines indicate ground-truth coupling. **b.** Comparison of OT, GW, LineageOT [210] and `moslin` on the four simulated topologies of (**a**) for true tree (left) and fitted tree (right) lineage distances; bar height indicates mean error between inferred and ground-truth couplings. **c.** Comparison of LineageOT and `moslin` on TedSim [354] simulated data for varying step size (left) and varying stochastic silencing rate (right). Computations run by Zoe Piran and Michal Klein; figure adapted from Lange et al. [15].

`moslin` outperforms competing approaches on all time points. Initially, we focused on the couplings between pairs of individual time points. We found that `moslin` outperformed LineageOT and OT for mapping cells across all developmental time points (Figure 4.5a). GW showed great diversity in performance, sometimes achieving the smallest mean error (390/450 min) while other times achieving by far the largest mean error (270/330 min). This highlights the sensitivity of GW to lineage distances as it does not consider gene expression in addition. `moslin` compensates for errors in lineage distances through gene expression similarity and derives accurate couplings for all time points.

We next zoomed in on the 210/270 min pair of time points where we identified a group of

cells for which `moslin`'s ancestor error was much lower than LineageOT's (Figure 4.5b). Specifically, while `moslin` accurately mapped cells from the "ABarapa" to the "ABarapapap" lineage, LineageOT incorrectly identified cells from the "ABaraaaaa", "ABaraaaap", "MSaaaap" and "MSaaapp" lineages as ancestors [24].

`moslin` combined with CellRank identifies putative driver genes. Taking a more global perspective, we used `moslin` to compute couplings for all pairs for time points and chained these together by matrix multiplication. This gave a global coupling across time points we used as input to CellRank's `RealtimeKernel`. On the basis of the global coupling, we used the `GPCCAEstimator` to compute fate probabilities towards a cluster of GLR cells (Figure 4.5c). Among the 15 highest correlated genes with GLR fate probability, we found *let-381* and *ceh-34*, both of which have been implicated in GLR development [24] (Figure 4.5d). This highlights the ability of `moslin` to identify putative decision driver genes for real scRNA-seq time course data.

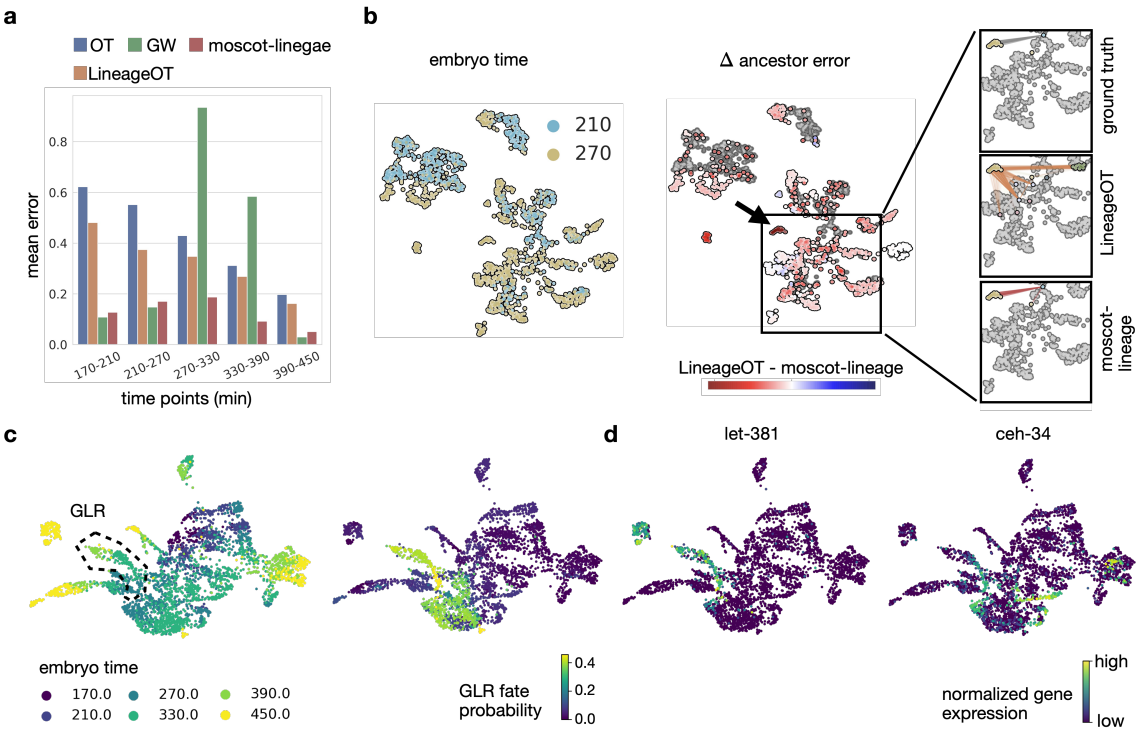

Figure 4.5: `moslin` **outperforms LineageOT in mapping *C. elegans* data. a.** Mean error shown as a bar chart for different methods across time points coupled. **b.** UMAP [121, 122] visualization of cells with embryo time $210 - 270$ min colored by embryo time (left) and the difference between LineageOT's and `moslin`'s ancestor error (right). Red (blue) values imply `moslin`(LinegeOT) performs better. The black arrow highlights a group of cells for which `moslin`'s mean error is much smaller. Inlets show ancestors of the marked population as predicted by the ground truth coupling (top), LineageOT (middle), and `moslin`(bottom). While `moslin` successfully recovered the correct ancestor, LineageOT predicts four incorrect ancestor populations. **c.** UMAP visualization colored by embryo time (left, dashed line marks GLR cells) and CellRank predicted fate probabilities towards GLR cells based on `moslin`'s coupling (right). **d.** UMAP visualizations colored by gene expression of putative decision driver genes towards GLR cells as identified by CellRank. Computations in (**a**) and (**b**) run by Zoe Piran and Michal Klein, collab for (**c**); figure adapted from Lange et al. [15].

4.5 Summary and discussion

In this chapter, we introduced moscot, a flexible framework for Optimal transport (OT) based modeling of single-cell data. While moscot is applicable to both spatial and temporal domains, we focus on the temporal applications in this thesis. In particular, we introduced `moscot-time`, an extension to the original WOT method [20] to link cells across time points which overcomes previous scalability limitations through linear time and memory

complexity in cell number (challenge iv and contribution i). Further, we showed how `moslin` increases mapping accuracy by combining within time point lineage similarity with across time point gene expression similarity for in vivo scLT experimental designs (challenge v and contribution ii). We showed how `moslin` outperforms simpler OT-based models as well as the competing LinegeOT [210] method in two simulation scenarios and how it can be combined with CellRank to recover putative driver genes in *C. elegans* embryogenesis [24] time course data (contribution iii).

4.5.1 moscot for OT in single-cell genomics.

moscot is a general framework to enable OT-based solutions to common problems in single-cell genomics. Its implementation follows a modular layout which offers two key advantages over competing OT solutions:

- improvements in the OT backend, e.g. faster approximate OT solvers, are immediately available to a range of biological problems defined in the frontend. These improvements are not limited to contributions from the single-cell community as the current OT backend is given by OTT [229] which is gaining popularity in the general ML community. Further, our backend-agnostic problem definitions in the frontend allow for further OT packages to be made available in the backend such as POT [344] or GeomLoss [355].

- extensions or additions to the problems frontend are possible with minimal effort; data loading/writing from AnnData objects [347], the solution of the OT problem, and many downstream analysis tasks are taken care of by moscot. The remaining task is the key modeling question: how can the biological problem at hand be translated into an OT problem? Thus, developers of OT-based solutions can focus on the main modeling task if they choose to include their contribution within the moscot framework.

We make sure that the resulting OT problem is solved fast and reliably with OTT in the backend, through GPU acceleration, just-in-time compilation, and low-rank factorizations.

Current limitations and outlook The entire moscot framework is currently based on the default notion of OT which is tied to the samples that are supplied in the data; the computed coupling is specific to these samples and cannot be generalized to new, incoming samples. In other words, OT does not generalize beyond the training data distribution.

While this is sufficient to address many open questions in both temporal and spatial domains, there exist applications such as perturbation mapping where one is interested in evaluating sample/perturbation combinations not observed in the training data.

OT can be extended towards generative mappings with Neural OT [356, 357]; the idea is to define a Monge map between two spaces based on the dual formulation of OT, i.e. using two potentials f and g that are learned by an input-convex neural network [358] (ICNN). The theoretical basis for this approach is given by Bernier's theorem which relates primal and dual OT formulations [359]. Neural OT has been applied to perturbation mapping for single-cell data with CellOT [343] and could be included as an alternative backend in moscot.

Beyond extensions and additions to our OT backend, we actively encourage additions to moscot's frontend for biological problems to provide a unified API to access OT solutions that have been suggested for data integration [237, 341], patient manifolds learning [223, 342] or perturbation mapping [343].

4.5.2 `moscot-time` for large scale time-series scRNA-seq datasets.

We showed how `moscot-time` outperforms the competing WOT model [20] when mapping cells across time-points in terms of both compute time and memory required to solve the OT problem; in particular, `moscot-time` required an order of magnitude less memory. In future work, we aim to demonstrate these advantages on a practical scRNA-seq time-series example containing millions of cells where previous approaches are no longer applicable. Further, an aspect we did not explore in this thesis is `moscot-time`'s flexible definition of the cost function which allows easy adaptation to other data modalities such as scATAC-seq or multi-modal SHARE-seq [70], CITE-seq [155] or DOGMA-seq [74] data. To adapt the cost function, we need a notion of cell-cell distance which is appropriate for the given data modality; this can be achieved using modality-specific representation learning techniques including TotalVI [145], PeakVI [153], MultiVI [156] or multigrate [157] (Section 2.1.2).

Current limitations and outlook. The main limitation in the current approach is that time points are linked pairwise; when linking t_i with t_{i+1}, we ignore all other time points. When we chain together these pairwise couplings to obtain a global coupling, this introduces discontinuities at each time point - an unrealistic model for real cellular dynamics. To overcome this limitation, Lavenant et al. [360] devised the *global* WOT (gWOT) model which formulates a joint optimization problem involving all time points and infers smooth trajectories. Future work could involve adapting the `moscot-time` model

to consider a joint optimization problem in a similar manner.

4.5.3 `moslin` for in-vivo scLT data.

We demonstrated how `moslin` can be applied to real scRNA-seq data in the *C. elegans* example; however, lineage distances were derived from the ground-truth lineage tree rather than from an actual CRISPR/Cas9 lineage tracing experiment. We are currently investigating an application to real experimental scLT data for zebrafish heart regeneration [186] assayed using the LINNAEUS method [177]. Our initial results look promising and recapitulate the lineage origin of transient fibroblast subpopulations suggested and validated in the original publication [186].

Current limitations and outlook. The main limitation we are currently facing for the `moslin` model is the difficulty to define within time point cost functions for the GW term in the presence of experimental replicates. The problem is that lineage barcodes are only comparable within each respective replicate. For just two replicates, a possible workaround consists in first solving an FGW problem across replicates within one time point and using the resulting coupling to heuristically define a cost function; however, this does not generalize well to more than two replicates as it requires solving many FGW problems which becomes prohibitively expensive. An alternative we are currently evaluating is the computation of a barycenter at each time point from all available replicates [221]; this represents an "average" sample in terms of both gene expression and lineage similarity. Given two barycenters, one at either time point, the default `moslin` model can be applied to compute a coupling between them.

Another limitation of `moslin` is scalability; by default, compute time and memory scale cubically and quadratically for an FGW problem, respectively. While we resolved the quadratic memory complexity in `moscot-time` through online cost function evaluation (in OTT), the equivalent is not possible for `moslin` as the mirror descent scheme leads to varying cost matrices between Sinkhorn updates. A possible solution is given by low-rank factorizations which have recently been extended from classic OT to (F)GW objective functions and are now implemented in OTT [262]. However, this requires careful evaluation of mapping performance as a function of the rank r, the balance parameter α, and entropic regularization strength ϵ which we are currently investigating.

Chapter 5

Summary and outlook

Technological innovations over the past few years have turned single-cell genomics into a powerful lens to study dynamical biological processes; this thesis presented computational tools to support this endeavor. In particular, we made contributions along two main categories: Markov-state and optimal transport-based modeling of cellular trajectories. The research question formulated in Chapter 1 is motivated by recent technological advances, i.e. RNA velocity [1, 16] and dynamic genetic in-vivo barcoding [176–178], and revolves around establishing modeling paradigms that jointly make use of these novel data modalities and existing concepts developed for scRNA-seq data. We implement our theoretical contributions in two python packages, CellRank and moscot, and demonstrate their advantages over existing methods across a range of simulated and real data examples. Further, we show how both approaches can be used to formulate novel hypotheses about biological mechanisms and we experimentally validate one such hypothesis for CellRank on lung regeneration data [11, 14]. We summarize our contributions and discuss open questions in this chapter.

5.1 CellRank for directed single-cell fate mapping

In Chapter 3, we introduced the CellRank [14] framework and showed how it derives a robust Markov-chain representation of RNA velocity [1, 16] and gene expression similarity through a combination of `VelocityKernel` and `ConnectivityKernel`. Building on this representation, we introduced the `GPCCAEstimator` to infer initial and terminal states, fate probabilities, and various downstream quantities such as putative decision driver genes. We demonstrated the success of this analysis pipeline on MEF reprogramming [22], pancreas development [23] and lung regeneration [11] data. However, there are limitations to this approach which stem from RNA velocity not being applicable to some biological systems. For example, we showed how RNA velocity infers a directionality for adult hematopoiesis which is opposite to the known ground truth [25]. CellRank's modular design enabled us to overcome this limitation by introducing new kernels which make use of pseudotime (`PseudotimeKernel`), the CytoTRACE score [26] (`CytoTRACEKernel`) or real-time informa-

tion (`RealtimeKernel`) to make CellRank widely applicable. We demonstrate the success of our new kernels on adult hematopoiesis [25], zebrafish development [333], and MEF reprogramming [20].

CellRank has been taken up enthusiastically by the community with over 55k downloads to date and published applications to various biological systems including lung development [361], relationships among splenic Treg precursors [362], immune response in chronic obstructive pulmonary disease [363], and endoderm formation during gastrulation [364].

Outlook. We discussed some directions for further research in Section 3.6; these included kernels for novel data modalities such as metabolic labeling [284–288] or multi-modal data [70–73], estimators to describe different paths through phenotypic space and fine-grained approaches to identify genes with specific dynamics. All of these can be tackled within the CellRank framework on the basis of discrete Markov chains.

However, a promising direction for future research is generative models that can attribute cellular fate decisions to particular gene regulatory events, i.e. transcription factor binding, in a more causal manner. While these regulatory interactions appear to be difficult to estimate from observational data alone, interventional data containing e.g. CRISPR knockouts with shared transcriptome readout [42, 43], represent exciting new possibilities. A use-case of generative fate-mapping tools is cell-fate engineering; given a desired target cell-type A, these tools could help in designing optimal reprogramming routes, i.e. combinations of transcription factors that need to be up or down-regulated at specific times to achieve an optimal yield of cell type A. Cell-fate engineering has important applications in designing faithful disease models [365, 366] and in curing diseases related to the loss of particular cell types such as neuronal subtypes for Alzheimer's disease [367, 368] or beta cells for diabetes type 1 [369]. The CellOracle [54] and Capybara [370] methods represent promising steps in this direction.

5.2 moscot for scalable applications of OT to temporal single-cell data

In Chapter 4, we introduced the moscot framework with a focus on temporal applications. In particular, we introduced `moscot-time`, a model to estimate probabilistic couplings for cells sequenced at different time points. We showed that `moscot-time` can be applied to datasets that were inaccessible to previous methods due to scalability limitations in

both compute time and memory. For complex cellular dynamics, gene expression similarity across time points might not be enough to recover accurate couplings. For this challenging setting, we introduced `moslin` which supplements gene expression information with lineage relationships recorded through dynamic genetic barcoding techniques [176–178]. We demonstrated in applications to simulated and real data how our FGW objective function efficiently combines within time point lineage information with across time point gene expression information. In particular, we showcased on *C.elegans* developmental data how `moslin` can be combined with CellRank's `RealtimeKernel` to infer putative decision driver genes. `moslin` is part of the moscot framework and benefits from its consistent and easy-to-use API.

Outlook. We discussed some directions for further research in Section 4.5; these included barycenter computations to handle experimental replicates in `moslin`, low-rank approaches to accelerate the optimization of the FGW objective function, and neural OT to generalize beyond the training distribution. From an experimental point of view, dynamic genetic lineage tracing approaches remain difficult to set up and are currently only available for a few model organisms; to make `moslin` more widely applicable, it could be adapted for mitochondrial lineage tracing data which is experimentally easier to obtain due to recent technological innovations [74, 201, 204, 206]. This entails adapting the cost function to account for the specific sources of noise encountered in mitochondrial lineage tracing data (Subsection 2.1.3).

From a modeling perspective, an important consideration for mitochondrial lineage tracing data is experimental design; the most exciting use-cases for the technology are regenerative systems in humans (e.g. blood [28, 206]) which cannot be studied using any prospective scLT technology. However, this corresponds to the clonal resampling rather than the independent clonal evolution setting. In an effort to describe such an experimental design with `moslin`, the model should be extended towards an alternative objective function which relates both lineage barcodes as well as gene expression across time points, inspired by recent methodological advances [338, 371].

Long-term, multi-modal temporal readout should be coupled to spatial readout to obtain a more holistic view of cellular development including various molecular layers as well as the effects of spatial proximity; such datasets are starting to emerge [372]. Spatio-temporal studies enable us to go beyond the isolated view of individual cells and allow us to study their interaction with surrounding cells, i.e. including spatially-dependent cell-cell communication. Models should be adapted to include both external stimuli as well as internal regulatory elements when modeling fate choice in a spatio-temporal context. More fun-

damentally, future single-cell assays may be able to overcome the limitation of destroying cells when measuring genome-wide expression levels; this would allow entirely new insights into the regulatory underpinnings of fate decisions and provide exciting opportunities for mathematical modeling. Improvements in fluidic-force microscopy have recently enabled the first steps in this direction with live-seq [373].

To conclude, this thesis introduced two new modeling frameworks based on Markov chains and optimal transport which generalize trajectory inference beyond normal development and enable more scalable and accurate couplings of cells over time points, respectively. The frameworks extend previous efforts towards multi-view single-cell data with temporal and lineage resolution as well as estimates of the current direction of differentiation. We anticipate these methods will play an important role in using single-cell genomics data to learn about cellular dynamics and fate choice.

Appendix A

Background theory

A.1 Perron-Frobenius Theorem

This section reproduces the Perron-Frobenius Theorem [216] which ensures the existence and uniqueness of invariant measures for Markov chains. The theorem itself is not only defined for Markov chains but more generally for real-square matrices $A \in \mathbb{R}^{N\times N}$. Therefore, before stating the theorem, we generalize some of the notions from Section 2.2 to real square matrices A.

Definition A.1 (Non-Negative Matrix)**.** *If $A_{i,j} \geq 0 \, \forall i, j \in \{1, ..., n\}$, then we call A a non-negative matrix.*

Definition A.2 (Irreducible Matrix)**.** *Let A be a real square matrix. Then A is irreducible if it cannot be conjugated into block upper triangular form using a permutation matrix P, i.e.*

$$PAP^{-1} \neq \begin{pmatrix} B & C \\ 0 & D \end{pmatrix} \tag{A.1}$$

where B and D are non-trivial, i.e. they have a size greater than zero.

If A is non-negative, we can associate it with a weighted, directed graph G with N vertices where edge weights between vertices i and j are given by $A_{i,j}$. In this case, A is irreducible if and only if G is strongly connected. We call A *reducible* if it is not irreducible.

Definition A.3 (Period of an Index)**.** *Let A be non-negative and fix an index i. Define the set $M := \{m : (A^m)_{i,i} > 0\}$. Then the period $h(i)$ of i is the greatest common divisor of the set M.*

Note that if A is non-negative and irreducible, $h(i)$ is the same for all i and we can define the period h of A as the period of any of its indices. If the period of A is one, we call it *aperiodic*. Further note that if any diagonal element of A is positive, A is aperiodic.

Theorem A.1 (Perron-Frobenius Theorem for irreducible matrices)**.** *Let $A \in \mathbb{R}^{N\times N}$ be irreducible and non-negative. Let further $r = \rho(A)$ be the spectral radius and h be the period of A. Then the following hold:*

(i) r is a positive, real eigenvalue of A called the Perron-Frobenius eigenvalue

(ii) r is simple and both left and right eigenspaces associated with r are one-dimensional

(iii) There exist left and right eigenvectors w and v associated with r which have only positive components

(iv) A has exactly h complex eigenvalues with absolute value r. Each of these has algebraic multiplicity one. Further, each of these eigenvalues has the form $r\exp(i2\pi l/h)$ for $l \in \{0, 1, ..., h-1\}$.

Proof. See Perron [216]. □

Existence and uniqueness of invariant measures. Consider an irreducible MC with transition matrix $T \in R_+^{N\times N}$ with the usual row normalization $\sum_j T_{ij} = 1\ \forall i$. From the row normalization, it follows that the vector $e = (1, ..., 1)^\top$ is a right eigenvector of T with eigenvalue 1. Suppose that there exists a right eigenvector v with eigenvalue $|\lambda| > 1$. It follows that the vector $\lambda^l v = T^l v$ has exponentially growing length for $n \to \infty$, thus there exist (i, j) with $T^l_{i,j} > 1$, a contradiction to the fact that T^l describes the l-step transition probabilities of the MC. Therefore, $\rho(T) = r = 1$ is the spectral radius of T and by Theorem A.1, there exists a unique (up to multiplication) non-negative left eigenvector w associated with r such that $w^\top T = w^\top$.

A.2 Deriving the CME for RNA velocity

A common way to derive CMEs is to relate the system state at two nearby time points t and $t + dt$. In particular, we consider events which contribute towards $P_{mn}(t + dt)$, i.e. events which lead to the system being in state (m, n) at time $t + dt$, conditional on the system state at time t. The probability $P_{mn}(t)$ is defined as in Equation (2.87) in Section 2.5 of the main text, in particular, we write $u(t) = m$ for unspliced molecules and $s(t) = n$ for spliced molecules to simplify equations. Note that we only consider processes which are linear in dt, i.e. first order contributions. For the RNA velocity model, these include

(i) the system was in state (m, n) at time t, no transcription, splicing or degradation happened in dt. This happens with probability $(1-\alpha dt)(1-\beta dt)^m(1-\gamma dt)^n$.

(ii) the system was in state $(m-1, n)$ at time t and one transcription event happened in dt. This happens with probability αdt.

(iii) the system was in state $(m, n+1)$ at time t and one degradation event happened in dt. This happens with probability $(n+1)\gamma dt$.

(iv) the system was in state $(m+1, n-1)$ at time t and one splicing event happened in dt. This happens wiht probability $(m+1)\beta dt$.

