Methods and applications in machine learning and computational biology for the analysis and the integration of high dimensional single-cell transcriptomics datasets

Aziz Fouché

PSL UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à l'Institut Curie – PSL (U900)

Méthodes et applications en apprentissage automatique et biologie computationnelle pour l'analyse et l'intégration de données transcriptomiques single-cell de haute dimension

-

Methods and applications in machine learning and computational biology for the analysis and the integration of high dimensional single-cell transcriptomics datasets

Soutenue par
Aziz FOUCHÉ
Le 10 novembre 2023

Ecole doctorale n° 515
Complexité du Vivant

Spécialité
Bio-informatique

Composition du jury :

Gabriel PEYRÉ Directeur de Recherche, ENS	*Président*
Valentina BOEVA Professeur, ETH Zurich	*Rapporteur*
Didier SURDEZ Professeur, Balgrist University Hospital	*Rapporteur*
Andrei ZINOVYEV Ingénieur de Recherche, Institut Curie	*Directeur de thèse*
Olivier DELATTRE Directeur de Recherche, Institut Curie	*Co-directeur de thèse*

institutCurie

Remerciements

Je souhaite commencer cette thèse de doctorat par exprimer quelques remerciements, sans ordre particulier d'importance. Tout d'abord à Andrei, mon directeur de thèse, qui a su m'encadrer et me transmettre au travers de nombreux échanges une part de sa curiosité mathématique et biologique. Également à Olivier et à toute son équipe qui m'ont toujours accueilli chaleureusement, et qui m'ont permis de profiter de leur grande expérience dans l'étude des sarcomes d'Ewing, ainsi que des données qu'ils ont générées; notamment à Lou et Karine qui ont réalisé toute la partie expérimentale sur les lignées cellulaires inductibles, sans qui le chapitre 5 de cette thèse n'aurait pas pu exister. Remerciements également aux membres de l'équipe SysBio passés et présents que j'ai pu côtoyer (malgré ma présence très occasionnelle dans les locaux de Curie), et en particulier Alexander, Cristobal, Jonathan, Loïc, Lucie, Marianyela, Marco et Nicolas, tant pour les collaborations que pour les échanges informels. Grand merci également à Kati et Caroline pour toute l'aide apportée sur le plan administratif sur lequel je ne brille guère – mais je me soigne. Sur le plan plus personnel, j'adresse mes remerciements à Gabrièle ainsi que mes parents, mes grand-parents, mon frère et mes amis, que je n'ai pas pu voir autant que je l'aurais voulu pendant ces trois ans. Finalement, un très grand merci à Valentina et Didier pour avoir accepté d'être rapporteurs de ce manuscrit et à tous les membres du jury ainsi qu'aux membres invités qui permettent la tenue de la soutenance de mon doctorat.

Résumé

Les cellules vivantes sont des éléments fondamentaux de la vie, jouant un rôle à la fois structurel et fonctionnel dans tous les types d'organismes. Décrites pour la première fois au XVIIe siècle par Robert Hooke et malgré la myriade de percées réalisées en biologie cellulaire depuis lors, de nombreux aspects de leur biologie sont encore inconnus aujourd'hui. Les cellules eucaryotes stockent leur information génétique dans des molécules d'ADN enfermées dans leur noyau, qui sont transcrites en molécules d'ARN messager qui servent de plans lors de la synthèse de protéines, une famille de molécules responsables de divers rôles fonctionnels et structurels au sein des cellules. Les progrès technologiques réalisés au cours de la dernière décennie, tels que le séquençage de nouvelle génération (NGS) et l'acquisition de données single-cell, ont ouvert la voie à des jeux de données incroyablement riches, capables de décrire quantitativement des populations cellulaires avec une extrême précision : en une seule expérience, on peut aujourd'hui analyser l'expression génique de dizaines de milliers de cellules sur des dizaines de milliers de gènes. Parallèlement, le domaine de l'apprentissage automatique a connu une vague d'approches nouvelles et revisitées (théorie des réseaux neuronaux profonds, transport optimal, noyaux...), rendue possible par les développements mathématiques et les progrès du matériel informatique. L'un des principaux objectifs de la biologie computationnelle aujourd'hui est de relier ces deux domaines en appliquant des approches d'apprentissage automatique à des ensembles de données biologiques complexes afin de répondre à des questions biologiques difficiles.

L'une des questions clés est celle de l'*intégration des données* qui consiste à concevoir des algorithmes capables de produire une représentation commune de plusieurs ensembles de données provenant de différentes sources ou mesurées selon différentes modalités biologiques, de sorte à ce que des cellules similaires se retrouvent proches les unes des autres indépendamment de leur ensemble de données d'origine. Ce problème est très difficile dans le cas général, et sa résolution a des applications très recherchées telles que la création d'atlas cellulaires complets pour une maladie en agrégeant les données de nombreux patients, ou l'inférence de modèles incluant des facteurs provenant de différentes modalités biologiques. De nombreuses approches ont été proposées pour aborder l'intégration de données au cours des dix dernières années, à tel point qu'en dépit d'études comparatives régulières, il est difficile de savoir ce qu'il convient d'utiliser pour une application donnée. Pour résoudre ce problème, nous avons développé un nouveau framework d'intégration de données appelé *transmorph*, qui fournit de nombreux algorithmes d'apprentissage automatique sous forme de blocs de construction qui peuvent être assemblés en des pipelines d'intégration de données complexes. Nous montrons que *transmorph* peut être utilisé pour construire des pipelines d'intégration de données qui fonctionnent aussi bien que les approches de l'état de l'art, tout en s'avérant utile pour déterminer quelle sous-unité algorithmique est la plus adaptée à une situation donnée. *Transmorph* est aujourd'hui distribué en tant que framework python open-source, et propose un écosystème de jeux de données de référence, de mesures d'évaluation de la qualité, d'outils graphiques ainsi qu'une API utilisateur complète pour construire des modèles d'intégration de données de bout en bout.

Un autre espoir suscité par ces données single-cell est de pouvoir améliorer notre compréhension du cancer, car les tumeurs sont des systèmes cellulaires hétérogènes intégrés dans un microenvironnement complexe. En particulier, des approches d'analyse facto-

rielle peuvent être appliquées pour découvrir des signaux multidimensionnels dans l'espace d'expression génique, qui peuvent ensuite être interprétés à l'aide de bases de données et reliés à des processus biologiques tels que la prolifération cellulaire, l'activité métabolique ou les métastases. En utilisant des lignées cellulaires de sarcome d'Ewing inductibles où l'effet de l'oncogène peut être contrôlé avec précision, nous avons mis en évidence une douzaine de ces processus et étudié leur dépendance à l'égard de l'activité de l'oncogène. Nous avons accordé une attention particulière aux signaux liés à la prolifération, et avons pu mettre en évidence dans de nombreux ensembles de données une trajectoire multidimensionnelle se déroulant dans l'espace d'expression génique correspondant au processus du cycle cellulaire. Nous avons finalement pu dériver de ces observations un modèle du cycle cellulaire segmenté, capable d'approximer l'état des cellules individuelles au sein de leur cycle ainsi que d'autres caractéristiques telles qu'une approximation du temps de doublement des cellules.

Abstract

Living cells are fundamental building blocks of Life, playing both structural and functional roles in all types of organisms. First described in the XVIIth century by Robert Hooke and despite the myriad of breakthroughs that were achieved in Cell Biology since then, many aspects of their biology are still unknown today. Eukaryotic cells store their genetic information within DNA molecules enclosed within their nucleus, which is transcribed into messenger RNA molecules which serve as blueprints for synthesizing proteins, which is a diverse family of molecules responsible for various functional and structural roles within cells. Technological advances that took place during the last decade such as next-generation sequencing (NGS) and single-cell assays opened the door to incredibly rich datasets able to quantitatively describe cell populations with extreme precision: in one experiment, one can today approximate the gene expression of tens of thousands of cells over tens of thousands of genes. In parallel, the machine learning field also witnessed a surge of new and revisited approaches (deep neural networks theory, optimal transport, kernels...), made possible by mathematical and hardware developments. One of the main goals of computational biology today is to link these two fields by applying machine learning approaches to complex biological datasets in order to answer challenging biological questions.

In cancer research, gathering tumor data from different patients yields datasets with intrinsic statistical biases linked to acquisition methods, genetic specificities, or environmental differences. For this reason, jointly analyzing data coming from different sources first requires identifying these biases, preferably automatically, and possibly correcting these biases. One key question referred to as *data integration* is to conceive algorithms able to yield a joint representation of several datasets coming from different sources or measured along different biological modalities, so that similar cells end up close to one another independently from their dataset of origin. This problem is highly challenging in the general case, and solving it has very sought-after applications such as creating comprehensive cell atlases for a disease by aggregating data from many patients, or inferring models including factors from different biological modalities. Many approaches have been proposed to tackle data integration over the last ten years, so much so that despite regular benchmark studies it is puzzling to know what to use for a given application. To tackle this issue we developed a new data integration framework named *transmorph*, that provides many machine learning algorithms as building blocks that can be assembled into complex data integration pipelines. We show that transmorph can be used to build data integration pipelines that work on par with state-of-the-art approaches, while also proving to be useful to determine which algorithmic subunit is more adapted to a given situation. Transmorph is today distributed as an open-source python framework, and embarks an ecosystem of benchmarking datasets, quality assessment metrics, plotting tools as well as a comprehensive user API to build end-to-end data integration models.

Another hope for these highly resolute single-cell assays is that they could improve our understanding of cancer, as tumors are highly heterogeneous cell formations embedded in a complex microenvironment. In particular, factor analysis approaches can be applied to discover multidimensional signals in the gene expression space that can then be enriched using databases, and related to interpretable biological processes such as cell proliferation, metabolic activity or metastasis. Using Ewing sarcoma inducible cell lines where the

oncogene presence can be precisely monitored, we highlighted a dozen of such processes and studied their dependence on the oncogene activity. We paid particular attention to proliferation-related signals, and we could highlight in many datasets a multidimensional trajectory taking place in the gene expression space corresponding to the cell cycle process. We were able to derive from these observations a segment-wise cell cycle model, able to approximate the state of individual cells within the cycle as well as other features such as cell doubling time.

Contents

Remerciements 1

Résumé 3

Abstract 5

Abbreviations and conventions 11

Manuscript organization and publications 13

1 Introduction 15
1.1 A new generation of biological data . . . 17
1.1.1 Bulk mRNA sequencing . . . 17
1.1.2 The single-cell revolution . . . 18
1.1.3 Processing of single-cell data . . . 20
1.1.4 Deciphering dynamical cell processes using scRNA-seq data . . . 21
1.2 Ewing sarcoma is an aggressive pediatric tumor . . . 23
1.2.1 Ewing sarcoma's features . . . 23
1.2.2 Multidimensional factor analysis of Ewing sarcoma cell processes . . . 25
1.3 Integration of single-cell data . . . 29
1.3.1 Data integration links biological datasets across batches or modalities 29
1.3.2 Horizontal integration (HI) links batches anchored by their common modality . . . 32
1.3.3 Vertical integration (VI) connects modalities measured in the same cells . . . 34
1.3.4 Diagonal and mosaic integration jointly embed non- or partially-anchored datasets . . . 36

2 *Transmorph*, a novel framework to perform integration of single-cell data 41
2.1 *Transmorph*: concept and architecture . . . 42
2.1.1 *Transmorph* allows conceiving end-to-end data integration models . 43
2.1.2 Package implementation . . . 45
2.2 *Transmorph* algorithms . . . 46
2.2.1 Transformations . . . 46
2.2.2 Matchings . . . 49
2.2.3 Mergings . . . 53
2.2.4 Embedding evaluator . . . 56
2.3 A few real-life applications of the *transmorph* framework . . . 58
2.3.1 Single-cell RNA-seq datasets . . . 58
2.3.2 *transmorph* models perform on par with other state-of-the-art tools . 58
2.3.3 Performing integration in gene space by using an appropriate embedding . . . 61
2.3.4 Gene space integration can be leveraged to annotate cell types reliably 63

2.3.5 Transferring cell cycle phase annotations across osteosarcoma and Ewing sarcoma datasets . . . 64
2.3.6 Data integration of Ewing sarcoma datasets . . . 67
2.4 Discussion . . . 68

3 Unsupervised weights selection for optimal transport-based dataset integration 71
3.1 General outline of the suggested single-cell dataset integration methodology 72
3.2 Method for kernel density uniformization . . . 74
3.3 Bandwidth selection . . . 75
3.4 Quadratic program greatly reduces kernel density empirical variance . . . 75
3.5 Weighted dataset integration . . . 76
3.6 Integration results on synthetic datasets . . . 77
3.7 Integration results in cell cycle space . . . 78
3.8 Integration results on balanced single-cell multi-omics datasets . . . 78
3.9 Integration results on unbalanced single-cell multi-omics datasets . . . 79
3.10 Materials and methods . . . 80
3.10.1 Datasets . . . 80
3.10.2 Synthetic datasets . . . 80
3.10.3 Ewing sarcoma single-cell datasets . . . 80
3.10.4 Multi-omics scSNAREseq dataset . . . 80
3.10.5 Optimal transport . . . 81
3.10.6 Gromov-Wasserstein problem . . . 81
3.10.7 Unbalanced optimal transport . . . 82
3.10.8 Gaussian kernel bandwidth selection . . . 82
3.10.9 Assessing integration quality in scSNAREseq data . . . 83
3.10.10 Assessing the computational time . . . 83
3.11 Discussion about this data integration approach . . . 83

4 Modeling progression of single cell populations through the cell cycle as a sequence of switchess 85
4.1 Background . . . 86
4.2 Methods and materials . . . 87
4.2.1 Single-cell data used in this study . . . 87
4.2.2 Definition of cell cycle genes . . . 88
4.2.3 Pooling reads from neighboring cells for compensating the technical drop-out . . . 88
4.2.4 Cell cycle trajectory-based single-cell data normalization . . . 88
4.2.5 Computing the cell cycle trajectory and quantifying pseudotime . . . 89
4.2.6 Curvature analysis of the cell cycle trajectory . . . 90
4.2.7 Estimating the effective dimensionality of a set of vectors . . . 90
4.3 Example of a cell cycle trajectory extracted from single-cell data . . . 90
4.4 Model of cell cycle with switches and divisions . . . 91
4.5 Simple example of dynamics with switches and cell division events . . . 94
4.6 Two-dimensional model of cell cycle progression . . . 96
4.7 Effective dimensionality versus number of states . . . 97
4.8 Kinetic model of cell cycle at transcriptomic level . . . 100
4.9 Fitting parameters of the kinetic cell cycle model . . . 102
4.10 Simulating cell cycle trajectories of various durations . . . 104
4.11 Predicting cell line doubling time from the geometrical properties of cell cycle trajectory . . . 106
4.12 Discussion . . . 107

5 Uncovering Ewing sarcoma cell processes using inducible cell lines 109
5.1 Inducible Ewing sarcoma cell lines, a time-resolved study 109
5.1.1 Experimental setup 109
5.1.2 Quality control and data exploration 110
5.2 Identifying Ewing sarcoma transcriptional signatures 111
5.2.1 Inducible Ewing sarcoma cell lines setup 112
5.2.2 Indentifying consensual independent components 112
5.2.3 From independent components to transcriptional signatures 113
5.3 New Ewing sarcoma cell processes have been identified 115
5.3.1 New signatures cover various Ewing sarcoma cell processes 116
5.3.2 Factors not directly related to the high expression of EF1 119
5.4 Summary of Ewing sarcoma cell processes 119

6 Discussion and conclusion 125
6.1 The future of data integration 125
6.2 Studying the cell cycle in the gene expression space: future challenges . . . 126
6.3 The single-cell asset 127

Bibliography 129

Appendix 149

Abbreviations and conventions

.
This section contains abbreviations and mathematical conventions used in this Ph.D. thesis, and apply unless explicitly stated otherwise.

Abbreviations

- **API:** Application Program Interface
- **ATAC-seq:** Assay for Transposase-Accessible Chromatin using sequencing
- **CCA:** Canonical Correlation Analysis
- **CCT:** Cell Cycle Trajectory
- **cDNA:** Complementary DNA
- **DAE:** Deep Autoencoder
- **DI:** Diagonal data Integration
- **DL:** Deep Learning
- **EMT:** Epithelial-Mesenchymal Transition
- **EWS:** Ewing Sarcoma
- **EF1:** Chimeric protein EWSR1-FLI1
- **HGP:** Human Genome Project
- **HI:** Horizontal data Integration
- **GEO:** Gene Expression Omnibus
- **GW:** Gromov-Wasserstein
- **ICA:** Independent Component Analysis
- **KNN:** k-Nearest Neighbors
- **LISI:** Local Inverse Simpson's Index
- **MF:** Matrix Factorization
- **MI:** Mosaic data Integration
- **ML:** Machine Learning
- **MNN:** Mutual Nearest Neighbors
- **MSC:** Mesenchymal Stem Cell

- **NGS:** Next-Generation Sequencing
- **OT:** Optimal Transport
- **PCA:** Principal Component Analysis
- **PDX:** Patient-Derived Xenograft
- **RNA-seq:** RNA-sequencing
- **scRNA-seq:** single-cell RNA-sequencing
- **shRNA:** Small Hairpin RNA
- **SNP:** Single-Nucleotide Polymorphism
- **TS:** Transcriptional Signature
- **UMI:** Unique Molecular Identifier
- **VI:** Vertical data Integration

Mathematical conventions

- S (uppercase) designates a set
- $\mathbb{N}$ designates the set of non-negative integers, $\mathbb{R}$ designates the set of real numbers
- For any set S and $k \in \mathbb{N}$, S^k represents the set of vectors of length k whose coordinates belong to S. For any $n \in \mathbb{N}$, $m \in \mathbb{N}$, $S^{n \times m}$ represents the set of rectangular matrices of n rows and m columns whose elements belong to S.
- $\mathbf{A}$ (uppercase, bold) represents a rectangular matrix. $\mathbf{A}^T$ is its transpose.
- $\mathbf{a}$ (lowercase, bold) represents a column vector. $\mathbf{a}^T$ is its transpose (it is a row vector).
- a (lowercase, italic) represents a scalar value.
- $\mathbf{X} \in \mathbb{N}^{n_0 \times d_0}$ represents a raw counts, unfiltered matrix of n_0 cells (in row) by d_0 transcripts (in columns).
- $\mathbf{X} \in \mathbb{N}^{n \times d}$ represents a raw counts matrix of n cells (in row) by d transcripts (in columns).
- $\mathbf{X} \in \mathbb{R}^{n \times d}$ represents a row-normalized, log-scaled gene expression matrix of n cells (in row) by d transcripts (in columns).
- $\{a, b, c\}$ represents the set of items a, b and c.
- $\{a_1, \ldots, a_k\}$ with k a positive integer represents the ordered set of k items a_1 to a_k.
- $\mathbf{0}_k$ represents the column vector of size k where all coefficients are set to 0.
- $\mathbf{1}_k$ represents the column vector of size k where all coefficients are set to 1.

Manuscript organization and publications

This manuscript contains a mixture of published articles, unpublished manuscripts, and original sections; this section details its organization, as well as my personal contributions to the articles.

Text from the paper (Aynaud et al., 2020) is not included in this manuscript, as it was published prior to my PhD and my contributions to the study were minor. The paper (Mirkes et al., 2022) is not included in the manuscript either, as I only contributed to software implementation, not to the algorithm conception.

- Sections 1.1 and 1.2 are original to this manuscript.
- Section 1.3 is the review (Fouché and Zinovyev, 2023) that I wrote for a special issue of *Frontiers in Bioinformatics*, in which I provide an overview of data integration methods from a machine learning point of view.
- Chapter 2 is an extended version of the paper (Fouché et al., 2023), published in *Nuclear Acids Research: Genomics and Bioinformatics*, where I present *transmorph* which is the data integration framework we developed.
- Chapter 3 contains (Fouché and Zinovyev, 2021), an unpublished manuscript where I propose a kernel-based data integration approach that leverages unbalanced optimal transport to align cell cycle trajectories.
- Chapter 4 is the paper (Zinovyev et al., 2022) which present a computational model of the cell cycle, where I participated in the model elaboration, the code implementation and the data analysis.
- Chapter 5 is original to this manuscript.

Chapter 1

Introduction

Section 1.3 adatped from (Fouché and Zinovyev, 2023).

The last decades have witnessed many exciting discoveries in the field of Biology, which transformed the way we study life and in particular how we study living cells. The Human Genome Project (HGP) was an ambitious scientific program that started during the 1990s (Council et al., 1988) and was completed in the early 2000s (Sequencing, 2004). It unveiled the first full sequence of a Human genome, containing more than three billion nucleotides, for a total cost of approximately three billion US dollars. The HGP alone had a huge impact on biology and medicine (Hood and Rowen, 2013) but in just two decades, DNA sequencing technologies have rapidly evolved to the point we can today sequence an entire human genome in less than a day and for less than 1,000 US dollars. In parallel, a myriad of other sequencing technologies and biological assays appeared, allowing researchers to gather data from many more biological modalities such as gene expression (Klein et al., 2015; Macosko et al., 2015), chromatin accessibility (Buenrostro et al., 2015a,b), DNA methylation (Guo et al., 2013), protein abundance (Aebersold and Mann, 2003; Westermeier and Marouga, 2005; Tibes et al., 2006), or lipidomics (Wenk, 2005). Finally, sequencing pipelines also progressed in terms of resolution, resulting in being able to perform analysis at the level of individual cells (Stuart and Satija, 2019). Thanks to these successive breakthroughs, the central molecular dogma of molecular biology (Fig. 1.1) formulated by Francis Crick 70 years ago (Crick, 1958) can now be appreciated with exquisite precision.

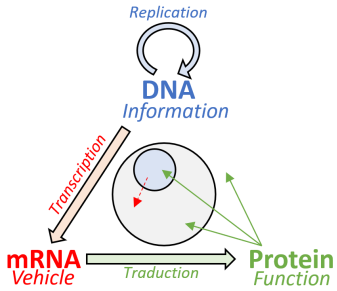

Figure 1.1: **Molecular biology's central dogma.** Proteins are molecules that carry out many biological functions such as structure, signaling, or biochemical reactions. These proteins are sequences of amino acids whose blueprints are stored within the cell's genome, as long DNA molecules enclosed within the cell's nucleus. This genetic information is carried out of the nucleus for protein synthesis via small intermediary nucleic acids, the messenger RNAs.

In this new era of ever-growing information, both in terms of quantity and complexity, computer algorithms have become essential to harness biological data. Biologists have used biostatistics for a long time to formulate and challenge their hypotheses. As data be-

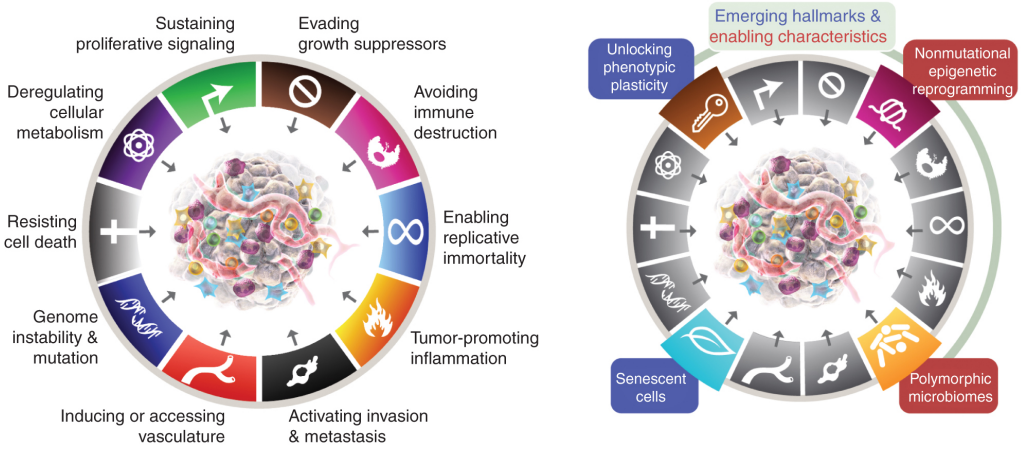

Figure 1.2: **Hallmarks of cancer as of today**, summarizing the various deregulations cells can undergo that favoritizes tumorigenesis. *From (Hanahan, 2022)*

came more and more complex, computational systems biology (Kitano, 2002) proved to be an invaluable tool to describe and predict the behavior of many biological systems thanks to advanced mathematical models and algorithms. In recent years, high-throughput sequencing allowed scientists to generate enormous amounts of data, causing machine learning (ML) concepts to occupy a large part of computational biology. Today, almost all ML paradigms have found practical uses in challenging fields such as neuroscience (Vu et al., 2018) or cancer research (Kourou et al., 2021).

Cancer is a worldwide issue with dramatic health, social and economic consequences. It is estimated every one out of five people will develop cancer during their life, which translates to several tens of millions of new patients per year; millions eventually die from it (WHO, 2018). Cancer occurrence can be challenging to predict as they are usually highly multifactorial diseases, with many factors having been identified as favoriting its onset: aging, genetics, chemical reagents, diet, comorbidities, pathogens, radiation, and hormones, to cite the main ones (Stein and Colditz, 2004). Furthermore, our understanding of tumorigenesis is still evolving today, as indicated by the frequent updates to the Hallmarks of Cancer (Hanahan and Weinberg, 2000, 2011; Hanahan, 2022) summarized in Fig. 1.2. Tumors are complex biological systems that are also capable of interacting with other systems such as the immune system or the vascular system, which makes it even harder to decipher the tumorigenesis process precisely.

Cancer can occur in children, teenagers, and young adults despite being frequently associated with aging. These pediatric cancers are rarer, but they still represent tens of thousands of new young patients yearly and unfortunately thousands of deaths (NCI, 2023). Thanks to the progress in cancer treatment, pediatric cancer prognosis has significantly improved in the last few decades, with a survival rate of 58% of children and 68% of adolescents in the 1970s to 85% of children and adolescents today. The most frequent pediatric cancer types are leukemia, brain and spinal cord tumors, neuroblastoma, Wilms tumor, lymphoma (including both Hodgkin and non-Hodgkin), rhabdomyosarcoma, retinoblastoma and bone cancers (including osteosarcoma and Ewing sarcoma) (ACS, 2023). The Curie Institute is a reference hospital and research center for pediatric cancers in France, and many important contributions in the field involve researchers from the institute.

This introduction will first present the nature of today's biological data types. We will detail in this first section the process of mRNA sequencing (RNA-seq) and single-cell data acquisition, and will present the basic principles of single-cell RNA-seq (scRNA-seq) data analysis. In the next section, we will introduce Ewing sarcoma, an aggressive pediatric tumor I studied during my Ph.D., and notably discuss its features and its well-identified oncogene, EWSR-FLI1 (EF1). We will introduce in the third section the core concept of cell process-associated transcriptional signatures, as well as its applications to study cell

processes like transcriptional cell cycle or tumoral heterogeneity. The final part of this introduction will be dedicated to questions related to single-cell data integration, which describes the critical problem of merging datasets across origins and modalities.

1.1 A new generation of biological data

Over the last decade, there has been a surge in the amount of biological data available to study biological systems. First, more and more data can be acquired from smaller and smaller systems. If, in the past, biology was the study of living individuals and their tissues, modern technologies allow scientists to study living cells and their components at the molecular level. This results in a steep increase in the information gathered per quantity of biological material available. Going hand-to-hand with this first observation, data collection, processing, and analysis is today easier than ever thanks to the myriad of technologies researchers have at their disposal, such as high-resolution assays doable in routine and efficient bioinformatics pipelines. Finally, biological data is now not only multi-dimensional (measuring several features from a single biological modality such as gene expression) but also multi-modal, which means several biological modalities can be acquired from a single biological sample. All these facts make it so research centers can produce daily gigabytes of biological data, which brings a crucial need for efficient data analysis methods to digest this raw data into understandable and interpretable information.

This section first presents the principles underlying mRNA sequencing, as this data modality is the main focus of my work. I will then introduce single-cell assays that proved to be invaluable in getting an insight into tissue heterogeneity. Finally, I will cover the state of single-cell RNA-seq data analysis, with the typical preprocessing steps, tools, and the limits of this type of analysis.

1.1.1 Bulk mRNA sequencing

Molecular biology's "central dogma", stated by Francis Crick in the 1950s, describes genes as elements of information stored within DNA molecules, expressed as messenger RNA (mRNA) molecules, further translated into proteins that carry out biological functions (Lodish et al., 2008) (Fig. 1.1). For this reason, knowing at a given time which genes are expressed within a living cell provides an insight into which biological processes are either occurring or about to occur within this cell.

Bulk mRNA sequencing describes a type of biological assay that is used to assess the mRNAs present in a mixture of many cells. This type of experiment is widely used today, as it is a relatively easy-to-carry-out assay to compare two cell populations at the molecular level. The main limitation of bulk RNA-seq is that it makes it difficult to account for the heterogeneity of cells within the mixture; its single-cell counterpart addresses this. The complete set of mRNA molecules present within a mixture at a given time are referred to as this cell's *transcriptome*, and can be represented as a long integer vector $\mathbf{x} \in \mathbb{N}^m$ with m being the total number of unique transcripts $\{g_1, \dots, g_m\}$ that can be expressed within this cell; in practice, m can be several tens of thousand. In this formalism, each coordinates x_i of the expression vector $\mathbf{x}$ represents the number of mRNA molecules sequenced corresponding to the gene g_i. This molecular profile can then be used to characterize cells in the mixture using statistical analyses, and to compare different cell populations at the gene expression level.