We combine these 4 contributions to yield an expression for $P_{mn}(t+dt)$,

$$\begin{aligned} P_{mn}(t+dt) = & P_{mn}(t)(1-\alpha dt)(1-\beta dt)^m(1-\gamma dt)^n \\ & + P_{m-1,n}(t)\alpha dt \\ & + P_{m,n+1}(n+1)\gamma dt \\ & + P_{m+1,n-1}(m+1)\beta dt \\ & + \mathcal{O}(dt^2)\,. \end{aligned}$$

Re-arranging terms, removing higher-order contributions and taking the limit $dt \to 0$ gives

$$\begin{aligned} \frac{dP_{mn}(t)}{dt} = & -P_{mn}(t)(\alpha + m\beta + n\gamma) \\ & + P_{m-1,n}\alpha \\ & + P_{m,n+1}(n+1)\gamma \\ & + P_{m+1,n-1}(m+1)\beta\,, \end{aligned} \tag{A.2}$$

which can be re-arranged into the final form of Li et al. [278] which was given in Section 2.5.

A.3 Solution to the moment equations for RNA velocity

The system of ODEs which results from computing moments up to a certain order $\langle m^k n^l \rangle$ of the CME of Equation (2.88) in Section 2.5 is closed, i.e. moments of a certain order $o = l+k$ do not depend on higher-order terms but only on lower (and same) order terms. As the differential equations are also linear, they can be solved in closed form, without making e.g. the usual moment closure approximation. We outline below the general procedure and show how to apply it to first-order moments.

General procedure. Define $\boldsymbol{x}$ to be the vector of all moments $\langle m^k n^l \rangle$ up to a certain order $o = l + k$ and write the system in matrix form as

$$\frac{d\boldsymbol{x}}{dt} = A\boldsymbol{x} + \boldsymbol{b}\,, \tag{A.3}$$

for coefficient matrix $A \in \mathbb{R}^{E \times E}$ for E, the number of equations and moments of given order, and $b \in \mathbb{R}^E$, a constant vector. Equations of this form can be solved by finding the general solution to the homogeneous equation, $d\boldsymbol{x}/dt = A\boldsymbol{x}$, and combining it with a particular solution to the nonhomogeneous equations, $d\boldsymbol{x}/dt = A\boldsymbol{x} + b$. The Ansatz $\boldsymbol{x} = \boldsymbol{v}\mathrm{e}^{\lambda}t$ shows the general solution to the homogeneous equation is given by the eigenvectors $\boldsymbol{v}^{(i)}$ and eigenvalues $\lambda^{(i)}$ of A, while a particular solution to the nonhomogeneous equation is given by a constant vector $\boldsymbol{x} = \boldsymbol{g}$. Combining the former two leads to the solution

$$\boldsymbol{x}(t) = \sum_i c_i \boldsymbol{v}^{(i)} \mathrm{e}^{\lambda^{(i)} t} + \boldsymbol{g}\,, \tag{A.4}$$

for constants c_i to be determined using the boundary conditions.

A.3.1 Solution to the first order moment equations for RNA velocity

As outlined above, define $\boldsymbol{x}(t) \coloneqq [\langle u(t) \rangle, \langle s(t) \rangle]^\top \in \mathbb{R}^2$. Further, suppose $\beta, \gamma > 0$ and $\beta \neq \gamma$. The first order moment equations from Section 2.5 may be written

$$\frac{d\boldsymbol{x}}{dt} = A\boldsymbol{x} + \boldsymbol{b}\,,$$

for matrix $A \in \mathbb{R}^2$ and vector $\boldsymbol{b} \in \mathbb{R}^2$ given by

$$A \coloneqq \begin{bmatrix} -\beta & 0 \\ \beta & -\gamma \end{bmatrix}, \quad b \coloneqq \begin{bmatrix} \alpha^{\text{on/off}} \\ 0 \end{bmatrix},$$

subject to the initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0 \coloneqq [\langle u(0) \rangle, \langle s(0) \rangle]^\top = [u_0, s_0]^\top$.

General solution to the homogeneous equation. Using the ansatz $\boldsymbol{x} = \boldsymbol{v}e^{\lambda t}$ yields

$$\begin{aligned} &\lambda \boldsymbol{v}\mathrm{e}^{\lambda t} = A\boldsymbol{v}\mathrm{e}^{\lambda t} \\ &\Leftrightarrow \lambda \boldsymbol{v} = A\boldsymbol{v}\,, \end{aligned}$$

as $\mathrm{e}^{\lambda t} > 0 \, \forall t$. Thus, λ and $\boldsymbol{v}$ are eigenvalues and eigenvectors, respectively, of A. Solving for these gives

$$\lambda^{(1)} = -\beta, \qquad \lambda^{(2)} = -\gamma$$
$$\boldsymbol{v}^{(1)} = \begin{bmatrix} \frac{\gamma-\beta}{\beta} \\ 1 \end{bmatrix}, \qquad \boldsymbol{v}^{(2)} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Accordingly, the general solution to the homogeneous equation reads

$$\boldsymbol{x} = c_1 \boldsymbol{v}^{(1)} \mathrm{e}^{\lambda^{(1)} t} + c_2 \boldsymbol{v}^{(2)} \mathrm{e}^{\lambda^{(2)} t},$$

for constants $c_1, c_2 \in \mathbb{R}$, to be determined later using the initial condition.

Particular solution to the the nonhomogeneous equation. As the nonhomogeneous part does not involve t, we use a constant vector as ansatz,

$$\boldsymbol{x}(t) = \boldsymbol{g},$$

for $g \in \mathbb{R}^2$. Substituting this into the nonhomogeneous equation yields

$$A\boldsymbol{g} = -\boldsymbol{b},$$

with solution given by $\boldsymbol{g} = [\alpha^{\mathrm{on/off}}/\beta, \alpha^{\mathrm{on/off}}/\gamma]^\top$.

Enforcing the initial condition. Combining the above two, the general solution to the nonhomogeneous equation is given by

$$\boldsymbol{x} = c_1 \boldsymbol{v}^{(1)} \mathrm{e}^{\lambda^{(1)} t} + c_2 \boldsymbol{v}^{(2)} \mathrm{e}^{\lambda^{(2)} t} + \boldsymbol{g}.$$

Using the initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0$ gives the following system of equations for the unknown $\boldsymbol{c} \coloneqq [c_1, c_2]^\top$:

$$V\boldsymbol{c} = \boldsymbol{x}_0 - \boldsymbol{g}$$

for matrix $V \coloneqq [\boldsymbol{v}^{(1)}, \boldsymbol{v}^{(2)}] \in \mathbb{R}^{2\times 2}$ with solution given by

$$\boldsymbol{c} = \begin{bmatrix} \frac{\alpha^{\mathrm{on/off}} - \beta u_0}{\beta - \gamma} \\ \frac{\alpha^{\mathrm{on/off}} \beta + \gamma(\gamma s_0 - \beta(s_0 + u_0))}{\gamma(\gamma - \beta)} \end{bmatrix}.$$

Combining all pieces gives the solution to the initial value problem,

$$x_1(t) = \left(\frac{\beta u_0 - \alpha^{\text{on/off}}}{\beta}\right) \mathrm{e}^{-\beta t} + \frac{\alpha^{\text{on/off}}}{\beta}\,,$$
$$x_2(t) = \left(\frac{\alpha^{\text{on/off}} - \beta u_0}{\beta - \gamma}\right) \mathrm{e}^{-\beta t} + \left(\frac{\alpha^{\text{on/off}}\beta + \gamma(\gamma s_0 - \beta(s_0 + u_0))}{\gamma(\gamma - \beta)}\right) \mathrm{e}^{-\gamma t} + \frac{\alpha^{\text{on/off}}}{\gamma}\,,$$

which may be simplified to read

$$x_1(t) = u_0 \mathrm{e}^{-\beta t} + \frac{\alpha^{\text{on/off}}}{\beta}(1 - \mathrm{e}^{-\beta t})\,, \tag{A.5}$$

$$\begin{aligned} x_2(t) &= \left(\frac{\alpha^{\text{on/off}} - \beta u_0}{\beta - \gamma}\right) \mathrm{e}^{-\beta t} + \left(\frac{\alpha^{\text{on/off}}\beta - \beta\gamma u_0}{\gamma(\gamma - \beta)}\right) \mathrm{e}^{-\gamma t} + s_0 \mathrm{e}^{-\gamma t} + \frac{\alpha^{\text{on/off}}}{\gamma} \\ &= \left(\frac{\alpha^{\text{on/off}} - \beta u_0}{\beta - \gamma}\right) \mathrm{e}^{-\beta t} + \left(\frac{\alpha^{\text{on/off}} - \beta u_0}{\gamma - \beta} - \frac{\alpha^{\text{on/off}}}{\gamma}\right) \mathrm{e}^{-\gamma t} + s_0 \mathrm{e}^{-\gamma t} + \frac{\alpha^{\text{on/off}}}{\gamma} \\ &= s_0 \mathrm{e}^{-\gamma t} + \frac{\alpha^{\text{on/off}}}{\gamma}\left(1 - \mathrm{e}^{-\gamma t}\right) + \frac{\alpha^{\text{on/off}} - \beta u_0}{\gamma - \beta}\left(\mathrm{e}^{-\gamma t} - \mathrm{e}^{-\beta t}\right)\,. \end{aligned} \tag{A.6}$$

A.4 Solution to the CME for RNA velocity

In Li et al. [278], the authors derive the results we reproduce in Theorem A.2 for both the off- and on-stages with $\alpha^{\text{on/off}} = 0$ and $\alpha^{\text{on/off}} = \alpha$, respectively.

Theorem A.2 (Solution to the CME of Equation (2.88))**.** *Suppose $\beta \neq \gamma$. In the off-stage, with initial data given by $P^{on}_{mn|MN}(0) = \delta_{mM}\delta_{nN}$, the solution to the CME is given by*

$$P^{off}_{mn|MN}(t) = Bin\,(m|M, p_1)\, C_n(M - m, p_2, N, p_3)\,, \tag{A.7}$$

for $Bin\,(k|M - m, p_2)$, the binominal distribution, and for C_n defined by

$$C_n(M - m, p_2, N, p_3) = \sum_{k=0}^{n} Bin\,(k|M - m, p_2)\, Bin\,(n - k|N, p_3)\,,$$

where p_1, p_2 and p_3 are defined as follows:

$$\begin{aligned} p_1(t) &= \mathrm{e}^{-\beta t}\,, \\ p_2(t) &= \frac{\beta}{\beta-\gamma}\left(\frac{\mathrm{e}^{-\gamma t}-\mathrm{e}^{-\beta t}}{1-\mathrm{e}^{-\beta t}}\right), \\ p_3(t) &= \mathrm{e}^{-\gamma t}\,. \end{aligned}$$

In the on-stage, with initial data given by $P^{on}_{mn|m_0n_0}(0) = \delta_{mm_0}\delta_{nn_0}$, the solution to the CME is given by

$$P^{on}_{mn|m_0n_0}(t) = \sum_{k=0}^{m}\sum_{l=0}^{n} P^{on}_{kl|00}(t) P^{off}_{m-k,n-l|m_0,n_0}(t)\,, \tag{A.8}$$

where $P^{on}_{kl|00}(t)$ is the distribution in the on-stage for zero initial counts, i.e. $P^{on}_{mn|00}(0) = \delta_{m0}\delta_{n0}$, for which the following expression holds:

$$P^{on}_{kl|00}(t) = \frac{x_1^m(t)\, x_2^n(t)}{m!\, n!}\mathrm{e}^{-a(t)-b(t)}\,, \tag{A.9}$$

where $x_1(t)$ and $x_2(t)$ are the solutions to the first-order moments equations given by Equations (A.5) *and* (A.6) *for initial condition $(u_0, s_0) = (0, 0)$. Thus, for zero initial counts, $u(t)$ and $s(t)$ are independently Poisson distributed around the mean value given by the solution to the first order moment equations.*

Proof. See the proofs to Theorems 2.1 and 2.2 as well as Corollary 2.2 in Li et al. [278]. □

Appendix B

Supplementary Figures

B.1 CellRank: pancreas development

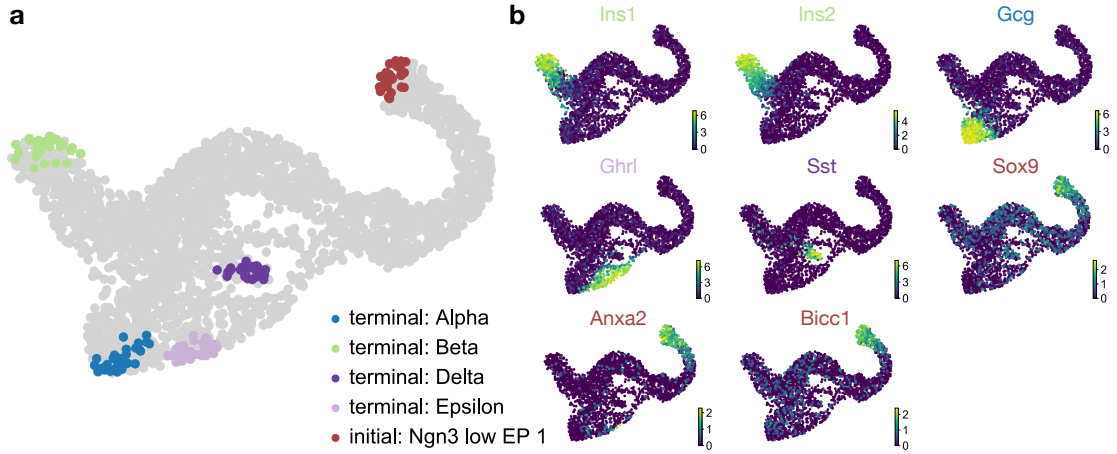

Figure B.1: Marker genes confirm CellRanks initial and terminal state annotations in the pancreas data. a. CellRank-computed initial and terminal states from Figure 3.7d in Section 3.4. **b.** We color cells based on the expression level of the indicated gene in each UMAP. Terminal states express key marker genes; showing for beta: *Ins1* and *Ins2* (insulin), alpha: *Gcg* (glucagon), epsilon: *Ghrl* (ghrelin), delta: *Sst* (somatostatin) [318]. For the initial state, we show the expression of ductal cell markers *Sox9*, *Anxa2*, and *Bicc1* [23, 318]. Figure reproduced from Lange et al. [14].

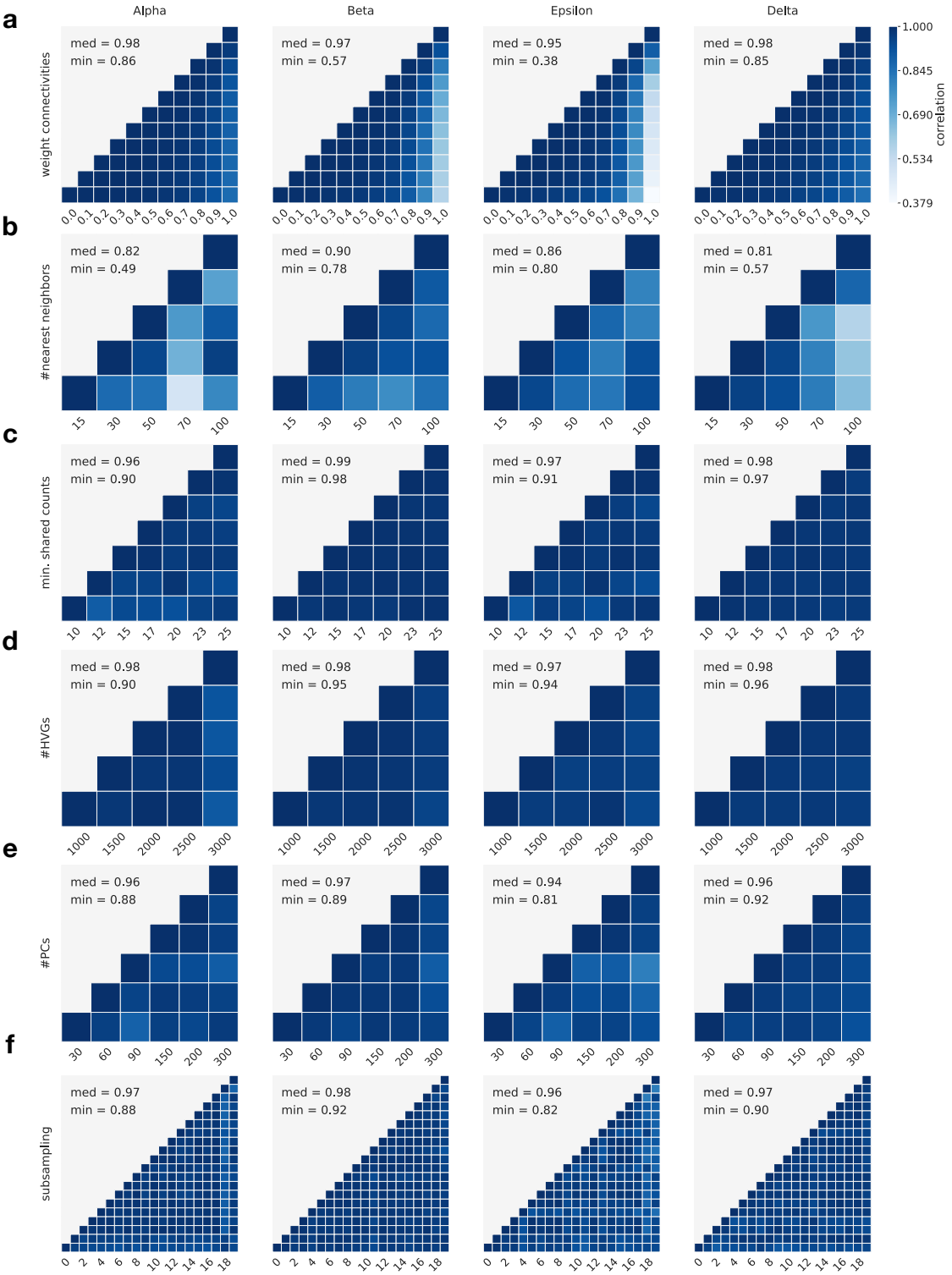

Figure B.2: CellRank is robust to parameter choice and random subsampling a-e. Pairwise correlations of fate probabilities per lineage when varying (**a**) the weight given to the `ConnectivityKernel` (**b**) the number of nearest neighbors in KNN graph construction, (**c**) the gene filtering parameter *min_shared_counts* which determines the minimum required number of spliced and unspliced counts, (**d**) the number of highly variable genes, (**e**) the number of principal components N_l used for KNN graph construction. **f.** Pairwise correlations of fate probabilities per lineage when randomly subsampling the data to 90% of cells. Figure reproduced from Lange et al. [14].

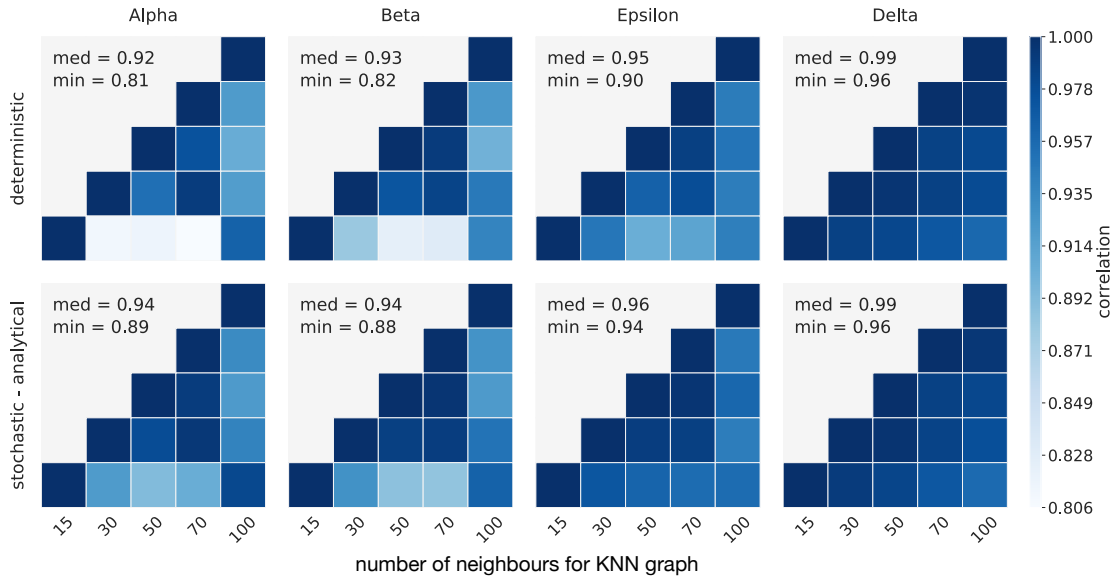

Figure B.3: Propagating uncertainty significantly increases the robustness of fate probabilities. We show this here for the alpha, beta, and epsilon lineages with respect to the number of neighbors used for KNN graph construction (one-sided Wilcoxon signed-rank test, $W = 55.0, P = 9.7 \times 10^{-4}$). For the delta lineage, no significant robustness increase was found. For similar results with respect to other parameters, see the original publication [14]. Figure reproduced from Lange et al. [14].

B.2 CellRank: lung regeneration

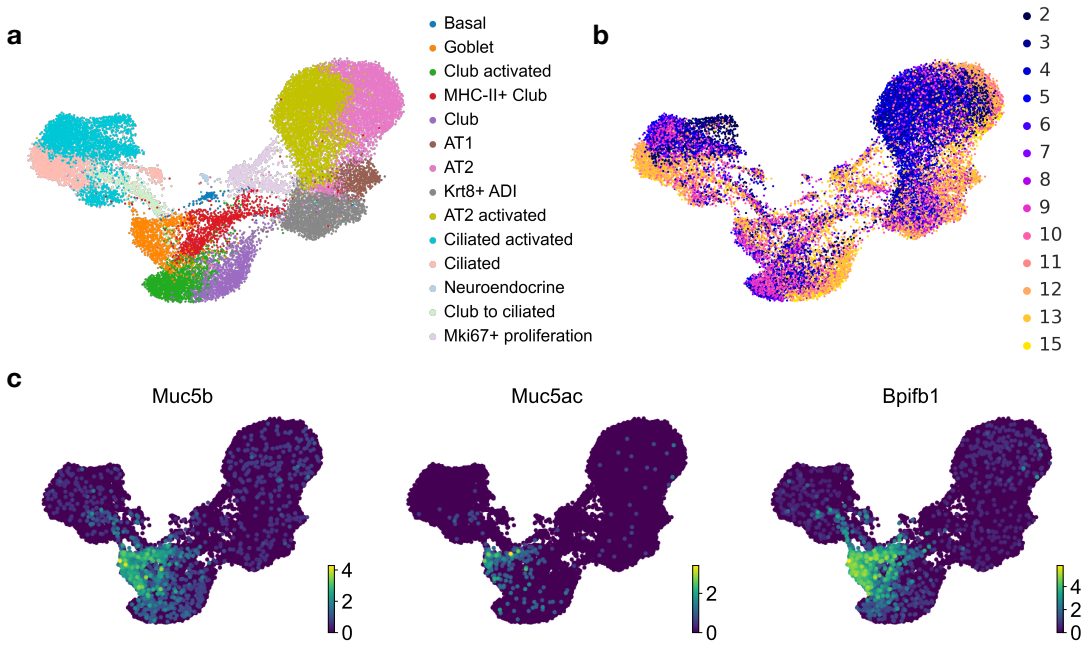

Figure B.4: Cluster labels and time point annotations for lung data. a. Original cluster labels for the lung regeneration data [11] in a UMAP projection. The dataset consists of 24,882 murine lung epithelial cells sequenced using Drop-seq [35] at 13 time points spanning days 2-15 past bleomycin injury. The 'activated' label refers to cell states that emerge after bleomycin injury. **b.** Same as (**a**) with time points colored in. Time points refer to time passed since bleomycin injury. **c.** Expression of goblet cell markers *Muc5b*, *Muc5ac* and *Bpifb1* agrees with the goblet annotation of (*a*). Figure reproduced from Lange et al. [14].

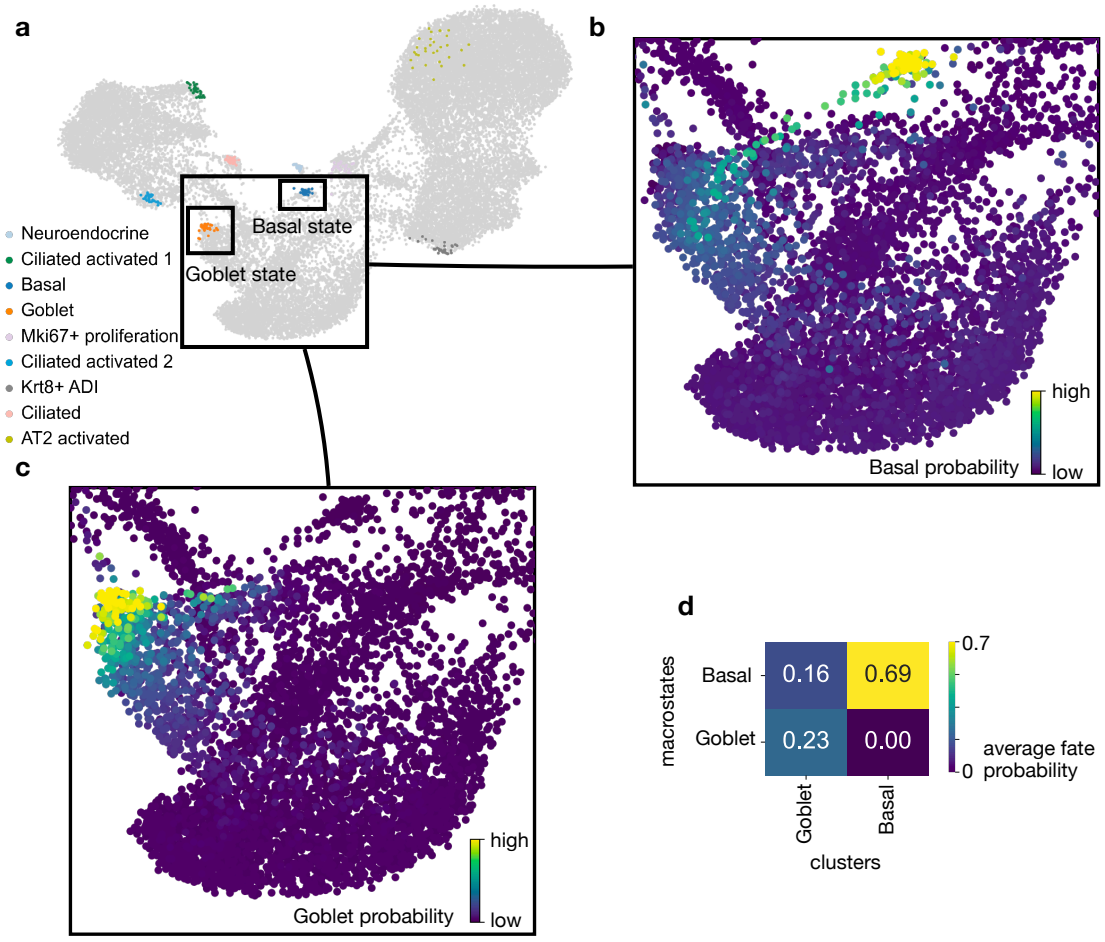

Figure B.5: CellRank predicts a goblet to basal dedifferentiation trajectory. a. Cellrank identifies 9 macrostates; we highlight airway cells, including club, goblet and basal cells. **b.** Fate probabilities of transitioning towards the basal state. A 'band' of cells within the goblet cluster exhibits high basal cell fate probability. **c.** Fate probabilities of transitioning towards the goblet state. Basal cells do not show any probability of transitioning towards the goblet state. **d.** Quantification of the results from (**b**) and (**c**). Goblet cells are likely to transition towards basal cells, but basal cells are extremely unlikely to transition towards the goblet state, confirming the direction of the recovered trajectory, from goblet to basal. Figure reproduced from Lange et al. [14].

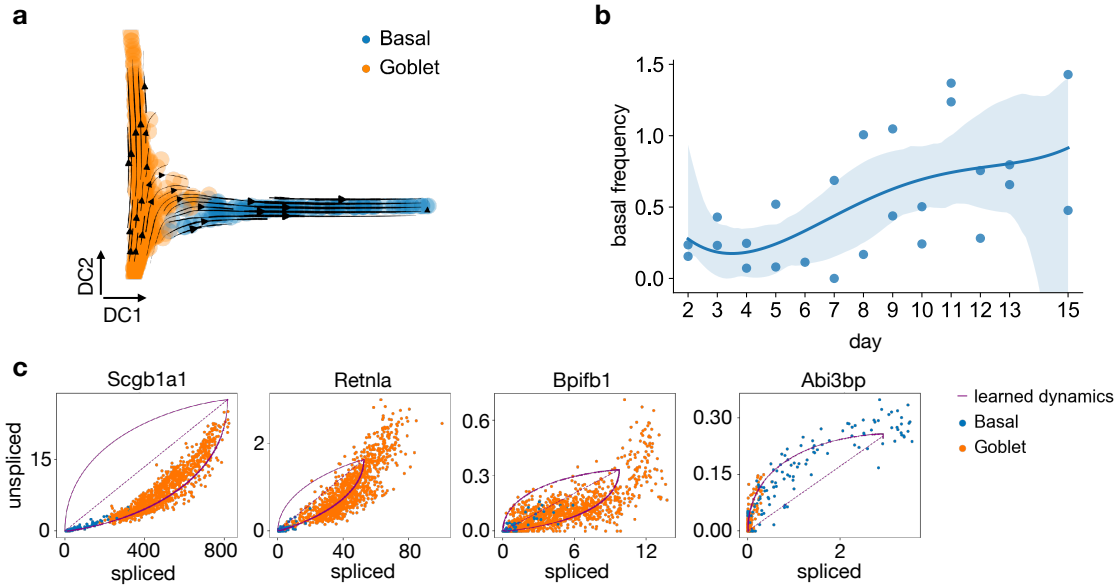

Figure B.6: Basal cell frequency increases over time and gene-wise velocities support dedifferentiation. a. Diffusion map computed on the subset of basal and goblet cells, showing scVelo computed velocities as streamlines. **b.** Proportion of basal cells per sample for each of the two samples available per time point. The blue line shows a 4^{th} order polynomial regression fit, shaded regions are 95% confidence intervals computed through bootstrap sampling. **c.** Scatter plots of spliced vs. unspliced counts for *Scgb1a1*, *Retnla*, *Bpifb1* and *Abi3bp*, all of which are among the top 30 likelihood genes according to scVelo's dynamical model of splicing kinetics15, colored by cell type. Purple line shows scVelo's fitted splicing dynamics which support the goblet to basal direction for all 4 genes. Both *Scbg1a176* as well as *Bpifb177* are known markers for secretory/goblet cells and are downregulated in the transition. The top 100 likelihood genes further include known goblet cell markers *Muc5b* and *Muc5ac78*, highlighting that velocities are driven by biologically meaningful genes (data now shown). Figure reproduced from Lange et al. [14].

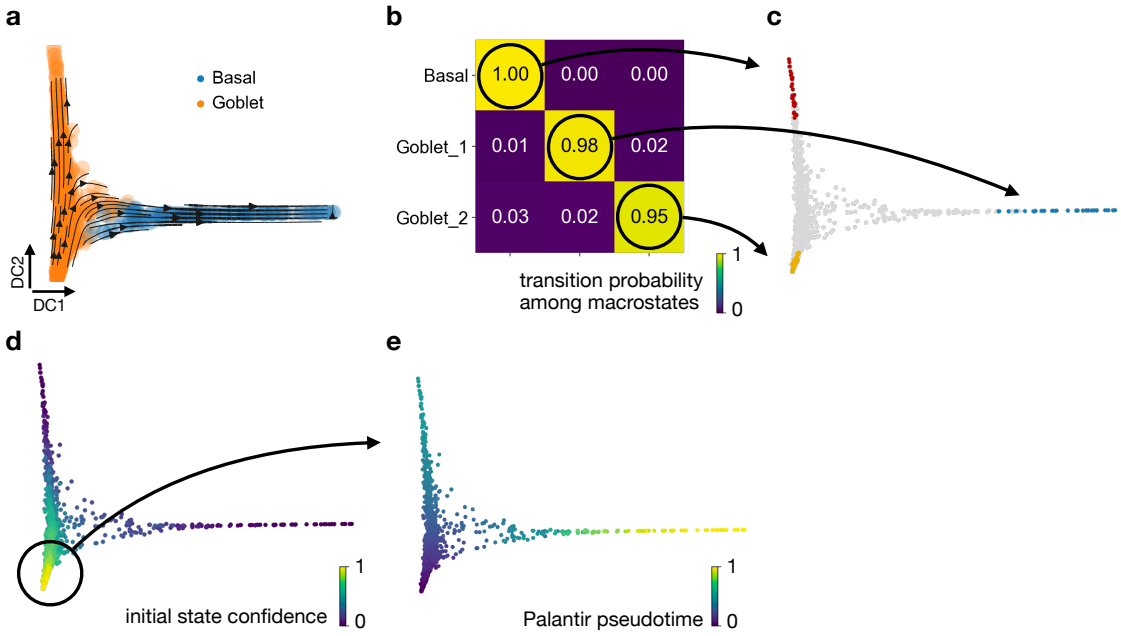

Figure B.7: Computing a pseudotime for the goblet to basal transitions. a. Diffusion map of a subset of the cells from the lung data of Figure 3.10d labeled as "Goblet" and "Basal" in the original publication [11]. **b.** Coarse-grained transition matrix $\tilde{T}$, computed for three macrostates. The macrostate labeled as 'Goblet_2' was automatically detected as initial by the `GPCCAEstimator` because it had the smallest value in the coarse-grained stationary distribution $\tilde{\pi}$. **c.** Showing the 30 cells most confidently assigned to their macrostate in the diffusion map. We kept the color for the basal state but created two new colors for the initial and terminal goblet states because they both overlap with the same transcriptomic goblet cluster and hence would both get the same color. **d.** Membership vector corresponding to the initial 'Goblet_2' state, here labeled as 'initial state confidence'. The cell which had the maximum value in the initial state confidence was used as initial cells to compute Palantir's pseudotime [25]. **e.** Palantir pseudotime. Figure reproduced from Lange et al. [14].

Bibliography

[1] G. L. Manno, R. Soldatov, A. Zeisel, E. Braun, H. Hochgerner, V. Petukhov, K. Lidschreiber, M. E. Kastriti, P. Lönnerberg, A. Furlan, J. Fan, L. E. Borm, Z. Liu, D. v. Bruggen, J. Guo, X. He, R. Barker, E. Sundström, G. Castelo-Branco, P. Cramer, I. Adameyko, S. Linnarsson, and P. V. Kharchenko. "RNA velocity of single cells". en. In: *Nature* (Aug. 2018), p. 1. DOI: 10.1038/s41586-018-0414-6.

[2] F. Tang, C. Barbacioru, Y. Wang, E. Nordman, C. Lee, N. Xu, X. Wang, J. Bodeau, B. B. Tuch, A. Siddiqui, K. Lao, and M. A. Surani. "mRNA-Seq whole-transcriptome analysis of a single cell". en. In: *Nature Methods* 6.5 (May 2009), pp. 377–382. DOI: 10.1038/nmeth.1315.

[3] J. Cao, D. R. O'Day, H. A. Pliner, P. D. Kingsley, M. Deng, R. M. Daza, M. A. Zager, K. A. Aldinger, R. Blecher-Gonen, F. Zhang, M. Spielmann, J. Palis, D. Doherty, F. J. Steemers, I. A. Glass, C. Trapnell, and J. Shendure. "A human cell atlas of fetal gene expression". eng. In: *Science (New York, N.Y.)* 370.6518 (Nov. 2020), eaba7721. DOI: 10.1126/science.aba7721.

[4] W. Sungnak, N. Huang, C. Bécavin, M. Berg, R. Queen, M. Litvinukova, C. Talavera-López, H. Maatz, D. Reichart, F. Sampaziotis, K. B. Worlock, M. Yoshida, and J. L. Barnes. "SARS-CoV-2 entry factors are highly expressed in nasal epithelial cells together with innate immune genes". en. In: *Nature Medicine* 26.5 (May 2020), pp. 681–687. DOI: 10.1038/s41591-020-0868-6.

[5] C. G. K. Ziegler et al. "SARS-CoV-2 Receptor ACE2 Is an Interferon-Stimulated Gene in Human Airway Epithelial Cells and Is Detected in Specific Cell Subsets across Tissues". en. In: *Cell* 181.5 (May 2020), 1016–1035.e19. DOI: 10.1016/j.cell.2020.04.035.

[6] B. Pijuan-Sala, J. A. Griffiths, C. Guibentif, T. W. Hiscock, W. Jawaid, F. J. Calero-Nieto, C. Mulas, X. Ibarra-Soria, R. C. V. Tyser, D. L. L. Ho, W. Reik, S. Srinivas, B. D. Simons, J. Nichols, J. C. Marioni, and B. Göttgens. "A single-cell molecular map of mouse gastrulation and early organogenesis". en. In: *Nature* 566.7745 (Feb. 2019), pp. 490–495. DOI: 10.1038/s41586-019-0933-9.

[7] S. Nowotschin, M. Setty, Y.-Y. Kuo, V. Liu, V. Garg, R. Sharma, C. S. Simon, N. Saiz, R. Gardner, S. C. Boutet, D. M. Church, P. A. Hoodless, A.-K. Hadjantonakis, and D. Pe'er. "The emergent landscape of the mouse gut endoderm at single-cell resolution". en. In: *Nature* 569.7756 (May 2019), pp. 361–367. DOI: 10.1038/s41586-019-1127-1.

[8] A. Regev, S. A. Teichmann, E. S. Lander, I. Amit, C. Benoist, E. Birney, B. Bodenmiller, P. Campbell, P. Carninci, M. Clatworthy, H. Clevers, B. Deplancke, I. Dunham, J. Eberwine, R. Eils, W. Enard, A. Farmer, L. Fugger, B. Göttgens, N. Hacohen, M. Haniffa, M. Hemberg, S. Kim, P. Klenerman, A. Kriegstein, E. Lein, S. Linnarsson, E. Lundberg, J. Lundeberg, P. Majumder, J. C. Marioni, M. Merad, M. Mhlanga, M. Nawijn, M. Netea, G. Nolan, D. Pe'er, A. Phillipakis, C. P. Ponting, S. Quake, W. Reik, O. Rozenblatt-Rosen, J. Sanes, R. Satija, T. N. Schumacher, A. Shalek, E. Shapiro, P. Sharma, J. W. Shin, O. Stegle, M. Stratton, M. J. T. Stubbington, F. J. Theis, M. Uhlen, A. van Oudenaarden, A. Wagner, F. Watt, J. Weissman, B. Wold, R. Xavier, N. Yosef, and Human Cell Atlas Meeting Participants. "The Human Cell Atlas". In: *eLife* 6 (Dec. 2017). Ed. by T. R. Gingeras, e27041. DOI: 10.7554/eLife.27041.

[9] N. Rajewsky, G. Almouzni, S. A. Gorski, S. Aerts, I. Amit, M. G. Bertero, C. Bock, A. L. Bredenoord, G. Cavalli, S. Chiocca, H. Clevers, B. De Strooper, A. Eggert, J. Ellenberg, X. M. Fernández, M. Figlerowicz, S. M. Gasser, N. Hubner, J. Kjems, J. A. Knoblich, G. Krabbe, P. Lichter, S. Linnarsson, J.-C. Marine, J. Marioni, M. A. Marti-Renom, M. G. Netea, D. Nickel, M. Nollmann, H. R. Novak, H. Parkinson, S. Piccolo, I. Pinheiro, A. Pombo, C. Popp, W. Reik, S. Roman-Roman, P. Rosenstiel, J. L. Schultze, O. Stegle, A. Tanay, G. Testa, D. Thanos, F. J. Theis, M.-E. Torres-Padilla, A. Valencia, C. Vallot, A. van Oudenaarden, M. Vidal, and T. Voet. "LifeTime and improving European healthcare through cell-based interceptive medicine". en. In: *Nature* (Sept. 2020), pp. 1–14. DOI: 10.1038/s41586-020-2715-9.

[10] C. Weinreb and A. M. Klein. "Lineage reconstruction from clonal correlations". en. In: *Proceedings of the National Academy of Sciences* (July 2020), p. 202000238. DOI: 10.1073/pnas.2000238117.

[11] M. Strunz, L. M. Simon, M. Ansari, J. J. Kathiriya, I. Angelidis, C. H. Mayr, G. Tsidiridis, M. Lange, L. F. Mattner, M. Yee, P. Ogar, A. Sengupta, I. Kukhtevich, R. Schneider, Z. Zhao, C. Voss, T. Stoeger, J. H. L. Neumann, A. Hilgendorff, J. Behr, M. O'Reilly, M. Lehmann, G. Burgstaller, M. Königshoff, H. A. Chapman, F. J. Theis, and H. B. Schiller. "Alveolar regeneration through a Krt8+ transitional stem cell state that persists in human lung fibrosis". en. In: *Nature Communications* 11.1 (July 2020), p. 3559. DOI: 10.1038/s41467-020-17358-3.

[12] W. Saelens, R. Cannoodt, H. Todorov, and Y. Saeys. "A comparison of single-cell trajectory inference methods". en. In: *Nature Biotechnology* (Apr. 2019). DOI: 10.1038/s41587-019-0071-9.

[13] S. Tritschler, M. Büttner, D. S. Fischer, M. Lange, V. Bergen, H. Lickert, and F. J. Theis. "Concepts and limitations for learning developmental trajectories from single cell genomics". en. In: *Development* 146.12 (June 2019), dev170506. DOI: 10.1242/dev.170506.

[14] M. Lange, V. Bergen, M. Klein, M. Setty, B. Reuter, M. Bakhti, H. Lickert, M. Ansari, J. Schniering, H. B. Schiller, D. Pe'er, and F. J. Theis. "CellRank for directed single-cell fate mapping". en. In: *Nature Methods* (Jan. 2022), pp. 1–12. DOI: 10.1038/s41592-021-01346-6.

[15] M. Lange, Z. Piran, M. Klein, B. Spanjaard, J. P. Junker, F. J. Theis, and M. Nitzan. "Mapping lineage-traced single-cells across time-points". In: *In preparation* (2022).

[16] V. Bergen, M. Lange, S. Peidli, F. A. Wolf, and F. J. Theis. "Generalizing RNA velocity to transient cell states through dynamical modeling". en. In: *Nature Biotechnology* (Aug. 2020), pp. 1–7. DOI: 10.1038/s41587-020-0591-3.

[17] D. E. Wagner and A. M. Klein. "Lineage tracing meets single-cell omics: opportunities and challenges". en. In: *Nature Reviews Genetics* 21.7 (July 2020), pp. 410–427. DOI: 10.1038/s41576-020-0223-2.

[18] S. C. Bendall, K. L. Davis, E.-a. D. Amir, M. D. Tadmor, E. F. Simonds, T. J. Chen, D. K. Shenfeld, G. P. Nolan, and D. Pe'er. "Single-Cell Trajectory Detection Uncovers Progression and Regulatory Coordination in Human B cell Development". In: *Cell* 157.3 (Apr. 2014), pp. 714–725. DOI: 10.1016/j.cell.2014.04.005.

[19] F. H. Crick. "On protein synthesis". eng. In: *Symposia of the Society for Experimental Biology* 12 (1958), pp. 138–163.

[20] G. Schiebinger, J. Shu, M. Tabaka, B. Cleary, V. Subramanian, A. Solomon, J. Gould, S. Liu, S. Lin, P. Berube, L. Lee, J. Chen, J. Brumbaugh, P. Rigollet, K. Hochedlinger, R. Jaenisch, A. Regev, and E. S. Lander. "Optimal-Transport Analysis of Single-Cell Gene Expression Identifies Developmental Trajectories in Reprogramming". English. In: *Cell* 0.0 (Jan. 2019). DOI: 10.1016/j.cell.2019.01.006.

[21] C. Villani. *Optimal transport: old and new*. en. Grundlehren der mathematischen Wissenschaften 338. Berlin Heidelberg: Springer, 2009.

[22] B. A. Biddy, W. Kong, K. Kamimoto, C. Guo, S. E. Waye, T. Sun, and S. A. Morris. "Single-cell mapping of lineage and identity in direct reprogramming". En. In: *Nature* 564.7735 (Dec. 2018), p. 219. DOI: 10.1038/s41586-018-0744-4.

[23] A. Bastidas-Ponce, S. Tritschler, L. Dony, K. Scheibner, M. Tarquis-Medina, C. Salinno, S. Schirge, I. Burtscher, A. Böttcher, F. J. Theis, H. Lickert, and M. Bakhti. "Comprehensive single cell mRNA profiling reveals a detailed roadmap for pancreatic endocrinogenesis". en. In: *Development* 146.12 (June 2019), dev173849. DOI: 10.1242/dev.173849.

[24] J. S. Packer, Q. Zhu, C. Huynh, P. Sivaramakrishnan, E. Preston, H. Dueck, D. Stefanik, K. Tan, C. Trapnell, J. Kim, R. H. Waterston, and J. I. Murray. "A lineage-resolved molecular atlas of C. elegans embryogenesis at single-cell resolution". en. In: *Science* 365.6459 (Sept. 2019). DOI: 10.1126/science.aax1971.

[25] M. Setty, V. Kiseliovas, J. Levine, A. Gayoso, L. Mazutis, and D. Pe'er. "Characterization of cell fate probabilities in single-cell data with Palantir". En. In: *Nature Biotechnology* 37.4 (Apr. 2019), p. 451. DOI: 10.1038/s41587-019-0068-4.

[26] G. S. Gulati, S. S. Sikandar, D. J. Wesche, A. Manjunath, A. Bharadwaj, M. J. Berger, F. Ilagan, A. H. Kuo, R. W. Hsieh, S. Cai, M. Zabala, F. A. Scheeren, N. A. Lobo, D. Qian, F. B. Yu, F. M. Dirbas, M. F. Clarke, and A. M. Newman. "Single-cell transcriptional diversity is a hallmark of developmental potential". en. In: *Science* 367.6476 (Jan. 2020), pp. 405–411. DOI: 10.1126/science.aax0249.

[27] C. Trapnell, D. Cacchiarelli, J. Grimsby, P. Pokharel, S. Li, M. Morse, N. J. Lennon, K. J. Livak, T. S. Mikkelsen, and J. L. Rinn. "The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells". en. In: *Nature Biotechnology* 32.4 (Apr. 2014), pp. 381–386. DOI: 10.1038/nbt.2859.