I will now describe a standard bulk RNA-seq acquisition procedure, based on the state-of-the-art review (Wang et al., 2009) (Fig. 1.3). Starting from a cell mixture, the first step consists in the preparation of a *complementary DNA* (cDNA) library from the transcripts that are found within the mixture. The isolation of RNA molecules is carried out by degrading DNA molecules using DNAses. Mature mRNA molecules are then selected by targetting their 3'-polyadenylated tails using poly-T oligomers bound to a substrate.

During the next step, isolated mRNAs are reverse-transcribed into cDNA molecules, that are subsequently amplified, then fragmented and selected according to a length criterion. This allows these molecules to be sequenced efficiently using standard high-throughput DNA sequencing technologies.

Once cDNA sequences have been obtained, transcriptome assembly can be carried out by aligning these sequences onto a reference genome in order to determine the corresponding genes. The main caveat at this step is that due to mRNA splicing, query cDNA sequences do not align contiguously with the reference genome because introns are missing from the cDNA. To circumvent this issue, non-contiguous sequence alignment algorithms such as TopHat (Trapnell et al., 2009) use a so-called seed-based heuristic that aligns subsequences of the query sequence onto the reference one, then extend from these seeds using dynamic programming to find potential matches. Estimated transcript counts can eventually be assessed in a final step, thanks to various tools such as HTseq (Anders et al., 2015).

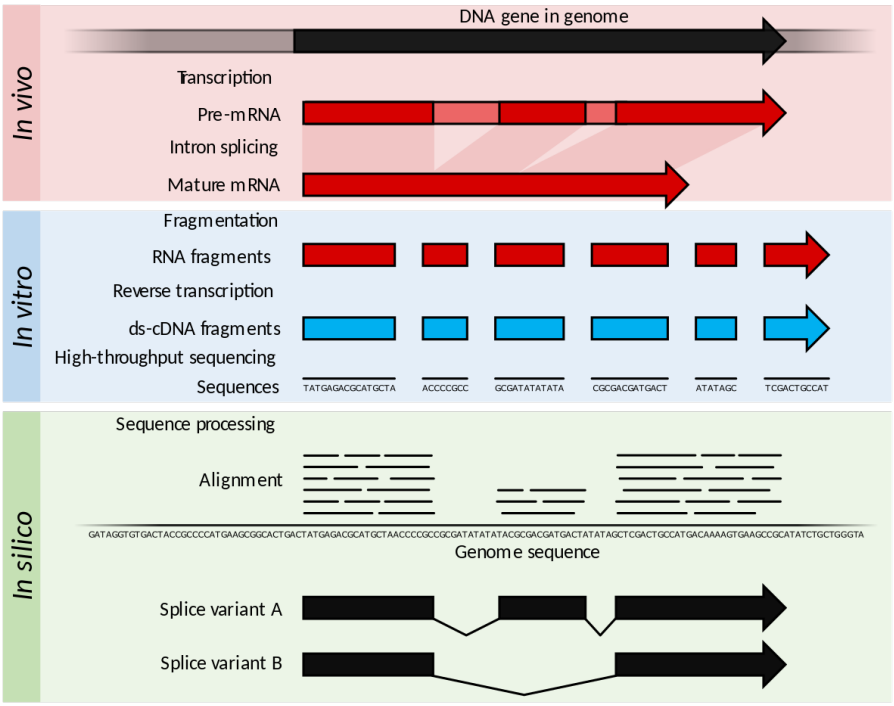

Figure 1.3: **Typical mRNA sequencing pipeline.** *Image credits: Thomas Shafee, Wikipedia.*

1.1.2 The single-cell revolution

Single-cell RNA sequencing

If bulk RNA-seq provides a strong basis to profile mixtures of cells and highlight strong gene expression differences between several experimental conditions, it is of limited interest to explore a single condition containing a heterogeneous cell population, for instance when multiple cell types are in the cell mixture. In this instance, the resulting bulk profile typically consists of a weighted average of the different cell types' profiles, which can be challenging to deconvolute properly. If measuring mRNA expression within single cells has been discussed for quite some time now (Eberwine et al., 1992; Kurimoto et al., 2006), it has only been a few years since platforms supporting this technology have been made available in research centers. Conducting single-cell RNA-seq (scRNA-seq) experiments allowed scientists to get an exquisite insight into the heterogeneity of the cells contained within a tissue of interest, both in terms of cell types and cell states.

The main difference in scRNA-seq assays compared to bulk assays is the execution of a preliminary step consisting in the isolation of single cells before mRNA sequencing occurs. Recent scRNA-seq platforms, such as the 10X pipeline available at the time of writing at Institut Curie, encapsulate individual cells into lipidic droplets endowed with DNA molecules composed of a unique nucleic acid sequence that serves as molecular barcodes to identify each cell in the analysis (Klein et al., 2015; Macosko et al., 2015). Reverse transcription using oligo-deoxythymine (oligo-dT) primers and cDNA library preparation occurs within each oil droplet, followed by the usual sequencing and alignment steps.

Using high-throughput scRNA-seq pipelines such as 10X typically yields a few hundred to several thousand cells per acquisition, with tens of thousands of genes measured. This provides a multidimensional, comprehensive picture of the cell population sample, which can then be analyzed in depth using advanced data science and machine learning techniques. The following section will cover the basics of scRNA-seq data analysis, as well as the primary tools data scientists have at their disposal to decipher the biological signals present in scRNA-seq datasets.

If scRNA-seq data has many advantages compared to bulk RNA-seq, it also presents several limitations. First and foremost, scRNA-seq experiments are heavier and costlier than bulk ones and necessitate specific platforms and pipelines. In addition, scRNA-seq datasets are typically much larger in storage as they can easily reach several gigabytes per experiment. For these reasons, acquiring scRNA-seq data should always be motivated by precise needs and be carried out on biological samples presenting high enough cell heterogeneity. Furthermore, due to the scarce amount of biological material available within a single cell compared to bulk assays, scRNA-seq yields way fewer transcripts per cell than its bulk counterpart. This results in important variations, especially for genes with low total counts. In the most extreme cases, notable side effects such as the so-called *dropout effect* (Kharchenko et al., 2014) can occur. This describes genes expressed in the cell but not detected in the experiment due to the small number of transcripts available, thus being falsely considered as non-expressed. Other artifacts can arise within scRNA-seq pipelines, such as the sequencing of apoptotic cells, or *doublets* records that characterize the event where two cells are captured within a single oil droplet. For these reasons and in order to make scRNA-seq data exploitable, several preprocessing steps are usually carried out to detect and correct these issues, and will be discussed in the next section.

Other single-cell assays

This last decade has also witnessed a sharp increase in the amount and in complexity of the data produced for cellular biology, thanks to an ever-growing number of bulk and single-cell profiling assays. These technologies allowed scientists to study heterogeneous cell populations through many biological feature spaces (or *modalities*) such as mRNA expression (Klein et al., 2015; Macosko et al., 2015), but also DNA methylation (Guo et al., 2013) and chromatin accessibility (Buenrostro et al., 2015a,b), and protein abundance (Aebersold and Mann, 2003; Westermeier and Marouga, 2005; Tibes et al., 2006). These assays can also be carried out either in bulk, which yields a single averaged molecular profile for each sample, or at the single-cell level, which provides an exquisite insight into cell states and types present in the cell population.

Thanks to this panel of biological modalities that are today available for studying living cells, it is now possible to study biological processes through the prism of many different molecular mechanisms. We already discussed how monitoring gene expression could give an insight into the molecular actors orchestrating a biological process, but other modalities can also bring additional complementary information such as genetic variants, epigenetic regulation, or kinase activity. The big challenge is then to figure out how the actors from the different modalities interact together to allow a biological process of interest to take place.

In addition, during the last few years, there have been several joint assays proposed

to profile single cells through several modalities simultaneously, such as scM&T-seq for transcriptome and methylome (Angermueller et al., 2016), sc-GEM for genotype, transcriptome and methylome (Cheow et al., 2016), CITE-seq for transcriptome and surface proteins (Stoeckius et al., 2017), or SNARE-seq for transcriptome and chromatin accessibility (Chen et al., 2019b).

It is also worth mentioning spatial transcriptomics, which yields for each well measurements from a small number of cells while also providing positional information of cells within the biological tissue (Ståhl et al., 2016). Finally, important phenotypical information can be obtained from microscopic imaging data, such as whole slide imaging (Pantanowitz et al., 2011). These two modalities bring precious insights into tissue structure that can be notably leveraged to infer cell-cell interactions, which is a piece of information that is typically difficult to access using other modalities.

1.1.3 Processing of single-cell data

Let us now go back to scRNA-seq data; once a scRNA-seq dataset has been acquired and processed into a raw count matrix $\mathbf{X}_{\text{raw}} \in \mathbb{N}^{n_0 \times d_0}$ of n_0 cells measured on d_0 transcripts, additional preprocessing steps must be carried out to make them exploitable for subsequent analyses. Several scRNA-seq data analysis tools can be used to facilitate these preprocessing steps, the main one being Seurat (Satija et al., 2015; Butler et al., 2018; Stuart et al., 2019; Hao et al., 2021) for the R language and Scanpy (Wolf et al., 2018) for the Python language. Both of these packages are quite interchangeable in terms of features, and I personally choose Scanpy for language convenience.

ScRNA-seq data preprocessing usually starts with the filtering of both cells and genes, which serves several purposes. First, low-quality cells are removed from the dataset which limits their impact during the statistical analyses:

- Cells with too few gene counts are taken out, as they can be artefactual and/or may not carry sufficient statistical power.
- Cells with too high gene counts are presumed to be doublets and are taken out, i.e. two or more cells captured within in a single oil droplet.
- Apoptotic cells, which can be detected using their high expression of mitochondrial genes, are also removed.

A second filtering pass is then applied to genes: typically, genes expressed in too few cells or genes that do not vary widely among the cells are removed from the analysis. In the end of this filtering step, the raw count matrix $\mathbf{X}_{\text{raw}} \in \mathbb{N}^{n_0 \times d_0}$ has been sliced into a filters count matrix $\mathbf{X}_{\text{filtered}} \in \mathbb{N}^{n \times d}$, with n selected cells and d selected transcripts. It is important to keep in mind that all these filtering steps are based on arbitrary *ad hoc* thresholds that are decided by the data analyst, often based on visual inspection of statistical plots. For this reason, it is always necessary to take them with a grain of salt, and to acknowledge the inevitable biases these steps introduce into the data.

The next preprocessing step consists in transforming raw count values contained in $\mathbf{X}_{\text{filtered}}$ into comparable, pseudo-continuous values to make statistical analyses easier. We often first rescale $\mathbf{X}_{\text{filtered}}$ rows so that for each cell, its counts sum up to a fixed value (typically 10^4). This normalizes the base transcriptional activity between all cells in the dataset, which facilitates comparing the expression of individual genes between cells. There are debates about the well-fondness of this step, notably because this variation of gene expression can carry relevant biological information, as pointed out in (Hafemeister and Satija, 2019). For instance, we showed that total counts measured in a cell are highly correlated to its position within the cell cycle (Zinovyev et al., 2022), which is important biological information. For these reasons, more advanced count normalization procedures are today available in state-of-the-art scRNA-seq analysis packages; additional information on this topic can be found in (Hafemeister and Satija, 2019).

Once gene counts have been normalized, it is possible to carry out a *pooling* step, where for every cell, its global counts are pooled towards the barycenter of this cell's k-nearest neighbors (including the cell itself), either using the average or the median for more robustness to extreme values. Pooling helps to correct outlier or artefactual values such as the ones caused by dropout, but should be carried out carefully by monitoring the k value in order to avoid blurring the data too much (for $k = 1$, data is untouched while for $k = n$, all cells are projected into the dataset's barycenter). More advanced denoising methods than barycentric pooling can also be mentioned, such as MAGIC (Van Dijk et al., 2018) or DCA (Eraslan et al., 2019). Counts are finally individually logarithmized via the mapping $x \mapsto \log(x + 1)$, facilitating the integration of genes expressed across different orders of magnitude.

This standard preprocessing pipeline outputs a processed expression matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$, and this algorithm can be adapted by removing or adding computational steps such as z-score computation or data denoising depending on the quality of the data, and needs of the application. Overall, it is crucial to remember that typically, if two data scientists process the same dataset, they generally will not end up with the same expression matrix in the end. Indeed, the final result depends on the choice of several *ad hoc* thresholds and decisions taken by the analyst throughout the process. In the two following sections, we'll review how preprocessed expression matrices $\mathbf{X}$ can be used to study tissue heterogeneity as well as dynamical processes that occur at the single-cell level.

1.1.4 Deciphering dynamical cell processes using scRNA-seq data

In addition to the benefits of providing insight into the heterogeneity within a cell population, single-cell assays also provide a way to follow the progress of a biological process at the molecular level. In a similar way as chronophotographs allow tracking the dynamics of a mechanical trajectory (Fig. 1.4), each cell in a single-cell dataset can be seen as a snapshot of one state of a biological process. This idea has been leveraged in diverse applications such as lineage tracing (Schiebinger et al., 2019), transcriptional dynamics (La Manno et al., 2018) or inference of transcriptional trajectories (Chen et al., 2019a).

Transcriptomic trajectories come in various types and shape, such as linear sections (e.g., apoptosis process), branching sections (e.g., immune cells differentiation), or cyclic sections (e.g., cell cycle); some processes can be explained by just one of these types while more complex ones may necessitate combining several of them. The first important problem in this topic is to infer trajectories followed by cells from a scRNA-seq dataset solely based on their position in the gene expression space. The general idea of most approaches proposed to this day is to learn a topology that best fits the geometry of the data. Elastic principal graphs (Gorban et al., 2007) provide a powerful framework to approximate point clouds with tree-like structures and have been shown to discover relevant trajectories from scRNA-seq data (Chen et al., 2019a).

The second important question is to determine the direction of cells along the trajectory as well as its irreversibility. Some transitions are irreversible like apoptosis or cell cycle, while others are not such as glucose metabolism regulation. This also brings the question of the detection of the starting points (or initial states) and ending points (or final states) of the trajectory, which is highly non-trivial.

Finally, the question of trajectory local dimensionality must be addressed. Indeed, not all trajectories are best explained by a local dimensionality of 1 like tree-like trajectories. In fact, we observe dimensionality can vary to higher orders at different points of the trajectory. This problem has notably been discussed in (Bac et al., 2021), where an algorithm is proposed to detect the local intrinsic dimensionality of cellular trajectories.

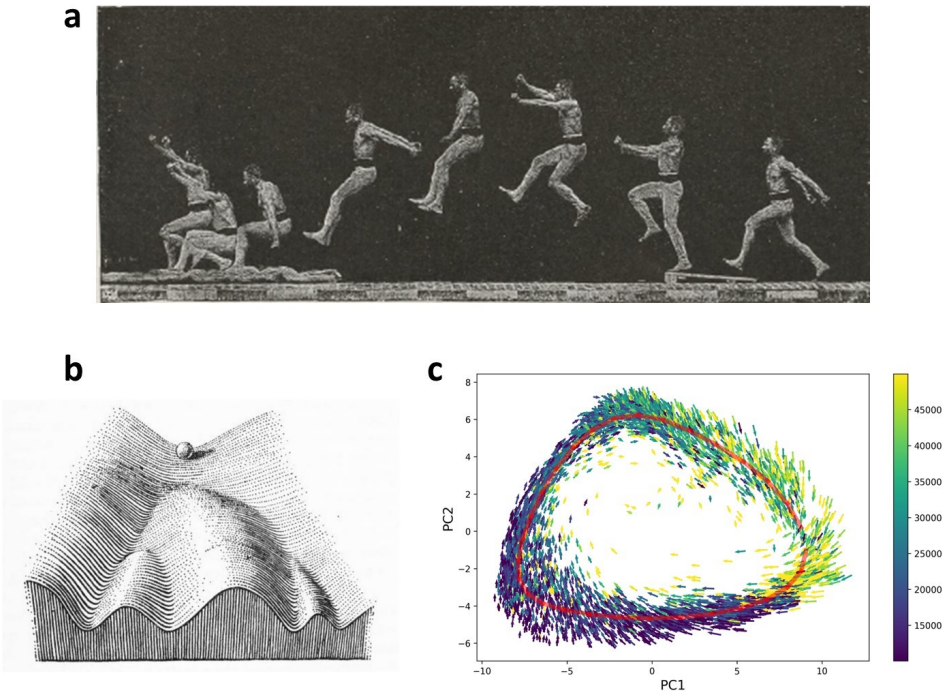

Figure 1.4: **Single-cell data analysis provides a snapshot of cells in different transcriptional states, from which trajectories can be derived.** (a) Man jumping, chronophotography by Etienne-Jules Marey, circo 1887. (b) Waddington landscape is a cell differentiation model where a cell is modeled as a ball rolling downhill following the curvature of an imaginary transcriptional landscape while differentiating. A scRNA-seq assay snapshots the ball position at a point of its trajectory. (c) CHLA9 Ewing sarcoma cell line embedded in a PCA space, each arrow represents a cell with the tip pointing towards the first derivative of its transcriptome based on RNA velocity (La Manno et al., 2018); cells are colored by number of transcripts measured. We clearly see the cell cycle trajectory appear in this representation.

1.2 Ewing sarcoma is an aggressive pediatric tumor

Ewing sarcoma is a pediatric bone tumor that was first described in the 1920s by James Ewing (Ewing, 1972), an American pathologist. Based on the primer (Grünewald et al., 2018), this section starts by presenting the general features of this disease (epidemiology, semiology, prognosis, and treatment). We will then focus on the mechanisms by which its well-characterized molecular oncogene, the chimeric protein *EF1*, deregulates the whole cell's transcriptome. We choose to dedicate the last part of this section to present a study we conducted shortly before starting this Ph.D., during which we identified a set of gene signatures associated with various Ewing sarcoma cell processes, notably leveraging multidimensional factor analysis.

1.2.1 Ewing sarcoma's features

Epidemiology

Ewing sarcoma is a malignant pediatric bone (mainly occurring in the pelvis, femur, tibia, humerus, fibula, and ribs) and soft tissue (mainly thoracic wall, gluteal muscle, pleural cavities, and cervical muscles) tumor, that ranks second in frequency among tumors that occur during childhood and adolescence (70% of the Ewing sarcoma cases occur between 5 and 25 years old, with peak incidence at age 15). It affects approximately 1.5 children, adolescents, and young adults per million, with 80 to 100 new patients per year in France (IGR, 2023), with a slightly superior occurrence in male individuals (Jawad et al., 2009). In 20 to 25% of the cases, tumors are already metastasized at the time of diagnosis (Gaspar et al., 2015), with primary metastasis targets being lungs and bone marrow.

Ewing sarcoma also occurs more frequently in individuals of European origin, and less frequently in individuals of Asian or African origin, which suggests the existence of genetic variants that increase the risk of developing this cancer (Jawad et al., 2009; Fraumeni and Glass, 1970; Jensen and Drake, 1970; Worch et al., 2011). There are rare cases reported of familial clustering, and no clear environmental factors have been highlighted to this day (Joyce et al., 1984).

Semiology and diagnosis

Ewing sarcoma symptoms are diverse, including local pain (notably when walking when the tumor is located in the legs) and swelling. It is frequent for the pain to be interpreted as being related to bone growth, or as an injury (Grünewald et al., 2018). Primary tumors developed by adolescents and young adults tend to occur more frequently in the pelvis and the axial skeleton, while also being prone to develop in soft tissues (Rochefort et al., 2017). According to the Grünewald primer, Ewing sarcoma is also not associated with typical B symptoms (fever, night sweats, and weight loss) until an advanced stage of the disease, or once the tumor has metastasized.

If, in many patients, the tumor can be detected by palpation, it can be left undiagnosed for a long time when it has developed more deeply within larger bones. According to (Widhe and Widhe, 2000), the median time to diagnosis is about 3 to 6 months. In general, the diagnosis of Ewing Sarcoma is performed using a radiological evaluation that can also reveal the presence of metastases. The tumor can be seen in radiography as a mass within the bone, with a multilayered appearance (in 'onion skin').

Prognosis and treatment

Ewing sarcoma is highly aggressive, with around 75% survival rate at 5 years when the cancer is not metastasized and in the absence of comorbidities, and 30% to 50% when the tumor has metastasized depending on the location of the metastasis; interestingly, time to diagnosis does not seem to influence the survival rate (Brasme et al., 2014). Tumors in

children tend to be associated with better outcomes, while those developed by adolescents and young adults are typically larger and more severe. Also, some locations are more severe than others: for instance, pelvic tumors are associated with a lower survival rate (Grünewald et al., 2018). According to (Gaspar et al., 2015), estimating the life expectancy of survivors is still a challenging question.

Localized and metastasized Ewing sarcoma can be treated clinically (SCC, 2023). Most of the time, chemotherapy is used in order to evaluate the tumor evolution afterward using RMI or PET scan. If the tumor has stopped growing, it is surgically removed if possible, treated with radiotherapy otherwise (for instance when the tumor is located in the spine or in the pelvis). If the tumor continues to grow or if it has metastasized, additional chemotherapies may be necessary, with the help of surgery and radiotherapy to control the tumor's growth.

Ewing sarcoma's oncogene, EF1, is well-identified

Ewing sarcoma is a cancer type caused in most cases (85%) by a well-characterized oncogene, a chimeric fusion protein called *EWSR1-FLI1* (also known as *EF1*, caused by a chromosomic fusion between the transcription factor *FLI1*, whose gene is located on the chromosome 11, and the protein *EWSR1*, whose gene is located on the chromosome 22 (Delattre et al., 1992). The *EF1* protein acts as a transcription factor with high affinity for some genome regions, notably those containing GGAA microsatellites, known as EWSR1-FLI1 response elements. It is thought that EF1 deregulates the expression of many genes by binding onto the genome in these various genomic regions, causing critical modifications to the cell's phenotype that eventually lead to tumorigenesis.

We studied in the past using single-cell RNA-seq data the deregulation of cell processes in the presence of EF1 (Aynaud et al., 2020). We notably observed in this study that EF1 seemed to modulate the expression of proliferation-related genes, as well as genes involved in other metabolic pathways such as oxidative phosphorylation, hypoxia, extracellular matrix organization, and mRNA transcription. We were also able to better characterize the transcriptomic intratumoral heterogeneity of Ewing sarcoma, notably via matrix factorization methods that allowed us to define sets of genes contributing to specific cellular processes. We will discuss this work in more detail in section 1.2.2.

Biological models of Ewing sarcoma

There exists several biological models of Ewing sarcoma that can help researchers study the transcriptomic landscape of this cancer type, either in bulk or at the single-cell level. In institutes such as Institut Curie where there is a close collaboration between research units and the hospital, fresh patient tumors can be worked with. These *in vivo* samples have the benefit of representing tumors grown in a realistic microenvironment, but they are hard to acquire as they require to perform a tumor biopsy which is painful for the patient. In particular, it makes time resolved experiments difficult to conduct as performing daily or weekly biopsies would be intolerable for the patient. Also, samples often contain cells from the tumor microenvironment (healthy epithelial and endothelial cells, immune cells...). This makes bulk single-cell analyses difficult to conduct as it results in mixed signals, and single-cell datasets require extra data anlysis work in order to isolate tumor cells.

On the other hand of the spectrum one can find Ewing sarcoma cell lines which can be grown *in vitro*. Due to the fact they develop in optimal growth condition and lack tumor microenvironement, they do not faithfully replicate real tumor cells. On the other hand, cell lines provide an ideal environment to apply very controlled perturbations and investigate transcriptional dynamics. In particular, we were able to work with inducible ASP14 cell lines for which EF1 expression can be regulated thanks to a TET genetic construct: introduction of doxycyclin into the growth medium triggers the expression of anti-EF1 small hairpin RNAs that cause the degradation of EF1 mRNAs, thus inhibiting

the effect of EF1. These cell lines can be used to conduct time-resolved experiments, and observe how depletion and reinduction of EF1 alters Ewing sarcoma cells' transcriptome.

Patient Derived Xenografts (PDXs) are very potent models for *in vivo* human tumors, that can partly overcome the aforementioned issues. They consist of cells or tissue gathered from a patient's tumor which have been implanted into an immunodeficient or a humanized mouse (Lai et al., 2017), to emulate a realistic tumor environment which can be used in personalized medicine for drug probing without putting the patient at risk. It is important to keep in mind PDX present some limitations, the major concern about this model being the fact human stroma in the tumor microenvironment is quickly replaced by murine stroma, which significantly changes tumor interactions with its microenvironment (Blomme et al., 2018).

1.2.2 Multidimensional factor analysis of Ewing sarcoma cell processes

Cells are complex systems in which many biological processes take place, such as cell cycle progression, differentiation processes or metabolic activity. Several regulation factors such as gene expression, chromatin conformation, protein phosphorylation, and regulatory RNAs subtly orchestrate these cell processes. Following the abundance of these factors therefore provides a natural way to determine the spectrum of cell processes happening within an individual cell. When several datasets are acquired at different time points, it it also possible to follow the evolution of cell processes throughout a time frame which proves to be highly valuable when following the development of a complex biological object such as a tumor.

In this section, we will limit ourselves to scRNA-seq transcriptional signatures, but it is important to remember that other modalities can provide additional information about cell processes. First, We will define the notion of transcriptional signatures and cover how they are identified. We will then expand on the types of transcriptional signatures that are relevant for the analysis of scRNA-seq tumoral datasets, namely cell cycle-related signatures as well as signatures associated with tumoral heterogeneity.

Cell process-associated transcriptional signatures

A *transcriptional signature* (TS) designates a set of genes associated with a specific biological process. The main advantage of using a TS over one specific gene is that it is more robust to biological or experimentation noise, and it makes it easier to compare cell states between different cells.

In practice, for a scRNA-seq dataset embedded in a gene space $\mathcal{G} = \{g_1, \dots, g_d\}$, a TS can be defined as a boolean vector $\mathbf{t} \in \{0,1\}^d$ where $t_i = 1$ if g_i belongs to the signature, 0 otherwise. Then, for any expression matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ its associated normalized signature scores is simply defined by $\mathbf{x}_t = \mathbf{X}\mathbf{t}/\mathbf{1}_d^T\mathbf{t} \in \mathbb{R}^n$, whose the i-th coordinate contains the signature score of the i-th cell, and can be used to assess the activity of the associated biological process within this cell.

It is important to notice that, unlike cell types, a cell can be involved in more than one cellular process and at different intensity levels in each of these processes. This suggests the space of biological states is continuous, with some regions corresponding to impossible conditions (for instance, a cell cannot be in hypoxia while carrying out oxidative phosphorylation). Therefore, exploring the space of possible cell states is a complex problem that can lead to exciting discoveries about cell biology, and we will show some interesting applications in this section.

Independent component analysis

Independent component analysis (ICA) is a matrix factorization (MF) approach where the signals captured by each individual matrix factors are optimized to become as mutually independent as possible. ICA was shown to be a useful tool for unraveling the complexity

of cancer biology from the analysis of different types of omics data. Such works highlight the use of ICA in dimensionality reduction, deconvolution, data pre-processing, meta-analysis, and others applied to different data types (transcriptome, methylome, proteome, single-cell data) (Sompairac et al., 2019).

In ICA we search for an approximation of the observed probability density function $P(x_1, x_2, .., x_n)$ by $\hat{P}(s_1, s_2, ..., s_n)$, where new s_i variables are some linear combinations of the initial variables x_i. We search for such linear transformation that $\hat{P}(s_1, s_2, ..., s_n)$ deviates as little as possible from the product of its marginal distributions $P(s_1) \times P(s_2), .., P(s_n)$ where the deviation is usually defined in terms of information geometry (e.g., as Kullback-Leibler divergence). It is shown that ICA is efficient in detecting and correcting the batch effects in omics datasets (Sompairac et al., 2019).

ICA is not a data dimensionality reduction technique *per se*: therefore, it is usually applied on top of reduced (e.g., by standard PCA) and whitened representation of the initial dataset. Therefore, the choice of the number of independent components is an important hyperparameter (Kairov et al., 2017). In the simplest approach, ICA solution represents a rotation of the whitened data point cloud such that each normalized coordinate deviates as much as possible from the standard Gaussian distribution (Hyvarinen, 1999).

In our experiments, we used the stabilized version of ICA (Captier et al., 2022) which is shown to be the optimal MF approach for reproducible analysis of transcriptomic data (Cantini et al., 2019). We applied it to cells labeled as T-cells from all datasets to prevent dataset-specific cell type imbalance from biasing the components.

Identification of transcriptional signatures

Now that TS have been properly defined, let us present the two categories of strategies that can be carried out to identify them: *top-down* approaches and *bottom-up* approaches. Top-down approaches consist in selecting a set of genes for a specific biological process by using prior knowledge, such as genes involved in known pathways of this biological process. These pathways can be taken from the literature, or databases such as KEGG (Kanehisa et al., 2012) or Reactome (Vastrik et al., 2007; Gillespie et al., 2022). The main benefit of this type of approach is that it tends to yield high quality, curated lists of genes that have been validated by the community as being clearly involved in the biological process of interest. Scores associated with such TS are quite robust, and are based on biological knowledge. On the other hand, these TS do not directly lead to identifying new processes, pathways or genes, which makes top-down approaches limited for the unbiased exploration of biological signals present in a dataset. Also, it is interesting to note that these TS are usually not specific of a particular biological system, for instance a particular cancer type, which can be both an upside (less biased analysis) or a downside (missing important additional effectors). TS determined using top-down approaches are widely used today in a variety of applications, and are highly valuable for hypothesis validation as well as data interpretation.

Bottom-up approaches rather try to learn TS in an unsupervised or semi-supervised manner directly from the data using deconvolution algorithms. Most popular approaches rely on matrix decomposition algorithms such as non-negative matrix factorization (NMF) (Lawton and Sylvestre, 1971; Lee and Seung, 1999; Brunet et al., 2004)) and independent component analysis (ICA) (Jutten and Herault, 1991; Liebermeister, 2002; Zinovyev et al., 2013; Sompairac et al., 2019). For $s \in \mathbb{N}$ a number of factors, these methods decompose a gene expression matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ into two matrices $\mathbf{A} \in \mathbb{R}^{n \times s}$ and $\mathbf{S} \in \mathbb{R}^{s \times d}$ so that $\mathbf{X} \approx \mathbf{AS}$. Doing so, the activity of each factor in a given cell can be read in the matrix $\mathbf{A}$, and the contribution of each gene to a given factor can be read in the matrix $\mathbf{S}$. $\mathbf{S}$ can then be used to convert the factors into TS using an *enrichment* process, where for each factor its top contributing genes are identified and linked to a known biological function based on the literature or databases. Notably, the ToPPGene suite (Chen et al., 2009) is very precious tool for candidate gene priorization.

The main differences between NMF and ICA are the optimality criteria and thus the optimization procedure. NMF constrains both **A** and **S** to be non-negative, while ICA constrains the rows of **S** to be statistically independent. This highly influences the factorization output: NMF will tend to have a more easily interpretable output thanks to the non-negativity of the matrices, while the independence criterion of ICA should allow for a more subtle disentangling of the underlying biological signals. Both of these approaches are used today with robust software implementations, and have led to the discovery of relevant TS.

Transcriptional signatures help decipher tumoral heterogeneity

Tumors are highly heterogeneous biological systems, composed of cells with various types and states surrounded by a microenvironment at the interface with other systems like the immune or vascular systems. For this reason, single-cell analysis is an invaluable asset to improve our understanding of tumorigenesis and metastasis. In particular, profiling individual tumor cells through scRNA-seq assays gives a precise picture of gene regulation and biological processes within them. Unfortunately, tumoral heterogeneity also causes complex interactions between cell types and states, making tumor analysis challenging. Notably, if the tumoral microenvironment presents clear cell types that are clustered in the gene expression space, tumor cells often mainly differ by their intrinsic cell states, which are more challenging to distinguish and classify.

We conducted in the past a study (Aynaud et al., 2020) during which independent component analysis (ICA) was used as an unsupervised approach capable of disentangling (TS) present in various cancer datasets. These single-cell datasets included Ewing sarcoma (EWS) tumors, EWS cell lines, EWS patient-derived xenografts (PDX), but also retinoblastoma and neuroblastoma tumors. These TS were enriched with the help of public databases, and were then used to characterize how cell processes conducted by tumoral cells differ from those carried out within non-tumoral ones. Diverse TS were identified, notably several TS associated with cell proliferation, one TS associated with the expression of the EWS oncogene (EF1), several TS associated with glucose catabolism (cell respiration and hypoxia), as well as diverse other biological functions such as mRNA splicing or extracellular matrix organization.

TS have been valuable tools for the interpretation of EWS datasets. In particular, they have been used to observe how EWS cell lines respond to the suppression of EF1 and how they recover once EF1 is reintroduced. Indeed, these TS allowed us to not only follow the dynamics of induction at the molecular level, but also at the level of biological processes. By doing so, we were able to follow how EWS cells behave once they do not have access to EF1, as well as observe the orchestration of cell processes throughout EF1 reinduction. During this PHD, we extend this study by generating new EWS inducible datasets using more recent scRNA-seq technologies. This allowed us to improve the characterization of EWS TS, and identify new ones. Details about this work are discussed in Chapter 5.

Cell cycle transcriptional signatures

The cell cycle is a fundamental biological process during which a mother cell grows and divides into two daughter cells. Most living cells can undergo cell cycle, from prokaryotic organisms like bacteria to plant cells or animal cells; some cells are incapable of cell division, such as neurons. Cell cycle is a finely regulated process split into ordered successive segments called *phases*, during which specific events occur. For the remainder of this section, we will only discuss the Eukaryotic cell cycle as it is relevant for studying tumorigenesis. The following explanation about cell cycle phases is based on the textbook (Lodish et al., 2008), and additional details can be found in it.

- **G0 phase.** Cells that do not undergo cell division are often referred to as *quiescent* and belong to a phase called G0. Cells enter G0 from G1, and can exit G0 into G1.

Some cells stay indefinitely in G0, like neurons for instance.

- **G1 phase.** G1 phase is the first cell cycle phase and it takes place before DNA replication. During G1, cells grow in size and synthesize proteins and RNA molecules necessary during the S phase. Once these conditions are satisfied, cell cycle goes through a first checkpoint called START that allows the cell to enter the S phase during which DNA is replicated.

- **S phase.** DNA replication occurs during S phase and is carried out by a DNA replication machinery. Without going into too much details, the origin replication complex (ORC), as well as CDT1 and CDC6 load the the replicative helicase complex MCM. MCM is then phosphorylated by S-phase-specific CDKs and DDKs, which initiates the opening process of the DNA double helix that allows DNA polymerases to start DNA replication.

- **G2 phase.** G2 phase occurs once chromosomes have been replicated, and consists in a phase where cells synthesize proteins and other factors necessary for cell division.

- **M phase.** M phase or *mitosis* describes the phase during which cells divide, and can be split into four consecutive subphases: the *prophase*, during which nuclear envelope is degraded, chromosomes are condensed and mitotic spindle is formed; the *metaphase*, during which the mitotic spindle attaches to the chromosomes' centromere; the *anaphase*, during which microtubules that form the mitotic spindle are shortened which pulls daughter chromatids toward each pole of the cell; the *telophase* (and *cytokinesis*), during which daughter cells are split, their nuclear envelope is reformed, and chromosomes are decondensed.

The cell cycle is a highly regulated process, with a variety of factors involved in order to orchestrate its progression. In the current eukaryotic cell cycle model, the main regulators are a family of proteins called *cyclin dependent kinases* (CDKs), that are controlled by *cyclins* proteins as well as other kinases. When a CDK is activated by its dimerization with its respective cyclin, its phosphorylative activity acts as an activator of a specific cell cycle phase. For this reason, the deactivation of cyclin/CDK complexes is mostly caused by cyclin degradation.

ScRNA-seq is a very powerful tool to investigate cell cycle progression of individual cells. Indeed, many cell cycle factors are regulated at the gene expression level, which makes it possible to determine whether a cell is cycling, and in which phase it currently is. Cell cycle TS have been proposed to help determining the state of cells within the cell cycle process, and one of the most popular list of such genes has been published in (Tirosh et al., 2016). This list of genes was determined *in vitro* by screening a set of 800 genes for the response to specific cyclins. Such TS can also be constructed using unsupervised algorithms such as ICA, and we proposed in the past a set of cell cycle TS from Ewing sarcoma cells (Aynaud et al., 2020).

In this framework, each cell is associated to a small set of scores (typically from 2 to 4) that identifies its position onto a cycling trajectory, around which cells revolve as they progress throughout the cell cycle. One score always measures the expression of genes specific to the G1 and S phases, and another measures the ones specific to the G2 and M phases. Then, there can be a score associated to histones that are proteins responsible for cromatin condensation that takes place during mitosis, and another score associated to genes responsible for the mitosis exit. Overall, monitoring these values allow for a very precise estimation of the cell cycle phase a cell is undergoing. We worked on improving this cell cycle model as well as characterizing the interplay between its different components, which we explain in further details in Chapter 4.

1.3 Integration of single-cell data

Hand-to-hand with the surge of biological modalities, there has been an explosion in the number of available datasets helped by various scientific initiatives to make biological data more easily available (Conesa and Beck, 2019), like with The Cancer Genome Atlas (TCGA) database. When tackling difficult biological questions, using data gathered across different sources or modalities is enticing. On the one hand, combining data from different sources helps to provide a comprehensive view of the biological object of interest. For example, it can facilitate the discovery of rare but relevant cell types or states, or help quantify the relative abundance of cell types across a collection of biological samples. On the other hand, having different modalities at their disposal allows scientists to link them together, possibly leading to exciting mechanistic discoveries. Finally, there can be an emergent property where analyzing a biological object through several modalities simultaneously could yield superior information compared to analyzing each modality individually.

Unfortunately, there are several obstacles to overcome before data from several sources and modalities can be used within an analysis pipeline. First, the multiplicity of sources comes at the price of all sorts of batch effects, as datasets can come from different replicas, technologies, individuals, or species. Then, combining datasets containing measurements from various biological modalities is a major computational challenge, especially when samples are not clearly paired across datasets, as there is no trivial common space to embed samples together. Therefore, there is a real need for methods and tools that would be able to tie together biological datasets across datasets (or *batches*) and modalities.

1.3.1 Data integration links biological datasets across batches or modalities

This last decade has witnessed a sharp increase in the amount and complexity of data produced for cellular biology, thanks to an ever-growing number of bulk and single-cell profiling assays. These technologies allowed scientists to study heterogeneous cell populations through many biological feature spaces (or *modalities*) such as mRNA expression (Klein et al., 2015; Macosko et al., 2015), DNA methylation (Guo et al., 2013) and chromatin accessibility (Buenrostro et al., 2015a,b), and protein abundance (Aebersold and Mann, 2003; Westermeier and Marouga, 2005; Tibes et al., 2006). These assays can be carried out either in bulk, which yields for each sample a single averaged molecular profile, or at the single-cell level, which provides an exquisite insight into cell states and types present in the cell population. In particular, carrying out biological assays at the single-cell level snapshots cells at various points of a dynamical process, which can then be leveraged for various applications such as lineage tracing (Schiebinger et al., 2019), transcriptional dynamics (La Manno et al., 2018), inference of transcriptional trajectories (Chen et al., 2019a) and many more.

In addition, during the last few years, there have been several joint assays proposed to profile single cells through several modalities simultaneously, such as scM&T-seq for transcriptome and methylome (Angermueller et al., 2016), sc-GEM for genotype, transcriptome and methylome (Cheow et al., 2016), CITE-seq for transcriptome and surface proteins (Stoeckius et al., 2017), or SNARE-seq for transcriptome and chromatin accessibility (Chen et al., 2019b). It is also worth mentioning spatial transcriptomics, which yields measurements from a small number of cells in each well while also providing positional information of cells within the biological tissue (Ståhl et al., 2016). Finally, important phenotypical information can be obtained from microscopic imaging data, such as whole slide imaging (Pantanowitz et al., 2011).

Hand-to-hand with the surge of biological modalities, there has been an explosion in the number of available datasets helped by various scientific initiatives to make biological data more easily available (Conesa and Beck, 2019); among these initiatives, one can

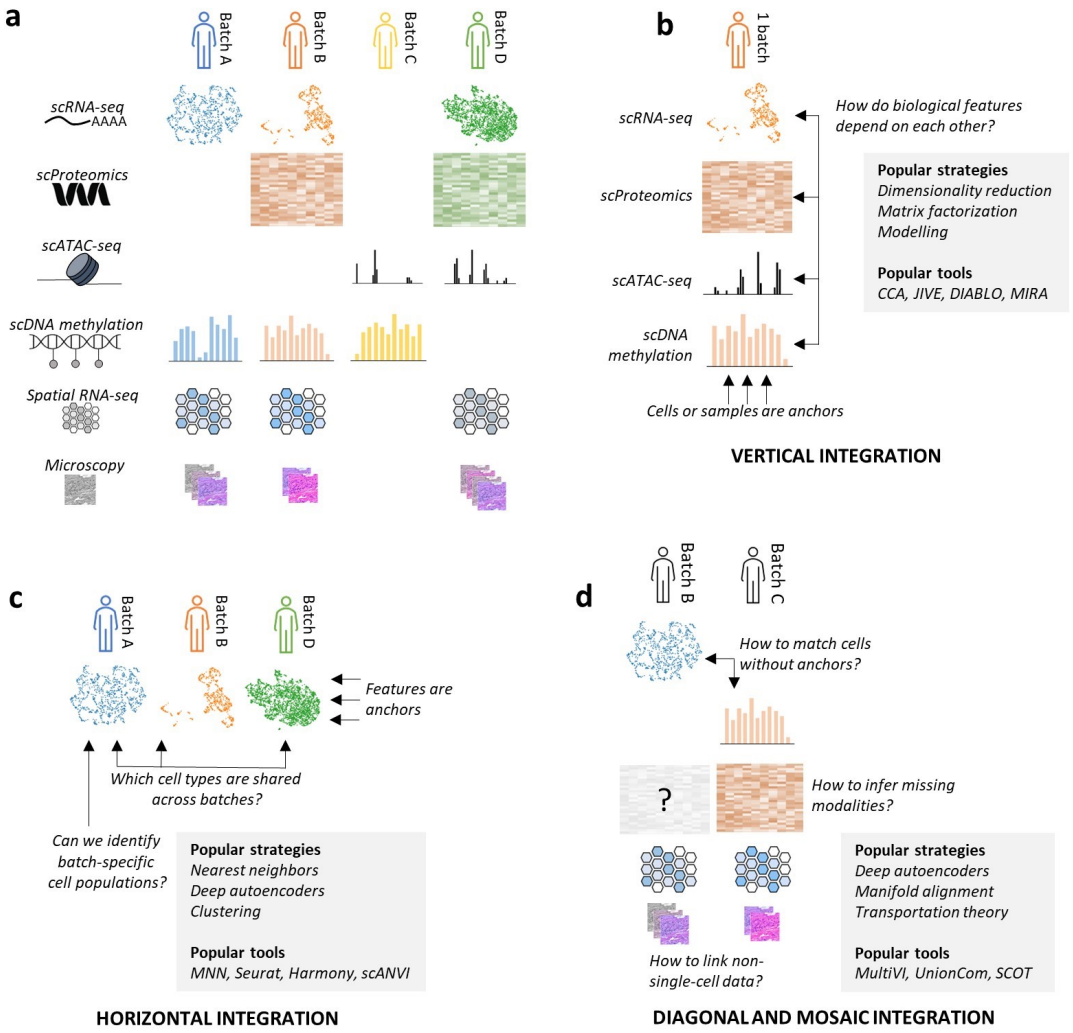

Figure 1.5: **Data integration describes a set of problems aiming to tie together data across different origins or modalities.** **(a)** A biological object can be profiled through multiple batches (columns) and modalities (rows), and not all batches necessarily contain measurements for all modalities. **(b)** Vertical integration (VI) consists in using cells or samples as anchors to deduce links between features across modalities. **(c)** Horizontal integration (HI) consists in using overlapping features as anchors to jointly analyze data coming from different sources. **(d)** Diagonal integration (DI) consists in embedding together several batches with non-overlapping modalities. Mosaic integration (MI) is the problem of missing modalities inference. *From (Fouché and Zinovyev, 2023).*

Tool	Strategy	Input	Output	Year	Reference
ComBAT	BA	RNA-seq	Gene space	2007	(Johnson et al., 2007)
MNN	NN	RNA-seq	Gene space	2018	(Haghverdi et al., 2018)
scmap	NN	RNA-seq	Clustering	2018	(Kiselev et al., 2018)
scvi	DAE	RNA-seq, spatial	Embedding	2018	(Lopez et al., 2018)
ingest	DR	RNA-seq	Embedding	2018	(Wolf et al., 2018)
CONOS	NN	RNA-seq	Graph	2019	(Barkas et al., 2019)
Scanorama	NN	RNA-seq	Embedding	2019	(Hie et al., 2019)
scAlign	DAE	RNA-seq	Embedding	2019	(Johansen and Quon, 2019)
Harmony	CL	RNA-seq	Embedding	2019	(Korsunsky et al., 2019)
Seurat v3	NN	RNA-seq	Gene space	2019	(Stuart et al., 2019)
LIGER	MF	RNA-seq	Embedding	2019	(Welch et al., 2017)
DESC	DAE	RNA-seq	Embedding	2020	(Li et al., 2020)
BBKNN	NN	RNA-seq	Graph	2020	(Polański et al., 2020)
SpaGE	NN	RNA-seq, spatial	Embedding	2020	(Abdelaal et al., 2020)
Tangram	DAE	RNA-seq, spatial	Embedding	2021	(Biancalani et al., 2021)
Canek	NN	RNA-seq	Embedding	2022	(Loza et al., 2022)
CAPITAL	MA	RNA-seq	Embedding	2022	(Sugihara et al., 2022)
SCISSOR	RE	RNA-seq	Graph	2022	(Sun et al., 2022)
DAPCA	MF	Any	Embedding	2023	(Mirkes et al., 2022)

Table 1.1: **A non-exhaustive list of horizontal integration (HI) tools aiming to jointly embed single-cell datasets measured in the same modality into a common space.** BA: Bayesian, NN: Nearest Neighbors, DAE: Deep Autoencoders, DR: Dimensionality Reduction, CL: Iterative Clustering, MF: Matrix Factorization, MA: Manifold Alignment, RE: Regression, FR: Framework

mention atlases of entire organisms such as the Tabula Muris (Schaum et al., 2018) and Human (Tabula Sapiens Consortium et al., 2022) Consortia. We would also like to talk about disease-based atlas such as The Cancer Genome Atlas (TCGA) database (Weinstein et al., 2013), and the IMMUcan database (Camps et al., 2023) which provides an exquisite insight into the nature of tumor microenvironment. When tackling difficult biological questions, using data gathered across different sources or modalities is enticing. On the one hand, combining data from different sources helps to provide a comprehensive view of the biological object of interest. For example, it can facilitate the discovery of rare but relevant cell types or states, or help quantify the relative abundance of cell types across a collection of biological samples. On the other hand, having different modalities at their disposal allows scientists to link them together, possibly leading to exciting mechanistic discoveries. Finally, there can be an emergent property where analyzing a biological object through several modalities simultaneously could yield superior information compared to analyzing each modality individually.

Unfortunately, there are several obstacles to overcome before data from several sources and modalities can be used within an analysis pipeline. First, the multiplicity of sources comes at the price of all sorts of batch effects, as datasets can come from different replicas, technologies, individuals, or even species. Then, combining datasets containing measurements from different modalities is a major computational challenge, especially when samples are not linked across datasets, as there is no trivial common space to embed samples together. Therefore, there is a real need for methods and tools that would be able to tie together biological datasets across datasets (or *batches*) and modalities. In this review, we investigate this question through the prism of machine learning paradigms, and present how a few of these concepts are today widely used within popular, state-of-the-art data integration methods.

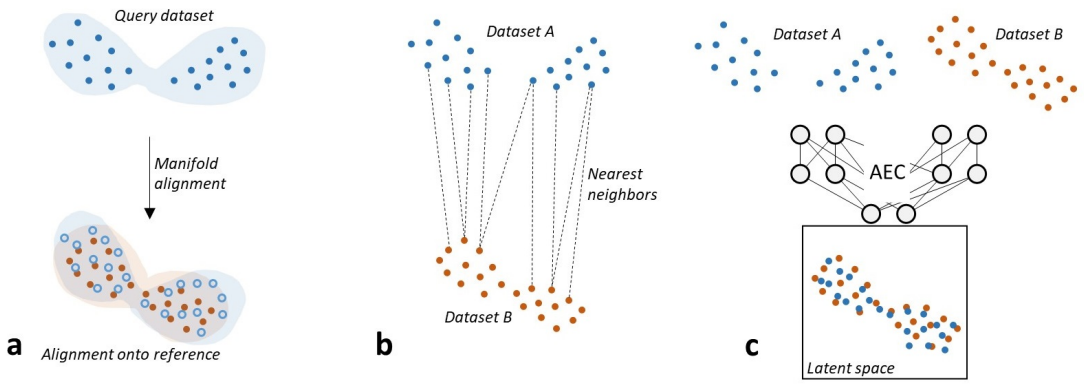

Figure 1.6: **Horizontal integration describes the problem of embedding together datasets measured along the same biological modality.** Different types of popular machine learning approaches are commonly used to match similar cells across batches. **(a)** Manifold alignment techniques find the projection that create the optimal overlap between two point clouds. **(b)** Nearest neighbors techniques identifies similar cells across datasets based on a similarity measure. **(c)** Deep autoencoders (AEC) learn a joint latent representation of the data in which batch effects are regressed out. *From (Fouché and Zinovyev, 2023).*

1.3.2 Horizontal integration (HI) links batches anchored by their common modality

Horizontal integration (HI) describes the situation where several batches are all gathered in a common modality with overlapping feature spaces. It is worth noting that depending on the tool, there may only suffice that each pair of datasets contains an overlapping feature space (e.g., dataset A containing features $\{f_1, f_2\}$, dataset B containing features $\{f_1, f_3\}$ and dataset C containing features $\{f_2, f_3\}$). HI is a convenient framework in which cells can directly be compared across different batches due to their feature space overlap, which allows the use of natural concepts such as distances, neighborhoods, or similarity measures. Many tools have been proposed to tackle HI, and we gathered a non-exhaustive list of them in (Table 1.1). As we can see, these methods employ various strategies to identify similar cells across batches and embed cells into a joint space. Some require additional information, such as reference datasets or cell labels. The remainder of this section is devoted to describing the main computational principles and machine learning paradigms HI methods rely on and providing some rationale and guidelines about each of them.

Many HI methods rely on manifold alignment strategies to integrate batches together [Fig. 1.6**a**], allowing them to consider the whole data structure instead of matching individual cells. Perhaps the oldest and most natural manifold alignment technique is Procrustes analysis (Gower, 1975), named after the mythical greek thug who cut or stretched his victims so that they fit the length of their bed. This is an old and intuitive machine learning paradigm mostly used for shape alignment that aims at projecting query datasets onto a reference one while only allowing simple transformations (rotation, rescaling, and shifting). Procrustes-based methods are not often used to integrate single-cell data, although some attempts can be found in the literature (Eto et al., 2018). First introduced to infer cell differentiation trajectories (Schiebinger et al., 2019), discrete optimal transport (OT) theory and its extensions (Gromov-Wasserstein, partial OT, unbalanced OT) is the most popular paradigm used for manifold alignment-based HI. It aims to align cells as discrete probability distributions represented as weighted point clouds in a metric space based on pairwise cell-cell cost matrices between batches that are often distance matrices. OT and

its extensions have been successfully applied to horizontal and diagonal data integration (Demetci et al., 2022; Cao et al., 2022b). Manifold alignment-based HI is a powerful paradigm, but it can sometimes struggle to solve complex alignment tasks (for instance, when the structure of a dataset presents ambiguous symmetries or when some batches contain specific cell types that must not be aligned).

Another class of HI methods seeks similar cells across batches, operating at the single-cell level rather than at a global level [Fig. 1.6**b**]. Some are based on the nearest neighbors approach like mutual nearest neighbors (MNN) (Haghverdi et al., 2018), CONOS (Barkas et al., 2019), Scanorama (Hie et al., 2019), Seurat (Satija et al., 2015; Butler et al., 2018; Stuart et al., 2019; Hao et al., 2021) that include different integration schemes such as CCA and robust PCA (RPCA), or BBKNN (Polański et al., 2020). All nearest neighbors-based methods rely on the hypothesis that batch effects are almost orthogonal to biological effects, which would allow identifying similar cells across batches through simple orthogonal projection. They then apply various strategies to end up with a joint representation of cells like correction vectors or joint graph construction. These methods tend to work best when facing slight to moderate batch effects and generally fail when batch effects are far from being orthogonal to relevant biological signals. They tend to scale well to large datasets thanks to various optimizations during nearest neighbors computation like nearest neighbors descent (Dong et al., 2011). Another metric-based approach is described in Harmony (Korsunsky et al., 2019), which is probably the most used tool in practice for HI of single-cell data. It uses an iterative algorithm of successive biased clustering across batches and correction. First, cells are clustered across datasets with such a bias that penalizes clusters of cells with a homogeneous batch of origin. Then, cells of a given cluster are pooled towards each other. An optimality criterion is tested at each iteration to assess whether batch mixing is sufficient, using a local purity metric called Local Inverse Simpson's Index (LISI). Due to its simplicity and availability with both Python and R packages, Harmony is widely used today and still achieves respectable results in benchmarks (Anaissi et al., 2022) despite being limited when facing strong batch effects (Luecken et al., 2022).

Deep autoencoders (DAEs) (and more recently variational autoencoders) have been popular tools in single-cell for a few years already and excel at performing a variety of complex preprocessing tasks, such as dimensionality reduction (Wang and Gu, 2018), or denoising and correcting dropouts (Eraslan et al., 2019), as well as acting as generative models (Trong et al., 2020). DAEs are neural networks that leverage a bottleneck structure to learn a compressed data representation in a low dimensional space, which can then be exploited for various tasks [Fig. 1.6**c**]. DAE is a powerful framework to carry out horizontal data integration with tools such as scvi (Lopez et al., 2018), scAlign (Johansen and Quon, 2019) or DESC (Li et al., 2020). In particular, scANVI, part of the scvi framework, is the top performer tool in the (Luecken et al., 2022) atlas-scale benchmark. DAEs generally have high computational capabilities thanks to the fact to be able to exploit GPU acceleration during training. The main downside of DAEs is the large amounts of data necessary for their training and their lack of interpretability, though there are efforts to improve on the latter point (Svensson et al., 2020; Treppner et al., 2022).

Despite the myriad approaches proposed to tackle HI, it remains challenging today to correct strong batch effects. For instance, (Tran et al., 2020; Luecken et al., 2022) showed that if several methods can satisfyingly remove moderate batch effects, integrating datasets across species remains difficult for unsupervised methods which do not require cell labeling information. Also, many methods rely on finding first an overlapping feature space between all datasets, which can be an obstacle when building large atlases combining many batches of varying quality, where the number of common features can shrink drastically. Finally, the problem of selecting appropriate metrics to assess data integration quality is still difficult. Most benchmarks use a mixture of metrics to measure different aspects of the data integration task such as batch mixture, label clustering or topology preservation,

depending on the information available:

- Batch mixture metrics such as batch-LISI are commonly used to measure how much the data integration procedure brought cells from different datasets close to one another. These metrics are popular because they do not require additional information, such as cell types or states, and can be used as unsupervised tools. Unfortunately, a good integration does not necessarily imply good batch mixture metrics, as two datasets without overlapping cell types should not be mixed after integration; similarly, projecting all datasets together onto a single point would result in perfect batch mixing, but all the biological information would be lost. For these reasons, even though batch mixture metrics are quite informative and widely used, most benchmarks also include other integration metrics to compensate for these limitations.

- Label clustering metrics, such as normalized mutual information or adjusted Rand index, provide an additional axis to measure data integration quality by assessing if cells of similar type cluster together after integration. Label clustering metrics are usually quite good for controlling the data integration quality if cell types can be identified confidently. The main downside of these metrics is the necessity to have high-confidence cell labels available before integration, which is often not the case (especially as one of the purposes of data integration is to be carried out before clustering and cell type inference).

- Finally, topology preservation metrics assess how data integration has preserved relations between the different cells and penalize cases where cells that were close before integration have been brought far apart by the algorithm (meaning cells that were initially similar but are dissimilar after integration). Topology can be biology-driven by observing the conservation of signals related to specific cell processes, such as cell cycle or other transcriptomic trajectories, or data-driven with algorithms as simple as comparing the k-nearest neighbors of a cell before and after integration and penalizing the differences.

Evaluating the quality of a HI can be daunting, as shown by the large variety of metrics that have been developed for it. In practice, we often use a batch mixture metric such as LISI, complemented by a secondary metric that can be either a label clustering metric if high-confidence labels are available and a topology preservation metric otherwise.

1.3.3 Vertical integration (VI) connects modalities measured in the same cells

Vertical integration (VI) uses several datasets containing individual measurements from the same cells obtained from joint single-cell assays measured through different biological features (e.g., gene expression and chromatin accessibility) to infer relations between the different modalities. VI is usually declined into two variants, namely *local* VI and *global* VI. Local VI identifies links between individual features (such as genes and methylated promoters), and can be used to formulate hypotheses of direct or indirect biological interactions between the omics layers (e.g., gene expression and accessibility of a chromatin region), with methods like LMM (Van Der Wijst et al., 2018) or Spearman's rank correlation coefficient (Cuomo et al., 2020). On the other hand, global VI links features across different modalities via global factors that can be related to biological processes (e.g., identifying a group of genes and chromatin regions to correspond to proliferation activity).