[28] L. Penter, S. H. Gohil, C. Lareau, L. S. Ludwig, E. M. Parry, T. Huang, S. Li, W. Zhang, D. Livitz, I. Leshchiner, L. Parida, G. Getz, L. Z. Rassenti, T. J. Kipps, J. R. Brown, M. S. Davids, D. S. Neuberg, K. J. Livak, V. G. Sankaran, and C. J. Wu. "Longitudinal single-cell dynamics of chromatin accessibility and mitochondrial mutations in chronic lymphocytic leukemia mirror disease history". en. In: *Cancer Discovery* (June 2021), candisc.0276.2021. DOI: 10.1158/2159-8290.CD-21-0276.

[29] D. E. Cohen and D. Melton. "Turning straw into gold: directing cell fate for regenerative medicine". en. In: *Nature Reviews Genetics* 12.4 (Apr. 2011), pp. 243–252. DOI: 10.1038/nrg2938.

[30] F. H. Biase, X. Cao, and S. Zhong. "Cell fate inclination within 2-cell and 4-cell mouse embryos revealed by single-cell RNA sequencing". In: *Genome Research* 24.11 (Nov. 2014), pp. 1787–1796. DOI: 10.1101/gr.177725.114.

[31] S. Semrau, J. E. Goldmann, M. Soumillon, T. S. Mikkelsen, R. Jaenisch, and A. van Oudenaarden. "Dynamics of lineage commitment revealed by single-cell transcriptomics of differentiating embryonic stem cells". en. In: *Nature Communications* 8.1 (Oct. 2017), p. 1096. DOI: 10.1038/s41467-017-01076-4.

[32] M. B. Elowitz, A. J. Levine, E. D. Siggia, and P. S. Swain. "Stochastic Gene Expression in a Single Cell". In: *Science* 297.5584 (Aug. 2002), pp. 1183–1186. DOI: 10.1126/science.1070919.

[33] D. Bode, A. H. Cull, J. A. Rubio-Lara, and D. G. Kent. "Exploiting Single-Cell Tools in Gene and Cell Therapy". In: *Frontiers in Immunology* 12 (July 2021), p. 702636. DOI: 10.3389/fimmu.2021.702636.

[34] A. M. Klein, L. Mazutis, I. Akartuna, N. Tallapragada, A. Veres, V. Li, L. Peshkin, D. A. Weitz, and M. W. Kirschner. "Droplet barcoding for single cell transcriptomics applied to embryonic stem cells". In: *Cell* 161.5 (May 2015), pp. 1187–1201. DOI: 10.1016/j.cell.2015.04.044.

[35] E. Z. Macosko, A. Basu, R. Satija, J. Nemesh, K. Shekhar, M. Goldman, I. Tirosh, A. R. Bialas, N. Kamitaki, E. M. Martersteck, J. J. Trombetta, D. A. Weitz, J. R. Sanes, A. K. Shalek, A. Regev, and S. A. McCarroll. "Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets". English. In: *Cell* 161.5 (May 2015), pp. 1202–1214. DOI: 10.1016/j.cell.2015.05.002.

[36] M. D. Luecken, M. Büttner, K. Chaichoompu, A. Danese, M. Interlandi, M. F. Mueller, D. C. Strobl, L. Zappia, M. Dugas, M. Colomé-Tatché, and F. J. Theis. "Benchmarking atlas-level data integration in single-cell genomics". en. In: *Nature Methods* 19.1 (Jan. 2022), pp. 41–50. DOI: 10.1038/s41592-021-01336-8.

[37] D. S. Fischer, L. Dony, M. König, A. Moeed, L. Zappia, L. Heumos, S. Tritschler, O. Holmberg, H. Aliee, and F. J. Theis. "Sfaira accelerates data and model reuse in single cell genomics". In: *Genome Biology* 22 (Aug. 2021), p. 248. DOI: 10.1186/s13059-021-02452-6.

[38] M. Lotfollahi, M. Naghipourfar, M. D. Luecken, M. Khajavi, M. Büttner, M. Wagenstetter, Ž. Avsec, A. Gayoso, N. Yosef, M. Interlandi, S. Rybakov, A. V. Misharin, and F. J. Theis. "Mapping single-cell data to reference atlases by transfer learning". en. In: *Nature Biotechnology* (Aug. 2021), pp. 1–10. DOI: 10.1038/s41587-021-01001-7.

[39] M. Lotfollahi, F. A. Wolf, and F. J. Theis. "scGen predicts single-cell perturbation responses". En. In: *Nature Methods* 16.8 (Aug. 2019), p. 715. DOI: 10.1038/s41592-019-0494-8.

[40] M. Lotfollahi, M. Naghipourfar, F. J. Theis, and F. A. Wolf. "Conditional out-of-distribution generation for unpaired data using transfer VAE". en. In: *Bioinformatics* 36.Supplement_2 (Dec. 2020), pp. i610–i617. DOI: 10.1093/bioinformatics/btaa800.

[41] M. Lotfollahi, A. K. Susmelj, C. D. Donno, Y. Ji, I. L. Ibarra, F. A. Wolf, N. Yakubova, F. J. Theis, and D. Lopez-Paz. "Compositional perturbation autoencoder for single-cell response modeling". en. In: *bioRxiv* (Apr. 2021), p. 2021.04.14.439903. DOI: 10.1101/2021.04.14.439903.

[42] B. Adamson, T. M. Norman, M. Jost, M. Y. Cho, J. K. Nuñez, Y. Chen, J. E. Villalta, L. A. Gilbert, M. A. Horlbeck, M. Y. Hein, R. A. Pak, A. N. Gray, C. A. Gross, A. Dixit, O. Parnas, A. Regev, and J. S. Weissman. "A Multiplexed Single-Cell CRISPR Screening Platform Enables Systematic Dissection of the Unfolded Protein Response". eng. In: *Cell* 167.7 (Dec. 2016), 1867–1882.e21. DOI: 10.1016/j.cell.2016.11.048.

[43] A. Dixit, O. Parnas, B. Li, J. Chen, C. P. Fulco, L. Jerby-Arnon, N. D. Marjanovic, D. Dionne, T. Burks, R. Raychowdhury, B. Adamson, T. M. Norman, E. S. Lander, J. S. Weissman, N. Friedman, and A. Regev. "Perturb-Seq: Dissecting Molecular Circuits with Scalable Single-Cell RNA Profiling of Pooled Genetic Screens". en. In: *Cell* 167.7 (Dec. 2016), 1853–1866.e17. DOI: 10.1016/j.cell.2016.11.038.

[44] Y. Ji, M. Lotfollahi, F. A. Wolf, and F. J. Theis. "Machine learning for perturbational single-cell omics". en. In: *Cell Systems* 12.6 (June 2021), pp. 522–537. DOI: 10.1016/j.cels.2021.05.016.

[45] G. Palla, H. Spitzer, M. Klein, D. Fischer, A. C. Schaar, L. B. Kuemmerle, S. Rybakov, I. L. Ibarra, O. Holmberg, I. Virshup, M. Lotfollahi, S. Richter, and F. J. Theis. "Squidpy: a scalable framework for spatial omics analysis". en. In: *Nature Methods* 19.2 (Feb. 2022), pp. 171–178. DOI: 10.1038/s41592-021-01358-2.

[46] R. R. Stickels, E. Murray, P. Kumar, J. Li, J. L. Marshall, D. J. Di Bella, P. Arlotta, E. Z. Macosko, and F. Chen. "Highly sensitive spatial transcriptomics at near-cellular resolution with Slide-seqV2". en. In: *Nature Biotechnology* (Dec. 2020), pp. 1–7. DOI: 10.1038/s41587-020-0739-1.

[47] S. Vickovic, G. Eraslan, F. Salmén, J. Klughammer, L. Stenbeck, D. Schapiro, T. Äijö, R. Bonneau, L. Bergenstråhle, J. F. Navarro, J. Gould, G. K. Griffin, Å. Borg, M. Ronaghi, J. Frisén, J. Lundeberg, A. Regev, and P. L. Ståhl. "High-definition spatial transcriptomics for in situ tissue profiling". en. In: *Nature Methods* (Sept. 2019), pp. 1–4. DOI: 10.1038/s41592-019-0548-y.

[48] V. Svensson, S. A. Teichmann, and O. Stegle. "SpatialDE: identification of spatially variable genes". en. In: *Nature Methods* (Mar. 2018). DOI: 10.1038/nmeth.4636.

[49] P. Bachireddy, E. Azizi, C. Burdziak, V. N. Nguyen, C. S. Ennis, K. Maurer, C. Y. Park, Z.-N. Choo, S. Li, S. H. Gohil, N. G. Ruthen, Z. Ge, D. B. Keskin, N. Cieri, K. J. Livak, H. T. Kim, D. S. Neuberg, R. J. Soiffer, J. Ritz, E. P. Alyea, D. Pe'er, and C. J. Wu. "Mapping the evolution of T cell states during response and resistance to adoptive cellular therapy". English. In: *Cell Reports* 37.6 (Nov. 2021). DOI: 10.1016/j.celrep.2021.109992.

[50] E. Stephenson, G. Reynolds, R. A. Botting, F. J. Calero-Nieto, M. D. Morgan, Z. K. Tuong, K. Bach, W. Sungnak, K. B. Worlock, M. Yoshida, N. Kumasaka, K. Kania, J. Engelbert, B. Olabi, J. S. Spegarova, N. K. Wilson, N. Mende, L. Jardine, L. C. S. Gardner, I. Goh, D. Horsfall, J. McGrath, S. Webb, M. W. Mather, R. G. H. Lindeboom, E. Dann, N. Huang, K. Polanski, E. Prigmore, F. Gothe, J. Scott, R. P. Payne, K. F. Baker, A. T. Hanrath, I. C. D. Schim van der Loeff, A. S. Barr, A. Sanchez-Gonzalez, L. Bergamaschi, F. Mescia, J. L. Barnes, E. Kilich, A. de Wilton, A. Saigal, A. Saleh, S. M. Janes, C. M. Smith, N. Gopee, C. Wilson, P. Coupland, J. M. Coxhead, V. Y. Kiselev, S. van Dongen, J. Bacardit, H. W. King, A. J. Rostron, A. J. Simpson, S. Hambleton, E. Laurenti, P. A. Lyons, K. B. Meyer, M. Z. Nikolić, C. J. A. Duncan, K. G. C. Smith, S. A. Teichmann, M. R. Clatworthy, J. C. Marioni, B. Göttgens, and M. Haniffa. "Single-cell multi-omics analysis of the immune response in COVID-19". en. In: *Nature Medicine* 27.5 (May 2021), pp. 904–916. DOI: 10.1038/s41591-021-01329-2.

[51] R. Browaeys, W. Saelens, and Y. Saeys. "NicheNet: modeling intercellular communication by linking ligands to target genes". en. In: *Nature Methods* (Dec. 2019), pp. 1–4. DOI: 10.1038/s41592-019-0667-5.

[52] R. Vento-Tormo, M. Efremova, R. A. Botting, M. Y. Turco, M. Vento-Tormo, K. B. Meyer, J.-E. Park, E. Stephenson, K. Polański, A. Goncalves, L. Gardner, S. Holmqvist, J. Henriksson, A. Zou, A. M. Sharkey, B. Millar, B. Innes, L. Wood, A. Wilbrey-Clark, R. P. Payne, M. A. Ivarsson, S. Lisgo, A. Filby, D. H. Rowitch, J. N. Bulmer, G. J. Wright, M. J. T. Stubbington, M. Haniffa, A. Moffett, and S. A. Teichmann. "Single-cell reconstruction of the early maternal–fetal interface in humans". En. In: *Nature* 563.7731 (Nov. 2018), p. 347. DOI: 10.1038/s41586-018-0698-6.

[53] D. Türei, A. Valdeolivas, L. Gul, N. Palacio-Escat, M. Klein, O. Ivanova, M. Ölbei, A. Gábor, F. Theis, D. Módos, T. Korcsmáros, and J. Saez-Rodriguez. "Integrated intra- and intercellular signaling knowledge for multicellular omics analysis". eng. In: *Molecular Systems Biology* 17.3 (Mar. 2021), e9923. DOI: 10.15252/msb.20209923.

[54] K. Kamimoto, C. M. Hoffmann, and S. A. Morris. "CellOracle: Dissecting cell identity via network inference and in silico gene perturbation". en. In: *bioRxiv* (Mar. 2020), p. 2020.02.17.947416. DOI: 10.1101/2020.02.17.947416.

[55] S. Chen and J. C. Mar. "Evaluating methods of inferring gene regulatory networks highlights their lack of performance for single cell gene expression data". In: *BMC Bioinformatics* 19.1 (June 2018), p. 232. DOI: 10.1186/s12859-018-2217-z.

[56] S. Aibar, C. B. González-Blas, T. Moerman, V. A. Huynh-Thu, H. Imrichova, G. Hulselmans, F. Rambow, J.-C. Marine, P. Geurts, J. Aerts, J. van den Oord, Z. K. Atak, J. Wouters, and S. Aerts. "SCENIC: single-cell regulatory network inference and clustering". en. In: *Nature Methods* 14.11 (Nov. 2017), pp. 1083–1086. DOI: 10.1038/nmeth.4463.

[57] M. H. Spitzer and G. P. Nolan. "Mass Cytometry: Single Cells, Many Features". In: *Cell* 165.4 (May 2016), pp. 780–791. DOI: 10.1016/j.cell.2016.04.019.

[58] D. R. Bandura, V. I. Baranov, O. I. Ornatsky, A. Antonov, R. Kinach, X. Lou, S. Pavlov, S. Vorobiev, J. E. Dick, and S. D. Tanner. "Mass cytometry: technique for real time single cell multitarget immunoassay based on inductively coupled plasma time-of-flight mass spectrometry". eng. In: *Analytical Chemistry* 81.16 (Aug. 2009), pp. 6813–6822. DOI: 10.1021/ac901049w.

[59] A. Rotem, O. Ram, N. Shoresh, R. A. Sperling, A. Goren, D. A. Weitz, and B. E. Bernstein. "Single-cell ChIP-seq reveals cell subpopulations defined by chromatin state". en. In: *Nature Biotechnology* 33.11 (Nov. 2015), pp. 1165–1172. DOI: 10.1038/nbt.3383.

[60] K. Grosselin, A. Durand, J. Marsolier, A. Poitou, E. Marangoni, F. Nemati, A. Dahmani, S. Lameiras, F. Reyal, O. Frenoy, Y. Pousse, M. Reichen, A. Woolfe, C. Brenan, A. D. Griffiths, C. Vallot, and A. Gérard. "High-throughput single-cell ChIP-seq identifies heterogeneity of chromatin states in breast cancer". en. In: *Nature Genetics* 51.6 (June 2019), pp. 1060–1066. DOI: 10.1038/s41588-019-0424-9.

[61] M. Bartosovic, M. Kabbe, and G. Castelo-Branco. "Single-cell CUT&Tag profiles histone modifications and transcription factors in complex tissues". en. In: *Nature Biotechnology* (Apr. 2021), pp. 1–11. DOI: 10.1038/s41587-021-00869-9.

[62] S. J. Wu, S. N. Furlan, A. B. Mihalas, H. S. Kaya-Okur, A. H. Feroze, S. N. Emerson, Y. Zheng, K. Carson, P. J. Cimino, C. D. Keene, J. F. Sarthy, R. Gottardo, K. Ahmad, S. Henikoff, and A. P. Patel. "Single-cell CUT&Tag analysis of chromatin modifications in differentiation and tumor progression". en. In: *Nature Biotechnology* (Apr. 2021), pp. 1–6. DOI: 10.1038/s41587-021-00865-z.

[63] H. S. Kaya-Okur, S. J. Wu, C. A. Codomo, E. S. Pledger, T. D. Bryson, J. G. Henikoff, K. Ahmad, and S. Henikoff. "CUT&Tag for efficient epigenomic profiling of small samples and single cells". en. In: *Nature Communications* 10.1 (Apr. 2019), p. 1930. DOI: 10.1038/s41467-019-09982-5.

[64] S. J. Clark, S. A. Smallwood, H. J. Lee, F. Krueger, W. Reik, and G. Kelsey. "Genome-wide base-resolution mapping of DNA methylation in single cells using single-cell bisulfite sequencing (scBS-seq)". en. In: *Nature Protocols* 12.3 (Mar. 2017), pp. 534–547. DOI: 10.1038/nprot.2016.187.

[65] C. Angermueller, S. J. Clark, H. J. Lee, I. C. Macaulay, M. J. Teng, T. X. Hu, F. Krueger, S. A. Smallwood, C. P. Ponting, T. Voet, G. Kelsey, O. Stegle, and W. Reik. "Parallel single-cell sequencing links transcriptional and epigenetic heterogeneity". en. In: *Nature Methods* 13.3 (Mar. 2016), pp. 229–232. DOI: 10.1038/nmeth.3728.

[66] S. A. Smallwood, H. J. Lee, C. Angermueller, F. Krueger, H. Saadeh, J. Peat, S. R. Andrews, O. Stegle, W. Reik, and G. Kelsey. "Single-cell genome-wide bisulfite sequencing for assessing epigenetic heterogeneity". en. In: *Nature Methods* 11.8 (Aug. 2014), pp. 817–820. DOI: 10.1038/nmeth.3035.

[67] J. D. Buenrostro, B. Wu, U. M. Litzenburger, D. Ruff, M. L. Gonzales, M. P. Snyder, H. Y. Chang, and W. J. Greenleaf. "Single-cell chromatin accessibility reveals principles of regulatory variation". en. In: *Nature* 523.7561 (July 2015), pp. 486–490. DOI: 10.1038/nature14590.

[68] M. Stoeckius, S. Zheng, B. Houck-Loomis, S. Hao, B. Z. Yeung, W. M. Mauck, P. Smibert, and R. Satija. "Cell Hashing with barcoded antibodies enables multiplexing and doublet detection for single cell genomics". In: *Genome Biology* 19.1 (Dec. 2018), p. 224. DOI: 10.1186/s13059-018-1603-1.

[69] V. M. Peterson, K. X. Zhang, N. Kumar, J. Wong, L. Li, D. C. Wilson, R. Moore, T. K. McClanahan, S. Sadekova, and J. A. Klappenbach. "Multiplexed quantification of proteins and transcripts in single cells". en. In: *Nature Biotechnology* 35.10 (Aug. 2017), pp. 936–939. DOI: 10.1038/nbt.3973.

[70] S. Ma, B. Zhang, L. M. LaFave, A. S. Earl, Z. Chiang, Y. Hu, J. Ding, A. Brack, V. K. Kartha, T. Tay, T. Law, C. Lareau, Y.-C. Hsu, A. Regev, and J. D. Buenrostro. "Chromatin Potential Identified by Shared Single-Cell Profiling of RNA and Chromatin". en. In: *Cell* 183.4 (Nov. 2020), 1103–1116.e20. DOI: 10.1016/j.cell.2020.09.056.

[71] C. Zhu, M. Yu, H. Huang, I. Juric, A. Abnousi, R. Hu, J. Lucero, M. M. Behrens, M. Hu, and B. Ren. "An ultra high-throughput method for single-cell joint analysis of open chromatin and transcriptome". en. In: *Nature Structural & Molecular Biology* 26.11 (Nov. 2019), pp. 1063–1070. DOI: 10.1038/s41594-019-0323-x.

[72] J. Cao, D. A. Cusanovich, V. Ramani, D. Aghamirzaie, H. A. Pliner, A. J. Hill, R. M. Daza, J. L. McFaline-Figueroa, J. S. Packer, L. Christiansen, F. J. Steemers, A. C. Adey, C. Trapnell, and J. Shendure. "Joint profiling of chromatin accessibility and gene expression in thousands of single cells". en. In: *Science* (Aug. 2018), eaau0730. DOI: 10.1126/science.aau0730.

[73] S. Chen, B. B. Lake, and K. Zhang. "High-throughput sequencing of the transcriptome and chromatin accessibility in the same cell". en. In: *Nature Biotechnology* 37.12 (Dec. 2019), pp. 1452–1457. DOI: 10.1038/s41587-019-0290-0.

[74] E. P. Mimitou, C. A. Lareau, K. Y. Chen, A. L. Zorzetto-Fernandes, Y. Hao, Y. Takeshima, W. Luo, T.-S. Huang, B. Z. Yeung, E. Papalexi, P. I. Thakore, T. Kibayashi, J. B. Wing, M. Hata, R. Satija, K. L. Nazor, S. Sakaguchi, L. S. Ludwig, V. G. Sankaran, A. Regev, and P. Smibert. "Scalable, multimodal profiling of chromatin accessibility, gene expression and protein levels in single cells". en. In: *Nature Biotechnology* (June 2021), pp. 1–13. DOI: 10.1038/s41587-021-00927-2.

[75] T. Stuart and R. Satija. "Integrative single-cell analysis". En. In: *Nature Reviews Genetics* (Jan. 2019), p. 1. DOI: 10.1038/s41576-019-0093-7.

[76] G. X. Y. Zheng, J. M. Terry, P. Belgrader, P. Ryvkin, Z. W. Bent, R. Wilson, S. B. Ziraldo, T. D. Wheeler, G. P. McDermott, J. Zhu, M. T. Gregory, J. Shuga, L. Montesclaros, J. G. Underwood, D. A. Masquelier, S. Y. Nishimura, M. Schnall-Levin, P. W. Wyatt, C. M. Hindson, R. Bharadwaj, A. Wong, K. D. Ness, L. W. Beppu, H. J. Deeg, C. McFarland, K. R. Loeb, W. J. Valente, N. G. Ericson, E. A. Stevens, J. P. Radich, T. S. Mikkelsen, B. J. Hindson, and J. H. Bielas. "Massively parallel digital transcriptional profiling of single cells". en. In: *Nature Communications* 8 (Jan. 2017), p. 14049. DOI: 10.1038/ncomms14049.

[77] C. Ziegenhain, B. Vieth, S. Parekh, I. Hellmann, and W. Enard. "Quantitative single-cell transcriptomics". en. In: *Briefings in Functional Genomics* (2018). DOI: 10.1093/bfgp/ely009.

[78] D. Grün, A. Lyubimova, L. Kester, K. Wiebrands, O. Basak, N. Sasaki, H. Clevers, and A. van Oudenaarden. "Single-cell messenger RNA sequencing reveals rare intestinal cell types". en. In: *Nature* 525.7568 (Sept. 2015), pp. 251–255. DOI: 10.1038/nature14966.

[79] M. Plass, J. Solana, F. A. Wolf, S. Ayoub, A. Misios, P. Glažar, B. Obermayer, F. J. Theis, C. Kocks, and N. Rajewsky. "Cell type atlas and lineage tree of a whole complex animal by single-cell transcriptomics". In: *Science* 360.6391 (May 2018), eaaq1723. DOI: 10.1126/science.aaq1723.

[80] D. P. Bartel. "Metazoan MicroRNAs". English. In: *Cell* 173.1 (Mar. 2018), pp. 20–51. DOI: 10.1016/j.cell.2018.03.006.

[81] S. Picelli, Å. K. Björklund, O. R. Faridani, S. Sagasser, G. Winberg, and R. Sandberg. "Smart-seq2 for sensitive full-length transcriptome profiling in single cells". en. In: *Nature Methods* 10.11 (Nov. 2013), pp. 1096–1098. DOI: 10.1038/nmeth.2639.

[82] M. Hagemann-Jensen, C. Ziegenhain, P. Chen, D. Ramsköld, G.-J. Hendriks, A. J. M. Larsson, O. R. Faridani, and R. Sandberg. "Single-cell RNA counting at allele and isoform resolution using Smart-seq3". en. In: *Nature Biotechnology* 38.6 (June 2020), pp. 708–714. DOI: 10.1038/s41587-020-0497-0.

[83] D. A. Jaitin, E. Kenigsberg, H. Keren-Shaul, N. Elefant, F. Paul, I. Zaretsky, A. Mildner, N. Cohen, S. Jung, A. Tanay, and I. Amit. "Massively Parallel Single-Cell RNA-Seq for Marker-Free Decomposition of Tissues into Cell Types". en. In: *Science* 343.6172 (Feb. 2014), pp. 776–779. DOI: 10.1126/science.1247651.

[84] J. Cao, J. S. Packer, V. Ramani, D. A. Cusanovich, C. Huynh, R. Daza, X. Qiu, C. Lee, S. N. Furlan, F. J. Steemers, A. Adey, R. H. Waterston, C. Trapnell, and J. Shendure. "Comprehensive single-cell transcriptional profiling of a multicellular organism". eng. In: *Science (New York, N.Y.)* 357.6352 (Aug. 2017), pp. 661–667. DOI: 10.1126/science.aam8940.

[85] A. B. Rosenberg, C. M. Roco, R. A. Muscat, A. Kuchina, P. Sample, Z. Yao, L. T. Graybuck, D. J. Peeler, S. Mukherjee, W. Chen, S. H. Pun, D. L. Sellers, B. Tasic, and G. Seelig. "Single-cell profiling of the developing mouse brain and spinal cord with split-pool barcoding". eng. In: *Science (New York, N.Y.)* 360.6385 (Apr. 2018), pp. 176–182. DOI: 10.1126/science.aam8999.

[86] P. Datlinger, A. F. Rendeiro, T. Boenke, M. Senekowitsch, T. Krausgruber, D. Barreca, and C. Bock. "Ultra-high-throughput single-cell RNA sequencing and perturbation screening with combinatorial fluidic indexing". en. In: *Nature Methods* 18.6 (June 2021), pp. 635–642. DOI: 10.1038/s41592-021-01153-z.

[87] S. Islam, A. Zeisel, S. Joost, G. La Manno, P. Zajac, M. Kasper, P. Lönnerberg, and S. Linnarsson. "Quantitative single-cell RNA-seq with unique molecular identifiers". en. In: *Nature Methods* 11.2 (Feb. 2014), pp. 163–166. DOI: 10.1038/nmeth.2772.

[88] C. Ziegenhain, B. Vieth, S. Parekh, B. Reinius, A. Guillaumet-Adkins, M. Smets, H. Leonhardt, H. Heyn, I. Hellmann, and W. Enard. "Comparative Analysis of Single-Cell RNA Sequencing Methods". In: *Molecular Cell* 65.4 (Feb. 2017), 631–643.e4. DOI: 10.1016/j.molcel.2017.01.023.

[89] X. Zhang, T. Li, F. Liu, Y. Chen, J. Yao, Z. Li, Y. Huang, and J. Wang. "Comparative Analysis of Droplet-Based Ultra-High-Throughput Single-Cell RNA-Seq Systems". eng. In: *Molecular Cell* 73.1 (Jan. 2019), 130–142.e5. DOI: 10.1016/j.molcel.2018.10.020.

[90] V. Svensson, K. N. Natarajan, L.-H. Ly, R. J. Miragaia, C. Labalette, I. C. Macaulay, A. Cvejic, and S. A. Teichmann. "Power analysis of single-cell RNA-sequencing experiments". en. In: *Nature Methods* 14.4 (Apr. 2017), pp. 381–387. DOI: 10.1038/nmeth.4220.

[91] E. Mereu, A. Lafzi, C. Moutinho, C. Ziegenhain, D. J. McCarthy, A. Álvarez-Varela, E. Batlle, Sagar, D. Grün, J. K. Lau, S. C. Boutet, C. Sanada, A. Ooi, R. C. Jones, K. Kaihara, C. Brampton, Y. Talaga, Y. Sasagawa, K. Tanaka, T. Hayashi, C. Braeuning, C. Fischer, S. Sauer, T. Trefzer, C. Conrad, X. Adiconis, L. T. Nguyen, A. Regev, J. Z. Levin, S. Parekh, A. Janjic, L. E. Wange, J. W. Bagnoli, W. Enard, M. Gut, R. Sandberg, I. Nikaido, I. Gut, O. Stegle, and H. Heyn. "Benchmarking single-cell RNA-sequencing protocols for cell atlas projects". en. In: *Nature Biotechnology* 38.6 (June 2020), pp. 747–755. DOI: 10.1038/s41587-020-0469-4.

[92] J. Cao, M. Spielmann, X. Qiu, X. Huang, D. M. Ibrahim, A. J. Hill, F. Zhang, S. Mundlos, L. Christiansen, F. J. Steemers, C. Trapnell, and J. Shendure. "The single-cell transcriptional landscape of mammalian organogenesis". En. In: *Nature* (Feb. 2019), p. 1. DOI: 10.1038/s41586-019-0969-x.

[93] Y. Sasagawa, H. Danno, H. Takada, M. Ebisawa, K. Tanaka, T. Hayashi, A. Kurisaki, and I. Nikaido. "Quartz-Seq2: a high-throughput single-cell RNA-sequencing method that effectively uses limited sequence reads". In: *Genome Biology* 19.1 (Mar. 2018), p. 29. DOI: 10.1186/s13059-018-1407-3.

[94] A. Wagner, A. Regev, and N. Yosef. "Revealing the vectors of cellular identity with single-cell genomics". en. In: *Nature Biotechnology* 34.11 (Nov. 2016), pp. 1145–1160. DOI: 10.1038/nbt.3711.

[95] C. A. Vallejos, D. Risso, A. Scialdone, S. Dudoit, and J. C. Marioni. "Normalizing single-cell RNA sequencing data: challenges and opportunities". en. In: *Nature Methods* 14.6 (June 2017), pp. 565–571. DOI: 10.1038/nmeth.4292.

[96] R. Lopez, J. Regier, M. B. Cole, M. I. Jordan, and N. Yosef. "Deep generative modeling for single-cell transcriptomics". en. In: *Nature Methods* 15.12 (Dec. 2018), pp. 1053–1058. DOI: 10.1038/s41592-018-0229-2.

[97] L. Amrhein, K. Harsha, and C. Fuchs. "A mechanistic model for the negative binomial distribution of single-cell mRNA counts". en. In: *bioRxiv* (June 2019), p. 657619. DOI: 10.1101/657619.

[98] G. Gorin and L. Pachter. *Direct simulation of a stochastically driven multi-step birth-death process*. en. Tech. rep. bioRxiv, Mar. 2021, p. 2021.01.20.427480. DOI: 10.1101/2021.01.20.427480.

[99] A. Sarkar and M. Stephens. "Separating measurement and expression models clarifies confusion in single-cell RNA sequencing analysis". en. In: *Nature Genetics* 53.6 (June 2021), pp. 770–777. DOI: 10.1038/s41588-021-00873-4.

[100] P. V. Kharchenko, L. Silberstein, and D. T. Scadden. "Bayesian approach to single-cell differential expression analysis". en. In: *Nature Methods* 11.7 (July 2014), pp. 740–742. DOI: 10.1038/nmeth.2967.

[101] G. Finak, A. McDavid, M. Yajima, J. Deng, V. Gersuk, A. K. Shalek, C. K. Slichter, H. W. Miller, M. J. McElrath, M. Prlic, P. S. Linsley, and R. Gottardo. "MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data". en. In: *Genome Biology* 16.1 (Dec. 2015). DOI: 10.1186/s13059-015-0844-5.

[102] E. Pierson and C. Yau. "ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis". In: *Genome Biology* 16.1 (Nov. 2015), p. 241. DOI: 10.1186/s13059-015-0805-z.

[103] P. Lin, M. Troup, and J. W. K. Ho. "CIDR: Ultrafast and accurate clustering through imputation for single-cell RNA-seq data". In: *Genome Biology* 18.1 (Mar. 2017), p. 59. DOI: 10.1186/s13059-017-1188-0.

[104] D. van Dijk, R. Sharma, J. Nainys, K. Yim, P. Kathail, A. J. Carr, C. Burdziak, K. R. Moon, C. L. Chaffer, D. Pattabiraman, B. Bierie, L. Mazutis, G. Wolf, S. Krishnaswamy, and D. Pe'er. "Recovering Gene Interactions from Single-Cell Data Using Data Diffusion". en. In: *Cell* 174.3 (July 2018), 716–729.e27. DOI: 10.1016/j.cell.2018.05.061.

[105] M. Huang, J. Wang, E. Torre, H. Dueck, S. Shaffer, R. Bonasio, J. I. Murray, A. Raj, M. Li, and N. R. Zhang. "SAVER: gene expression recovery for single-cell RNA sequencing". en. In: *Nature Methods* 15.7 (July 2018), pp. 539–542. DOI: 10.1038/s41592-018-0033-z.

[106] D. Risso, F. Perraudeau, S. Gribkova, S. Dudoit, and J.-P. Vert. "A general and flexible method for signal extraction from single-cell RNA-seq data". en. In: *Nature Communications* 9.1 (Jan. 2018), p. 284. DOI: 10.1038/s41467-017-02554-5.

[107] B. Vieth, C. Ziegenhain, S. Parekh, W. Enard, and I. Hellmann. "powsimR: power analysis for bulk and single cell RNA-seq experiments". eng. In: *Bioinformatics (Oxford, England)* 33.21 (Nov. 2017), pp. 3486–3488. DOI: 10.1093/bioinformatics/btx435.

[108] W. Tang, F. Bertaux, P. Thomas, C. Stefanelli, M. Saint, S. Marguerat, and V. Shahrezaei. "bayNorm: Bayesian gene expression recovery, imputation and normalization for single-cell RNA-sequencing data". In: *Bioinformatics* 36.4 (Feb. 2020), pp. 1174–1181. DOI: 10.1093/bioinformatics/btz726.

[109] W. Chen, Y. Li, J. Easton, D. Finkelstein, G. Wu, and X. Chen. "UMI-count modeling and differential expression analysis for single-cell RNA sequencing". In: *Genome Biology* 19.1 (May 2018), p. 70. DOI: 10.1186/s13059-018-1438-9.

[110] F. W. Townes, S. C. Hicks, M. J. Aryee, and R. A. Irizarry. "Feature selection and dimension reduction for single-cell RNA-Seq based on a multinomial model". In: *Genome Biology* 20.1 (Dec. 2019), p. 295. DOI: 10.1186/s13059-019-1861-6.

[111] V. Svensson. "Droplet scRNA-seq is not zero-inflated". en. In: *Nature Biotechnology* 38.2 (Feb. 2020), pp. 147–150. DOI: 10.1038/s41587-019-0379-5.

[112] Y. Cao, S. Kitanovski, R. Küppers, and D. Hoffmann. "UMI or not UMI, that is the question for scRNA-seq zero-inflation". en. In: *Nature Biotechnology* (Feb. 2021), pp. 1–2. DOI: 10.1038/s41587-020-00810-6.

[113] V. Svensson. "Reply to: UMI or not UMI, that is the question for scRNA-seq zero-inflation". en. In: *Nature Biotechnology* (Feb. 2021), pp. 1–1. DOI: 10.1038/s41587-020-00811-5.

[114] G. Eraslan, Ž. Avsec, J. Gagneur, and F. J. Theis. "Deep learning: new computational modelling techniques for genomics". En. In: *Nature Reviews Genetics* (Apr. 2019), p. 1. DOI: 10.1038/s41576-019-0122-6.

[115] F. A. Wolf, P. Angerer, and F. J. Theis. "SCANPY: large-scale single-cell gene expression data analysis". eng. In: *Genome Biology* 19.1 (2018), p. 15. DOI: 10.1186/s13059-017-1382-0.

[116] R. Satija, J. A. Farrell, D. Gennert, A. F. Schier, and A. Regev. "Spatial reconstruction of single-cell gene expression data". en. In: *Nature Biotechnology* 33.5 (May 2015), pp. 495–502. DOI: 10.1038/nbt.3192.

[117] T. Stuart, A. Butler, P. Hoffman, C. Hafemeister, E. Papalexi, W. M. Mauck, Y. Hao, M. Stoeckius, P. Smibert, and R. Satija. "Comprehensive Integration of Single-Cell Data". en. In: *Cell* 177.7 (June 2019), 1888–1902.e21. DOI: 10.1016/j.cell.2019.05.031.

[118] Y. Hao, S. Hao, E. Andersen-Nissen, W. M. Mauck, S. Zheng, A. Butler, M. J. Lee, A. J. Wilk, C. Darby, M. Zager, P. Hoffman, M. Stoeckius, E. Papalexi, E. P. Mimitou, J. Jain, A. Srivastava, T. Stuart, L. M. Fleming, B. Yeung, A. J. Rogers, J. M. McElrath, C. A. Blish, R. Gottardo, P. Smibert, and R. Satija. "Integrated analysis of multimodal single-cell data". English. In: *Cell* 0.0 (May 2021). DOI: 10.1016/j.cell.2021.04.048.

[119] A. Butler, P. Hoffman, P. Smibert, E. Papalexi, and R. Satija. "Integrating single-cell transcriptomic data across different conditions, technologies, and species". en. In: *Nature Biotechnology* 36.5 (May 2018), pp. 411–420. DOI: 10.1038/nbt.4096.

[120] D. P. Kingma and M. Welling. "Auto-Encoding Variational Bayes". In: *arXiv:1312.6114 [cs, stat]* (May 2014).

[121] L. McInnes and J. Healy. "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction". In: *arXiv:1802.03426 [cs, stat]* (Feb. 2018).

[122] E. Becht, L. McInnes, J. Healy, C.-A. Dutertre, I. W. H. Kwok, L. G. Ng, F. Ginhoux, and E. W. Newell. "Dimensionality reduction for visualizing single-cell data using UMAP". en. In: *Nature Biotechnology* (Dec. 2018). DOI: 10.1038/nbt.4314.

[123] M. S. Jain, K. Polanski, C. D. Conde, X. Chen, J. Park, L. Mamanova, A. Knights, R. A. Botting, E. Stephenson, M. Haniffa, A. Lamacraft, M. Efremova, and S. A. Teichmann. "MultiMAP: dimensionality reduction and integration of multimodal data". In: *Genome Biology* 22.1 (Dec. 2021), p. 346. DOI: 10.1186/s13059-021-02565-y.

[124] L. v. d. Maaten and G. Hinton. "Visualizing Data using t-SNE". In: *Journal of Machine Learning Research* 9.Nov (2008), pp. 2579–2605.

[125] E.-a. D. Amir, K. L. Davis, M. D. Tadmor, E. F. Simonds, J. H. Levine, S. C. Bendall, D. K. Shenfeld, S. Krishnaswamy, G. P. Nolan, and D. Pe'er. "viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia". In: *Nature biotechnology* 31.6 (June 2013), pp. 545–552. DOI: 10.1038/nbt.2594.

[126] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker. "Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps". en. In: *Proceedings of the National Academy of Sciences of the United States of America* 102.21 (May 2005), pp. 7426–7431. DOI: 10.1073/pnas.0500334102.

[127] R. R. Coifman and S. Lafon. "Diffusion maps". In: *Applied and Computational Harmonic Analysis*. Special Issue: Diffusion Maps and Wavelets 21.1 (July 2006), pp. 5–30. DOI: 10.1016/j.acha.2006.04.006.

[128] L. Haghverdi, F. Buettner, and F. J. Theis. "Diffusion maps for high-dimensional single-cell analysis of differentiation data". eng. In: *Bioinformatics (Oxford, England)* 31.18 (Sept. 2015), pp. 2989–2998. DOI: 10.1093/bioinformatics/btv325.

[129] L. Zappia and F. J. Theis. "Over 1000 tools reveal trends in the single-cell RNA-seq analysis landscape". In: *Genome Biology* 22.1 (Oct. 2021), p. 301. DOI: 10.1186/s13059-021-02519-4.

[130] M. D. Luecken and F. J. Theis. "Current best practices in single-cell RNA-seq analysis: a tutorial". In: *Molecular Systems Biology* 15.6 (June 2019), e8746. DOI: 10.15252/msb.20188746.

[131] B. Vieth, S. Parekh, C. Ziegenhain, W. Enard, and I. Hellmann. "A systematic evaluation of single cell RNA-seq analysis pipelines". en. In: *Nature Communications* 10.1 (Oct. 2019), pp. 1–11. DOI: 10.1038/s41467-019-12266-7.

[132] K. P. Murphy. *Machine learning: a probabilistic perspective*. Adaptive computation and machine learning series. Cambridge, MA: MIT Press, 2012.

[133] R. Bacher, L.-F. Chu, N. Leng, A. P. Gasch, J. A. Thomson, R. M. Stewart, M. Newton, and C. Kendziorski. "SCnorm: robust normalization of single-cell RNA-seq data". en. In: *Nature Methods* 14.6 (June 2017), pp. 584–586. DOI: 10.1038/nmeth.4263.

[134] J. Breda, M. Zavolan, and E. van Nimwegen. "Bayesian inference of gene expression states from single-cell RNA-seq data". en. In: *Nature Biotechnology* (Apr. 2021), pp. 1–9. DOI: 10.1038/s41587-021-00875-x.

[135] C. Hafemeister and R. Satija. "Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression". In: *Genome Biology* 20.1 (Dec. 2019), p. 296. DOI: 10.1186/s13059-019-1874-1.

[136] L. Haghverdi, A. T. L. Lun, M. D. Morgan, and J. C. Marioni. "Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors". en. In: *Nature Biotechnology* 36.5 (May 2018), pp. 421–427. DOI: 10.1038/nbt.4091.

[137] I. Korsunsky, N. Millard, J. Fan, K. Slowikowski, F. Zhang, K. Wei, Y. Baglaenko, M. Brenner, P.-r. Loh, and S. Raychaudhuri. "Fast, sensitive and accurate integration of single-cell data with Harmony". en. In: *Nature Methods* 16.12 (Dec. 2019), pp. 1289–1296. DOI: 10.1038/s41592-019-0619-0.

[138] H. T. N. Tran, K. S. Ang, M. Chevrier, X. Zhang, N. Y. S. Lee, M. Goh, and J. Chen. "A benchmark of batch-effect correction methods for single-cell RNA sequencing data". en. In: *Genome Biology* 21.1 (Dec. 2020), p. 12. DOI: 10.1186/s13059-019-1850-9.

[139] W. E. Johnson, C. Li, and A. Rabinovic. "Adjusting batch effects in microarray expression data using empirical Bayes methods". en. In: *Biostatistics* 8.1 (Jan. 2007), pp. 118–127. DOI: 10.1093/biostatistics/kxj037.

[140] B. Hie, B. Bryson, and B. Berger. "Efficient integration of heterogeneous single-cell transcriptomes using Scanorama". en. In: *Nature Biotechnology* 37.6 (June 2019), pp. 685–691. DOI: 10.1038/s41587-019-0113-3.

[141] K. Polański, M. D. Young, Z. Miao, K. B. Meyer, S. A. Teichmann, and J.-E. Park. "BBKNN: fast batch alignment of single cell transcriptomes". In: *Bioinformatics* 36.3 (Feb. 2020), pp. 964–965. DOI: 10.1093/bioinformatics/btz625.

[142] A. S. Booeshaghi and L. Pachter. "Normalization of single-cell RNA-seq counts by log(x + 1) or log(1 + x)". In: *Bioinformatics* 37.15 (Aug. 2021), pp. 2223–2224. DOI: 10.1093/bioinformatics/btab085.

[143] D. P. Kingma and M. Welling. "An Introduction to Variational Autoencoders". In: *Foundations and Trends® in Machine Learning* 12.4 (2019), pp. 307–392. DOI: 10.1561/2200000056.

[144] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. "Variational Inference: A Review for Statisticians". In: *Journal of the American Statistical Association* 112.518 (Apr. 2017), pp. 859–877. DOI: 10.1080/01621459.2017.1285773.

[145] A. Gayoso, Z. Steier, R. Lopez, J. Regier, K. L. Nazor, A. Streets, and N. Yosef. "Joint probabilistic modeling of single-cell multi-omic data with totalVI". en. In: *Nature Methods* (Feb. 2021), pp. 1–11. DOI: 10.1038/s41592-020-01050-x.

[146] D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". In: *arXiv:1412.6980 [cs]* (Jan. 2017).

[147] V. Svensson, A. Gayoso, N. Yosef, and L. Pachter. "Interpretable factor models of single-cell RNA-seq via variational autoencoders". In: *Bioinformatics* 36.11 (June 2020), pp. 3418–3421. DOI: 10.1093/bioinformatics/btaa169.

[148] S. Rybakov, M. Lotfollahi, F. J. Theis, and F. A. Wolf. "Learning interpretable latent autoencoder representations with annotations of feature sets". en. In: *bioRxiv* (Dec. 2020), p. 2020.12.02.401182. DOI: 10.1101/2020.12.02.401182.

[149] C. Xu, R. Lopez, E. Mehlman, J. Regier, M. I. Jordan, and N. Yosef. "Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models". eng. In: *Molecular Systems Biology* 17.1 (Jan. 2021), e9620. DOI: 10.15252/msb.20209620.

[150] T. Stuart, A. Srivastava, S. Madad, C. A. Lareau, and R. Satija. "Single-cell chromatin state analysis with Signac". en. In: *Nature Methods* 18.11 (Nov. 2021), pp. 1333–1341. DOI: 10.1038/s41592-021-01282-5.

[151] A. Danese, M. L. Richter, K. Chaichoompu, D. S. Fischer, F. J. Theis, and M. Colomé-Tatché. "EpiScanpy: integrated single-cell epigenomic analysis". en. In: *Nature Communications* 12.1 (Sept. 2021), p. 5228. DOI: 10.1038/s41467-021-25131-3.

[152] J. M. Granja, M. R. Corces, S. E. Pierce, S. T. Bagdatli, H. Choudhry, H. Y. Chang, and W. J. Greenleaf. "ArchR is a scalable software package for integrative single-cell chromatin accessibility analysis". en. In: *Nature Genetics* 53.3 (Mar. 2021), pp. 403–411. DOI: 10.1038/s41588-021-00790-6.

[153] T. Ashuach, D. A. Reidenbach, A. Gayoso, and N. Yosef. "PeakVI: A deep generative model for single-cell chromatin accessibility analysis". en. In: *Cell Reports Methods* 2.3 (Mar. 2022), p. 100182. DOI: 10.1016/j.crmeth.2022.100182.

[154] D. Bredikhin, I. Kats, and O. Stegle. "MUON: multimodal omics analysis framework". In: *Genome Biology* 23.1 (Feb. 2022), p. 42. DOI: 10.1186/s13059-021-02577-8.

[155] M. Stoeckius, C. Hafemeister, W. Stephenson, B. Houck-Loomis, P. K. Chattopadhyay, H. Swerdlow, R. Satija, and P. Smibert. "Simultaneous epitope and transcriptome measurement in single cells". en. In: *Nature Methods* 14.9 (Sept. 2017), pp. 865–868. DOI: 10.1038/nmeth.4380.

[156] T. Ashuach, M. I. Gabitto, M. I. Jordan, and N. Yosef. *MultiVI: deep generative model for the integration of multi-modal data*. en. Tech. rep. bioRxiv, Sept. 2021, p. 2021.08.20.457057. DOI: 10.1101/2021.08.20.457057.

[157] M. Lotfollahi, A. Litinetskaya, and F. J. Theis. *Multigrate: single-cell multi-omic data integration*. en. Tech. rep. bioRxiv, Mar. 2022, p. 2022.03.16.484643. DOI: 10.1101/2022.03.16.484643.