A family of global VI tools are based on a methodology inspired by canonical correlation analysis (CCA) (Hotelling, 1992), which use joint feature measurements across datasets to identify correlated features across modalities [Fig. 1.7**a**]. RGCCA (Tenenhaus and

Tool	Strategy	Input	Year	Reference
CCA	FC	Any	1936	(Hotelling, 1992)
RGCCA	FC	Any	2011	(Tenenhaus and Tenenhaus, 2011)
JIVE	MD	Any	2013	(Lock et al., 2013)
SGCCA	FC	Any	2014	(Tenenhaus et al., 2014)
MOFA	MD	Any	2018	(Argelaguet et al., 2018, 2021)
DIABLO	FC	Any	2019	(Singh et al., 2019)
scAI	MD	RNA-seq, epigenomic	2020	(Jin et al., 2020)
Seurat v4	NN	Any	2021	(Hao et al., 2021)
scMM	DAE	Any	2021	(Minoura et al., 2021)
SMILE	DAE	Any	2021	(Xu et al., 2022b)
MIRA	TM	RNA-seq, chromatin state	2022	(Lynch et al., 2022)

Table 1.2: **A non-exhaustive list of global vertical integration (VI) tools that can be used to learn relations between features across modalities from joint single-cell assays.** FC: Feature Correlation, MD: Matrix Decomposition, NN: Nearest Neighbors, DAE: Deep Autoencoders, TM: Topic Modelling

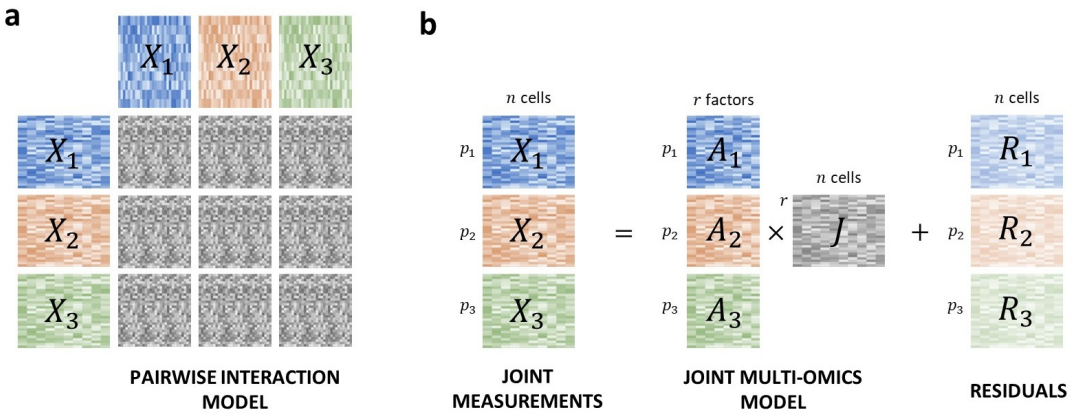

Figure 1.7: **Two main strategies are used for vertical integration of joint assays.** **a** Local strategies link features across modalities via pairwise correspondence. **b** Global strategies link features across modalities via common biological factors. *From (Fouché and Zinovyev, 2023).*

Tool	Strategy	Input	Output	Year	Reference
MATCHER	MA	RNA-seq, epigenetic	Gen. Model	2017	(Welch et al., 2017)
CoupledNMF	MF	RNA-seq, ATAC-seq	Clustering	2018	(Duren et al., 2018)
MMD-MA	MMD	Any	Embedding	2019	(Liu et al., 2019)
LIGER	MF	RNA, ATAC, scMethyl	Embedding	2019	(Welch et al., 2019)
UnionCom	MA	Any	Embedding	2020	(Cao et al., 2020a)
bindSC	NN	Any	Embedding	2020	(Dou et al., 2020)
SCIM	DAE	Any	Embedding	2020	(Stark et al., 2020)
MultiVI	DAE	RNA-seq, ATAC-seq	Embedding	2021	(Ashuach et al., 2021)
COBOLT	DAE	Any	Embedding	2021	(Gong et al., 2021)
Pamona	OT	Any	Embedding	2022	(Cao et al., 2022b)
Polarbear	DAE	RNA-seq, ATAC-seq	Embedding	2022	(Zhang et al., 2022a)
GLUE	DAE	Any	Embedding	2022	(Cao and Gao, 2022)
SCOT	GW	Any	Embedding	2022	(Demetci et al., 2022)
scJoint	DAE	RNA-seq, ATAC-seq	Embedding	2022	(Lin et al., 2022)
sciCAN	DAE	RNA-seq, ATAC-seq	Embedding	2022	(Xu et al., 2022a)
scDART	DAE	RNA-seq, ATAC-seq	Embedding	2022	(Zhang et al., 2022b)
StabMap	LI	Any	Embedding	2022	(Ghazanfar et al., 2022)
UINMF	MF	RNA, ATAC, spatial	Embedding	2022	(Kriebel and Welch, 2022)

Table 1.3: **A non-exhaustive list of diagonal (DI) and mosaic integration (MI) tools that integrate single-cell datasets gathered across different biological samples and modalities.** MA: Manifold Alignment, MF: Matrix Factorization, MMD: Maximum Mean Discrepancy, NN: Nearest Neighbors, DAE: Deep Autoencoders, OT: Optimal Transport, GW: Gromov-Wasserstein, LI: Linear Inference

Tenenhaus, 2011) extended this framework to simultaneously allow the analysis of more than 2 datasets. These concepts have been refined in (Tenenhaus et al., 2014) and DIABLO (Singh et al., 2019) to achieve better feature selection.

On the other hand, other popular global VI tools are based on matrix decomposition algorithms [Fig. 1.7**b**] (Lock et al., 2013; Argelaguet et al., 2018, 2020; Jin et al., 2020). These tools generally aim to decompose each data matrix into a component explained by global factors, a component containing dataset-specific and modality-specific factors, and a noise term. They mostly differ by their exact decomposition model and specific strategies used to infer its parameters.

If deep autoencoders did wonders for HI, they were also successfully applied to VI problems (Minoura et al., 2021) by using two distinct encoders and decoders using a shared latent space into which both modalities are projected. This strategy notably allows the network to "translate" a modality into another. We can also mention the recent MIRA method (Lynch et al., 2022), which leverages a variational autoencoder approach to learn gene expression and chromatin accessibility shared topics.

Overall, the VI framework has allowed the growth of methods taking advantage of the powerful sample anchoring across datasets, with many approaches proposed inspired by statistics and machine learning. A few important benchmarks have been carried out to assess the quality of VI tools, notably (Cantini et al., 2021) which focuses on joint dimensionality reduction (jDR) methods. Due to the difficulty of setting up joint assays and the inability of these methods to function without matched cells, there is a crucial need for diagonal integration (DI) tools that aim to integrate datasets across batches and modalities.

1.3.4 Diagonal and mosaic integration jointly embed non- or partially-anchored datasets

Diagonal integration (DI) and mosaic integration (MI) are two data integration frameworks for single-cell data that do not require datasets to be acquired through matched biological assays. In this paragraph, we use DI indistinguishably from MI. The goal is to leverage datasets structure and possibly external information, such as genomic locations,

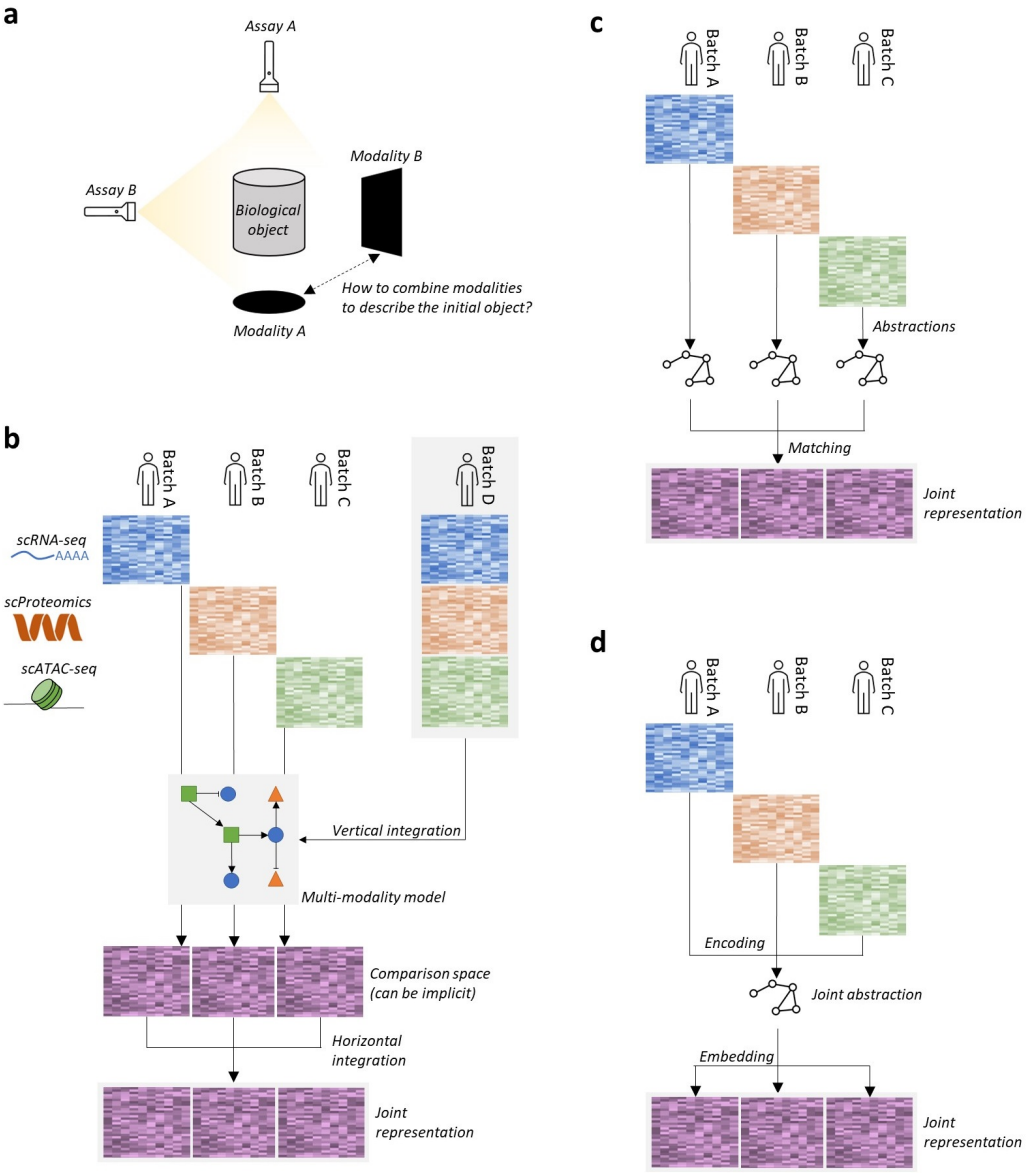

Figure 1.8: **Several strategies can be carried out to tackle the diagonal integration computational challenge. (a)** A biological object (e.g. a population of cells) can be profiled using different assays, without obvious means to link both representations. **(b)** Knowledge of interaction between features across modalities can be obtained from vertical integration of external datasets generated using joint assays. This information can then be leveraged to compare cells between batches even if they are not expressed in the same modality, which allows to use horizontal integration tools. **(c)** Datasets can be independently encoded into abstractions that can then be matched in an unsupervised fashion to build a joint representation of datasets. **(d)** Datasets can be jointly encoded into a unique abstraction, for instance through a learning process using a deep autoencoder framework, that can then be used as a joint embedding of datasets. *From (Fouché and Zinovyev, 2023).*

pathways, or partial sample or modality overlap to infer complete bonds between cells across modalities without relying on explicit sample anchoring [Fig. 1.8**a,b**]. DI generally aims to build a joint embedding of datasets into a common latent space, while MI focuses on inferring missing modalities from partially anchored datasets. Let us focus on the two main families of methods that exist for tackling DI: manifold alignment and deep autoencoders. These two machine learning paradigms can handle high levels of abstraction, which seems required to tackle DI in the general case.

Manifold alignment methods (Welch et al., 2017; Liu et al., 2019; Cao et al., 2020a; Demetci et al., 2022; Cao et al., 2022b) for DI operate similarly as in the HI case and work under the assumption stating that smooth point clouds alignment corresponds to meaningful biological correspondence [Fig. 1.8**c**]. This allows them to work in an unsupervised fashion without requiring additional knowledge other than data matrices. Despite working accurately in some cases, it has been shown this hypothesis is far from being universal (Xu and McCord, 2022). In this article, the authors show that under some simple data tweaking, such as missing cell types or different sample sizes, manifold alignment DI methods can generate erroneous embeddings featuring clusters with mixed cell types. This is concerning, as validating DI is a challenging task, given that it is rarely the case to have reliable cell type labels across modalities at disposal. Therefore, we suggest that these unsupervised manifold alignment methods must be used carefully and only when integration quality control is feasible. In other cases, it is preferable to choose another DI method that allows the user to provide additional information that helps bridge the gap across modalities.

As for HI and VI, deep autoencoders are powerful tools for solving DI tasks, with several advantages. First, they can take advantage of GPU acceleration built in deep learning libraries to greatly speed up the training process, and naturally scale to very large datasets. The second benefit of using these neural networks is that they offer the possibility to train a separate encoder and decoder for each biological modality, which helps capture modality-specific factors compared to manifold alignment algorithms where all omics layers are treated similarly. These separate encoders generally share a joint latent space [Fig. 1.8**d**], with some form of penalty to force latent representations to overlap. They also present an algorithmic structure that facilitates the introduction of external biological guidance, like in the GLUE tool (Cao and Gao, 2022), which uses a guidance graph as prior knowledge about functional relationships between features across modalities. We would also like to mention in this category the Polarbear tool (Zhang et al., 2022a), which leverages deep autoencoders to notably translate single-cell data between RNA-seq and ATAC-seq.

To the best of our knowledge, there do not exist at the time of writing a large-scale, independent benchmark of DI methods like for HI (Luecken et al., 2022). This is arguably difficult to set up due to the number of single-cell modalities available today, given the fact that, in addition, not all methods can deal with all modalities. Some may also require specific prior knowledge, and output type may vary. Furthermore, there is a lack of reliable metrics for assessing the quality of DI methods and real-life benchmarking datasets. A first breakthrough is to note in this direction, with a NIPS single-cell analysis competition organized recently which gave access to a public multimodal dataset containing single-cell gene expression, protein expression, and chromatin accessibility using CITE-seq and Multiome (Lance et al., 2022). With the democratization of such datasets, benchmarking DI methods will become more accessible, which will help standardize the field and identify the best-performing methods for each scenario.

To finish, there is a growing interest in integrating single-cell data with other related data modalities, such as whole slide images or spatial transcriptomics. There is a particular interest in deconvoluting spatial transcriptomic spots by integrating them with a single-cell RNA-seq dataset obtained from a similar same tissue. This is a current challenge, and several methods have been proposed for this task, notably benchmarked in (Li et al.,

2022).

There is always an urgent need for large-scale, independent benchmarks like the HI benchmark proposed in (Luecken et al., 2022), or the VI benchmark carried out in (Cantini et al., 2021). To the best of our knowledge, there is still a lack of large-scale independent DI and MI benchmarks. Two things are necessary to carry out such benchmarks: high-quality datasets and reliable metrics. A list of potential datasets can be found in (Argelaguet et al., 2021). There is no clear consensus about which quality assessment metric to use, and most benchmarks like (Luecken et al., 2022) opt for a mixture of metrics that cover several aspects of data integration: conservation of biological variance (CBV) metrics which measure how close similar cells (type or state) are after integration, and removal of batch effects (RBE) metrics. Some CBV metrics are label-based, such as normalized mutual information (NMI), adjusted Rand index (ARI), average silhouette width (ASW), class local inverse Simpson's index (cLISI), isolated label F1 (ILF) and isolated label silhouette (ILS), others are label-free and generally assess the conservation of biological processes such as cell cycle, highly variable genes, and transcriptomic trajectories. RBE metrics include batch-PC regression, batch-ASW, graph connectivity, iLISI, and kBet. We often observe a tradeoff between CBV and RBE, which can lead to different methods choice depending on the application, whether it is preferable to have good dataset mixing or conservation of subtle biological signals.

Overall, DI is arguably the most challenging data integration problem, and solving it is still a very active research area. This very convenient data integration paradigm is very versatile, as it theoretically does not need any anchoring (cells or features) between the different datasets. In practice, if many DI tools indeed work in a completely unsupervised way leveraging data topology such as MMD-MA (Liu et al., 2019), Pamona (Cao and Gao, 2022) or SCOT (Demetci et al., 2022), others require additional information to bridge the gap between modalities like GLUE (Cao et al., 2022a) or MultiVI (Ashuach et al., 2021) which can take a covariate design matrix as an optional parameter. For the moment, it appears that these biased methods offer more control on the results, as data topology can be misleading in practice and yield aberrant results (Xu and McCord, 2022). Therefore, using DI tools that can be enriched with biological context seems to be the best choice in the applications where such context can be obtained in a reliable way, typically when integrating datasets where strong covariates exist between modalities.

Chapter 2

Transmorph, a novel framework to perform integration of single-cell data

Adapted from (Fouché et al., 2023), extended.

Batch effects occur in most applications involving datasets gathered across multiple sources or experiments, and describe strong dataset-specific signals which are often irrelevant to the studied biological questions. Data integration is a computational paradigm aiming to learn a joint embedding of datasets in which batch effects are regressed out, meaning only dataset-independent factors are expressed. The idea is to combine information contained in several datasets, each of those being supposedly biased by its own specific batch effects. We focus here on the so-called *horizontal data integration* (Argelaguet et al., 2021) which seeks to integrate datasets obtained within the same domain with overlapping feature spaces. This is different from *vertical* and *diagonal data integration* where cells are measured in different domains, also known as multi-omics data integration. This scenario involves specific strategies and algorithms which are beyond the scope of this project (see for instance (Hao et al., 2021)).

Data integration is an important preprocessing step for applications involving several datasets (Fig. 2.1a). In some cases, something as simple as centering and normalization/scaling of features may suffice, but more complex batch effects often require more subtle, dedicated algorithms to be satisfactorily removed. Data integration can serve various purposes. The most common usage is to embed items from all datasets into a joint low dimensional space like in Harmony (Korsunsky et al., 2019), which can then be used to carry out various techniques such as clustering, label transfer, or visualization. Another use case is to directly perform integration in gene space like in MNN (Haghverdi et al., 2018) so that algorithms needing interpretable features such as matrix factorization methods can be used. Finally, integration can be carried out without embedding data points into an explicit feature space, for instance by outputting a joint graph of cells across datasets like in BBKNN (Polański et al., 2020).

Data integration finds particularly important applications in single-cell biology. Starting with a biological tissue, a single-cell dataset is generated and contains individual molecular measurements (for instance gene expression, SNPs, or chromatin accessibility) about single cells of the tissue. The strength of single-cell analysis is its ability to both provide an insight into intrinsic cell state, while also giving access to population-level information that can for instance be used to estimate cell types distribution within a tissue, which makes this technology relevant for analyzing patient samples in medicine. Due to genetic and environmental differences between individuals, batch effects are very prone to appear when dealing with single-cell datasets coming from different patients. The intertwining of batch-dependent and batch-independent factors is therefore an obstacle for

the analysis of large comprehensive datasets built by aggregating data from different individuals, notably when building cell atlases (see for instance (Angelidis et al., 2019)). Data integration is consequently a necessary technology to develop in order to mitigate dataset-specific signals while preserving relevant biological signals proper to the system of interest. This chapter contains the extended version of the paper (Fouché et al., 2023) which presents *transmorph*, a computational framework we developed to design data integration algorithms.

2.1 *Transmorph*: concept and architecture

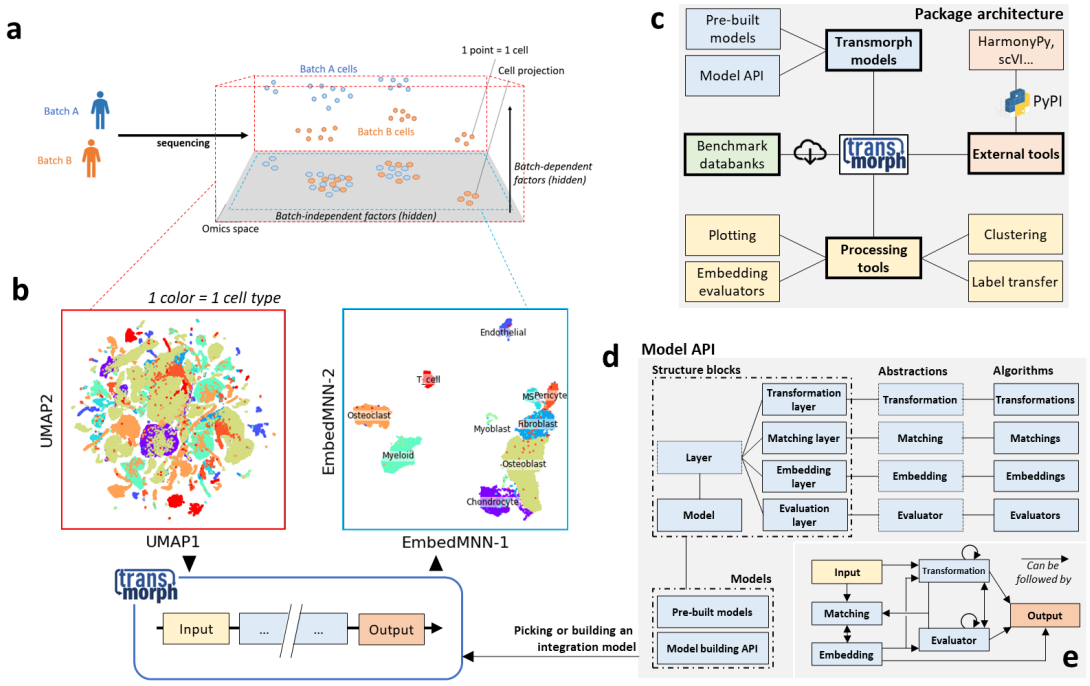

Figure 2.1: ***Transmorph* is a framework for scRNA-seq data integration.** **(a)** Schematic representation of the data integration problem. **(b)** *Transmorph* integration models conduct data integration of scRNA-seq datasets. Once the integration has been performed, cells cluster by type or state instead of origin. **(c)** *Transmorph* global package architecture, featuring internal and external models, benchmarking scRNA-seq databanks, and analysis tools. **(d)** Architecture of the model API, which allows engineering new data integration models using basic building blocks. **(e)** Directed compatibility chart of the model API modules, with arrows indicating how algorithms can be articulated within *transmorph* pipelines. *From (Fouché et al., 2023).*

As mentioned in the introduction, integration of single-cell RNA-seq data has been a prolific topic for the last decade, and dozens of methods still appear each year in the literature (Argelaguet et al., 2021). We believe this surge of methods comes with an urgent need for organization, classification, and large-scale benchmarks (see for instance (Luecken et al., 2022)) of both (a) end-to-end data integration pipelines, in order to guide the computational biologists that need to apply data integration to their research projects, and (b) algorithms, to guide methodologists who conceive new data integration methods. For this reason we introduce *transmorph*, an intuitive computational framework that aims to decompose data integration methods into basic algorithmic units that can be freely rewired to make emerge new data integration pipelines (Fouché et al., 2023). These units can either be extracted from the literature, such as a nearest neighbors search or a PCA

projection, or can be customized at will by the user. This flexibility allows scientists to harness integration methods adapted to the specificities of their data, for instance by choosing an output algorithm that produces an integration directly in gene space, or by selecting a matching algorithm that takes data topology into account, such as optimal transport (Fig. 2.1b).

In addition, *transmorph* is endowed with other features that facilitate its integration within real-life workflows of scRNA-seq data analysis (Fig. 2.1c). It is fully compatible with the standard scRNA-seq library *scanpy* (Wolf et al., 2018) as they handle the same type of objects, is interfaced with other first-class data integration tools such as *Harmony* (Korsunsky et al., 2019) and *scvi* (Lopez et al., 2018), and contains several annotated scRNA-seq datasets as well as standard quality assessment metrics and plotting functions to validate data integration algorithms. Finally, a comprehensive *model API* is available to allow the user to implement their own algorithmic units, using an object-oriented specification (Fig. 2.1d). For these reasons, we think *transmorph* is both an original and a powerful asset to design, apply, and benchmark data integration methods.

2.1.1 *Transmorph* allows conceiving end-to-end data integration models

Despite potentially achieving very good integration results in specific use cases, we believe that existing data integration algorithms are flawed by their intrinsic rigidity. By constraining the user to a fixed algorithm, they tend to excel in some use cases while struggling in others, as we show in the next sections. Also, the lack of access to their internal algorithms can make results difficult to interpret. Furthermore, these internal algorithms cannot be easily modified when needed, notably when the user needs a particular output type that the algorithm is not able to provide. For instance, despite the fact is usually yields high-quality embeddings, the Harmony algorithm cannot perform data integration in gene expression space, which can be a downside for the subsequent application of deconvolution methods such as independent component analysis. Another issue we can mention is the fact some matching paradigms are not suited for certain datasets topologies, as we show in the last subsection where nearest neighbors-based algorithms have trouble matching cycling cells. Finally, some algorithms do not scale as well as others to large datasets, which can disqualify certain tools from being applied in these situations, such as optimal transport-based methods.

To address these limitations we present *transmorph*, a novel and ambitious data integration framework. It features a modular way to create data integration algorithms using basic algorithmic and structural blocks, as well as analysis tools including embedding quality assessment and plotting functions. The framework also provides annotated, high quality and ready-to-use datasets to benchmark algorithms (Fig. 2.1c). Finally, it is meant to be easily expansible by allowing the user to define new algorithmic modules if necessary. In this framework, data integration models can be assembled by combining four classes of algorithms: transformations, matchings, embeddings, and evaluators (Fig. 2.1d-e, Tab. 2.1).

- **Transformation** algorithms take as input a set of datasets and return a new representation for each of them, embedded in some feature space (there can be one separate feature space per dataset or one common feature space). Transformations are generally used during preprocessing: classic examples are PCA, neighborhood-based data pooling, or common highly variable genes selection.

- **Matching** algorithms estimate a similarity measure between cells across datasets. They are the core component of our integration framework, as their quality directly influences cell-cell proximity in the final embedding. *Transmorph* uses three main categories of matching; (a) label-based matchings which require datasets to be labeled beforehand and match items of similar label; (b) neighbor-based matchings which match items close items with respect to some metric; (c) transport-based

Transformations	Matchings	Embeddings	Models
Common Features: Selects and orders common genes between either all datasets or pairs of datasets. **Standardization:** Normalize expression values per gene or per cell in order to improve the quality of geometric methods. **Pooling:** Pools each cell vector towards an average of its neighbors to reduce the effect of outliers. **PCA:** Linearly projects cell vectors into a variance-preserving, low dimensional basis to reduce the curse of dimensionality effect. **ICA:** Linarly projects cell vectors into a low dimensional basis of statistically independent vectors to reduce the curse of dimensionality effect.	**KNN:** Matches nearest neighbors of each cell across batches. **MNN:** Matches cells that mutually belong to the nearest neighbors of the other across batches. **Optimal Transport:** Matches cells across datasets using an optimal transport approach, with each dataset viewed as a mixture of Dirac distributions. This algorithm performs best when datasets topologies are similar, and penalizes translations, scaling and rotations. **Gromov-Wasserstein:** Matches cells across datasets using a Gromov-Wasserstein algorithm which only accounts for data topology, without penalizing isometric transformations. **Fused Gromov-Wasserstein:** Matches cells across datasets using a linear mixture of optimal transport and Gromov-Wasserstein in order to balance the penalty between geometry and topology. **Combined:** Combines several matchings into a single one.	**Barycenter:** Projects each cell in a *query dataset* to the average value of its matches in a *reference batch* that must be specified by the user. This embedding can produce a result in gene space, that can then be treated as scRNA-seq data. It necessitates that all cells in the *query* dataset have a match. **Graph Embedding:** Links cells from all datasets into a single common weighted graph. This weighted graph is then embedded in a space whose dimensionality is chosen by the user, a space that is used as this module's output. Due to the nonlinearity of this approach, the final representation can be used for clustering or other topological analyses. **LinearCorrection:** Computes correction vectors from cells in the *query dataset* and their match in the *reference dataset*. Unmatched cells are then attributed to a mixture of vectors of the nearest matched cells. All cells are eventually translated according to the correction vector that has been computed.	**TransportCorrection** Takes as input two or more scRNA-seq datasets, with one chosen as an alignment reference. It computes optimal transport between each dataset and the reference. It then uses the barycentric embedding to align each dataset to the reference and can output the result either in PC space or in gene expression space. **EmbedMNN:** Takes as input two or more scRNA-seq datasets, without requiring a reference to be specified. Cells are matched using a nearest neighbor scheme chosen by the user (KNN or MNN), and are organized within a joint weighted graph whose specification is described in *Material and Methods*. This graph is finally embedded using UMAP or MDE. **MNNCorrection:** Takes as input two or more scRNA-seq datasets, with one chosen as an alignment reference. Cells are matched using the nearest neighbor scheme chosen by the user (KNN or MNN). It then uses a linear correction module to align cells from each query dataset onto the reference one.

Table 2.1: Presentation of the main algorithmic modules available in the *transmorph* framework that can be used to build data integration pipelines. A brief explanation is given for each of them, additional information as well as algorithm parameters are available in section 2.2 and in the *transmorph* documentation.

matchings which leverage a distance metric between items within or across datasets to compute a similarity between items relying on topological correspondence.

- **Embedding** algorithms are a special class of transformations that take as additional input similarity relationships between samples that were estimated via a matching. They return an integrated view of all datasets jointly embedded in a common feature space, so that matched items tend to be close to one another in the final representation. The embedding step is in general the last step in an integration model and is chosen depending on the required output type. For instance, a joint embedding of datasets in an abstract space is suited for applications like visualization or clustering, while matrix factorization algorithms often require the embedding to be performed in an expressive feature space.

- **Checking** algorithms are special quality control points that can be added to a pipeline in order to test a condition. They are used to either set a branching point that leads to different outcomes or create an iterative structure within a model ("*repeat until the integrated representation satisfies this property*"). This type of strategy is notably used within the Harmony algorithm, where an iterative clustering and correction procedure is applied until an integration metric (Local Inverse Simpson's Index in this case) is considered to be satisfactory.

This expressive framework allows the building of complex data integration models suited for many applications with high computational efficiency and integration quality because each algorithmic module can be optimized independently. It also provides an objective comparison between algorithmic modules for a given application. Finally, it is supported by a sound software ecosystem with benchmarking databanks, pre-built models, and post-analysis tools, which allows one to carry out data integration within a scRNA-seq analysis workflow efficiently. Our framework is provided as an open-source Python package, and the following results showcase its capabilities to solve various challenging real-life problems of single-cell RNA-seq data integration while being on par with existing tools in terms of performance. It has been developed to be easily used in notebook environments, with a strong focus on computational efficiency so that models can be run on small machines in a few minutes, even in applications involving tens of thousands of cells and more than ten different datasets and cell types.

2.1.2 Package implementation

Conceiving *transmorph* from an implementation point of view has been challenging on many aspects. First, its modular design forced us to ensure all the modules can be freely articulated together as long as the interaction makes sense in theory (for instance, running an embedding without matching cells does not work). On the other hand, the implementation must allow the user to provide various hyperparameters to the algorithmic modules that can be of various types (booleans, scalars, vectors, matrices, functions...). Therefore, *transmorph* must be at the same time flexible in order to allow many types of algorithms to be expressed while also retaining some rigidity to ensure maximal compatibility between modules. Finally, *transmorph* must be usable for real-life, large-scale applications, which means it must guarantee a reasonable computational efficiency so that it can be used to integrate tens of thousands of cells in a few minutes on a laptop.

To satisfy all these constraints, we decided to implement the *transmorph* framework using a fully object-oriented approach. Every algorithmic module is implemented within a class, and the hyperparameters for the algorithm are provided by the user as member attributes when the module is instantiated. Each class can implement various independent traits using multiple inheritance. For instance, modules such as Principal Component Analysis or Graph Embedding are endowed with the *isRepresentable* trait, which allows them to provide a new data embedding. This design also allows us to factorize some

redundant code, for instance all modules relying on the pairwise distance between points are endowed with the *usesMetric* trait, which caches metric matrices in order not to recompute them at every step of the pipeline if the representation is unchanged. Every module also implements one of the four barebones interfaces (Transformation, Matching, Embedding or Checking), which follow clear specifications to ensure compatibility between the different components; this interface generally consists of at least a constructor method and a fitting method. Using this system of basic interfaces that can be expanded with complex traits and custom functions allows *transmorph* to provide great flexibility while staying well specified with robust data flows. Furthermore, code factorization based on these traits allows us to maintain only one optimized version of each basic algorithm, such as nearest neighbors search, and make it affect every module.

An integration model is then represented as a directed graph of layers, each layer being endowed with one or more modules that are executed sequentially. These layers are in charge of managing a clean data flow, by receiving datasets processed by the upstream layer, processing them with their internal modules, and passing them to the next layer. The user can freely articulate layers and modules to define a custom data integration pipeline, and all module interfaces are public so that adding new algorithms following the specifications is straightforward. Now that we have described the global *transmorph* philosophy and some implementation details let us dive into the available *transmorph* modules.

2.2 *Transmorph* algorithms

This section details all computational bricks that can be used in our framework to design data integration pipelines. As aforementioned, we separate them into four algorithmic categories:

- *Transforming algorithms*, which take as input a set of batches and transform their geometry, possibly into a new space. This category contains for instance dimensionality reduction algorithms and statistical standardization procedures.
- *Matching algorithms*, which compute for every pair of batches the (possibly weighted) bipartite matching graph between samples from one batch and samples from the other. This graph's edges are weighted, corresponding to the similarity confidence between two samples.
- *Merging algorithms* are a special type of transforming algorithms, which take as input matching between batches in addition to datasets embedding. These algorithms are used to compute the joint embedding of batches.
- *Checking algorithms*, which takes a joint embedding of batches and computes an integration quality statistic. These algorithms are used to assess integration quality, and can be associated with decision branches.

2.2.1 Transformations

A *transformation* algorithm takes a set of datasets, each embedded in their respective space as input, and returns for each of these batches a new representation, which can preserve initial feature space or not. Let us present the transformation modules currently available within transmorph.

Common feature space embedding

Many geometrical algorithms operating across batches, such as batch nearest neighbors and optimal transport, require datasets to be first embedded within a common features

space – it is in particular crucial when distances are involved. Let $X_1, X_2, \ldots, X_K$ be batches of the same data type (e.g., RNA-seq or ATAC-seq) with respective features sets $F_1, F_2, \ldots, F_K$. A common feature embedding is typically found by projecting every batch in the common feature space $\tilde{F} = \bigcap_{i=1}^{K} F_K$, given this intersection is nonempty. In the case of strong batch effects, common feature space embedding can be followed by a standardization step, where all embedding features are corrected batch-wise to mean 0 and variance 1 in order to emphasize relative feature variations between batches rather than absolute signal.

This transformation is easy to carry out efficiently and gives good results in practice but suffers from a few downsides. First, it cannot be used in vertical integration or when batches are expressed in disjoint feature spaces. Furthermore, with an increasing number of datasets, the cardinal of $\tilde{F}$ tends to shrink, and just one batch of lesser quality can reduce its size drastically. This procedure might also introduce biases especially when applied to batches from different biological samples, for instance RNA-seq batches with different transcriptomic dynamics. In this data type, genes are typically filtered first to keep only highly variable genes. When embedding batches in a common features space, only common variable genes are then selected, which could make important variation signals disappear.

Barycentric pooling

Barycentric pooling is a smoothing technique that moves every point in a noisy dataset towards the average of its k nearest neighbors. Let $(\mathcal{X}, d)$ be a metric space $\mathcal{X}$ endowed with a distance d, and $X \in \mathcal{X}^n$ a dataset. For any set $S \subset \mathcal{X}$, the barycenter of S is denoted by $\overline{S} \in \mathcal{X}$, and is defined as

$$\overline{S} = \arg\min_{x \in \mathcal{X}} \sum_{s \in S} d^2(x, s) \tag{2.1}$$

If $\mathcal{X}$ is a vector space and d is the Euclidean distance, it is easy to verify that $\overline{S} = |S|^{-1} \sum_{\mathbf{s} \in S} \mathbf{s}$. We use this trick in practice to efficiently compute the Euclidean barycenter of points in vectorized datasets. For every $x \in X$ and k a positive integer, we first define the set S_x defined as the k-nearest neighbors of x in X; then, every x is projected onto S_x.

Barycentric pooling is a very useful preprocessing step. First of all, it helps mitigate the "dropout" effect in RNA-seq datasets, describing the phenomenon of transcripts being falsely not detected during sequencing while being present in the mixture. Pooling helps here by replacing missing counts with the average of this count within similar cells. It also generally reinforces local dataset geometry and communities by reducing variance and correcting outliers but can cause excessive fuzziness with large neighborhood sizes.

Principal component analysis

Component analysis techniques change the basis with respect to which a dataset is expressed into a new basis with more relevant features. These methods can be used to reduce dataset dimensionality, which notably helps metric-based algorithms deal with the curse of dimensionality, but also to separate signal from noise and group together features which depend on a common factor. The most famous component analysis method in the field is, with no doubt, Principal Component Analysis (PCA) (Hotelling, 1992). This optimization problem seeks a set of orthogonal vectors that maximize the variance of data points when projected orthogonally onto each component. PCA is widely used in many fields for explanatory data analysis, and as a dimensionality reduction tool during preprocessing in various algorithms. In the case of scRNA-seq datasets, PCA allows to reduce their dimensionality from a few tens of thousands features to only a few tens, while preserving a large part of variance. It is therefore an almost mandatory preprocessing step used by

most algorithms and tools developed in the field, from k-nearest neighbors computation to clustering, non-linear dimensionality reduction like UMAP and most data integration pipelines. For this reason, we decided to include PCA as a *transmorph* module and it is included as a preprocessing step in all pre-built *transmorph* models.

UMAP

Uniform Manifold Approximation and Projection (UMAP) is a widely used, non-linear dimensionality reduction method (Becht et al., 2019). It first computes an underlying weighted graph between close points, then optimizes a representation of this graph in a typically very low dimensional space driven by edge weights. This technique is mainly used for explanatory data analysis and is highly potent for community detection and trajectory analysis in single-cell data. We won't cover the theoretical motivations behind the UMAP algorithm as they are beyond the scope of our framework. Still, we'll describe the main computational steps as they are reused in some *transmorph* modules.

The graph-building process first identifies edges, then weights them using a local metric that simulates a uniform distribution of points along the hypothetical underlying manifold. Let $(\mathcal{X}, d)$ be a metric space endowed with a metric d, and $X \subset \mathcal{X}$ be a dataset embedded in this space with n samples. For every sample $x_i \in X$, and given a positive integer k, we compute the k-nearest neighbors of x_i in X denoted S_{x_i}. Once the neighborhood of x_i has been identified, we must weight the edges from x_i to each neighbor. To simulate a uniform distribution of points along the underlying manifold, edge weights must follow two rules:

- For every point x_i, the edge weight from x_i to its closest, non-identical neighbor must be equal to 1.
- For every point x_i, the sum of edge weights from x_i to its neighbors must be equal to $\log_2(k)$.

To fulfill both properties, an adaptation of the Gaussian kernel was proposed. For each x_i, we define ρ_i as the distance from x_i to its closest, non-identical neighbor. For $\sigma_i > 0$, edge weight between x_i and $x_j \in S_{x_i}$ is defined as

$$w_{\rho_i,\sigma_i}(x_i, x_j) = \exp\left(\frac{-\max\{0, d(x_i, x_j)\} - \rho_i}{\sigma_i}\right). \tag{2.2}$$

σ_i is defined so that $\sum_{x_j \in S_{x_i}} w_{\rho_i,\sigma_i}(x_i, x_j) = \log_2(k)$, and is approximated in practice using simple binary search. At the end of the procedure, we obtain a weighted graph $G_{X,\rho,\sigma} = (X, V, w_{\rho,\sigma})$ where graph vertices are elements of X, graph edges are $V = \bigcup_{x_i \in X} S_{x_i}$ and $w_{\rho,\sigma}$ is the edge weighting function. This graph can be described by a square adjacency matrix $\mathbf{A}$ where

$$\mathbf{A}_{ij} = w_{\rho_i,\sigma_i}(x_i, x_j) \text{ if } (x_i, x_j) \in V, 0 \text{ otherwise.} \tag{2.3}$$

The last step is to symmetrize $\mathbf{A}$ using the following interpretation: the probability that an undirected edge between x_i and x_j exists is the probability that a directed edge exists from x_i to x_j *or* from x_j to x_i. Assuming these two events are independent, the symmetrized adjacency matrix $\hat{\mathbf{A}}$ is given by

$$\hat{\mathbf{A}} = \mathbf{A} + \mathbf{A}^\intercal - \mathbf{A} \circ \mathbf{A}^\intercal \tag{2.4}$$

where $\circ$ is the matrix Hadamard product (component-wise). This matrix represents a symmetrical weighted graph of X samples and can be embedded in a low-dimensional space. The objective function for the embedding is defined using a set of attractive forces along edges and repulsive forces between a subsampling of non-linked vertices. Given the graph is fully connected, an initial embedding can be computed using a spectral layout, which is

then iteratively optimized using the aforementioned attractive and repulsive forces. These forces decrease in magnitude after each iteration, guaranteeing convergence. We use the `umap-learn` python implementation of UMAP in our framework.

Minimum distortion embedding

Minimum distortion embedding (MDE) is an intuitive framework for non-linear, low dimensionality embedding of datasets (Agrawal et al., 2021). It has been developed to unify embedding methods into a common framework, making them more easily interpretable and customizable. Let $(\mathcal{X}, d)$ be a metric space endowed with a metric d and $X \subset \mathcal{X}$ be a dataset embedded in $\mathcal{X}$. For $x_i, x_j \in X^2$, a *distortion* f_{ij} is defined as a function of a distance $d_{ij} \geq 0$. The idea is that given a distance d_{ij} between x_i and x_j in a hypothetical embedding, $f_{ij}(d_{ij})$ penalizes d_{ij} being too small or too large according to some criteria.

For instance, given a similarity measure $w_{ij} \geq 0$ between each pair of points, defining $f_{ij}(d_{ij}) = w_{ij}d_{ij}$ is a reasonable distortion choice: it penalizes high distances between similar items, with little to no constraint between dissimilar items. Negative weightings can be attributed to pairs of non-similar points, or the embedding can be constrained (for instance, standardized) to avoid a trivial solution. Instead of using a similarity measure, distortion functions can also penalize the discrepancy between the embedding distance d_{ij} and $d(x_i, x_j)$. For well-behaved distortion functions, MDE can be solved as a constrained optimization problem, either exactly if the objective function is quadratic or using a gradient descent scheme otherwise. We use the `pymde` python implementation of MDE in our framework.

2.2.2 Matchings

A *matching* is an algorithm that takes a set of batches as input, all embedded in some vector spaces, and returns for every pair of batches and every pair of cells between these batches, a scalar value we call *matching strength*. For every pair of batches X_a and X_b of respective sizes n and m, a matching can therefore be represented as a matrix $\mathbf{P}_{ab} \in \mathbb{R}^{n \times m}$ where the $\mathbf{P}_{ab,ij}$ coefficient is the matching strength between item X_a from batch a and item j from batch X_b. This section describes in detail the matching algorithms available in transmorph.

Label matching

Label matching is the simplest supervised matching algorithm we can think of, necessitating each batch to be associated with a labels vector of same size from a label set $\mathcal{L}$: $\mathbf{X}_a$ is endowed with labels $\mathbf{l}_a \in \mathcal{L}^n$, and $\mathbf{X}_b$ is endowed with labels $\mathbf{l}_b \in \mathcal{L}^m$. Label matching then simply associates a matching strength of 1 between samples of same label, 0 otherwise. For every pair of samples $\mathbf{x}_{a,i} \in \mathbf{X}_a$ and $\mathbf{x}_{b,j} \in \mathbf{X}_b$,

$$M^{\text{label}}_{\mathbf{l}_a,\mathbf{l}_b}(\mathbf{x}_{a,i}, \mathbf{x}_{b,j}) = \delta_{\mathbf{l}_{a,i}\mathbf{l}_{b_j}} \tag{2.5}$$

where δ is the Kronecker symbol ($\delta_{xy} = 1$ if $x = y$, 0 otherwise). This matching can be computed efficiently and does not depend on dataset embedding, which can be an advantage when no clear metric can be defined between them and cannot match cells with different cell labels. On the other hand, it requires assessing cell labels beforehand, which is a strong bias and necessitates a third-party algorithm. It also tends to generate a very high number of matching edges (of the order of $|\mathcal{L}|^{-1}nm$ considering labels are balanced within batches), which can severely reduce the performance of subsequent pipeline steps. This also implies every sample is matched to all samples with its label, meaning this matching is insensitive to variations within a label (e.g., cellular subtypes), which tends to blur results. For these reasons, label matching is not recommended in the general case.

Still, it could see use in particular scenarios, such as pipeline testings or when precise labels are available with high confidence.

Batch k-nearest neighbors

For k a positive integer, the batch k-nearest neighbors algorithm is derived from the well-known k-nearest neighbors algorithm (Fix and Hodges, 1989; Cover and Hart, 1967). It requires two sets X_a and X_b with items embedded in a common metric space $(\mathcal{X}, d)$, and works best when the batch effect is orthogonal to the biological signal of interest, which appears to be a reasonable assumption in practice for most single-cell data.

For every item $x_{a,i} \in X_a$, we denote by $r_{a,i}(k)$ the distance to its k-th nearest neighbor in X_b. We then define the batch k-nearest neighbors of $x_{a,i}$ in X_b as $BNN_k(x_{a,i}, X_b) = X_b \cap \bar{\mathcal{B}}(x_{a,i}, r_{a,i}(k))$ where $\bar{\mathcal{B}}(x, r)$ denotes the closed ball centered in $x \in \mathcal{X}$ of radius $r \in \mathbb{R}^+$. For every pair of samples $x_{a,i} \in X_a$ and $x_{b,j} \in X_b$,

$$M^{\mathrm{B}k\mathrm{NN}}(x_{a,i}, x_{b,j}) = \mathbb{1}_{BNN_k(x_{a,i}, X_b)}(x_{b,j}) \tag{2.6}$$

where $\mathbb{1}_S$ is the indicator function of set S. Batch k-nearest neighbors derived algorithm have been successfully applied to dataset integration in the single-cell field, for instance in the BBKNN tool (Polański et al., 2020). It tends to yield high-quality matching results when the orthogonality of batch effect hypothesis is verified, and only needs the tuning of the k parameter (typically set between 10 and 50 in our applications). Furthermore, it also returns a much smaller number of edges compared to label matching of the order of kn, which greatly improves the performance of subsequent pipeline steps.

Mutual k-nearest neighbors

For k a positive integer, k-mutual nearest neighbors (Haghverdi et al., 2018) is an alternative to batch k-nearest neighbors. It tends to provide higher quality edges than batch k-nearest neighbors, at the cost of increased computation time and lesser edges number. The idea is to compute reciprocal batch k-nearest neighbors between two batches, and only keep the intersection of both edge sets.

Given two sets X_a and X_b with items embedded in a common metric space $(\mathcal{X}, d)$ and two samples $x_a \in X_a$ and $x_b \in X_b$, we first compute $\mathrm{BNN}_k(x_a, X_b)$ and $\mathrm{BNN}_k(x_b, X_a)$. Then,

$$\begin{aligned} M^{\mathrm{M}k\mathrm{NN}}(x_a, x_b) &= \mathbb{1}_{BNN_k(x_a, X_b)}(x_b)\mathbb{1}_{BNN_k(x_b, X_a)}(x_a) \\ &= M^{\mathrm{B}k\mathrm{NN}}(x_a, x_b)M^{\mathrm{B}k\mathrm{NN}}(x_b, x_a) \end{aligned} \tag{2.7}$$

The mutual k-nearest neighbors approach tends to yield high-quality matchings, inheriting all good properties from batch k-nearest neighbors while also being symmetrical. It also works under the assumption batch effect is orthogonal to biological effect, and always returns a smaller number of edges compared to batch k-nearest neighbors ($M^{\mathrm{M}k\mathrm{NN}}(x_a, x_b) = 1 \Rightarrow M^{\mathrm{B}k\mathrm{NN}}(x_a, x_b) = 1$). This fact often induces in practice the need to tune up the k parameter in order to have enough matching edges for subsequent pipeline steps to be stable.

Discrete optimal transport

Discrete optimal transport (OT) problem can be naturally pictured as follows (Peyré et al., 2019). Assuming a set of n warehouses containing goods to deliver to m factories, the optimal transport problem consists in finding the cheapest way to transport all goods to factories knowing the cost of transporting goods is proportional to both mass carried and distance traveled. Originally brought into the field as a way to predict cell fate (Schiebinger et al., 2019), it has more recently been shown to be an interesting asset for

matching cells across datasets in integration tools like SCOT (Demetci et al., 2020) and Pamona (Cao et al., 2022b).

Formally, let $\mathbf{a} \in \mathbb{R}^n$ and $\mathbf{b} \in \mathbb{R}^m$ be two histograms, meaning $\mathbf{a}$ and $\mathbf{b}$ coefficients are non-negative and each vector sums up to 1. In our analogy, $\mathbf{a}$ represents the quantity of goods stored in each warehouse, and $\mathbf{b}$ is the capacity of each factory. We are provided a cost matrix $\mathbf{C} \in \mathbb{R}^{n \times m}$, where $\mathbf{C}_{ij}$ is the cost of moving one unit of mass from a_i to b_j. The optimal transport problem from $\mathbf{a}$ to $\mathbf{b}$ given cost $\mathbf{C}$ can then be expressed as follows:

$$\begin{aligned} L_{\mathbf{C}}(\mathbf{a}, \mathbf{b}) = \min_{\mathbf{P} \in (\mathbb{R}^+)^{n \times m}} \quad & \sum_{ij} \mathbf{C}_{ij} \mathbf{P}_{ij} \\ \text{s.t.} \quad & \mathbf{P}\mathbf{1}_m = \mathbf{a} \\ & \mathbf{P}^\intercal \mathbf{1}_n = \mathbf{b} \end{aligned} \tag{2.8}$$

$L_{\mathbf{C}}(\mathbf{a}, \mathbf{b})$ is called the *Wasserstein* distance between $\mathbf{a}$ and $\mathbf{b}$ for transport cost $\mathbf{C}$, and is the cheapest cost to transport all mass from $\mathbf{a}$ to $\mathbf{b}$ in this setup. The optimal *transport plan* $\mathbf{P}^*$ is the valid $\mathbf{P}$ minimizing **Eq. 2.8**, and can be row-normalized to $\mathbf{1}_n$ to be used as a probabilistic matching between $\mathbf{a}$ and $\mathbf{b}$.

In practice, optimal transport can be computed between two datasets $\mathbf{X}_a \in \mathbb{R}^{n \times d_a}$ and $\mathbf{X}_b \in \mathbb{R}^{m \times d_b}$ with vectorized samples in rows. In this case, it is common to define $\mathbf{a} = \frac{1}{n}\mathbf{1}_n$ and $\mathbf{b} = \frac{1}{m}\mathbf{1}_m$, and to transform $\mathbf{X}_a$ and $\mathbf{X}_b$ so that they are expressed in the same feature space d. It can be achieved for instance by selecting the genes expressed in both datasets, yielding $\hat{\mathbf{X}}_a \in \mathbb{R}^{n \times d}$ and $\hat{\mathbf{X}}_b \in \mathbb{R}^{m \times d}$. The cost matrix $\mathbf{C} \in \mathbb{R}^{n \times m}$ is then typically defined as the pairwise distance matrix between $\hat{\mathbf{X}}_a$ and $\hat{\mathbf{X}}_b$, for a given distance. If $\mathbf{X}_a$ and $\mathbf{X}_b$ cannot easily be embedded in a common features space, the Gromov-Wasserstein approach is generally a better alternative. We use here optimal transport as a matching algorithm, by considering the row-normalized $\frac{\mathbf{P}_{ij}}{\mathbf{P}_i^\intercal \mathbf{1}_m}$ as the probability that cell i from dataset $\mathbf{X}_a$ is similar to cell j from dataset $\mathbf{X}_b$.

There are obvious limitations to this procedure, notably the mass conservation issue: OT will always move *all* mass from $\mathbf{a}$ to $\mathbf{b}$, regardless of the possible batch-specific samples. Consequently, all $\mathbf{X}_a$ cells will be mapped to at least one cell in $\mathbf{X}_b$, even though some cells from $\mathbf{X}_a$ may belong to a cell type missing in $\mathbf{X}_b$. Even worse, if there is a class imbalance between datasets (e.g. 50% of cell type A in dataset $\mathbf{X}_a$, and 25% of cell type A in dataset $\mathbf{X}_b$), there will necessarily be wrong assignments using this method. Exact computation of optimal transport is furthermore computationally expensive, of the order of $\mathcal{O}((n+m)^3)$ which makes it inefficient for large-scale problems (typically above 10^4 points). The supplementary note contains an alternate approximate and unbalanced formulation which provides a good approximation of the solution at a more reasonable cost, while also dealing with the class imbalance issue.

Entropic regularization of optimal transport

The optimal transport problem can be approximated using an additional entropy term (Cuturi, 2013; Peyré et al., 2019), which allows the minimization to be carried out using an efficient iterative procedure. For a given transport plan $\mathbf{P}$, we define its entropy as

$$H(\mathbf{P}) = -\sum_{ij} \mathbf{P}_{ij}(\log(\mathbf{P}_{ij}) - 1). \tag{2.9}$$

$H(\mathbf{P})$ is 1-strongly concave given its Hessian $\partial^2 H(\mathbf{P}) = -\operatorname{diag}(1/\mathbf{P}_{ij})$ and $\mathbf{P}_{ij} \leq 1$. $-H(\mathbf{P})$ can then be used as a regularizer term in Eq. 2.8 with a regularization term $\varepsilon > 0$ (Wilson, 1969), making the objective ε-strongly convex:

$$L^{\varepsilon}_{\mathbf{C}}(\mathbf{a},\mathbf{b}) = \min_{\mathbf{P}\in(\mathbb{R}^+)^{n\times m}} \sum_{ij}\mathbf{C}_{ij}\mathbf{P}_{ij} - \varepsilon H(\mathbf{P}) \quad \text{s.t.} \quad \mathbf{P}\mathbf{1}_m = \mathbf{a} \quad \mathbf{P}^\intercal\mathbf{1}_n = \mathbf{b} \tag{2.10}$$

Sinkhorn-Knopp algorithm can be used to optimize the objective, we invite the reader to refer to (Cuturi, 2013) for details. In short, the goal is to decompose a transport plan $\mathbf{P} = \text{diag}(\mathbf{u})\mathbf{K}\,\text{diag}(\mathbf{v})$, where $\mathbf{u}$ and $\mathbf{v}$ are the unknown *scaling variables* and $\mathbf{K}$ can be derived from the parameters. $\mathbf{u}$ and $\mathbf{v}$ can be approached using an iterative two-step normalization procedure. The smaller ε, the closer the objective is from unregularized formulation, at a cost of decreased convergence rate. According to (Altschuler et al., 2017) and assuming $n = m$ for simplicity, this algorithm computes a τ-approximate solution of the original optimal transport problem in $\mathcal{O}(n^2\log(n)\tau^{-3})$ operations. It allows to tackle larger scale problems in reasonable time. In practice, the resulting transport plan is often more fuzzy and less sparse than the exact solution, which necessitates filtering small values to stay efficient.

Unbalanced optimal transport

As stated previously, one of the major drawbacks of optimal transport is its constraint to always move all mass from source distribution to target distribution. As there is almost always class imbalance between single-cell datasets, this hard constraint necessarily causes matchings between cells of different cell type. This bad property can be worked around using an alternative unbalanced optimal transport problem (Liero et al., 2018). The idea is to relax the hard mass conservation constraint, by rather penalizing mass discrepancy via a divergence D_φ. Given two penalty coefficients τ_1 and τ_2, the objective function is written as

$$L^{\tau}_{\mathbf{C}}(\mathbf{a},\mathbf{b}) = \min_{\mathbf{P}\in(\mathbb{R}^+)^{n\times m}} \sum_{ij}\mathbf{C}_{ij}\mathbf{P}_{ij} + \tau_1 D_\varphi(\mathbf{P}\mathbf{1}_m|\mathbf{a}) + \tau_2 D_\varphi(\mathbf{P}^\intercal\mathbf{1}_n|\mathbf{b}) \tag{2.11}$$

This objective function can also be optimized using an adaptation of the Sinkhorn-Knopp algorithm. Removing the hard mass conservation constraint helps in practice to deal with situations of class imbalance between datasets, while staying entirely unsupervised.

Gromov-Wasserstein

OT-based matchings all share two common weaknesses. First of all, defining a cost matrix between two datasets can be non-trivial, especially if they are not embedded in the same features space. There may be workarounds such as using a latent space embedding method first, but this is not an easy task in the general case. Furthermore, OT-based matchings are not invariant to important families of transformations, such as scaling, shifting and rotation. The Gromov-Wasserstein problem is a natural extension of OT which does not suffer from these issues. Instead of requiring a cost matrix *between* datasets, it rather needs for each dataset *inner* pairwise costs between samples.

Let $\mathbf{X}_a \in (\mathcal{X}_a, d_a)$ and $\mathbf{X}_b \in (\mathcal{X}_b, d_b)$ be two datasets embedded in two possibly distinct metric spaces, containing respectively n and m samples. We first compute $\mathbf{D}_a \in \mathbb{R}^{n\times n}$ (resp. $\mathbf{D}_b \in \mathbb{R}^{m\times m}$) the pairwise inner distance matrix of $\mathbf{X}_a$ (resp. $\mathbf{X}_b$) where $\mathbf{D}_{a,ij} = d_a(\mathbf{x}_{a,i}, \mathbf{x}_{a,j})$ (resp. $\mathbf{D}_{b,ij} = d_b(\mathbf{x}_{b,i}, \mathbf{x}_{b,j})$). We also endow dataset $\mathbf{X}_a$ with histogram $\mathbf{a} \in \mathbb{R}^n$, and dataset $\mathbf{X}_b$ with histogram $\mathbf{b} \in \mathbb{R}^m$. The Gromov-Wasserstein problem between $\mathbf{X}_a$ and $\mathbf{X}_b$ is then defined as

$$
\begin{aligned}
GW((\mathbf{a},\mathbf{D}_a),(\mathbf{b},\mathbf{D}_b))^2 = \min_{\mathbf{P}\in(\mathbb{R}^+)^{n\times m}} \quad & \sum_{i_a i_b j_a j_b} |\mathbf{D}_{a,i_a j_a} - \mathbf{D}_{b,i_b j_b}| \mathbf{P}_{i_a i_b} \mathbf{P}_{j_a j_b} \\
\text{s.t.} \quad & \mathbf{P}\mathbf{1}_m = \mathbf{a} \\
& \mathbf{P}^\intercal \mathbf{1}_n = \mathbf{b}
\end{aligned}
\tag{2.12}
$$

This problem being equivalent to a graph matching problem, it is NP-hard (Lyzinski et al., 2015) thus being difficult to solve in practice. It can be entropy-regularized similarly to the OT problem using Sinkhorn-Knopp iterations (Peyré et al., 2019).