[158] D. I. Spivak. "METRIC REALIZATION OF FUZZY SIMPLICIAL SETS". en. In: *Preprint* (2009), p. 4.

[159] Y. A. Malkov and D. A. Yashunin. "Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs". In: *arXiv:1603.09320 [cs]* (Aug. 2018).

[160] W. Dong, C. Moses, and K. Li. "Efficient k-nearest neighbor graph construction for generic similarity measures". In: *Proceedings of the 20th international conference on World wide web*. WWW '11. New York, NY, USA: Association for Computing Machinery, Mar. 2011, pp. 577–586. DOI: 10.1145/1963405.1963487.

[161] T. Chari, J. Banerjee, and L. Pachter. *The Specious Art of Single-Cell Genomics*. en. Tech. rep. bioRxiv, Sept. 2021, p. 2021.08.25.457696. DOI: 10.1101/2021.08.25.457696.

[162] C. N. Heiser and K. S. Lau. "A Quantitative Framework for Evaluating Single-Cell Data Structure Preservation by Dimensionality Reduction Techniques". eng. In: *Cell Reports* 31.5 (May 2020), p. 107576. DOI: 10.1016/j.celrep.2020.107576.

[163] D. Kobak and G. C. Linderman. "Initialization is critical for preserving global data structure in both t -SNE and UMAP". en. In: *Nature Biotechnology* (Feb. 2021), pp. 1–2. DOI: 10.1038/s41587-020-00809-z.

[164] S. M. Cooley, T. Hamilton, E. J. Deeds, and J. C. J. Ray. "A novel metric reveals previously unrecognized distortion in dimensionality reduction of scRNA-Seq data". en. In: *bioRxiv* (July 2019), p. 689851. DOI: 10.1101/689851.

[165] V. Y. Kiselev, K. Kirschner, M. T. Schaub, T. Andrews, A. Yiu, T. Chandra, K. N. Natarajan, W. Reik, M. Barahona, A. R. Green, and M. Hemberg. "SC3: consensus clustering of single-cell RNA-seq data". en. In: *Nature Methods* 14.5 (May 2017), pp. 483–486. DOI: 10.1038/nmeth.4236.

[166] J. Chen, A. Schlitzer, S. Chakarov, F. Ginhoux, and M. Poidinger. "Mpath maps multi-branching single-cell trajectories revealing progenitor cell progression during development". en. In: *Nature Communications* 7.1 (June 2016), p. 11988. DOI: 10.1038/ncomms11988.

[167] B. Tasic, V. Menon, T. N. Nguyen, T. K. Kim, T. Jarsky, Z. Yao, B. Levi, L. T. Gray, S. A. Sorensen, T. Dolbeare, D. Bertagnolli, J. Goldy, N. Shapovalova, S. Parry, C. Lee, K. Smith, A. Bernard, L. Madisen, S. M. Sunkin, M. Hawrylycz, C. Koch, and H. Zeng. "Adult mouse cortical cell taxonomy revealed by single cell transcriptomics". en. In: *Nature Neuroscience* 19.2 (Feb. 2016), pp. 335–346. DOI: 10.1038/nn.4216.

[168] Z. Ji and H. Ji. "TSCAN: Pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis". In: *Nucleic Acids Research* 44.13 (July 2016), e117–e117. DOI: 10.1093/nar/gkw430.

[169] T. A. Geddes, T. Kim, L. Nan, J. G. Burchfield, J. Y. H. Yang, D. Tao, and P. Yang. "Autoencoder-based cluster ensembles for single-cell RNA-seq data analysis". In: *BMC Bioinformatics* 20.19 (Dec. 2019), p. 660. DOI: 10.1186/s12859-019-3179-5.

[170] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. "Fast unfolding of communities in large networks". In: *Journal of statistical mechanics: theory and experiment* 2008.10 (2008), P10008.

[171] V. Traag, L. Waltman, and N. J. van Eck. "From Louvain to Leiden: guaranteeing well-connected communities". In: *arXiv:1810.08473 [physics]* (Oct. 2018).

[172] J. H. Levine, E. F. Simonds, S. C. Bendall, K. L. Davis, E.-a. D. Amir, M. D. Tadmor, O. Litvin, H. G. Fienberg, A. Jager, E. R. Zunder, R. Finck, A. L. Gedman, I. Radtke, J. R. Downing, D. Pe'er, and G. P. Nolan. "Data-Driven Phenotypic Dissection of AML Reveals Progenitor-like Cells that Correlate with Prognosis". English. In: *Cell* 162.1 (July 2015), pp. 184–197. DOI: 10.1016/j.cell.2015.05.047.

[173] V. Y. Kiselev, T. S. Andrews, and M. Hemberg. "Challenges in unsupervised clustering of single-cell RNA-seq data". En. In: *Nature Reviews Genetics* (Jan. 2019), p. 1. DOI: 10.1038/s41576-018-0088-9.

[174] V. Traag. "Community Detection". en. In: *Algorithms and Dynamical Models for Communities and Reputation in Social Networks*. Cham: Springer International Publishing, 2014, pp. 11–47. DOI: 10.1007/978-3-319-06391-1_2.

[175] F. A. Wolf, F. K. Hamey, M. Plass, J. Solana, J. S. Dahlin, B. Göttgens, N. Rajewsky, L. Simon, and F. J. Theis. "PAGA: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells". In: *Genome Biology* 20.1 (Mar. 2019), p. 59. DOI: 10.1186/s13059-019-1663-x.

[176] B. Raj, D. E. Wagner, A. McKenna, S. Pandey, A. M. Klein, J. Shendure, J. A. Gagnon, and A. F. Schier. "Simultaneous single-cell profiling of lineages and cell types in the vertebrate brain". en. In: *Nature Biotechnology* 36.5 (May 2018), pp. 442–450. DOI: 10.1038/nbt.4103.

[177] B. Spanjaard, B. Hu, N. Mitic, P. Olivares-Chauvet, S. Janjuha, N. Ninov, and J. P. Junker. "Simultaneous lineage tracing and cell-type identification using CRISPR–Cas9-induced genetic scars". en. In: *Nature Biotechnology* 36.5 (May 2018), pp. 469–473. DOI: 10.1038/nbt.4124.

[178] A. Alemany, M. Florescu, C. S. Baron, J. Peterson-Maduro, and A. van Oudenaarden. "Whole-organism clone tracing using single-cell sequencing". en. In: *Nature* 556.7699 (Apr. 2018), pp. 108–112. DOI: 10.1038/nature25969.

[179] J. Sulston, E. Schierenberg, J. White, and J. Thomson. "The embryonic cell lineage of the nematode Caenorhabditis elegans". en. In: *Developmental Biology* 100.1 (Nov. 1983), pp. 64–119. DOI: 10.1016/0012-1606(83)90201-4.

[180] C. S. Baron and A. van Oudenaarden. "Unravelling cellular relationships during development and regeneration using genetic lineage tracing". en. In: *Nature Reviews Molecular Cell Biology* (Nov. 2019). DOI: 10.1038/s41580-019-0186-3.

[181] R. Moreno-Ayala and J. P. Junker. "Single-cell genomics to study developmental cell fate decisions in zebrafish". In: *Briefings in Functional Genomics* 20.6 (Nov. 2021), pp. 420–426. DOI: 10.1093/bfgp/elab018.

[182] P. Olivares-Chauvet and J. P. Junker. "Inclusion of temporal information in single cell transcriptomics". en. In: *The International Journal of Biochemistry & Cell Biology* 122 (May 2020), p. 105745. DOI: 10.1016/j.biocel.2020.105745.

[183] D. E. Wagner, C. Weinreb, Z. M. Collins, J. A. Briggs, S. G. Megason, and A. M. Klein. "Single-cell mapping of gene expression landscapes and lineage in the zebrafish embryo". en. In: *Science* (Apr. 2018), eaar4362. DOI: 10.1126/science.aar4362.

[184] M. M. Chan, Z. D. Smith, S. Grosswendt, H. Kretzmer, T. M. Norman, B. Adamson, M. Jost, J. J. Quinn, D. Yang, M. G. Jones, A. Khodaverdian, N. Yosef, A. Meissner, and J. S. Weissman. "Molecular recording of mammalian embryogenesis". en. In: *Nature* (May 2019). DOI: 10.1038/s41586-019-1184-5.

[185] S. Bowling, D. Sritharan, F. G. Osorio, M. Nguyen, P. Cheung, A. Rodriguez-Fraticelli, S. Patel, W.-C. Yuan, Y. Fujiwara, B. E. Li, S. H. Orkin, S. Hormoz, and F. D. Camargo. "An Engineered CRISPR-Cas9 Mouse Line for Simultaneous

Readout of Lineage Histories and Gene Expression Profiles in Single Cells". English. In: *Cell* 181.6 (June 2020), 1410–1422.e27. DOI: 10.1016/j.cell.2020.04.048.

[186] B. Hu, S. Lelek, B. Spanjaard, M. G. Simões, H. Aliee, R. Schäfer, F. Theis, D. Panáková, and J. P. Junker. *Cellular drivers of injury response and regeneration in the adult zebrafish heart*. en. Tech. rep. bioRxiv, Jan. 2021, p. 2021.01.07.425670. DOI: 10.1101/2021.01.07.425670.

[187] K. Hurley, J. Ding, C. Villacorta-Martin, M. J. Herriges, A. Jacob, M. Vedaie, K. D. Alysandratos, Y. L. Sun, C. Lin, R. B. Werder, J. Huang, A. A. Wilson, A. Mithal, G. Mostoslavsky, I. Oglesby, I. S. Caballero, S. H. Guttentag, F. Ahangari, N. Kaminski, A. Rodriguez-Fraticelli, F. Camargo, Z. Bar-Joseph, and D. N. Kotton. "Reconstructed Single-Cell Fate Trajectories Define Lineage Plasticity Windows during Differentiation of Human PSC-Derived Distal Lung Progenitors". en. In: *Cell Stem Cell* 26.4 (Apr. 2020), 593–608.e8. DOI: 10.1016/j.stem.2019.12.009.

[188] M. Ratz, L. von Berlin, L. Larsson, M. Martin, J. O. Westholm, G. La Manno, J. Lundeberg, and J. Frisén. "Clonal relations in the mouse brain revealed by single-cell and spatial transcriptomics". en. In: *Nature Neuroscience* (Feb. 2022), pp. 1–10. DOI: 10.1038/s41593-022-01011-x.

[189] B. Raj, J. A. Farrell, J. Liu, J. E. Kholtei, A. N. Carte, J. N. Acedo, L. Y. Du, A. McKenna, Đ. Relić, J. M. Leslie, and A. F. Schier. "Emergence of Neuronal Diversity during Vertebrate Brain Development". English. In: *Neuron* 108.6 (Dec. 2020), 1058–1074.e6. DOI: 10.1016/j.neuron.2020.09.023.

[190] D. Yang, M. G. Jones, S. Naranjo, W. M. Rideout, K. H. (Min, R. Ho, W. Wu, J. M. Replogle, J. L. Page, J. J. Quinn, F. Horns, X. Qiu, M. Z. Chen, W. A. Freed-Pastor, C. S. McGinnis, D. M. Patterson, Z. J. Gartner, E. D. Chow, T. G. Bivona, M. M. Chan, N. Yosef, T. Jacks, and J. S. Weissman. *Lineage Recording Reveals the Phylodynamics, Plasticity and Paths of Tumor Evolution*. en. Tech. rep. bioRxiv, Oct. 2021, p. 2021.10.12.464111. DOI: 10.1101/2021.10.12.464111.

[191] J. J. Quinn, M. G. Jones, R. A. Okimoto, S. Nanjo, M. M. Chan, N. Yosef, T. G. Bivona, and J. S. Weissman. "Single-cell lineages reveal the rates, routes, and drivers of metastasis in cancer xenografts". en. In: *Science* 371.6532 (Feb. 2021). DOI: 10.1126/science.abc1944.

[192] K. P. Simeonov, C. N. Byrns, M. L. Clark, R. J. Norgard, B. Martin, B. Z. Stanger, J. Shendure, A. McKenna, and C. J. Lengner. "Single-cell lineage tracing of metastatic cancer reveals selection of hybrid EMT states". en. In: *Cancer Cell* (June 2021). DOI: 10.1016/j.ccell.2021.05.005.

[193] N. Navin, J. Kendall, J. Troge, P. Andrews, L. Rodgers, J. McIndoo, K. Cook, A. Stepansky, D. Levy, D. Esposito, L. Muthuswamy, A. Krasnitz, W. R. McCombie, J. Hicks, and M. Wigler. "Tumour evolution inferred by single-cell sequencing". en. In: *Nature* 472.7341 (Apr. 2011), pp. 90–94. DOI: 10.1038/nature09807.

[194] Y. Wang, J. Waters, M. L. Leung, A. Unruh, W. Roh, X. Shi, K. Chen, P. Scheet, S. Vattathil, H. Liang, A. Multani, H. Zhang, R. Zhao, F. Michor, F. Meric-Bernstam, and N. E. Navin. "Clonal evolution in breast cancer revealed by single nucleus genome sequencing". en. In: *Nature* 512.7513 (Aug. 2014), pp. 155–160. DOI: 10.1038/nature13600.

[195] A. K. Casasent, A. Schalck, R. Gao, E. Sei, A. Long, W. Pangburn, T. Casasent, F. Meric-Bernstam, M. E. Edgerton, and N. E. Navin. "Multiclonal Invasion in Breast Tumors Identified by Topographic Single Cell Sequencing". eng. In: *Cell* 172.1-2 (Jan. 2018), 205–217.e12. DOI: 10.1016/j.cell.2017.12.007.

[196] A. S. Nam, K.-T. Kim, R. Chaligne, F. Izzo, C. Ang, J. Taylor, R. M. Myers, G. Abu-Zeinah, R. Brand, N. D. Omans, A. Alonso, C. Sheridan, M. Mariani, X. Dai, E. Harrington, A. Pastore, J. R. Cubillos-Ruiz, W. Tam, R. Hoffman, R. Rabadan, J. M. Scandura, O. Abdel-Wahab, P. Smibert, and D. A. Landau. "Somatic mutations and cell identity linked by Genotyping of Transcriptomes". en. In: *Nature* 571.7765 (July 2019), pp. 355–360. DOI: 10.1038/s41586-019-1367-0.

[197] T. Biezuner, A. Spiro, O. Raz, S. Amir, L. Milo, R. Adar, N. Chapal-Ilani, V. Berman, Y. Fried, E. Ainbinder, G. Cohen, H. M. Barr, R. Halaban, and E. Shapiro. "A generic, cost-effective, and scalable cell lineage analysis platform". en. In: *Genome Research* 26.11 (Nov. 2016), pp. 1588–1599. DOI: 10.1101/gr.202903.115.

[198] H. M. Kang, M. Subramaniam, S. Targ, M. Nguyen, L. Maliskova, E. McCarthy, E. Wan, S. Wong, L. Byrnes, C. M. Lanata, R. E. Gate, S. Mostafavi, A. Marson, N. Zaitlen, L. A. Criswell, and C. J. Ye. "Multiplexed droplet single-cell RNA-sequencing using natural genetic variation". en. In: *Nature Biotechnology* 36.1 (Jan. 2018), pp. 89–94. DOI: 10.1038/nbt.4042.

[199] R. Li, A. Montpetit, M. Rousseau, S. Y. M. Wu, C. M. T. Greenwood, T. D. Spector, M. Pollak, C. Polychronakos, and J. B. Richards. "Somatic point mutations occurring early in development: a monozygotic twin study". en. In: *Journal of Medical Genetics* 51.1 (Jan. 2014), pp. 28–34. DOI: 10.1136/jmedgenet-2013-101712.

[200] J. B. Stewart and P. F. Chinnery. "The dynamics of mitochondrial DNA heteroplasmy: implications for human health and disease". en. In: *Nature Reviews Genetics* 16.9 (Sept. 2015), pp. 530–542. DOI: 10.1038/nrg3966.

[201] L. S. Ludwig, C. A. Lareau, J. C. Ulirsch, E. Christian, C. Muus, L. H. Li, K. Pelka, W. Ge, Y. Oren, A. Brack, T. Law, C. Rodman, J. H. Chen, G. M. Boland, N. Hacohen, O. Rozenblatt-Rosen, M. J. Aryee, J. D. Buenrostro, A. Regev, and V. G. Sankaran. "Lineage Tracing in Humans Enabled by Mitochondrial Mutations and Single-Cell Genomics". en. In: *Cell* 176.6 (Mar. 2019), 1325–1339.e22. DOI: 10.1016/j.cell.2019.01.022.

[202] J. Xu, K. Nuno, U. M. Litzenburger, Y. Qi, M. R. Corces, R. Majeti, and H. Y. Chang. "Single-cell lineage tracing by endogenous mutations enriched in transposase accessible mitochondrial DNA". In: *eLife* 8 (Apr. 2019). Ed. by R. L. Levine, M. E. Bronner, and R. L. Levine, e45105. DOI: 10.7554/eLife.45105.

[203] C. A. Lareau, F. M. Duarte, J. G. Chew, V. K. Kartha, Z. D. Burkett, A. S. Kohlway, D. Pokholok, M. J. Aryee, F. J. Steemers, R. Lebofsky, and J. D. Buenrostro. "Droplet-based combinatorial indexing for massive-scale single-cell chromatin accessibility". en. In: *Nature Biotechnology* 37.8 (Aug. 2019), pp. 916–924. DOI: 10.1038/s41587-019-0147-6.

[204] T. E. Miller, C. A. Lareau, J. A. Verga, E. A. K. DePasquale, V. Liu, D. Ssozi, K. Sandor, Y. Yin, L. S. Ludwig, C. A. El Farran, D. M. Morgan, A. T. Satpathy, G. K. Griffin, A. A. Lane, J. C. Love, B. E. Bernstein, V. G. Sankaran, and P. van Galen. "Mitochondrial variant enrichment from high-throughput single-cell RNA sequencing resolves clonal populations". en. In: *Nature Biotechnology* (Feb. 2022), pp. 1–5. DOI: 10.1038/s41587-022-01210-8.

[205] M. A. Walker, C. A. Lareau, L. S. Ludwig, A. Karaa, V. G. Sankaran, A. Regev, and V. K. Mootha. "Purifying Selection against Pathogenic Mitochondrial DNA in Human T Cells". In: *New England Journal of Medicine* 383.16 (Oct. 2020), pp. 1556–1563. DOI: 10.1056/NEJMoa2001265.

[206] C. A. Lareau, L. S. Ludwig, C. Muus, S. H. Gohil, T. Zhao, Z. Chiang, K. Pelka, J. M. Verboon, W. Luo, E. Christian, D. Rosebrock, G. Getz, G. M. Boland, F. Chen, J. D. Buenrostro, N. Hacohen, C. J. Wu, M. J. Aryee, A. Regev, and V. G. Sankaran. "Massively parallel single-cell mitochondrial DNA genotyping and chromatin profiling". en. In: *Nature Biotechnology* 39.4 (Apr. 2021), pp. 451–461. DOI: 10.1038/s41587-020-0645-6.

[207] M. G. Jones, A. Khodaverdian, J. J. Quinn, M. M. Chan, J. A. Hussmann, R. Wang, C. Xu, J. S. Weissman, and N. Yosef. "Inference of single-cell phylogenies from lineage tracing data using Cassiopeia". In: *Genome Biology* 21.1 (Apr. 2020), p. 92. DOI: 10.1186/s13059-020-02000-8.

[208] W. Gong, A. A. Granados, J. Hu, M. G. Jones, O. Raz, I. Salvador-Martínez, H. Zhang, K.-H. K. Chow, I.-Y. Kwak, R. Retkute, A. Prusokas, A. Prusokas, A. Khodaverdian, R. Zhang, S. Rao, R. Wang, P. Rennert, V. G. Saipradeep, N. Sivadasan, A. Rao, T. Joseph, R. Srinivasan, J. Peng, L. Han, X. Shang, D. J. Garry, T. Yu, V. Chung, M. Mason, Z. Liu, Y. Guan, N. Yosef, J. Shendure, M. J. Telford, E. Shapiro, M. B. Elowitz, and P. Meyer. "Benchmarked approaches for reconstruction of in vitro cell lineages and in silico models of C. elegans and M. musculus developmental trees". eng. In: *Cell Systems* 12.8 (Aug. 2021), 810–826.e4. DOI: 10.1016/j.cels.2021.05.008.

[209] K. Ouardini, R. Lopez, M. G. Jones, S. Prillo, R. Zhang, M. I. Jordan, and N. Yosef. "Reconstructing unobserved cellular states from paired single-cell lineage tracing and transcriptomics data". en. In: *bioRxiv* (May 2021), p. 2021.05.28.446021. DOI: 10.1101/2021.05.28.446021.

[210] A. Forrow and G. Schiebinger. "LineageOT is a unified framework for lineage tracing and trajectory inference". en. In: *Nature Communications* 12.1 (Aug. 2021), p. 4940. DOI: 10.1038/s41467-021-25133-1.

[211] A. Tolver. *An introduction to Markov chains*. en. Lecture Notes in Stochastic Processes. Copenhagen: University of Copenhagen, 2016.

[212] P. C. Bressloff. "Stochastic switching in biology: from genotype to phenotype". en. In: *Journal of Physics A: Mathematical and Theoretical* 50.13 (Mar. 2017), p. 133001. DOI: 10.1088/1751-8121/aa5db4.

[213] L. Haghverdi, M. Büttner, F. A. Wolf, F. Buettner, and F. J. Theis. "Diffusion pseudotime robustly reconstructs lineage branching". en. In: *Nature Methods* 13.10 (Oct. 2016), pp. 845–848. DOI: 10.1038/nmeth.3971.

[214] C. Weinreb, S. Wolock, B. K. Tusi, M. Socolovsky, and A. M. Klein. "Fundamental limits on dynamic inference from single-cell snapshots". en. In: *Proceedings of the National Academy of Sciences* (Feb. 2018), p. 201714723. DOI: 10.1073/pnas.1714723115.

[215] R. Tarjan. "Depth-First Search and Linear Graph Algorithms". In: *SIAM Journal on Computing* 1.2 (June 1972), pp. 146–160. DOI: 10.1137/0201010.

[216] O. Perron. "Zur Theorie der Matrices". de. In: *Mathematische Annalen* 64.2 (June 1907), pp. 248–263. DOI: 10.1007/BF01449896.

[217] U. Von Luxburg. "A tutorial on spectral clustering". In: *Statistics and computing* 17.4 (2007), pp. 395–416.

[218] L. Haghverdi. "Geometric Diffusions for Reconstruction of Cell Differentiation Dynamics". PhD thesis. TUM, 2016. URL: https://mediatum.ub.tum.de/doc/1325451/1325451.pdf (visited on 10/10/2018).

[219] M. Belkin and P. Niyogi. "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation". en. In: *Neural Computation* 15.6 (June 2003), pp. 1373–1396. DOI: 10.1162/089976603321780317.

[220] M. Nitzan, N. Karaiskos, N. Friedman, and N. Rajewsky. "Gene expression cartography". en. In: *Nature* 576.7785 (Dec. 2019), pp. 132–137. DOI: 10.1038/s41586-019-1773-3.

[221] G. Peyré and M. Cuturi. "Computational Optimal Transport". In: *arXiv:1803.00567 [stat]* (Mar. 2020).