Fused Gromov-Wasserstein

Fused Gromov-Wasserstein (Vayer et al., 2019) is a natural extension of Wasserstein and Gromov-Wasserstein mapping. The first one focus on the metric related to the feature space and the second on the structure of the relations between samples within a dataset. Therefore, it may be desirable to consider both aspect but combining the Wasserstein and Gromov-Wasserstein problem with a trade-off parameter $\alpha \in [0,1]$.

With the previous notations, this problem can be formulated as:

$$
\begin{aligned}
FGW_\alpha((\mathbf{a},\mathbf{D}_a),(\mathbf{b},\mathbf{D}_b),C)^2 = \min_{\mathbf{P}\in(\mathbb{R}^+)^{n\times m}} \quad & (1-\alpha)\sum_{ij} \mathbf{C}_{ij}\mathbf{P}_{ij} \\
& + \alpha \sum_{i_a i_b j_a j_b} |\mathbf{D}_{a,i_a j_a} - \mathbf{D}_{b,i_b j_b}| \mathbf{P}_{i_a i_b} \mathbf{P}_{j_a j_b} \\
\text{s.t.} \quad & \mathbf{P}\mathbf{1}_m = \mathbf{a} \\
& \mathbf{P}^\intercal \mathbf{1}_n = \mathbf{b}
\end{aligned}
\tag{2.13}
$$

2.2.3 Mergings

Mergings are a class of transformations that takes as extra input matchings between batches. A merging F can be seen as a function taking as input a set of datasets $\{X_1,\dots,X_K\}$ with respectively $n_1,\dots,n_K$ items and a set of matchings $\{\mathbf{M}_{ij}\}_{i,j\leq K}$ expressed as non-negative square matrices, embedding all batches in a common space $\mathcal{Y}$. Every matching matrix $\mathbf{M}_{ij}$ must be row-normalized so that every nonzero row sums up to 1. If $i=j$, then $\mathbf{M}_{ii}$ is defined as the identity matrix $\mathbf{I}_{n_i}$.

Barycentric merging

Barycentric merging is the simplest merging to set up. It works under three assumptions, (1) one batch $\mathbf{X}_r$ is defined as *reference* and all batches will be corrected towards it; (2) reference batch $\mathbf{X}_r$ must be expressed in a vector space; (3) for every matching $\mathbf{M}_{sr} \in \mathbb{R}^{n_s\times n_r}$, every row must have at least one nonzero element ($\|\mathbf{M}_{sr}\mathbf{1}_{n_r}\|_0 = n_s$). Assumption (1) is usually specified by the user, and necessitates choosing a good quality batch with representers in every sample type. Reference choice always introduces a bias in the integration, which should not be overlooked in results interpretation. Assumption (2) is easy to verify in practice, as datasets are often vectorized and represented as $n\times d$ real-valued matrices. Assumption (3) necessitates choosing a *semicomplete* matching, which maps every sample from batch X_i to at least one sample from batch X_r. Transportation-based matchings usually verify this assumption, while nearest neighbor-based matchings do not. Failing to verify assumption (3) will cause non-matched points to be projected to the $\mathbf{0}$ of $\mathbf{X}_r$ feature space.

Let X_s be a batch to correct with respect to a reference batch $\mathbf{X}_r$ given a semicomplete, row-normalized matching matrix $\mathbf{M}_{sr}$. For every sample $x_k \in X_s$, the k-th row $\alpha_k = \mathbf{M}_{sr,k}$

provides a weighting vector which assesses the likelihood of x_k corresponding to any sample of $\mathbf{X}_r$. Barycentric merging F^{Bary} will then project x_k into $\mathbf{X}_r$ feature space $\mathcal{X}_r$ so that

$$F^{\text{Bary}}_{\mathbf{X}_r,\alpha}(\mathbf{x}_k) = \arg\min_{\mathbf{x}\in\mathcal{X}_r} \sum_{i\leq n_r} \alpha_i \left\|\mathbf{X}_{r,i} - \mathbf{x}\right\|_2^2. \tag{2.14}$$

The rationale behind this optimization problem is to bias the classic barycenter problem (see eq. 2.1) to penalize distances between more similar items. We quickly show $\mathbf{x}_k = \sum_{i\leq n_r} \alpha_i \mathbf{X}_{r,i}$ is the solution to this problem. Therefore, F^{Bary} can be easily generalized to project the whole X_s dataset onto $\mathbf{X}_r$ given $\mathbf{M}_{sr}$ via

$$F^{\text{Bary}}_{\mathbf{X}_r,\mathbf{M}_{sr}}(X_s) = \mathbf{M}_{sr}\mathbf{X}_r. \tag{2.15}$$

Barycentric merging has been used in several data integration pipelines such as Seurat (Stuart et al., 2019), SCOT (Demetci et al., 2022) and Pamona (Cao et al., 2022b), and generally yields good results. However, there are a few downsides to consider. First, choosing a reference introduces a high bias in the integration, and in some applications, there may be no natural option to choose as a reference; for instance, every batch could miss at least one sample class. The barycenter problem also intrinsically relies on a metric. This is an issue for high dimensional problems, for instance, in scRNA-seq datasets where the curse of dimensionality is a real concern; in this case, barycenter has little to no interpretable sense. A common solution is first to reduce the dimensionality of $\mathbf{X}_r$ using component analysis or non-linear methods such as UMAP (Becht et al., 2019) or MDE (Agrawal et al., 2021). One of the other uses of this method is to use a matching computed in a different space than the final embedding. Typically, one computes a matching in a lower dimensional representation (e.g., PC space) but uses total feature space for the embedding. This notably allows obtaining corrected feature counts for all batches with respect to a reference. Combined with a high-quality matching and reference batch, barycenter merging can nonetheless provide an efficient, high-quality integration without necessitating batches to be originally in the same space.

Linear correction

Linear correction is a linear merging based on first computing a set of correction vectors and then using them to correct batches with respect to a reference batch $\mathbf{X}_r$. It not only necessitates choosing a reference batch but also that all batches are initially embedded within a common vector space. Compared to barycentric merging, it can work with *incomplete* matchings, meaning not every matching row necessitates containing at least one nonzero element. The algorithm follows a two-step process to correct a given batch: it first computes correction vectors from matched samples to reference samples. Then it extrapolates the correction vectors to the unmatched samples. Other tools have used similar algorithms, notably in (Stuart et al., 2019).

Let $\mathbf{X}_1, \ldots, \mathbf{X}_K$ be K batches each represented in a common vector space $\mathcal{X} = \mathbb{R}^d$, respectively containing $n_1, \ldots, n_K$ samples. Let $1 \leq r \leq K$ be the reference dataset index, and $\mathbf{M}_1 \in \mathbb{R}^{n_1\times n_r}, \ldots, \mathbf{M}_K \in \mathbb{R}^{n_K\times n_r}$ be K matching matrices between each batch and the reference - by convention $\mathbf{M}_r = \mathbf{I}_{n_r}$. Each batch $\mathbf{X}_s$ is corrected independently towards the reference $\mathbf{X}_r$. Let $\mathbf{X}_s^m$ be the matched samples of $\mathbf{X}_s$ ($\mathbf{X}_s$ rows so that corresponding row in $\mathbf{M}_s$ contains at least one nonzero element), and $\mathbf{X}_s^u$ be the unmatched samples (the other rows). The first step is to compute the projection of each matched sample to its barycenter $\mathbf{Y}_s^m = F^{\text{Bary}}_{\mathbf{X}_r,\mathbf{M}_s}(\mathbf{X}_s^m)$. For every matched sample, we compute each corection vector

$$\mathbf{C}_s^m = \mathbf{Y}_s^m - \mathbf{X}_s^m. \tag{2.16}$$

Total correction vectors $\mathbf{C}_s$ are then computed as follows. We set the correction vector for every matched sample to the corresponding $\mathbf{C}_s^m$ entry. Otherwise, we set the correction

vector to one of its closest matched samples along the edges of a k-nearest neighbors graph. If there is no matched sample in the sample's connected component, the closest matched sample in terms of distance is selected. Variants exist for this step, for instance, averaging correction vectors among sets of points (e.g., neighborhood or clustering) instead of selecting just one to smooth the final representation. In the end, merging is performed as

$$F^{\mathrm{LC}}_{\mathbf{X}_r,\mathbf{M}_s}(\mathbf{X}_s) = \mathbf{X}_s + \mathbf{C}_s. \tag{2.17}$$

Like barycentric merging, linear correction can be used to correct feature counts of all batches with respect to a reference by computing correction vectors in the full feature space. It can also work with incomplete matchings, where only a subset of samples are matched to reference samples. Specific sample classes in the source batch remain an issue, as relevant matches cannot exist in this case; these samples end up either corrected towards an incorrect sample class, or are linearly translated into empty parts of space. Therefore, choosing an appropriate reference batch is crucial when using linear correction merging, and should not be overlooked.

Graph embedding algorithm

Joint graph embedding is an algorithm capable of building a joint weighted graph of cells from all batches, where two cells are linked together if they appear similar. This graph is weighted according to a UMAP-like methodology, meaning it can be embedded in a low dimensional space using UMAP (Becht et al., 2019) or MDE (Agrawal et al., 2021) optimizers upon minor tuning. The joint embedding algorithm consists of four major steps. More details can be found in the supplementary note.

1. For each batch, compute its k-nn graph weighted according to UMAP membership methodology.
2. For each pair of batches, weight matching edges according to UMAP membership methodology.
3. Build a joint graph combining the edges of steps 1 and 2, possibly selecting only the most impactful edges.
4. Embed the joint graph in an abstract feature space using a graph embedding optimizer such as UMAP or MDE.

Let $\{X_i \subset (\mathcal{X}_i, d_i), |X_i| = n_i\}_{i \leq K}$ be a set of finite datasets to integrate, each expressed in a metric space. Let us also assume we are provided for every pair of datasets X_i and X_j a matching $\mathbf{M}_{ij}$ between X_i samples and X_j samples. We start by computing, for each batch X_i its directed k-nearest neighbors graph $G_i = (X_i, E_i)$, weighted using UMAP methodology for computing membership strength [ref] (see section 2.2.1) resulting in an adjacency matrix $\mathbf{K}_i$ describing a *k-nearest neighbors strength graph*. This guarantees every sample contains at least one edge of weight 1, and helps to uniformize weights regardless of batch-specific point density; note that this graph is not symmetrized yet.

The next step is to convert all matching matrices to membership strength matrices so that edge weights are of the same nature as $\mathbf{K}_i$ k-nearest neighbors graphs. For two batches X_i and X_j associated with a matching matrix $\mathbf{M}_{ij} \in \mathbb{R}^{n_i \times n_j}$, we distinguish two cases.

- If $(\mathcal{X}_i, d_i) = (\mathcal{X}_j, d_j)$ which can often be achieved between datasets of the same data type using common features selection, weights can be chosen using UMAP methodology on the bipartite matching graph $\mathbf{M}_{ij}$ using the distance between matched points, yielding *matching strength graph* described by matrix $\mathbf{S}_{ij}$.

- In the general case $(\mathcal{X}_i, d_i) \neq (\mathcal{X}_j, d_j)$, we can leverage matching strength contained in $\mathcal{M}_{ij}$ to produce a dissimilarity measure, for instance using the trick $\mathbf{D}_{ij} = \text{inv}(\mathbf{M}_{ij} + 1)$ where $\text{inv}(\mathbf{M})$ denotes coordinate-wise matrix inversion. This dissimilarity can then be used to apply the computation of the membership matrix as described in the previous case.

Once all k-nearest neighbors strength matrices $\mathbf{K}_i$ and matching strength matrices $\mathbf{S}_{ij}$ have been computed, they can be assembled in a joint graph G of all batches described by an adjacency matrix $\mathbf{G}$ whose blocks are defined as

$$\mathbf{G} = \begin{bmatrix} \mathbf{K}_1 & \mathbf{S}_{12} & \mathbf{S}_{13} & \dots & \mathbf{S}_{1K} \\ \mathbf{S}_{21} & \mathbf{K}_2 & \mathbf{S}_{23} & \dots & \mathbf{S}_{2K} \\ \mathbf{S}_{31} & \mathbf{S}_{32} & \mathbf{K}_3 & \dots & \mathbf{S}_{3K} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{S}_{K1} & \mathbf{S}_{K2} & \mathbf{S}_{K3} & \dots & \mathbf{K}_K \end{bmatrix}. \tag{2.18}$$

This matrix typically contains a very large number of edges, especially in large-scale applications. This tends to increase the convergence time of the graph embedding step. Furthermore, vertices tend to have a very variable number of edges in $\mathbf{G}$, which can result in embedding instability. To counterbalance these properties, we choose first to carry out an edge pruning step based on $\mathbf{G}$. Given a target number of neighbors $k_t > K$, for every $\mathbf{G}$ row, all values below the k_t-th largest are set to 0 in order to only account for the high-confidence matches. $\mathbf{G}$ is eventually symmetrized into $\hat{\mathbf{G}}$ using

$$\hat{\mathbf{G}} = \mathbf{G} + \mathbf{G}^\intercal - \mathbf{G} \circ \mathbf{G}^\intercal. \tag{2.19}$$

$\mathbf{G}$ can eventually be embedded in an abstract feature space (we often use 2D or 3D in practice) using a graph embedding optimizer such as UMAP or MDE.

2.2.4 Embedding evaluator

An *embedding evaluator* algorithm takes in input an embedding of a set of batches in a common space. It then evaluates embedding quality based on some criteria (e.g., point communities preservation or local label purity) and returns an embedding quality value either per point or global.

Local inverse Simpson's index (LISI)

Local inverse Simpson's index (LISI) is an integration metric first introduced in Harmony (Korsunsky et al., 2019), which assesses neighborhood heterogeneity of a data point in terms of a given label. Simpson's diversity index is a diversity metric notably used in ecology to measure class diversity in a set of objects by computing the probability for two randomly selected items to share the same class. We have chosen to simplify LISI in our implementation by removing the custom UMAP-like cell-cell weighting introduced in Harmony in order to both improve its computational performance and make it less dependent on local geometry. For any set of objects $S = \{x_i\}_{i \leq n}$ endowed with labels $y_i \in \mathcal{L}^n$, we denote by n_l for $l \in \mathcal{L}$ the number of samples in S with label l. Then, Simpson's index of set S is given by

$$D_{\mathcal{L}}(S) = \sum_{l \in \mathcal{L}} \left(\frac{n_l}{n} \right)^2. \tag{2.20}$$

For $k > 0$ a *perplexity* parameter (we use $k = 90$) and x an embedded point, we compute its k-nearest neighbors k-nn(x) which is used as set S. $\text{LISI}_{\mathcal{L}}(x, k) = D_{\mathcal{L}}(k\text{-nn}(x))^{-1}$ is defined as the inverse of Simpson's index and estimates, for a given embedded point x, label diversity in its k-nearest neighborhood. As suggested in Harmony, we can use LISI in two modes:

- Batch-LISI, where points are labeled by their initial batch. This metric measures local batch diversity in embedding, higher diversity meaning higher batch mixing.
- Class-LISI, where points are labeled by their class. This metric measures local class diversity in the embedding, lower diversity meaning lower mixing between cell types.

Monitoring these two values allows objective comparison of different integration pipelines, ideally increasing batch-LISI while avoiding increasing class-LISI.

Local label entropy (LLE)

Local label entropy (LLE) is another objective integration metric based on Shannon entropy (Shannon, 1948), and works similarly as LISI. Given an embedded point x, LLE uses Shannon entropy as a measure of label diversity in its neighborhood k-nn(x) for a perplexity parameter $k > 0$. Let $\mathcal{L}$ be the set of data points labels, and for a point set S let $f_l(S), l \in \mathcal{L}$ be the frequency of label l in S. Then, for $l > 0$ we define the entropic coefficient $h(f) = -f\log(f)$, and we set $h(0) = 0$ by continuous extension as $\lim_{h\to 0^+} f(h) = 0$. LLE is then computed as

$$\mathrm{LLE}_{\mathcal{L}}(x,k) = \sum_{l\in\mathcal{L}} h(f_l(k\text{-nn}(x))). \tag{2.21}$$

As for LISI, we can define batch-LLE (bLLE) and class-LLE (cLLE) to be able to measure either batch or class heterogeneity.

Local topology preservation (LTP)

Local topology preservation (LTP) is another unsupervised integration metric which measures how much the local geometry of batches is affected by integration. Ideally, we would like integration to preserve the datasets' topology, meaning initially similar samples are embedded nearby after integration. LTP compares the samples' neighborhoods before and after integration in order to penalize changes in local geometry.

Let $X_1, \dots, X_K$ be K different batches to integrate into a common embedding space $\mathcal{Y}$. LTP consists of first computing the K nearest neighbors matrices $\mathbf{N}_1, \dots, \mathbf{N}_K$ given neighborhood sizes $k_1, \dots, k_K$. Let $f : \left(\bigcup_{k=1}^K X_k\right) \to Y$ be the integration function, we then compute nearest neighbor matrices $\mathbf{N}'_1, \dots, \mathbf{N}'_K$ on the embeddings $f(X_1), \dots, f(X_K)$. LTP then uses the norms

$$\|\mathbf{N}_1 - \mathbf{N}'_1\|_2^2, \dots, \|\mathbf{N}_K - \mathbf{N}'_K\|_2^2$$

as a measure of local topology distortion after integration. Ideally, a good data integration algorithm should preserve local topology and keep these values as close to zero as possible. On the other hand, using $f(X) = X$ guarantees LTPs to be equal to zero while not performing any form of integration, which proves that LTP cannot be used as a sufficient measure of integration quality.

Cluster label purity

The last type of data integration quality assessment metric I would like to discuss is cluster label purity. Single-cell RNA-seq data tends to form clusters in the gene expression space that regroup cells of similar type or state. For this reason, a reasonable quality assessment approach in cases where high-quality cell type labels are available is cluster label purity. The idea is simple, as cell types provide a natural partition of cells $C_1, \dots, C_K$ with $K \in N$ and $\bigsqcup_{i=1}^K C_i$ containing all cells from all datasets, $\bigsqcup$ denoting the disjoint union. We perform a similarity-based clustering after integration, for instance using the Louvain

Traag et al. (2019) algorithm, and measure how this new partition $D_1, \dots, D_L$ differs from the cell types one with $L \in \mathbb{N}$.

In order to penalize clusters that contain mixed cell types, we compute the joint partition matrix $\mathbf{J} \in [0,1]^{K \times L}$ where $\mathbf{J}_{i,j}$ contains the fraction of cells from cluster j that are labeled i – ideally, each column of $\mathbf{J}$ contains only one nonzero value. The idea is finally to compute the cluster-wise purity vector $\mathbf{p} \in [0,1]^L$ defined as the maximum of $\mathbf{J}$ column-wise, and cluster label purity can be computed as $\|\mathbf{p} - \mathbf{1}_L\|_2^2$, with low values being associated to high cluster purity.

2.3 A few real-life applications of the *transmorph* framework

2.3.1 Single-cell RNA-seq datasets

We used public datasets to benchmark our framework and compare its capabilities with other state-of-the-art integration pipelines. They were chosen to mimic various real-life scenarios, with total dataset sizes in the tens of thousands. All datasets contain RNA-seq data, acquired using 10X technology.

- The Zhou databank was collected from (Zhou et al., 2020) through the Curated Cancer Cell Atlas (3CA) website and contains osteosarcoma data from 11 different patients, ranging from 866 to 14,322 cells for a total of 64,557 cells. Each cell was annotated by the authors with a cell type among chondrocyte, endothelial, fibroblast, mesenchymal stem cell (MSC), myeloid, myoblast, osteoblast, osteoclast, pericyte, T cell.

- The Chen databank was collected from (Chen et al., 2019b) using the 3CA website and contains 61,870 nasopharyngeal cancer single-cell RNA-seq data from 14 different patients, ranging from 1,087 to 11,210 cells. Each cell was annotated by the authors with a cell type among B cell, endothelial, epithelial, macrophage, malignant, NK cell, plasma, and T cell.

Raw counts have been preprocessed following standard guidelines using the *scanpy* python package (Wolf et al., 2018). First, cells with low gene counts or high mitochondrial gene expression were filtered. Raw counts were then normalized to 10,000 per cell, followed by neighborhood pooling using 5 nearest neighbors. Counts were then $\log(1+x)$ transformed, and for each dataset, the top 10,000 most variable genes were kept. All these preprocessed annotated databanks can be automatically downloaded through our framework, in order to serve for benchmarking integration methods.

2.3.2 *transmorph* models perform on par with other state-of-the-art tools

We will first present how the *transmorph* framework can be used to create data integration models able to compute a low dimensional joint embedding of two or more datasets so that similar cells end up close to one another independently from their source. This type of task is typically used for visual data exploration or as a preprocessing step before carrying out a clustering algorithm, allowing clusters to only depend on cell type rather than on the original batch (Fig. 2.2a). A good joint dataset embedding algorithm should be able to function in a fully unsupervised fashion while being improved by additional labeling information, and should not require choosing a reference dataset as this induces an important bias. Ideally, it should also be able to tackle the joint embedding of more than two datasets simultaneously, with reasonable computational efficiency. We built a *transmorph* model for this application, called *EmbedMNN*, described in (Fig. 2.2b).

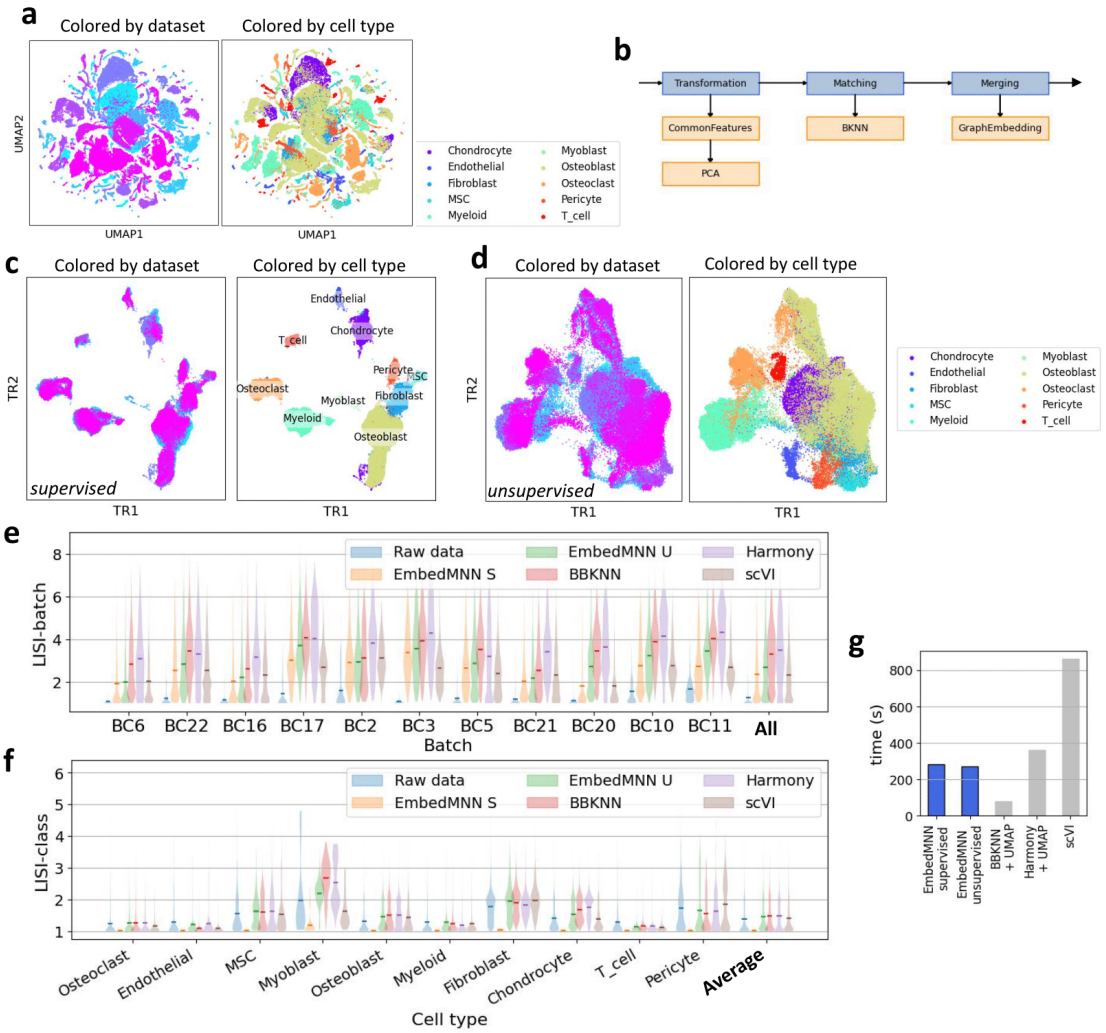

Figure 2.2: **Integration of 11 osteosarcoma scRNA-seq datasets (n=64,557) from different patients.** **(a)** Initial UMAP representation of the osteosarcoma datasets in their common genes space. **(b)** Architecture of the pre-built EmbedMNN integration model, computational modules are executed from left to right and from top to bottom. **(c)** Integration results with the supervised version of EmbedMNN. **(d)** Integration results with the unsupervised version of EmbedMNN. **(e)** LISI-batch score of various integration algorithms (higher is better), mean is marked. **(f)** LISI-class score of various integration algorithms (lower is better), mean is marked. **(g)** Execution time of various integration algorithms. *From (Fouché et al., 2023).*

EmbedMNN is conceptually inspired by CONOS (Barkas et al., 2019), and starts with a few preprocessing steps (normalizations and dimensionality reduction). It then combines a nearest neighbors-based joint graph construction step with a low dimensional graph construction, followed by an embedding step using either UMAP (Becht et al., 2019) or minimum distortion embedding (MDE) (Agrawal et al., 2021). This allows EmbedMNN to work without requiring a reference and with datasets of various topologies, and to output an embedding in a latent space that will be exploitable for clustering and visualization. Furthermore, EmbedMNN can work either in a fully unsupervised fashion or can take into account label information to prune matching edges between samples of different labels; we test both variants in this application.

Even though *transmorph* is not a data integration algorithm *per se*, but rather a framework to conceive data integration methods, we decided to benchmark the EmbedMNN model against other state-of-the-art horizontal integration algorithms. For this benchmark, we selected three algorithms designed to solve the joint embedding problem: Harmony (Korsunsky et al., 2019), which uses a clustering-driven, iterative strategy to optimize the embedded representation. scvi (Lopez et al., 2018), a deep learning framework that uses variational autoencoders to compute a latent integrated representation of datasets. BBKNN (Polański et al., 2020), which builds a weighted joint graph of datasets together using a batch-balanced variant of k-nearest neighbors. We embedded Harmony, scvi latent representation, and BBKNN results into a 2D space using UMAP (Becht et al., 2019) so that all methods' output space is comparable.

The benchmarking databank consists of 11 single-cell osteosarcoma datasets gathered from (Zhou et al., 2020), containing approximately 65,000 cells in total, which have been annotated by the authors with 10 different cell types (Fig. 2.2a). We will use this author annotation as the "ground truth" for this application, and measure how the different methods deviate from it. This use case is quite challenging due to dataset size, number of batches, and number of classes, but illustrates a reasonable real-life use case of data integration. Integration performance can be objectively measured through four integration metrics: batch and class mixing using a lightened version of local inverse Simpson's index (LISI) introduced in Harmony, clustering specificity using Louvain or Leiden community detection algorithm (Blondel et al., 2008; Traag et al., 2019), and computation time. It is to note that Harmony directly uses batch-LISI as a stopping criterion during its optimization procedure, so we have to expect it to have superior batch-LISI scores.

All methods could compute the integrated embedding in a reasonable amount of time given the number of data points (Fig. 2.2g), with the best performer being BBKNN + UMAP with 1min10s, taking advantage of the highly optimized *C++* nearest neighbors approximation library *annoy*. Both supervised and unsupervised versions of EmbedMNN algorithms could finish in under 5 minutes. At the same time, Harmony took 5min30s plus an extra 30s of UMAP computation to obtain a 2D embedding. scvi was the longest to complete, with around 10 minutes in total, but in all fairness, the minimum loss seemed to be reached between the 2 and 3 minutes mark.

Computed joint representations were reasonable overall for all methods, with effective batch mixing and cell type clustering (Fig. 2.2c-d). Nonetheless, no method achieved both excellent batch mixing and cell type separation, which is to be expected on such complex datasets (a large number of cells, patients, and cell types). Unsurprisingly, the supervised version of EmbedMNN outperformed all other methods by a large margin both in terms of local cell types homogeneity and clustering purity (Fig. 2.2f), with a very low LISI-class score for all cell types and a near-100% cluster purity, as it leveraged complete label information. This allowed it to prune edges between cells of different types during the matching step, which resulted in a very clean cells graph to embed. On the other hand, supervised EmbedMNN is associated with inferior batch mixing (Fig. 2.2e), and more explicit cluster delimitation after integration which can be an obstacle for some trajectory inference algorithms. The unsupervised version of EmbedMNN appears to be on par with

the other methods, with good LISI-class and LISI-batch scores (Fig. 2.2e-f) and good clustering purity (Fig. 6.3).