[222] L. V. Kantorovich. "On the Translocation of Masses". en. In: *Journal of Mathematical Sciences* 133.4 (Mar. 2006), pp. 1381–1382. DOI: 10.1007/s10958-006-0049-2.

[223] W. S. Chen, N. Zivanovic, D. van Dijk, G. Wolf, B. Bodenmiller, and S. Krishnaswamy. "Uncovering axes of variation among single-cell cancer specimens". en. In: *Nature Methods* 17.3 (Mar. 2020), pp. 302–310. DOI: 10.1038/s41592-019-0689-z.

[224] D. P. Bertsekas. "A new algorithm for the assignment problem". en. In: *Mathematical Programming* 21.1 (Dec. 1981), pp. 152–171. DOI: 10.1007/BF01584237.

[225] M. Cuturi. "Sinkhorn Distances: Lightspeed Computation of Optimal Transport". In: *Advances in Neural Information Processing Systems*. Vol. 26. Curran Associates, Inc., 2013.

[226] R. Cominetti and J. S. Martín. "Asymptotic analysis of the exponential penalty trajectory in linear programming". en. In: *Mathematical Programming* 67.1 (Oct. 1994), pp. 169–187. DOI: 10.1007/BF01582220.

[227] G. U. Yule. "On the Methods of Measuring Association Between Two Attributes". In: *Journal of the Royal Statistical Society* 75.6 (1912), pp. 579–652. DOI: 10.2307/2340126.

[228] R. Sinkhorn. "A Relationship Between Arbitrary Positive Matrices and Doubly Stochastic Matrices". In: *The Annals of Mathematical Statistics* 35.2 (June 1964), pp. 876–879. DOI: 10.1214/aoms/1177703591.

[229] M. Cuturi, L. Meng-Papaxanthos, Y. Tian, C. Bunne, G. Davis, and O. Teboul. "Optimal Transport Tools (OTT): A JAX Toolbox for all things Wasserstein". In: *arXiv:2201.12324 [cs, stat]* (Jan. 2022).

[230] R. Frostig, M. J. Johnson, and C. Leary. "Compiling machine learning programs via high-level tracing". en. In: *Systems for Machine Learning* (2018), pp. 23–24.

[231] M. Liero, A. Mielke, and G. Savaré. "Optimal Entropy-Transport problems and a new Hellinger–Kantorovich distance between positive measures". en. In: *Inventiones mathematicae* 211.3 (Mar. 2018), pp. 969–1117. DOI: 10.1007/s00222-017-0759-8.

[232] L. Chizat, G. Peyré, B. Schmitzer, and F.-X. Vialard. "Scaling algorithms for unbalanced optimal transport problems". en. In: *Mathematics of Computation* 87.314 (Nov. 2018), pp. 2563–2609. DOI: 10.1090/mcom/3303.

[233] F. Mémoli. "Gromov–Wasserstein Distances and the Metric Approach to Object Matching". en. In: *Foundations of Computational Mathematics* 11.4 (Aug. 2011), pp. 417–487. DOI: 10.1007/s10208-011-9093-5.

[234] M. Gromov. *Metric structures for Riemannian and non-Riemannian spaces*. eng. Modern Birkhäuser classics. Boston: Birkhäuser, 2001.

[235] G. Peyré, M. Cuturi, and J. Solomon. "Gromov-Wasserstein Averaging of Kernel and Distance Matrices". en. In: *International Conference on Machine Learning*. PMLR (2016), p. 9.

[236] J. Solomon, G. Peyré, V. G. Kim, and S. Sra. "Entropic metric alignment for correspondence problems". en. In: *ACM Transactions on Graphics* 35.4 (July 2016), pp. 1–13. DOI: 10.1145/2897824.2925903.

[237] P. Demetci, R. Santorella, B. Sandstede, W. S. Noble, and R. Singh. "SCOT: Single-Cell Multi-Omics Alignment with Optimal Transport". eng. In: *Journal of Computational Biology: A Journal of Computational Molecular Cell Biology* 29.1 (Jan. 2022), pp. 3–18. DOI: 10.1089/cmb.2021.0446.

[238] G. Palla, D. S. Fischer, A. Regev, and F. J. Theis. "Spatial components of molecular tissue biology". en. In: *Nature Biotechnology* (Feb. 2022), pp. 1–11. DOI: 10.1038/s41587-021-01182-1.

[239] N. Moriel, E. Senel, N. Friedman, N. Rajewsky, N. Karaiskos, and M. Nitzan. "NovoSpaRc: flexible spatial reconstruction of single-cell gene expression with optimal transport". en. In: *Nature Protocols* 16.9 (Sept. 2021), pp. 4177–4200. DOI: 10.1038/s41596-021-00573-7.

[240] T. Vayer, L. Chapel, R. Flamary, R. Tavenard, and N. Courty. "Fused Gromov-Wasserstein Distance for Structured Objects". In: *Algorithms* 13.9 (Sept. 2020), p. 212. DOI: 10.3390/a13090212.

[241] J.-D. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré. "Iterative Bregman Projections for Regularized Transportation Problems". In: *arXiv:1412.5154 [math]* (Dec. 2014).

[242] L. Deconinck, R. Cannoodt, W. Saelens, B. Deplancke, and Y. Saeys. "Recent advances in trajectory inference from single-cell omics data". en. In: *Current Opinion in Systems Biology* (May 2021), S2452310021000299. DOI: 10.1016/j.coisb.2021.05.005.

[243] X. Qiu, Q. Mao, Y. Tang, L. Wang, R. Chawla, H. A. Pliner, and C. Trapnell. "Reversed graph embedding resolves complex single-cell trajectories". en. In: *Nature Methods* 14.10 (Aug. 2017), pp. 979–982. DOI: 10.1038/nmeth.4402.

[244] M. Setty, M. D. Tadmor, S. Reich-Zeliger, O. Angel, T. M. Salame, P. Kathail, K. Choi, S. Bendall, N. Friedman, and D. Pe'er. "Wishbone identifies bifurcating developmental trajectories from single-cell data". en. In: *Nature Biotechnology* 34.6 (June 2016), pp. 637–645. DOI: 10.1038/nbt.3569.

[245] K. Street, D. Risso, R. B. Fletcher, D. Das, J. Ngai, N. Yosef, E. Purdom, and S. Dudoit. "Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics". In: *BMC Genomics* 19 (June 2018). DOI: 10.1186/s12864-018-4772-0.

[246] D. S. Fischer, A. K. Fiedler, E. Kernfeld, R. M. J. Genga, J. Hasenauer, R. Maehr, and F. J. Theis. "Beyond pseudotime: Following T-cell maturation in single-cell RNAseq time series". en. In: *bioRxiv* (Nov. 2017), p. 219188. DOI: 10.1101/219188.

[247] U. v. Luxburg, A. Radl, and M. Hein. "Hitting and Commute Times in Large Random Neighborhood Graphs". In: *Journal of Machine Learning Research* 15.52 (2014), pp. 1751–1798.

[248] K. S. Booth and G. S. Lueker. "Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms". en. In: *Journal of Computer and System Sciences* 13.3 (Dec. 1976), pp. 335–379. DOI: 10.1016/S0022-0000(76)80045-1.

[249] Q. Mao, L. Wang, I. W. Tsang, and Y. Sun. "Principal Graph and Structure Learning Based on Reversed Graph Embedding". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.11 (Nov. 2017), pp. 2227–2241. DOI: 10.1109/TPAMI.2016.2635657.

[250] T. Hastie and W. Stuetzle. "Principal Curves". In: *Journal of the American Statistical Association* 84.406 (June 1989), pp. 502–516. DOI: 10.1080/01621459.1989.10478797.

[251] Q. Mao, L. Yang, L. Wang, S. Goodison, and Y. Sun. "SimplePPT: A Simple Principal Tree Algorithm". en. In: *Proceedings of the 2015 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, June 2015, pp. 792–800. DOI: 10.1137/1.9781611974010.89.

[252] J. D. Buenrostro, M. R. Corces, C. A. Lareau, B. Wu, A. N. Schep, M. J. Aryee, R. Majeti, H. Y. Chang, and W. J. Greenleaf. "Integrated Single-Cell Analysis Maps the Continuous Regulatory Landscape of Human Hematopoietic Differentiation". English. In: *Cell* 173.6 (May 2018), 1535–1548.e16. DOI: 10.1016/j.cell.2018.03.074.

[253] A. B. Stergachis, S. Neph, A. Reynolds, R. Humbert, B. Miller, S. L. Paige, B. Vernot, J. B. Cheng, R. E. Thurman, R. Sandstrom, E. Haugen, S. Heimfeld, C. E. Murry, J. M. Akey, and J. A. Stamatoyannopoulos. "Developmental Fate and Cellular Maturity Encoded in Human Regulatory DNA Landscapes". English. In: *Cell* 154.4 (Aug. 2013), pp. 888–903. DOI: 10.1016/j.cell.2013.07.020.

[254] M. R. Corces, J. D. Buenrostro, B. Wu, P. G. Greenside, S. M. Chan, J. L. Koenig, M. P. Snyder, J. K. Pritchard, A. Kundaje, W. J. Greenleaf, R. Majeti, and H. Y. Chang. "Lineage-specific and single-cell chromatin accessibility charts human hematopoiesis and leukemia evolution". en. In: *Nature Genetics* 48.10 (Oct. 2016), pp. 1193–1203. DOI: 10.1038/ng.3646.

[255] C. H. Waddington. *The Strategy of the Genes*. en. Routledge, Apr. 2014.

[256] D. Ting, L. Huang, and M. Jordan. "An Analysis of the Convergence of Graph Laplacians". In: *arXiv:1101.5435 [stat]* (Jan. 2011).

[257] H. Hochgerner, A. Zeisel, P. Lönnerberg, and S. Linnarsson. "Conserved properties of dentate gyrus neurogenesis across postnatal development revealed by single-cell RNA sequencing". En. In: *Nature Neuroscience* 21.2 (Feb. 2018), p. 290. DOI: 10.1038/s41593-017-0056-2.

[258] C. Guibentif, J. A. Griffiths, I. Imaz-Rosshandler, S. Ghazanfar, J. Nichols, V. Wilson, B. Göttgens, and J. C. Marioni. "Diverse Routes toward Early Somites in the Mouse Embryo". English. In: *Developmental Cell* 56.1 (Jan. 2021), 141–153.e6. DOI: 10.1016/j.devcel.2020.11.013.

[259] J.-D. Benamou and Y. Brenier. "A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem". en. In: *Numerische Mathematik* 84.3 (Jan. 2000), pp. 375–393. DOI: 10.1007/s002110050002.

[260] W. Grathwohl, R. T. Q. Chen, J. Bettencourt, I. Sutskever, and D. Duvenaud. "FFJORD: Free-form Continuous Dynamics for Scalable Reversible Generative Models". In: *arXiv:1810.01367 [cs, stat]* (Oct. 2018).

[261] A. Tong, J. Huang, G. Wolf, D. van Dijk, and S. Krishnaswamy. "TrajectoryNet: A Dynamic Optimal Transport Network for Modeling Cellular Dynamics". In: *Proceedings of machine learning research* 119 (July 2020), pp. 9526–9536.

[262] M. Scetbon, G. Peyré, and M. Cuturi. "Linear-Time Gromov Wasserstein Distances using Low Rank Couplings and Costs". In: *arXiv:2106.01128 [cs, stat]* (June 2021).

[263] M. Scetbon, M. Cuturi, and G. Peyré. "Low-Rank Sinkhorn Factorization". In: *arXiv:2103.04737 [cs, stat]* (Mar. 2021).

[264] H. Liu, M. Luo, and J.-k. Wen. "mRNA stability in the nucleus". In: *Journal of Zhejiang University. Science. B* 15.5 (May 2014), pp. 444–454. DOI: 10.1631/jzus.B1400088.

[265] T. Li. "On the Mathematics of RNA Velocity I: Theoretical Analysis". en. In: *CSIAM Transactions on Applied Mathematics* 2.1 (June 2021), pp. 1–55. DOI: 10.4208/csiam-am.SO-2020-0001.

[266] A. P. Dempster, N. M. Laird, and D. B. Rubin. "Maximum likelihood from incomplete data via the EM-algorithm". In: *Journal of the Royal Statistical Society. Series B (Methodological)* (1977), pp. 1–38.

[267] X. Qiu, Y. Zhang, J. D. Martin-Rufino, C. Weng, S. Hosseinzadeh, D. Yang, A. N. Pogson, M. Y. Hein, K. H. (Min, L. Wang, E. I. Grody, M. J. Shurtleff, R. Yuan, S. Xu, Y. Ma, J. M. Replogle, E. S. Lander, S. Darmanis, I. Bahar, V. G. Sankaran, J. Xing, and J. S. Weissman. "Mapping transcriptomic vector fields of single cells". English. In: *Cell* 0.0 (Feb. 2022). DOI: 10.1016/j.cell.2021.12.045.

[268] J. C. Kimmel, N. Yi, M. Roy, D. G. Hendrickson, and D. R. Kelley. "Differentiation reveals latent features of aging and an energy barrier in murine myogenesis". en. In: *Cell Reports* 35.4 (Apr. 2021), p. 109046. DOI: 10.1016/j.celrep.2021.109046.

[269] Z. Zhang and X. Zhang. "Inference of high-resolution trajectories in single-cell RNA-seq data by using RNA velocity". en. In: *Cell Reports Methods* 1.6 (Oct. 2021), p. 100095. DOI: 10.1016/j.crmeth.2021.100095.

[270] J. Ma, J. Zhao, J. Tian, X. Bai, and Z. Tu. "Regularized vector field learning with sparse approximation for mismatch removal". en. In: *Pattern Recognition* 46.12 (Dec. 2013), pp. 3519–3532. DOI: 10.1016/j.patcog.2013.05.017.

[271] N. J. Proudfoot, A. Furger, and M. J. Dye. "Integrating mRNA processing with transcription". eng. In: *Cell* 108.4 (Feb. 2002), pp. 501–512. DOI: 10.1016/s0092-8674(02)00617-7.

[272] G. Gorin and L. Pachter. *Length Biases in Single-Cell RNA Sequencing of pre-mRNA*. en. preprint. Biophysics, July 2021. DOI: 10.1101/2021.07.30.454514.

[273] A. Selewa, R. Dohn, H. Eckart, S. Lozano, B. Xie, E. Gauchat, R. Elorbany, K. Rhodes, J. Burnett, Y. Gilad, S. Pott, and A. Basu. "Systematic Comparison of High-throughput Single-Cell and Single-Nucleus Transcriptomes during Cardiomyocyte Differentiation". en. In: *Scientific Reports* 10.1 (Jan. 2020), p. 1535. DOI: 10.1038/s41598-020-58327-6.

[274] V. Bergen, R. Soldatov, P. V. Kharchenko, and F. J. Theis. "RNA velocity—current challenges and future perspectives". In: *Molecular Systems Biology* 17.8 (Aug. 2021), e10282. DOI: 10.15252/msb.202110282.

[275] G. Gorin, M. Fang, T. Chari, and L. Pachter. *RNA velocity unraveled*. en. Tech. rep. bioRxiv, Feb. 2022, p. 2022.02.12.480214. DOI: 10.1101/2022.02.12.480214.

[276] V. Marot-Lassauzaie, B. J. Bouman, F. D. Donaghy, and L. Haghverdi. *Towards reliable quantification of cell state velocities*. en. Tech. rep. bioRxiv, Mar. 2022, p. 2022.03.17.484754. DOI: 10.1101/2022.03.17.484754.

[277] G. Gorin, V. Svensson, and L. Pachter. "Protein velocity and acceleration from single-cell multiomics experiments". In: *Genome Biology* 21.1 (Feb. 2020), p. 39. DOI: 10.1186/s13059-020-1945-3.

[278] C. Li, M. Virgilio, K. L. Collins, and J. D. Welch. *Single-cell multi-omic velocity infers dynamic and decoupled gene regulation*. en. Tech. rep. bioRxiv, Dec. 2021, p. 2021.12.13.472472. DOI: 10.1101/2021.12.13.472472.

[279] M. Tedesco, F. Giannese, D. Lazarević, V. Giansanti, D. Rosano, S. Monzani, I. Catalano, E. Grassi, E. R. Zanella, O. A. Botrugno, L. Morelli, P. Panina Bordignon, G. Caravagna, A. Bertotti, G. Martino, L. Aldrighetti, S. Pasqualato, L. Trusolino, D. Cittaro, and G. Tonon. "Chromatin Velocity reveals epigenetic dynamics by single-cell profiling of heterochromatin and euchromatin". en. In: *Nature Biotechnology* (Oct. 2021), pp. 1–10. DOI: 10.1038/s41587-021-01031-1.

[280] V. A. Herzog, B. Reichholf, T. Neumann, P. Rescheneder, P. Bhat, T. R. Burkard, W. Wlotzka, A. von Haeseler, J. Zuber, and S. L. Ameres. "Thiol-linked alkylation of RNA to assess expression dynamics". en. In: *Nature Methods* 14.12 (Dec. 2017), pp. 1198–1204. DOI: 10.1038/nmeth.4435.

[281] J. A. Schofield, E. E. Duffy, L. Kiefer, M. C. Sullivan, and M. D. Simon. "TimeLapse-seq: adding a temporal dimension to RNA sequencing through nucleoside recoding". en. In: *Nature Methods* 15.3 (Mar. 2018), pp. 221–225. DOI: 10.1038/nmeth.4582.

[282] C. Riml, T. Amort, D. Rieder, C. Gasser, A. Lusser, and R. Micura. "Osmium-Mediated Transformation of 4-Thiouridine to Cytidine as Key To Study RNA Dynamics by Sequencing". eng. In: *Angewandte Chemie (International Ed. in English)* 56.43 (Oct. 2017), pp. 13479–13483. DOI: 10.1002/anie.201707465.

[283] C. Jürges, L. Dölken, and F. Erhard. "Dissecting newly transcribed and old RNA using GRAND-SLAM". en. In: *Bioinformatics* 34.13 (July 2018), pp. i218–i226. DOI: 10.1093/bioinformatics/bty256.

[284] F. Erhard, M. A. P. Baptista, T. Krammer, T. Hennig, M. Lange, P. Arampatzi, C. S. Jürges, F. J. Theis, A.-E. Saliba, and L. Dölken. "scSLAM-seq reveals core features of transcription dynamics in single cells". En. In: *Nature* 571.7765 (July 2019), p. 419. DOI: 10.1038/s41586-019-1369-y.

[285] G.-J. Hendriks, L. A. Jung, A. J. M. Larsson, M. Lidschreiber, O. Andersson Forsman, K. Lidschreiber, P. Cramer, and R. Sandberg. "NASC-seq monitors RNA synthesis in single cells". en. In: *Nature Communications* 10.1 (July 2019), p. 3138. DOI: 10.1038/s41467-019-11028-9.

[286] N. Battich, J. Beumer, B. d. Barbanson, L. Krenning, C. S. Baron, M. E. Tanenbaum, H. Clevers, and A. v. Oudenaarden. "Sequencing metabolically labeled transcripts in single cells reveals mRNA turnover strategies". en. In: *Science* 367.6482 (Mar. 2020), pp. 1151–1156. DOI: 10.1126/science.aax3072.

[287] J. Cao, W. Zhou, F. Steemers, C. Trapnell, and J. Shendure. "Sci-fate characterizes the dynamics of gene expression in single cells". en. In: *Nature Biotechnology* (Apr. 2020), pp. 1–9. DOI: 10.1038/s41587-020-0480-9.

[288] Q. Qiu, P. Hu, X. Qiu, K. W. Govek, P. G. Cámara, and H. Wu. "Massively parallel and time-resolved RNA sequencing in single cells with scNT-seq". en. In: *Nature Methods* 17.10 (Oct. 2020), pp. 991–1001. DOI: 10.1038/s41592-020-0935-4.

[289] C. Soneson, A. Srivastava, R. Patro, and M. B. Stadler. "Preprocessing choices affect RNA velocity results for droplet scRNA-seq data". en. In: *PLOS Computational Biology* 17.1 (Jan. 2021), e1008585. DOI: 10.1371/journal.pcbi.1008585.

[290] D. T. Gillespie. "The chemical Langevin equation". In: *The Journal of Chemical Physics* 113.1 (July 2000), pp. 297–306. DOI: 10.1063/1.481811.

[291] C. Kreutz, A. Raue, D. Kaschek, and J. Timmer. "Profile likelihood in systems biology". en. In: *FEBS Journal* 280.11 (June 2013), pp. 2564–2571. DOI: 10.1111/febs.12276.

[292] J. A. Nelder and R. Mead. "A Simplex Method for Function Minimization". en. In: *The Computer Journal* 7.4 (Jan. 1965), pp. 308–313. DOI: 10.1093/comjnl/7.4.308.

[293] B. Reuter, M. Weber, K. Fackeldey, S. Röblitz, and M. E. Garcia. "Generalized Markov State Modeling Method for Nonequilibrium Biomolecular Dynamics: Exemplified on Amyloid β Conformational Dynamics Driven by an Oscillating Electric Field". en. In: *Journal of Chemical Theory and Computation* 14.7 (July 2018), pp. 3579–3594. DOI: 10.1021/acs.jctc.8b00079.

[294] B. Reuter, K. Fackeldey, and M. Weber. "Generalized Markov modeling of nonreversible molecular kinetics". In: *The Journal of Chemical Physics* 150.17 (May 2019), p. 174103. DOI: 10.1063/1.5064530.

[295] B. Reuter. *Generalisierte Markov-Modellierung: Modellierung irreversibler β-Amyloid-Peptid-Dynamik unter Mikrowelleneinfluss*. de. Springer Spektrum, 2020. DOI: 10.1007/978-3-658-29712-1.

[296] B. Reuter. *GPCCA: Generalized Perron Cluster Cluster Analysis program to coarse-grain reversible and non-reversible Markov State Models*. original-date: 2020-04-14T12:12:40Z. May 2020. URL: http://github.com/msmdev/gpcca (visited on 11/20/2020).

[297] S. Röblitz and M. Weber. "Fuzzy spectral clustering by PCCA+: application to Markov state models and data classification". en. In: *Advances in Data Analysis and Classification* 7.2 (June 2013), pp. 147–179. DOI: 10.1007/s11634-013-0134-6.

[298] K. Fackeldey and M. Weber. *GenPCCA: Markov state Models for Non-Equilibrium Steady States*. en. WIAS Report 29. Weierstrass Institute, 2017.

[299] G. H. Golub and C. F. Van Loan. *Matrix computations*. en. Fourth edition. Johns Hopkins studies in the mathematical sciences. Baltimore: The Johns Hopkins University Press, 2013.

[300] R. A. Horn and C. R. Johnson. *Matrix analysis*. en. 2nd ed. Cambridge ; New York: Cambridge University Press, 2012.

[301] S. Kube and M. Weber. "A coarse graining method for the identification of transition rates between molecular conformations". In: *The Journal of Chemical Physics* 126.2 (Jan. 2007), p. 024103. DOI: 10.1063/1.2404953.

[302] M. Weber. "Meshless Methods in Conformation Dynamics". PhD thesis. Zuse-Institut Berlin, Freie Universitaet Berlin, 2006. URL: https://www.zib.de/weber/Promotion.pdf (visited on 02/23/2021).

[303] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, and P. van Mulbregt. "SciPy 1.0: fundamental algorithms for scientific computing in Python". en. In: *Nature Methods* 17.3 (Mar. 2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[304] B. Reuter. *pyGPCCA - python GPCCA: Generalized Perron Cluster Cluster Analysis program to coarse-grain reversible and non-reversible Markov State Models*. Jan. 2021. URL: https://github.com/msmdev/pyGPCCA.

[305] V. Hernandez, J. E. Roman, and V. Vidal. "SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems". In: *ACM Transactions on Mathematical Software* 31.3 (Sept. 2005), pp. 351–362. DOI: 10.1145/1089014.1089019.

[306] L. D. Dalcin, R. R. Paz, P. A. Kler, and A. Cosimo. "Parallel distributed computing using Python". en. In: *Advances in Water Resources*. New Computational Methods and Software Tools 34.9 (Sept. 2011), pp. 1124–1139. DOI: 10.1016/j.advwatres.2011.04.013.

[307] Y. Saad and M. H. Schultz. "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems". In: *SIAM Journal on scientific and statistical computing* 7.3 (1986), pp. 856–869.

[308] L. Velten, S. F. Haas, S. Raffel, S. Blaszkiewicz, S. Islam, B. P. Hennig, C. Hirche, C. Lutz, E. C. Buss, D. Nowak, T. Boch, W.-K. Hofmann, A. D. Ho, W. Huber, A. Trumpp, M. A. G. Essers, and L. M. Steinmetz. "Human haematopoietic stem cell lineage commitment is a continuous process". en. In: *Nature Cell Biology* 19.4 (Apr. 2017), pp. 271–281. DOI: 10.1038/ncb3493.

[309] T. Hastie and R. Tibshirani. "Generalized Additive Models". EN. In: *Statistical Science* 1.3 (Aug. 1986), pp. 297–310. DOI: 10.1214/ss/1177013604.

[310] J. S. DeSalvo. "Standard Error of Forecast in Multiple Regression: Proof of a Useful Result". en. Publisher: RAND Corporation. Jan. 1970. URL: https://www.rand.org/pubs/papers/P4365.html (visited on 09/25/2020).

[311] S. Wood. *mgcv: Mixed GAM Computation Vehicle with Automatic Smoothness Estimation*. Aug. 2020. URL: https://CRAN.R-project.org/package=mgcv (visited on 09/23/2020).

[312] S. N. Wood. *Generalized Additive Models: An Introduction with R, Second Edition*. en. CRC Press, 2017.

[313] D. Servén and C. Brummit. *pyGAM: Generalized Additive Models in Python*. Oct. 2018. DOI: 10.5281/zenodo.1476122. URL: https://zenodo.org/record/1476122#.X2nIBpMza3I (visited on 09/22/2020).

[314] S. A. Morris, P. Cahan, H. Li, A. M. Zhao, A. K. San Roman, R. A. Shivdasani, J. J. Collins, and G. Q. Daley. "Dissecting engineered cell types and enhancing cell fate conversion via CellNet". In: *Cell* 158.4 (Aug. 2014), pp. 889–902. DOI: 10.1016/j.cell.2014.07.021.

[315] S. Sekiya and A. Suzuki. "Direct conversion of mouse fibroblasts to hepatocyte-like cells by defined factors". eng. In: *Nature* 475.7356 (June 2011), pp. 390–393. DOI: 10.1038/nature10263.

[316] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, and D. Cournapeau. "Scikit-learn: Machine Learning in Python". en. In: *the Journal of machine Learning research* 12 (2011), pp. 2825–2830.

[317] T. Fawcett. "An introduction to ROC analysis". en. In: *Pattern Recognition Letters* 27.8 (June 2006), pp. 861–874. DOI: 10.1016/j.patrec.2005.10.010.

[318] A. Bastidas-Ponce, K. Scheibner, H. Lickert, and M. Bakhti. "Cellular and molecular mechanisms coordinating pancreas development". en. In: *Development* 144.16 (Aug. 2017), pp. 2873–2888. DOI: 10.1242/dev.140756.

[319] A. Bastidas-Ponce, S. S. Roscioni, I. Burtscher, E. Bader, M. Sterr, M. Bakhti, and H. Lickert. "Foxa2 and Pdx1 cooperatively regulate postnatal maturation of pancreatic β-cells". en. In: *Molecular Metabolism* 6.6 (June 2017), pp. 524–534. DOI: 10.1016/j.molmet.2017.03.007.

[320] J. Zhang, L. B. McKenna, C. W. Bogue, and K. H. Kaestner. "The diabetes gene Hhex maintains δ-cell differentiation and islet function". In: *Genes & Development* 28.8 (Apr. 2014), pp. 829–834. DOI: 10.1101/gad.235499.113.

[321] N. A. J. Krentz, M. Y. Y. Lee, E. E. Xu, S. L. J. Sproul, A. Maslova, S. Sasaki, and F. C. Lynn. "Single-Cell Transcriptome Profiling of Mouse and hESC-Derived Pancreatic Progenitors". English. In: *Stem Cell Reports* 11.6 (Dec. 2018), pp. 1551–1564. DOI: 10.1016/j.stemcr.2018.11.008.

[322] J. S. Herman, Sagar, and D. Grün. "FateID infers cell fate bias in multipotent progenitors from single-cell RNA-seq data". en. In: *Nature Methods* 15.5 (May 2018), pp. 379–386. DOI: 10.1038/nmeth.4662.

[323] D. Stoffers, N. Zinkin, V. Stanojevic, W. Clarke, and J. Habener. "Pancreatic agenesis attributable to a single nucleotide deletion in the human IPF1 gene coding sequence". In: *Nature Genetics* 15.1 (1997), pp. 106–110. DOI: 10.1038/ng0197-106.

[324] J. Jonsson, L. Carlsson, T. Edlund, and H. Edlund. "Insulin-promoter-factor 1 is required for pancreas development in mice". In: *Nature* 371.6498 (1994), pp. 606–609. DOI: 10.1038/371606a0.

[325] P. R. Tata, H. Mou, A. Pardo-Saganta, R. Zhao, M. Prabhu, B. M. Law, V. Vinarsky, J. L. Cho, S. Breton, A. Sahay, B. D. Medoff, and J. Rajagopal. "Dedifferentiation of committed epithelial cells into stem cells in vivo". en. In: *Nature* 503.7475 (Nov. 2013), pp. 218–223. DOI: 10.1038/nature12777.

[326] P. W. Tetteh, H. F. Farin, and H. Clevers. "Plasticity within stem cell hierarchies in mammalian epithelia". eng. In: *Trends in Cell Biology* 25.2 (Feb. 2015), pp. 100–108. DOI: 10.1016/j.tcb.2014.09.003.

[327] S. Weinberger, B. Cockrill, and J. Mandel. *Principles of Pulmonary Medicine - 7th Edition*. 7th ed. Elsevier, 2017.

[328] B. L. M. Hogan, C. E. Barkauskas, H. A. Chapman, J. A. Epstein, R. Jain, C. C. W. Hsia, L. Niklason, E. Calle, A. Le, S. H. Randell, J. Rock, M. Snitow, M. Krummel, B. R. Stripp, T. Vu, E. S. White, J. A. Whitsett, and E. E. Morrisey. "Repair and regeneration of the respiratory system: complexity, plasticity, and mechanisms of lung stem cell function". eng. In: *Cell Stem Cell* 15.2 (Aug. 2014), pp. 123–138. DOI: 10.1016/j.stem.2014.07.012.

[329] J. R. Rock, S. H. Randell, and B. L. M. Hogan. "Airway basal stem cells: a perspective on their roles in epithelial homeostasis and remodeling". eng. In: *Disease Models & Mechanisms* 3.9-10 (Oct. 2010), pp. 545–556. DOI: 10.1242/dmm.006031.

[330] M. Barile, I. Imaz-Rosshandler, I. Inzani, S. Ghazanfar, J. Nichols, J. C. Marioni, C. Guibentif, and B. Göttgens. "Coordinated changes in gene expression kinetics underlie both mouse and human erythroid maturation". In: *Genome Biology* 22.1 (July 2021), p. 197. DOI: 10.1186/s13059-021-02414-y.

[331] S. V. Stassen, G. G. K. Yip, K. K. Y. Wong, J. W. K. Ho, and K. K. Tsia. "Generalized and scalable trajectory inference in single-cell omics data with VIA". en. In: *Nature Communications* 12.1 (Sept. 2021), p. 5528. DOI: 10.1038/s41467-021-25773-3.

[332] N. Schaum et al. "Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris". en. In: *Nature* 562.7727 (Oct. 2018), pp. 367–372. DOI: 10.1038/s41586-018-0590-4.

[333] J. A. Farrell, Y. Wang, S. J. Riesenfeld, K. Shekhar, A. Regev, and A. F. Schier. "Single-cell reconstruction of developmental trajectories during zebrafish embryogenesis". en. In: *Science* (Apr. 2018), eaar3131. DOI: 10.1126/science.aar3131.

[334] S. Zhang, A. Afanassiev, L. Greenstreet, T. Matsumoto, and G. Schiebinger. "Optimal transport analysis reveals trajectories in steady-state systems". eng. In: *PLoS computational biology* 17.12 (Dec. 2021), e1009466. DOI: 10.1371/journal.pcbi.1009466.

[335] W. E and E. Vanden-Eijnden. "Transition-Path Theory and Path-Finding Algorithms for the Study of Rare Events". In: *Annual Review of Physical Chemistry* 61.1 (2010), pp. 391–420. DOI: 10.1146/annurev.physchem.040808.090412.

[336] K. Van den Berge, H. Roux de Bézieux, K. Street, W. Saelens, R. Cannoodt, Y. Saeys, S. Dudoit, and L. Clement. "Trajectory-based differential expression analysis for single-cell sequencing data". en. In: *Nature Communications* 11.1 (Mar. 2020), pp. 1–13. DOI: 10.1038/s41467-020-14766-3.

[337] K. D. Yang, K. Damodaran, S. Venkatachalapathy, A. C. Soylemezoglu, G. V. Shivashankar, and C. Uhler. "Predicting cell lineages using autoencoders and optimal transport". en. In: *PLOS Computational Biology* 16.4 (Apr. 2020). Ed. by J. Ma, e1007828. DOI: 10.1371/journal.pcbi.1007828.

[338] N. Prasad, K. Yang, and C. Uhler. "Optimal Transport using GANs for Lineage Tracing". In: *arXiv:2007.12098 [cs, stat]* (July 2020).

[339] Z. Cang and Q. Nie. "Inferring spatial and signaling relationships between cells from single cell transcriptomic data". en. In: *Nature Communications* 11.1 (Apr. 2020), p. 2084. DOI: 10.1038/s41467-020-15968-5.

[340] R. Zeira, M. Land, and B. J. Raphael. "Alignment and Integration of Spatial Transcriptomics Data". en. In: *bioRxiv* (Mar. 2021), p. 2021.03.16.435604. DOI: 10.1101/2021.03.16.435604.

[341] K. Cao, Q. Gong, Y. Hong, and L. Wan. *uniPort: a unified computational framework for single-cell data integration with optimal transport*. en. Tech. rep. bioRxiv, Feb. 2022, p. 2022.02.14.480323. DOI: 10.1101/2022.02.14.480323.

[342] A. Tong, G. Huguet, A. Natik, K. MacDonald, M. Kuchroo, R. Coifman, G. Wolf, and S. Krishnaswamy. "Diffusion Earth Mover's Distance and Distribution Embeddings". In: *arXiv:2102.12833 [cs]* (July 2021).

[343] C. Bunne, S. G. Stark, G. Gut, J. S. d. Castillo, K.-V. Lehmann, L. Pelkmans, A. Krause, and G. Rätsch. "Learning Single-Cell Perturbation Responses using Neural Optimal Transport". en. In: *bioRxiv* (Dec. 2021), p. 2021.12.15.472775. DOI: 10.1101/2021.12.15.472775.

[344] R. Flamary, N. Courty, A. Gramfort, M. Z. Alaya, A. Boisbunon, S. Chambon, L. Chapel, A. Corenflos, K. Fatras, N. Fournier, L. Gautheron, N. T. H. Gayraud, H. Janati, A. Rakotomamonjy, I. Redko, A. Rolet, A. Schutz, V. Seguy, D. J. Sutherland, R. Tavenard, A. Tong, and T. Vayer. "POT: Python Optimal Transport". In: *Journal of Machine Learning Research* 22.78 (2021), pp. 1–8.

[345] B. Charlier, J. Feydy, J. A. Glaunès, F.-D. Collin, and G. Durif. "Kernel Operations on the GPU, with Autodiff, without Memory Overflows". In: *Journal of Machine Learning Research* 22.74 (2021), pp. 1–6.

[346] A. Gayoso, R. Lopez, G. Xing, P. Boyeau, V. Valiollah Pour Amiri, J. Hong, K. Wu, M. Jayasuriya, E. Mehlman, M. Langevin, Y. Liu, J. Samaran, G. Misrachi, A. Nazaret, O. Clivio, C. Xu, T. Ashuach, M. Gabitto, M. Lotfollahi, V. Svensson, E. da Veiga Beltrame, V. Kleshchevnikov, C. Talavera-López, L. Pachter, F. J. Theis, A. Streets, M. I. Jordan, J. Regier, and N. Yosef. "A Python library for probabilistic analysis of single-cell omics data". en. In: *Nature Biotechnology* (Feb. 2022), pp. 1–4. DOI: 10.1038/s41587-021-01206-w.

[347] I. Virshup, S. Rybakov, F. J. Theis, P. Angerer, and F. A. Wolf. "anndata: Annotated data". en. In: *bioRxiv* (Dec. 2021), p. 2021.12.16.473007. DOI: 10.1101/2021.12.16.473007.

[348] A. Forrow, J.-C. Hutter, M. Nitzan, P. Rigollet, and G. Schiebinger. "Statistical Optimal Transport via Factored Couplings". en. In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019, p. 12.

[349] J. Altschuler, F. Bach, A. Rudi, and J. Niles-Weed. "Massively scalable Sinkhorn distances via the Nystroem method". In: *arXiv:1812.05189 [cs, math, stat]* (Oct. 2019).

[350] C. Williams and M. Seeger. "Using the Nyström Method to Speed Up Kernel Machines". In: *Advances in Neural Information Processing Systems 13*. MIT Press, 2001, pp. 682–688.

[351] M. Scetbon and M. Cuturi. "Linear Time Sinkhorn Divergences using Positive Features". In: *arXiv:2006.07057 [cs, stat]* (Oct. 2020).

[352] A. Bakshi and D. P. Woodruff. "Sublinear Time Low-Rank Approximation of Distance Matrices". In: *arXiv:1809.06986 [cs, math]* (Sept. 2018).

[353] P. Indyk, A. Vakilian, T. Wagner, and D. Woodruff. "Sample-Optimal Low-Rank Approximation of Distance Matrices". In: *arXiv:1906.00339 [cs]* (June 2019).

[354] X. Pan, H. Li, and X. Zhang. "TedSim: temporal dynamics simulation of single-cell RNA sequencing data and cell division history". In: *Nucleic Acids Research* (Apr. 2022), gkac235. DOI: 10.1093/nar/gkac235.

[355] J. Feydy, T. Séjourné, F.-X. Vialard, S.-i. Amari, A. Trouvé, and G. Peyré. "Interpolating between Optimal Transport and MMD using Sinkhorn Divergences". In: *arXiv:1810.08278 [math, stat]* (Oct. 2018).

[356] A. V. Makkuva, A. Taghvaei, S. Oh, and J. D. Lee. "Optimal transport mapping via input convex neural networks". In: *arXiv:1908.10962 [cs, stat]* (June 2020).

[357] A. Taghvaei and A. Jalali. "2-Wasserstein Approximation via Restricted Convex Potentials with Application to Improved Training for GANs". In: *arXiv:1902.07197 [cs, math, stat]* (Feb. 2019).

[358] B. Amos, L. Xu, and J. Z. Kolter. "Input Convex Neural Networks". In: *arXiv:1609.07152 [cs, math]* (June 2017).

[359] C. Villani. *Topics in Optimal Transportation*. Vol. 58. American Mathematical Soc., 2021.

[360] H. Lavenant, S. Zhang, Y.-H. Kim, and G. Schiebinger. "Towards a mathematical theory of trajectory inference". In: *arXiv:2102.09204 [cs, math, stat]* (Feb. 2021).

[361] N. M. Negretti, E. J. Plosa, J. T. Benjamin, B. A. Schuler, A. C. Habermann, C. S. Jetter, P. Gulleman, C. Bunn, A. N. Hackett, M. Ransom, C. J. Taylor, D. Nichols, B. K. Matlock, S. H. Guttentag, T. S. Blackwell, N. E. Banovich, J. A. Kropski, and J. M. S. Sucre. "A single-cell atlas of mouse lung development". In: *Development* 148.24 (Dec. 2021), dev199512. DOI: 10.1242/dev.199512.

[362] C. Li, A. R. Muñoz-Rojas, G. Wang, A. O. Mann, C. Benoist, and D. Mathis. "PPARγ marks splenic precursors of multiple nonlymphoid-tissue Treg compartments". en. In: *Proceedings of the National Academy of Sciences* 118.13 (Mar. 2021). DOI: 10.1073/pnas.2025197118.

[363] G. Günes Günsel, T. M. Conlon, A. Jeridi, R. Kim, Z. Ertüz, N. J. Lang, M. Ansari, M. Novikova, D. Jiang, M. Strunz, M. Gaianova, C. Hollauer, C. Gabriel, I. Angelidis, S. Doll, J. C. Pestoni, S. L. Edelmann, M. S. Kohlhepp, A. Guillot, K. Bassler, H. P. Van Eeckhoutte, Ö. Kayalar, N. Konyalilar, T. Kanashova, S. Rodius, C. Ballester-López, C. M. Genes Robles, N. Smirnova, M. Rehberg, C. Agarwal, I. Krikki, B. Piavaux, S. E. Verleden, B. Vanaudenaerde, M. Königshoff, G. Dittmar, K. R. Bracke, J. L. Schultze, H. Watz, O. Eickelberg, T. Stoeger, G. Burgstaller,

F. Tacke, V. Heissmeyer, Y. Rinkevich, H. Bayram, H. B. Schiller, M. Conrad, R. Schneider, and A. Ö. Yildirim. "The arginine methyltransferase PRMT7 promotes extravasation of monocytes resulting in tissue injury in COPD". en. In: *Nature Communications* 13.1 (Mar. 2022), p. 1303. DOI: 10.1038/s41467-022-28809-4.

[364] K. Scheibner, S. Schirge, I. Burtscher, M. Büttner, M. Sterr, D. Yang, A. Böttcher, Ansarullah, M. Irmler, J. Beckers, F. M. Cernilogar, G. Schotta, F. J. Theis, and H. Lickert. "Epithelial cell plasticity drives endoderm formation during gastrulation". en. In: *Nature Cell Biology* 23.7 (July 2021), pp. 692–703. DOI: 10.1038/s41556-021-00694-x.

[365] Z. Guo, L. Zhang, Z. Wu, Y. Chen, F. Wang, and G. Chen. "In vivo direct reprogramming of reactive glial cells into functional neurons after brain injury and in an Alzheimer's disease model". eng. In: *Cell Stem Cell* 14.2 (Feb. 2014), pp. 188–202. DOI: 10.1016/j.stem.2013.12.001.

[366] J. Drouin-Ouellet, K. Pircs, R. A. Barker, J. Jakobsson, and M. Parmar. "Direct Neuronal Reprogramming for Disease Modeling Studies Using Patient-Derived Neurons: What Have We Learned?" In: *Frontiers in Neuroscience* 11 (Sept. 2017). DOI: 10.3389/fnins.2017.00530.

[367] C. Mollinari, J. Zhao, L. Lupacchini, E. Garaci, D. Merlo, and G. Pei. "Transdifferentiation: a new promise for neurodegenerative diseases". In: *Cell Death & Disease* 9.8 (Aug. 2018). DOI: 10.1038/s41419-018-0891-4.

[368] Y. Chen, J. Pu, and B. Zhang. "Progress and Challenges of Cell Replacement Therapy for Neurodegenerative Diseases Based on Direct Neural Reprogramming". In: *Human Gene Therapy* 27.12 (Aug. 2016), pp. 962–970. DOI: 10.1089/hum.2016.078.

[369] C. Heinrich, F. M. Spagnoli, and B. Berninger. "In vivo reprogramming for tissue repair". en. In: *Nature Cell Biology* 17.3 (Mar. 2015), pp. 204–211. DOI: 10.1038/ncb3108.

[370] W. Kong, Y. C. Fu, E. M. Holloway, G. Garipler, X. Yang, E. O. Mazzoni, and S. A. Morris. "Capybara: A computational tool to measure cell identity and fate transitions". en. In: *Cell Stem Cell* (Mar. 2022). DOI: 10.1016/j.stem.2022.03.001.

[371] S.-W. Wang, M. J. Herriges, K. Hurley, D. N. Kotton, and A. M. Klein. "CoSpar identifies early cell fate biases from single-cell transcriptomic and lineage information". en. In: *Nature Biotechnology* (Feb. 2022), pp. 1–9. DOI: 10.1038/s41587-022-01209-1.

[372] A. Uzquiano, A. J. Kedaigle, M. Pigoni, B. Paulsen, X. Adiconis, K. Kim, T. Faits, S. Nagaraja, N. Antón-Bolaños, C. Gerhardinger, A. Tucewicz, E. Murray, X. Jin, J. Buenrostro, F. Chen, S. Velasco, A. Regev, J. Z. Levin, and P. Arlotta. *Single-cell multiomics atlas of organoid development uncovers longitudinal molecular programs of cellular diversification of the human cerebral cortex*. en. Tech. rep. bioRxiv, Mar. 2022, p. 2022.03.17.484798. DOI: 10.1101/2022.03.17.484798.

[373] W. Chen, O. Guillaume-Gentil, R. Dainese, P. Y. Rainer, M. Zachara, C. G. Gäbelein, J. A. Vorholt, and B. Deplancke. "Genome-wide molecular recording using Live-seq". en. In: *bioRxiv* (Mar. 2021), p. 2021.03.24.436752. DOI: 10.1101/2021.03.24.436752.

Nomenclature

$\mathrm{Ber}(\cdot)$	Bernoulli distribution.
$\mathrm{NB}(\cdot)$	Negative binomial distribution.
$\mathcal{N}(\cdot)$	Normal or multivariate normal distribution.
X	Cell-state matrix, usually containing gene expression ($X^{(R)}$) or chromatin accessibility ($X^{(A)}$) of dimensions #cells times #features.
Z	Low-dimensional cell-state matrix.
N_c	Number of cells.
N_g	Number of genes in a scRNA-seq experiment.
N_l	Number of latent dimensions in a low-dimensional embedding.
N_p	Number of peaks in a scATAC-seq experiment.
N_m	Number of macrostates in the GPCCA method.
W	Graph adjacency matrix.
λ_i	Eigenvalues, often of the transition matrix T.
$\mathcal{G}$	Graph with vertices V and edges E. Typically, V represent cells and E represent nearest-neighbor relations among them.
T	Right-stochastic cell-cell transition matrix.
$L^{(\mathrm{rw})}$	Random-walk graph Laplacian.
$L^{(\mathrm{sym})}$	Symmetric graph Laplacian.
$\mathbb{R}_+$	Non-negative real numbers.
Δ_K	Probability simplex in K dimensions.
P	Optimal transport coupling matrix.
τ	Pseudotime.
χ	GPCCA membership matrix.