Overall, this shows that *transmorph* provides a framework capable of creating data integration models of sufficient quality to tackle joint dataset integration of challenging scRNA-seq datasets in terms of computational efficiency and integration quality. In the next section, we will show that its modularity allows the user to modify a *transmorph* model to change its output space (from an abstract space to a gene expression space), which is not possible to our knowledge with the other tools presented in this first scenario.

All benchmarks have been run on a laptop equipped with 32GB of RAM, an Intel CPU i7-10750H (12 cores) processor at 5GHz and an NVIDIA GPU GeForce GTX 1650 Ti Mobile.

- EmbedMNN was used on preprocessed counts with *transmorph* v0.2.0, using default parameters: "bknn" matching, 10 matching neighbors, 10 embedding neighbors, UMAP optimizer and 2 dimensions.
- BKNNCorrection was used on preprocessed counts with *transmorph* v0.2.0, using default parameters: "bknn" matching, 30 matching neighbors, 10 linear correction neighbors.
- TransportCorrection was used on preprocessed counts with *transmorph* v0.2.0, using solver="unbalanced", entropy_epsilon=0.02, unbalanced_reg=5.
- Harmony was used with default parameters directly on preprocessed counts using the *rpy2* python interface. We also tried the *harmonypy* python implementation, interfaced *via scanpy*. It successfully converged in under 10 iterations in both cases and produced comparable results.
- scvi was used on raw counts following the authors' guidelines with n_layers=2 and n_latent=30, and was optimized during 124 epochs (automatically chosen by the software).
- We used the *scanpy* implementation of BBKNN on preprocessed counts. We carried out BBKNN with default parameters on a 50-PC representation of datasets using default parameters, using neighbors_within_batch=3 and 10 annoy trees.
- Seurat was used in RStudio after converting AnnData datasets to h5seurat using the SeuratDisk package. We carried out the integration using SelectIntegrationFeatures, FindIntegrationAnchors and IntegrateData with default parameters. We were not able to complete the last integration step despite our efforts due to memory usage issues.

2.3.3 Performing integration in gene space by using an appropriate embedding

In some applications, providing a joint embedding of datasets into an abstract space is not suited, as original features (i.e. genes) do carry important information for output interpretability. This is for instance the case when performing matrix factorization algorithms such as independent component analysis (ICA) or non-negative matrix factorization (NMF), or when annotating cells with appropriate cell types. In this case, it is necessary to perform the integration directly within gene space, which brings some technical difficulties. Notably, gene spaces are often very large which is detrimental to the scalability of distance-based algorithms due to the curse of dimensionality. In this scenario, EmbedMNN, Harmony or BBKNN are not adapted, as they are unable to return their output in full gene space. This would normally imply we need to find another integration tool to carry out the integration in gene space, which would come with important time

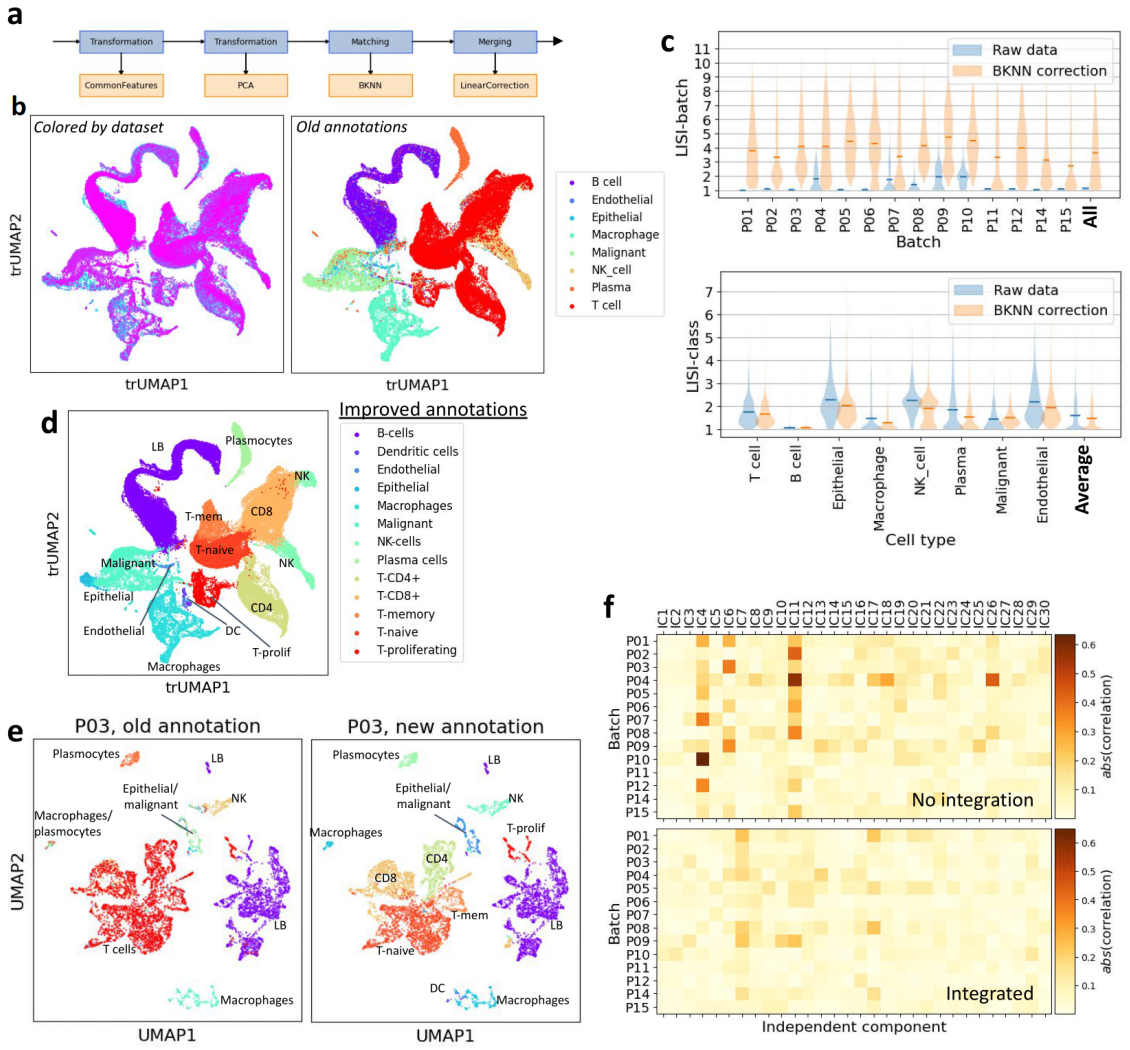

Figure 2.3: **Gene space integration of 14 nasopharyngeal carcinomas scRNA-seq datasets from different patients (n=61,870). (a)** Architecture of the BKNNCorrection pre-built integration model performing integration in gene space. Computational modules are executed from left to right and from top to bottom.. **(b)** UMAP visualization of the integration result, colored by dataset (left) and by original cell type annotations (right). **(c)** LISI-batch (top, higher is better) and LISI-class (bottom, lower is better) before and after integration, mean is marked. **(d)** UMAP representation of the integration result, endowed with new cell type annotations determined within the integrated gene space. **(e)** UMAP representation of the dataset P03, with old (left) and improved (right) cell type annotations. Comparative plots for other datasets can be found as supplementary figure. **(f)** Absolute value of the correlation between each independent component and each batch among T-cells, before (top) and after (bottom) integration. *From (Fouché et al., 2023).*

costs (package installation, data processing, workflow adaptation...). In this example, we demonstrate how the modular nature of the *transmorph* library can instead provide a way to adapt an existing model to suit a new application easily. We first identify that the embedding step of EmbedMNN is by design not adapted to a full gene space application. To tackle this limitation, we can swap this module for something more adapted like a linear correction step in gene space (Fig. 2.3a), which instead leverages correction vectors in a similar fashion to what is used within the MNN (Haghverdi et al., 2018) and Seurat (Butler et al., 2018) tools, and can handle the property of neighbor-based matchings that do not provide a match to every cell from the query dataset. Given a reference dataset, the linear correction approach consists in first, finding some matchings between query and reference items, then computing correction vectors from these queries to their references, to finally propagating these correction vectors along the query dataset to end up with corrected profiles. This last step allows for the alignment of query cells that have no match in the reference dataset. Furthermore, contrarily to graph embedding, linear correction step can be carried out in gene space to obtain a gene expression matrix as output. This makes it a natural choice for this application.

We use 14 nasopharyngeal carcinoma datasets gathered from (Chen et al., 2020) to benchmark the strategy (Fig. 2.3b). The goal is to embed these datasets in the space defined as the intersection of their common most variable genes so that cells sharing the same annotation end up in close proximity after integration. This is once again a challenging task as the datasets are quite large (more than 60,000 cells to embed), there are 8 different cell annotations, some datasets do not contain cells from all types, and the embedding space is large for a geometrical approach (more than 900 genes). To measure integration quality from another angle, we carry out ICA on T-cells from all datasets, which allows us to observe dataset-specific gene expression signals without the bias of cell type imbalance between datasets. As we can see, before integration the dataset-specific signal appears to be strongly correlated with several independent components (ICs) computed by ICA (Fig. 2.3f top).

BKNNCorrection completes in a very reasonable time of 1 minute and 33 seconds and provides a convincing correction (Fig. 2.3b) by being associated with great improvements in LISI-batch (Fig. 2.3c top) while maintaining low levels of LISI-class (Fig. 2.3c bottom). We were not able to successfully carry out Seurat integration on these datasets in a reasonable time and memory usage on this dataset using our machine. Overall, this showcases how *transmorph* provides a new way to easily tweak models, allowing them to tackle different scenarios with good efficiency and integration quality. We also eventually ensure most of the dataset-specific signal has disappeared after integration (Fig. 2.3f bottom), resulting in a weak correlation with any of the ICs recomputed by ICA on the integrated dataset. This is a desired property for subsequent accurate interpretation of the independent components through, for example, functional enrichment analysis.

2.3.4 Gene space integration can be leveraged to annotate cell types reliably

Gene space integration can be leveraged in a very natural way to perform cell type annotation. As integration outputs new gene counts for each cell, these new molecular profiles can be used within the integration space to perform clustering and cell type annotation via differential gene expression analysis. These newly found annotations can be expected to be more precise than annotations performed on each dataset individually and can allow rarer cell types to be identified with high statistical confidence. In particular, most cell type annotation strategies rely on prior cell clustering to label each cluster with a cell type according to marker genes. Frequently, rare cell types do not form a separate cluster in the original datasets due to their limited population size, while they should constitute a larger cluster once datasets have been integrated together. Newly found annotations can eventually be mapped back to the individual datasets. We will use this methodology to

improve annotations found in the previously used nasopharyngeal carcinoma scRNA-seq datasets.

We performed a clustering of datasets integrated into the space of their common genes and performed a differential gene expression on these clusters (Fig. 6.4). We then determined cell types by combining initial annotations, well-known marker genes as well as PanglaoDB (Franzén et al., 2019). Doing so allowed us to confidently annotate 13 different cell types, greatly refining initial annotations (Fig. 2.3b-d). Comparing old and new annotations for each cell shows most annotations have been made more precise rather than corrected (Fig. 6.4), notably splitting the "T cell" label into the various lymphoid lineage-associated labels "T-naive", "T-CD4+", "T-CD8+", "T-memory" and "T-proliferating", and the "macrophage" label into the myeloid lineage-associated labels "macrophages" and "dendritic cells". The only different annotations were among "epithelial", "endothelial" and "malignant", which is to be expected as nasopharyngeal carcinomas are endothelial tumors, making these types hard to strictly separate. All the annotations were eventually be mapped back into the original datasets (Fig. 2.3e), and convincingly annotated clusters that can be seen in exploratory data analysis. This notably allowed the identification of a very small subpopulation of dendritic cells notably characterized by the expression of CCR7 and CCLE9A genes as well as proliferating T lymphocytes, expressing high levels of proliferation markers like MKI67 and PCNA. These subpopulations were too rare in each dataset to form a distinct cluster, which explains why they could not be annotated initially. It is to note that the CD4 gene was not highly variable within all datasets and therefore it was missing in the integrated gene space. We validated the LT-CD4+ cluster by checking the CD4 expression in datasets in which the gene is present (Fig. 6.5). This application shows how the output of *transmorph* gene space models can be used to improve cell type annotations by integrating several datasets directly in gene space.

2.3.5 Transferring cell cycle phase annotations across osteosarcoma and Ewing sarcoma datasets

Cell cycle is one of the most fundamental biological processes through which biological cells grow and divide, but is yet to be fully understood. Single-cell transcriptomics offers great insight into its properties and dynamics, as gene expression regulation is a key factor for cell cycle progression. Gene expression modulation during the cell cycle can be visualized and interpreted by looking at the so-called cell cycle plots. In these plots, each cell is reduced to a small set of coordinates (typically between 2 and 4 (Zinovyev et al., 2022)), each of those corresponding to the average transcription activity of genes associated with a specific cell cycle signal (e.g. G1/S phase, G2/M phase, histones) (Fig. 2.4a). In this configuration, cells revolve along a one-dimensional looping trajectory throughout their progression in the cell cycle. Studying the geometry of these trajectories and cell distribution along them can provide exquisite insight into cell cycle speed, cell growth, or even eventual cell cycle arrest.

A challenging question when studying the cell cycle at the single-cell level is the automatic annotation of cells with cell cycle phases. Some phases like mitosis can be accurately identified by looking at markers such as the total number of raw counts which drops by a factor of two after cell division, but other phases are fuzzier, especially for lower-quality datasets, or fast-cycling cell types. Annotation of scRNA-seq data with cell cycle phases was studied experimentally in (Mahdessian et al., 2021b), where the authors used genetic constructs to follow the abundance of key cell cycle proteins which they can then relate to cell cycle phases, but doing so comes with important costs and experimenter time; a natural idea would be to transfer labels from datasets annotated using this methodology to other unlabeled ones. Unfortunately, this is not as easy as it seems: differences in preprocessing, cell types, and cell cycle properties can quite drastically affect a dataset topology and geometry, making many proximity-based methods irrelevant. A natural label transfer strategy can be pictured as follows (Fig. 2.4b). First, we carry out data integration of all

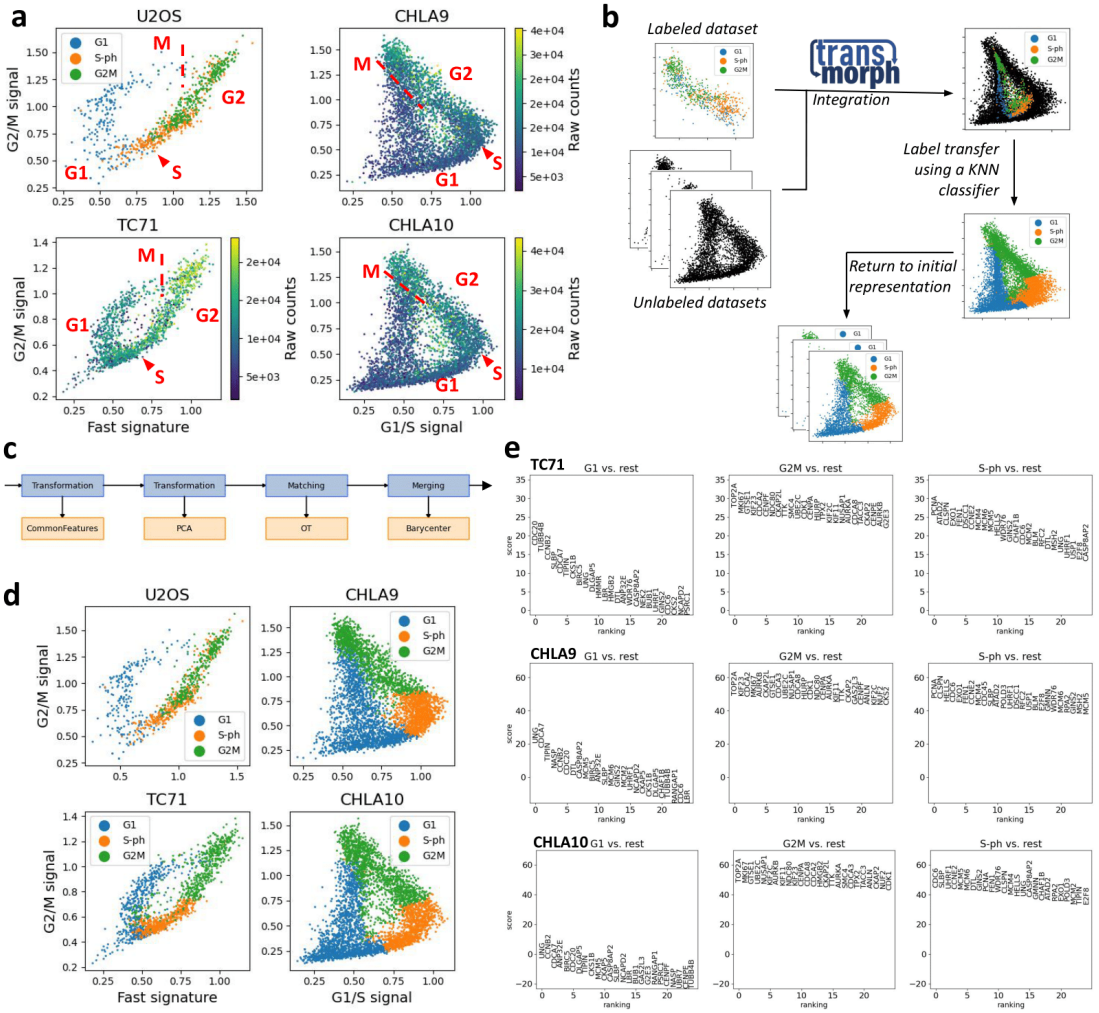

Figure 2.4: **Transferring cell cycle phase annotations between osteosarcoma (U2OS, TC71) and Ewing sarcoma (CHLA9, CHLA10) scRNA-seq datasets.** **(a)** Visualizing the cell cycle loop of each dataset, approximate positions of cell cycle phases are annotated. U2OS annotations are provided by the authors, other datasets are colored according to the number of read counts. **(b)** Schematic strategy for the data integration-based label transfer. **(c)** Architecture of the TransportCorrection pre-built integration model performing integration in gene space. Computational modules are executed from left to right and from top to bottom.. **(d)** Automatically transferred annotations using the TransportCorrection model. **(e)** Differential gene expression was performed using a Wilcoxon rank-sum test, showing the most specific genes associated with cells of each label. *From (Fouché et al., 2023).*

datasets into a common embedding space. Then, we predict cell cycle labels of unlabeled datasets in this common space using a supervised learning approach. Finally, the learned labels can be transferred back to the original representations to be interpreted.

In this experiment, we seek to automatically annotate three single-cell RNA-seq Ewing sarcoma datasets (CHLA9, CHLA10, and TC71) gathered from (Miller et al., 2020b) (Fig. 2.4a). To do so, we will attempt to transfer cell cycle phase, author-provided annotations contained in an osteosarcoma dataset (U2OS) gathered from (Mahdessian et al., 2021b), onto the three Ewing sarcoma datasets.

Raw counts for Ewing sarcoma cell lines datasets CHLA9, CHLA10, and TC71 were obtained from (Miller et al., 2020b). Raw counts and annotations for the osteosarcoma U2OS dataset were obtained from (Mahdessian et al., 2021b). They were preprocessed according to state-of-the-art guidelines. Raw counts per cell were normalized to 10,000 to account for differences in global expression and were then $\log(1+x)$ transformed; the top 10,000 variable genes were kept in each dataset. Data points were eventually pooled to reduce noise, by setting every cell counts vector to the average of its 5 nearest neighbors (neighbors were determined using euclidean distance in a 30-PC space). We used cell cycle genes identified in (Tirosh et al., 2016) to characterize G1/S and G2/M signals. For fast cell cycle datasets TC71 and U2OS, we used only a subset of informative G1/S genes which helped to retrieve a proper loop signal (CDK1, UBE2C, TOP2A, TMPO, HJURP, RRM1, RAD51AP1, RRM2, CDC45, BLM, BRIP1, E2F8 and HIST2H2AC). Integration was carried out using full gene space.

Preprocessing differences, geometrical specificities and apparent S/G2M label mixing within the U2OS reference dataset are tough difficulties to overcome both for integration and label transfer methods. We first attempt to perform the integration using BKNNCorrection, setting CHLA10 as the reference dataset considering its good quality and representativity (cells are scattered uniformly around the trajectory, and the central "hole" is well resolved). Unfortunately, predicted cell cycle labels are not satisfying (Fig. 6.2): post-mitotic cells are associated with the G2/M label, S-phase is labeled too late on the trajectory, and some early G1 cells are labeled as S. This disappointing performance may be caused by a lack of orthogonality between cell cycle factors and batch effects. This is a crucial hypothesis for all neighbors-based dataset integration, not satisfying it results in a poor matching quality making integration unreliable.

This motivates the need to seek a more appropriate matching algorithm for this situation. We choose here a transportation-based matching, which is robust for applications where information is contained in data topology. It relies on discrete optimal transport that has been brought into the scRNA-seq field a few years ago in (Schiebinger et al., 2019), which can be pictured as looking for the most economical way to move mass in a metric space from a point cloud onto another. This class of problems yields a natural and harmonious way to match cells across batches, by operating at the dataset level instead of operating at the cell level like in MNN. We can use the *transmorph* pre-built model TransportCorrection inspired from SCOT (Demetci et al., 2020) and Pamona (Cao et al., 2022b), which consists of a few preprocessing steps followed by a transport-based matching, used to project every query item onto the barycenter of its matches (Fig. 2.4c). In this case, we had to use the unbalanced formulation of optimal transport (Liero et al., 2018; Peyré et al., 2019) to account for cell cycle phase imbalance between "standard" and "fast" cell cycle datasets; this variant is also implemented in our framework. Label transfer using this model instead of BKNNCorrection yields much better labeling, entirely interpretable and in line with the patterns we expect for the "standard" and "fast" cell cycle (Fig. 2.4d). We see mitosis point is now well identified by the automatic annotation, and S-phase labels are better located. Differential gene expression between the different identified labels yields well-known cell cycle genes specific to each phase, which shows annotation is accurate (Fig. 2.4e). Among these genes we notably see a few well-known ones appear in all profiles such as the TOP2A gene which is associated with the G2/M phases,

PCNA with the S phase, and CDC20 with the G1 phase. Therefore in this scenario, the transportation-based matching was clearly better suited than the nearest neighbors-based one and allowed an accurate cell cycle label transfer. This shows how important choosing the right matching can be, and how *transmorph* addresses it.

In chapter 4, we will present another optimal transport-based algorithm we developed to perform horizontal integration of single-cell data in the space of cell cycle genes. Instead of optimizing an unbalanced optimal transport problem, we solve a kernel optimization task to weight cells so that those in crowded regions of the space weight less than the ones in sparser regions. We will discuss the methodology and the rationale in more details in chapter 4.

2.3.6 Data integration of Ewing sarcoma datasets

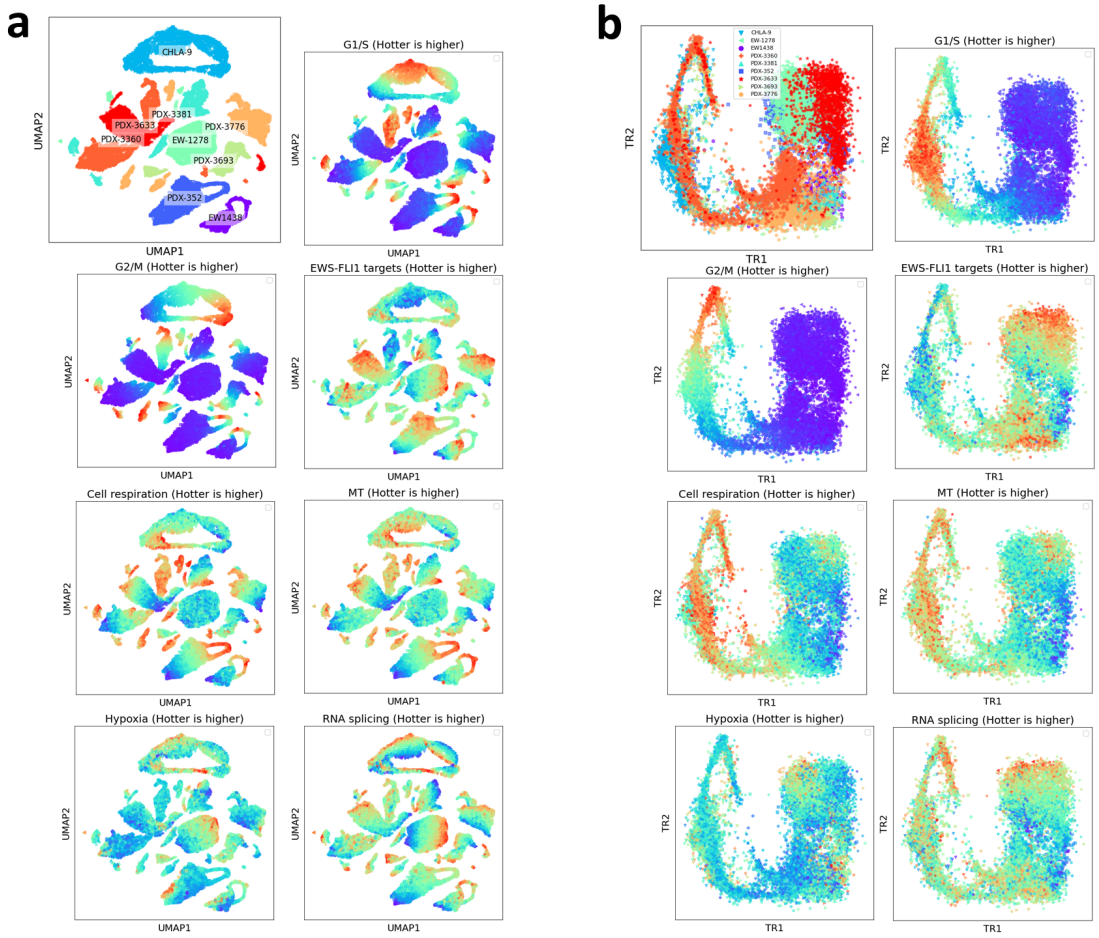

Figure 2.5: **Using *transmorph* to perform data integration of Ewing sarcoma datasets (PDX, cell lines, tumors). In (a) and (b), top left plot is coloured by original dataset. Other plots are couloured by cell process signal, purple is low, red is high.** (a) Common genes UMAP plot of the datasets before integration. (b) Datasets embedding after applying the EmbedMNN data integration pipeline, showing cell states are better localized.

We would like to showcase another application of the *transmorph* package for the horizontal integration of public Ewing sarcoma (EWS) scRNA-seq datasets. Integrating cancer data is challenging for several reasons: first, cell states are highly heterogeneous in such datasets, and are more subtle than cell types; as we can see in Fig. 2.5a, there is no clear clustering within each EWS dataset. Furthermore, we attempt here to integrate data

from different EWS models: Patient-Derived Xenografts (PDX352, PDX3360, PDX3381, PDX3633, PDX3693, PDX3776), cell lines (CHLA9), and tumors (EW1278 and EW1438). All these EWS models have their biological specificities discussed in the introduction, which bring important biological biases. Finally, all these datasets originally come from different patients, with all the inter-individual biases we can think of (age, sex, ethnicity, environment...). For this reason, such an integration task is highly challenging in practice, and we can see in Fig. 2.5a a strong clustering per dataset.

We used the gene signatures identified in (Aynaud et al., 2020) to estimate the activity of 7 cell processes in those datasets via the expression of their respective genes: G1/S cell cycle phase, G2/M cell cycle phase, EF1 (the EWS oncogene) targets expression, cell respiration, mitochondrial genes, hypoxia and RNA splicing. We observe in Fig. 2.5a some activity of these target signals in various parts of the plot, and we would like to co-embed all these cells so that cells with similar signals end up close to one another. We chose to use the EmbedMNN algorithm presented in section 2.3.2, as it is a natural data integration pipeline to pick when we just need a co-embedding of several datasets.

The embedding results after applying EmbedMNN are shown in Fig. 2.5b. We can see that all of the datasets are embedded into a single cloud with a clear looping part. Upon closer inspection, this looping structure is associated with cells that present high cell cycle gene expression, and also metabolic signals (glucose metabolism and mitochondrial genes), which suggests that proliferating cells have been correctly grouped together. We also notice the loop contains a region associated with the expression of G1/S genes, and another region associated with the expression of G2/M genes, which suggests that these more subtle proliferation signals have also been accounted for by the algorithm. We also observe good colocalization of other gene signature levels in the embedding. All this suggests that EmbedMNN was able to satisfyingly integrate these Ewing sarcoma datasets into a common space, and was able to respect Ewing sarcoma cell states while doing so.

2.4 Discussion

Horizontal data integration and batch effect correction are key computational challenges, especially in computational biology to be able to properly analyze single-cell data from different batches or patients (Argelaguet et al., 2021). We identified the need for modular methods to tackle this problem, and demonstrated the necessity to carefully combine trustworthy cell-cell similarity algorithms with relevant embedding algorithms. We also clearly showed how deceiving data integration can be when carried out improperly, which can be detrimental to subsequent analyses. This alone motivates the need for more modular tools, where every algorithmic step can be controlled if necessary. To address this need and instead of introducing yet another data integration technique we present *transmorph*, a novel modular computational framework for data integration, implemented as an open-source *python* library. We provided a robust implementation for it and demonstrated its value through various real-life applications both in terms of efficiency, quality, and versatility. We would like to highlight that EmbedMNN and TransportCorrection models represent original and previously not proposed combinations of base algorithms that were connected into complete data integration methods, using *transmorph* as a toolbox for fast building and testing of data integration models. Furthermore, these pre-built models can easily be transformed into to a combinatorial number of alternative models by changing their constructor parameters (preprocessing steps, matching type, optimal transport flavor, supervised or unsupervised behavior, gene space output, or linear subspace output).

If *transmorph* is an expressive data integration framework that provides a way to articulate multiple algorithmic modules together in order to shape data integration pipelines, there still exists some expressiveness limitations to overcome. In particular, if trained deep learning models such as deep autoencoders (DAE) can be used as custom *transfor-*

mation modules, *transmorph* does not provide a way to either train or fine-tune them without relying on external libraries. For this reason, we think it is useful to mention the development of some recent DAE-based data integration algorithms, that use different approaches to couple several algorithmic paradigms such as Uniport (Cao et al., 2022a) and MATHCLOT (Gossi et al., 2023) that combine DAE and optimal transport, or SMILE (Xu et al., 2022b) that replaces the decoder part by an information-based evaluator. Even if these different tools do not provide as much modularity as *transmorph* to deal with very different biological applications of horizontal data integration, they are certainly better suited for cases necessitating higher levels of abstraction such as cross-modality (vertical, diagonal, and mosaic) data integration.

We feel that increasing modularity and user agency often leads to bloated, over-engineered, and impractical pieces of software. For this reason, we provide via *transmorph* several pre-built integration models ready to be used in daily workflows, with high efficiency and integration quality. For more advanced and specific applications, our framework also allows building integration models from scratch by combining a variety of algorithmic modules, all of which are implemented and optimized inside our library. We eventually provide complete interfaces which allow users to implement their own computational modules if they need to. All this is endowed with a rich software ecosystem including benchmarking datasets, integration metrics, monitoring, and plotting tools as well as interfaces with other state-of-the-art data integration tools like Harmony (Korsunsky et al., 2019) and scvi (Gayoso et al., 2022).

We plan to continue maintaining *transmorph* in the future, in order to keep it up to speed with the ever-growing field of data integration methods. We will continue expanding it with new algorithms, either already existing or to come. We also would also like to add more support for vertical and diagonal integration, as for now the only diagonal matching is based on Gromov-Wasserstein which has a hard time scaling to the size of current data integration problems. For instance, we plan to use gene space transformation to deal with specific vertical integration cases such as integration between RNA-seq and ATAC-seq data. We would eventually like to add domain adaptation methods to our framework (for instance by including supervised PCA (Barshan et al., 2011) or domain adaptation PCA (Mirkes et al., 2022) to our preprocessing steps), in order to tighten the bridge towards this growing research field which presents many similarities with data integration.

There are still crucial questions to be answered in order to provide trustable data integration methods, especially in single-cell biology. Among these questions are the definition of relevant metrics to measure dissimilarity between cells (even more importantly across different domains), the research of sound and unbiased ways to measure integration quality, and the necessity to continue to carry out exhaustive benchmarks to identify the most appropriate data integration methods and algorithms for a given use case.

Chapter 3

Unsupervised weights selection for optimal transport-based dataset integration

Adapted from (Fouché and Zinovyev, 2021).

Recent democratization and flourishing of biological assays at the single-cell level raise important challenges in subsequent analysis pipelines (Lähnemann et al., 2020). One of those, known as dataset integration, is of particular interest and aims at tying data together across different datasets, samples, and modalities (Argelaguet et al., 2021).

Any single cell omics dataset contains biases that may be related to interindividual specificities, tissue composition and preparation, sequencing technology, experimental variation, or pipeline parameters. Depending on the study context, some or all of these factors can be irrelevant to answering the biological questions of interest. On the other hand, there exist differences between datasets, such as dataset-specific cell populations or differences in cell type proportions. The dataset integration techniques try to address a problem consisting of regressing out the irrelevant biases, while preserving insightful specificities to avoid "overcorrection", ideally in an unsupervised fashion. For this reason, dataset integration is a critical step in any analysis pipeline that includes a step where data from different sources (e.g., collected from different tumor samples) are combined. Without a well-developed dataset integration methodology, in the single-cell data analysis field, we are doomed to deal with one biological sample at a time. Failing to eliminate improper biases or removing dataset key specificities typically compromises the success of subsequent visualization, clustering, dimensionality reduction, and prediction techniques; this usually results in misleading interpretations and altered biological insights.

As mentioned in the introductions, several approaches have been proposed to solve horizontal integration problems. In single-cell biology, mutual nearest neighbors-based methods have become popular notably in Seurat (Adey, 2019; Barkas et al., 2019). There also exists techniques based on generative adversarial networks like MAGAN (Amodio and Krishnaswamy, 2018), variational autoencoders (Simidjievski et al., 2019), or integral probability metrics like MMD-MA (Liu et al., 2019). Among those, a class of methods uses optimal transport (OT) to evaluate the similarity between cells across datasets. These OT-based methods are generally derived from an algorithm proposed for histogram transfer in image processing (Ferradans et al., 2013). This was recently brought into the single-cell field with the SCOT (Demetci et al., 2020) and Pamona (Cao et al., 2020b) tools, which are built upon an OT variant called Gromov-Wasserstein (GW). We propose an extension of this class of techniques, focusing on tackling heterogeneity in cell types and phenotypes between datasets.

We propose in this chapter an original density-based extension of the OT-based integration pipeline geared towards tackling cell type imbalance issues (Fig. 3.1, **d.**). We

challenge our approach against both balanced and unbalanced OT techniques implemented in SCOT on four pairs of synthetic and biological datasets, with balanced or unbalanced datasets. We demonstrate in these benchmarks our method to be more robust than the original one, with reasonable extra cost in terms of computer memory and time. We also provide a reasoned comparison between OT- and GW-based methods. We eventually discuss the limitations of this class of methods, and possible extensions.

3.1 General outline of the suggested single-cell dataset integration methodology

OT integration pipelines generally start by computing a discrete optimal transport plan between a source and a reference dataset, which describes how to displace source dataset samples onto reference ones at a minimal cost. This transport plan, hypothesized to reflect sample-sample similarity between datasets, is then used to compute a barycentric projection of source samples in the reference space (Ferradans et al., 2013).

We extend this OT integration pipeline with an extra preprocessing step that adjusts sample weights depending on local point density (Fig. 3.1d). The intuition behind this approach is to tackle the issue of cell type imbalance, when proportions in cell types differ between datasets. By increasing sample weights in sparse regions and decreasing sample weights in populated regions, we hope to uniformize weighting per cell type over both datasets, correcting cell type imbalance and emphasizing geometric constraints. These custom weights are used during the optimal transport and integration steps.

We propose a kernel method to adjust weights, seeking weights in the probability simplex that minimize the empirical variance of the weighted sum of kernels over dataset samples. We formulate this optimization problem as a standard quadratic optimization problem, that can be dealt with using state-of-the-art QP solvers.

The whole pipeline is summarized in (Fig. 3.1, **d**). OT and GW can be computed with the help of the `python` package `pot` (Flamary and Courty, 2017). It notably features `C`-accelerated implementations of both Wasserstein and GW distances with very good performance, associated with computation times typically between one and ten seconds for all datasets we use.

In order to test the suggested methods, we defined the following datasets (see section 3.10 for a more detailed description, Fig 3.1a-c):

1. A pair of synthetic 1D datasets embedded in a 3D spiraling shape
2. A pair of public Ewing's sarcoma datasets embedded in cell cycle genes space
3. A pair of datasets obtained with the scSNARE-seq assay (Chen et al., 2019b), one with chromatin accessibility profiles and the other containing gene counts with matched cells

A spiraling domain is interesting for several reasons. First, the underlying domain is continuous, which is a good stress test for integration methods that must preserve manifold continuity. Also, a spiraling pattern can potentially challenge the integration techniques (some points close to the spiral center can be mapped to the "external" regions of the reference spiral). Quality assessment is eventually performed by comparing, for each integrated sample, its position in the initial spiral versus in the integrated one.

We chose CHLA9 Ewing sarcoma cell line scRNASeq dataset ($n = 3752$) from (Miller et al., 2020b) as a reference, because the differences in cell cycle phases comprised the most important source of transcriptomic heterogeneity in this dataset (Fig. 3.1, **b**). PDX352 patient-derived xenograft profiled using scRNASeq ($n = 1937$) was chosen as the query dataset among those published in (Aynaud et al., 2020), for its high proportion of non-proliferative cells and with a clear cyclic structure corresponding to proliferating cells and

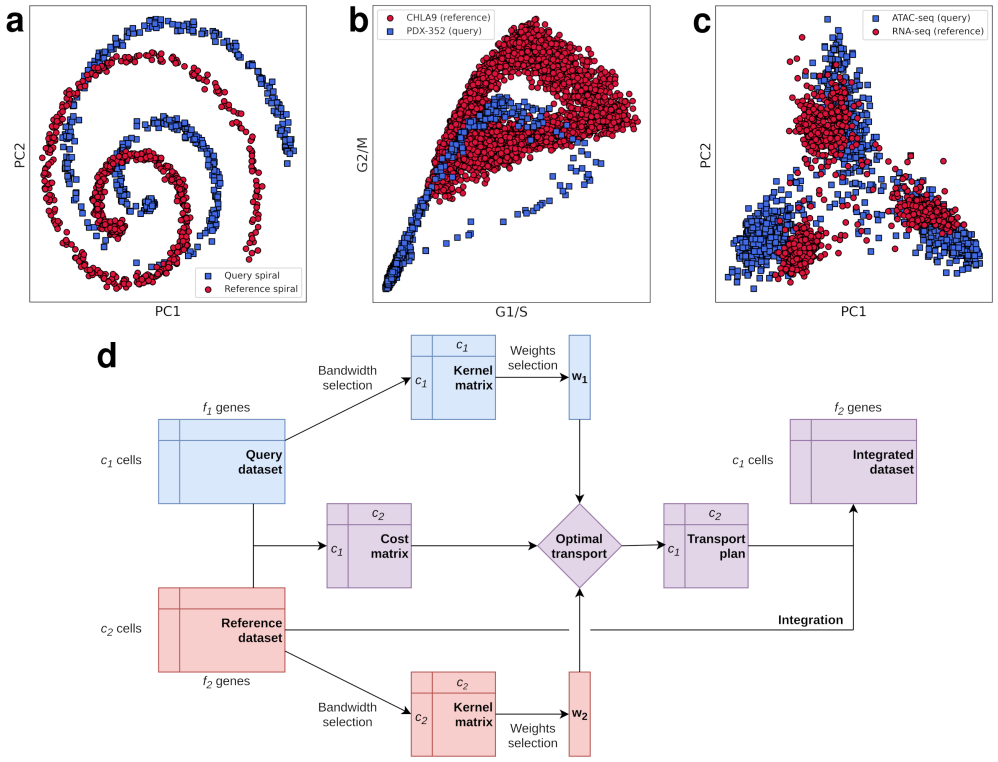

Figure 3.1: **Datasets overview and algorithm architecture.** (a) Synthetic spiraling datasets embedded in a 2-PC space. (b) Ewing sarcoma single-cell datasets embedded in the cell space (x-axis: G1/S genes signal, y-axis: G2/M genes signal). (c) Cell lines mixture acquired through scSNAREseq embedded in a 2-PC space. (d) A schematic view of the weighted optimal transport-based integration pipeline. *From (Fouché and Zinovyev, 2021).*

its high number of cells. This pair of datasets is a good stress test for integration methods, as they have quite similar support domains but with very different cell state distributions; they were also the largest PDX and cell line datasets in the collection.

Eventually, the scATACseq case is interesting as an application of multi-omics dataset integration, frequently used in dataset integration benchmarks. Indeed, cells being matched between datasets facilitates integration quality assessment.

3.2 Method for kernel density uniformization

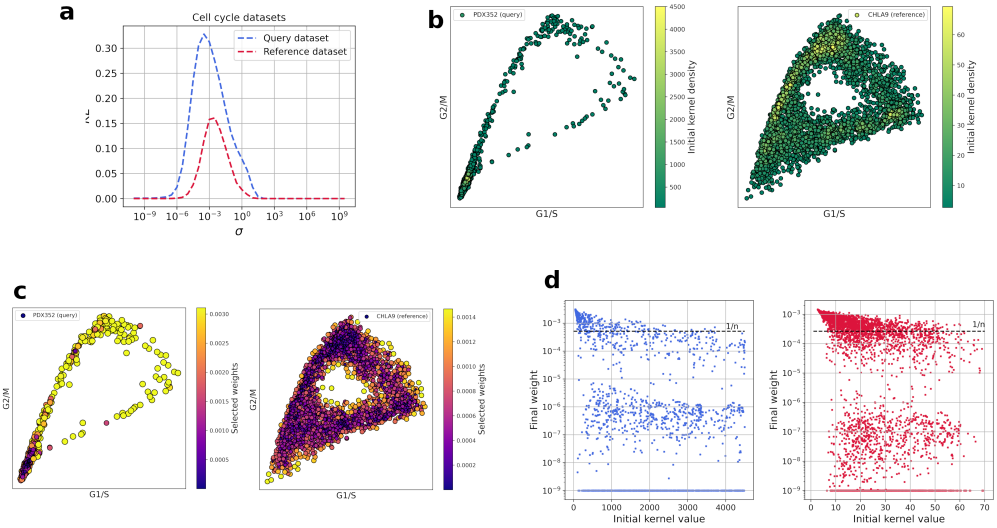

Figure 3.2: **Bandwidth and weighting selection of cell cycle datasets.** (a) Bandwidth choice over cell cycle datasets. Blue: Query dataset. Red: Reference dataset. (b) Point-wise Gaussian kernel density of each dataset before weights selection. Right: Query dataset. Left: Reference dataset. (c) Point-wise weights selection, illustrated by color and dot area. Right: Query dataset. Left: Reference dataset. (d) Relationship between initial point-wise Gaussian kernel density and selected weight. Right: Query dataset. Left: Reference dataset. *From (Fouché and Zinovyev, 2021).*

We propose a density uniformization method to adjust sample weights before unbalanced OT or GW dataset integration, increasing the weights of samples in sparse regions and decreasing the weights of samples in populated regions.

Let $\{\boldsymbol{x_i}\}_{i\leq n}$ be a dataset consisting of n samples in a vector space $\mathcal{X}$ endowed with n distance-based positive semi-definite kernels $\mathcal{K}_i(\boldsymbol{x}) = f_i(\|\boldsymbol{x}-\boldsymbol{x_i}\|_2)$. Let us further assume that for all $i \leq n$, $\int_{\mathcal{X}} \mathcal{K}_i(\boldsymbol{x})dx = 1$. For every $\boldsymbol{x} \in \mathcal{X}$, we define the weighted sum of kernels at $\boldsymbol{x}$

$$w_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \sum_{i=1}^{n} \boldsymbol{\alpha}_i \mathcal{K}_i(\boldsymbol{x}) \tag{3.1}$$

with $\boldsymbol{\alpha} \in \mathbb{R}^n$. We furthermore constraint $\boldsymbol{\alpha}$ to be contained in the probability simplex, so that $\sum_{i=1}^{n} \boldsymbol{\alpha}_i \mathcal{K}_i(.)$ is the PDF of a probability distribution of $\mathcal{X}$. This means $\boldsymbol{\alpha} \succeq \mathbf{0}_n$ (coordinate-wise comparison) and $\boldsymbol{\alpha}^T \mathbf{1}_n = 1$.

We can write an expression for the empirical variance of $w_{\boldsymbol{\alpha}}$ over the dataset,

$$v(\boldsymbol{\alpha}) = \frac{1}{n}\sum_{i=1}^{n}(w_{\boldsymbol{\alpha}}(\boldsymbol{x_i}) - \mu_{\boldsymbol{\alpha}})^2 \tag{3.2}$$

with $\mu_{\boldsymbol{\alpha}} = \frac{1}{n}\sum_{j=1}^{n} w_{\boldsymbol{\alpha}}(\boldsymbol{x_j})$.

The kernel uniformization problem can then be stated as minimizing $v(\boldsymbol{\alpha})$ over the probability simplex,

$$\begin{aligned} \min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \quad & \frac{1}{n}\sum_{i=1}^{n}(w_{\boldsymbol{\alpha}}(\boldsymbol{x_i}) - \mu_{\boldsymbol{\alpha}})^2 \\ \text{s.t.} \quad & \boldsymbol{\alpha} \succeq \mathbf{0}_n \\ & \boldsymbol{\alpha}^T\mathbf{1}_n = 1 \end{aligned} \tag{3.3}$$

Let $\boldsymbol{K} \in (\mathbb{R}^+)^{n\times n}$ be the kernel matrix defined as $K_{ij} = \mathcal{K}_j(\boldsymbol{x_i})$, Eq. 3.2 can be rewritten as a quadratic form of $\boldsymbol{\alpha}$ with $\boldsymbol{M} = \boldsymbol{K}^T\mathbf{1}_n\mathbf{1}_n^T$

$$v(\alpha) = \frac{1}{n}\boldsymbol{\alpha}^T(\boldsymbol{K}-\boldsymbol{M})^T(\boldsymbol{K}-\boldsymbol{M})\boldsymbol{\alpha} \tag{3.4}$$

Eq. 3.3 defines a quadratic program constrained on a simplex, also known as *standard quadratic optimization problem* (Bomze, 1998) that cannot be solved analytically, motivating the usage of interior point methods. We used the Python implementation of `osqp` (Stellato et al., 2020) to carry out the computation, using a Gaussian kernel based on the Euclidean distance matrix (variance-normalized).

3.3 Divergence maximization is a proper heuristic for bandwidth selection

We used a Kullback-Leibler (KL)-divergence maximization heuristic to select in an unsupervised fashion a reasonable bandwidth for a Gaussian kernel used in computing weights (see section 3.10 for details).

For all the datasets we used for testing, the KL-divergence with respect to sigma appears to contain a single maximum with σ in the interval $[10^{-10}, 10^{10}]$ (Fig. 3.2a, Fig. 6.10-6.11-6.12a). These maxima correspond to the ones we find by fine-tuning the bandwidth by hand, and yield quite close to uniform density profiles with small variability (Fig. 3.2b, Fig. 6.10-6.11-6.12b). Our automatic bandwidth selection algorithm implementation converged to a solution in a reasonable time (under a minute for all datasets).

3.4 Quadratic program greatly reduces kernel density empirical variance

We first verify as a sanity check that carrying out the weighting procedure reduces density variance as expected over our eight datasets, and by how much. We compare empirical density variance over all datasets using uniform weights (with $\alpha_i = n^{-1}$ for every sample), and using weights minimizing density variance over the dataset (Eq. 3.3). Tab. 3.1 shows empirical variance with and without adjusted weights; we see using weights minimizing the quadratic program decreases the empirical variance by several orders of magnitude in all eight datasets. Computation time on our setup varies from 250ms for spiral datasets to a few tens of seconds for CHLA9. Computation time is directly related to dataset size, which conditions the quadratic program's dimensionality.

We can visualize the result of density uniformization on Ewing sarcoma datasets in Fig. 3.2, **c**, where chosen weights are represented by both color gradient and dot area. As expected, samples in populated regions are associated with below-average coefficients, while samples in sparse regions like loop borders are associated with above-average coefficients. This directly implies the reference dataset needs to be of high quality, without outliers as they would be associated with high coefficients and probably fool downstream analyses. However, based on such estimation, it may be possible to identify outliers as abnormally large weights and set their weights to zero using a threshold. Similar plots

Dataset	Variance, uniform weights	Variance, QP weights	Change ratio
Spiral (Query)	1.43×10^{-5}	4.53×10^{-8}	3.16×10^{-3}
Spiral (Reference)	1.81×10^{-5}	4.01×10^{-8}	2.21×10^{-3}
PDX352 (Query)	1.80×10^{6}	3.07	1.70×10^{-6}
CHLA9 (Reference)	2.05×10^{2}	1.64×10^{-2}	7.97×10^{-5}
scATACseq bal. (Query)	1.20×10^{-1}	1.25×10^{-3}	1.05×10^{-2}
scRNAseq bal. (Reference)	2.16×10^{1}	2.32×10^{-2}	1.08×10^{-3}
scATACseq imb. (Query)	1.06×10^{-1}	1.88×10^{-3}	1.18×10^{-2}
scRNAseq imb. (Reference)	3.33×10^{1}	2.60×10^{-2}	7.80×10^{-4}

Table 3.1: Gaussian density variance before and after reweighting over each dataset.

for the six other datasets can be examined in Fig. 6.10-6.11-6.12c, and display a similar trend.

Let us eventually examine the relationship between initial density at a point, and its estimated weight. We expect points with high initial density values to be associated with low weights and vice-versa. Fig. 3.2d confirms this trend, but with some interesting extra observations. First, we notice that the reweighting rule does not seem to be uniquely determined by initial density at point, as we observe a wide range of weights for a given initial kernel value. We also observe a quite clear gap between "large weight points" ($\boldsymbol{\alpha}_i \geq 10^{-5}$) and "small weight points" ($\boldsymbol{\alpha}_i \leq 10^{-5}$), suggesting the weighting method to adopt a kind of "all or nothing" strategy. These observations suggest naive weighting methods, such as weighting points in an inverse proportional fashion with respect to initial density, not to coincide with a variance-minimizer weighting. All these observations can be repeated for Ewing's sarcoma datasets (6.10d). Interestingly, for scSNAREseq datasets, the final weight of each point seems to exactly match the normalized inverse of the initial weighted kernel sum at this point (6.11-6.12d).

3.5 Weighted dataset integration

OT or GW can in the discrete case be used as an integration technique for vectorized datasets, originally proposed for histogram color transfer in image processing (Ferradans et al., 2013). We propose to extend this approach to the custom weights case.

Let $\boldsymbol{X}$ and $\boldsymbol{Y}$ be two matrices of $\mathbb{R}^{n \times d_X}$ and $\mathbb{R}^{m \times d_Y}$ representing two vectorized datasets containing respectively n and m samples. Distance information is necessary to carry out OT or GW. For OT, let $\boldsymbol{C_{XY}} \in \mathbb{R}^{n \times m}$ be the matrix containing pairwise distances between $\boldsymbol{X}$ and $\boldsymbol{Y}$. For GW, let $\boldsymbol{C_X} \in \mathbb{R}^{n \times n}$ and $\boldsymbol{C_Y} \in \mathbb{R}^{m \times m}$ be two matrices containing pairwise distances in $\boldsymbol{X}$ and in $\boldsymbol{Y}$. Let $\boldsymbol{P^*}$ be the optimal transport plan from $\boldsymbol{X}$ to $\boldsymbol{Y}$, computed either using OT or GW, assuming samples from $\boldsymbol{X}$ (resp. $\boldsymbol{Y}$) are associated either to uniform weights, or weights obtained via the uniformization procedure described in subsection 3.2. We denote these weights by $\boldsymbol{\alpha_X}$ and $\boldsymbol{\alpha_Y}$. The idea is then to consider each row in

$$\boldsymbol{T} = \mathrm{diag}(\boldsymbol{U}\mathbf{1}_m)^{-1}\boldsymbol{U} \tag{3.5}$$

as a probability distribution, with $\boldsymbol{U} = \boldsymbol{P^*}\,\mathrm{diag}(\boldsymbol{\alpha_Y})^{-1}$ – by construction, each row sums up to 1. Namely, $\boldsymbol{T_{ij}}$ is interpreted as a probability, $\mathbb{P}(\boldsymbol{x}_i$ corresponds to $\boldsymbol{y}_j)$. We can then derive an expression for the predicted $\boldsymbol{X}$ dataset position after a weighted barycentric integration,

$$\boldsymbol{X'} = \boldsymbol{TY}. \tag{3.6}$$

As a last step, we choose to apply an individual small stochastic coefficient to each integrated point (typically sampled from a normal distribution with mean 1 and standard deviation 1%) in order to avoid the case where several points are exactly mapped onto the same location. We found in our tests that this trick greatly improves the convergence time of some downstream algorithms such as UMAP (Becht et al., 2019).

3.6 Integration results on synthetic datasets

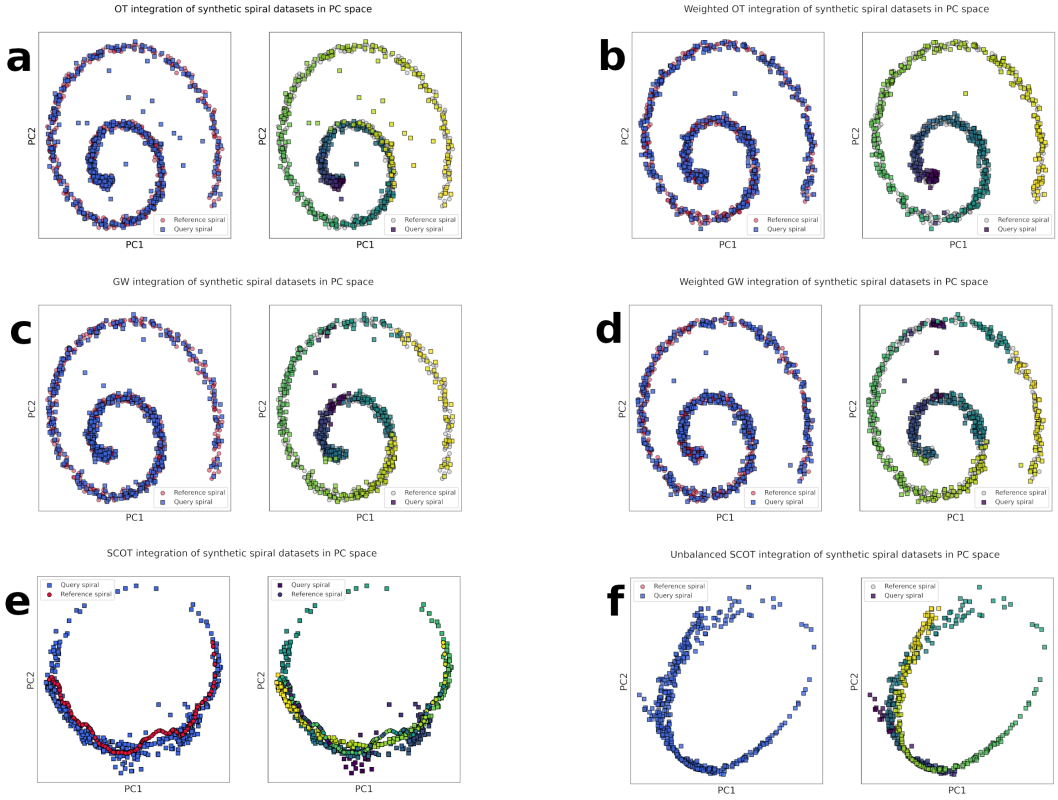

Figure 3.3: Comparison of integration methods on synthetic spiraling datasets. Left subpanels: colored by original dataset. Right subpanels: colored by their initial position in the spiral, integration should preserve the color gradient. **a.** Unweighted optimal transport-based integration. **b.** Weighted optimal transport-based integration. **c.** Unweighted GW-based transport-based integration. **d.** Weighted GW-based transport-based integration. **e.** Balanced SCOT integration. **f.** Unbalanced SCOT integration.

We first use two synthetic unbalanced datasets characterized by intrinsic dimensionality one embedded in a 3D spiraling domain to assess the effectiveness of the different OT-based integration methods (Fig. 3.3). We focused on two dataset integration evaluation criteria: domain overlapping (left panes) and manifold structure preservation (right panes), which are two crucial features for integration methods. By domain overlapping, we mean the ability of a dataset integration method to result in a data distribution being similar for two or more integrated distributions. The OT framework does not guarantee this, as a source point equally transported to two target points will be mapped exactly in between. By manifold structure preservation here, we mean that the resulting dataset integration will reproduce the structure of geodesic distances along the two initial data manifolds. For example, in the spiral example shown in Fig. 3.3, the color denotes the geodesic distance along the spiral from its central point (blue - close to the center and yellow - far from the center). In the ideal integration of manifolds, the data points from the query dataset should find themselves in the position corresponding to their initial geodesic distance.

As expected, the unweighted optimal transport integration is quite successful with respect to the first criterion, with most query points falling on the reference spiral, but fails the second one with a lot of query ”exterior” points being mapped on the reference ”interior” (3.3, **a**). On the other hand, weighted optimal transport integration passes both tests convincingly (3.3, **b**).

Both unweighted and weighted GW-based methods pass the overlapping test, but quite miserably fail the one for manifold preservation, as witnessed by the query data points

integrated into the wrong position on the reference spiral (3.3, **c, d**). In other words, several points with a blueish color are found where the reference data points are yellow.

We could not achieve a satisfying integration with the SCOT tool, using both the balanced (3.3, **e**) and the unbalanced (3.3, **f**) formulations, despite trying to fine-tune several parameters in a large range.

Running times on this example were 1.7s for unweighted optimal transport, 2.5s for weighted optimal transport, 2.5s for unweighted GW, 4s for weighted GW, 0.9s for balanced SCOT, and 0.4s for unbalanced SCOT.

3.7 Integration results on single-cell dataset embedded in cell cycle space

As a first real-life example, we choose to integrate Ewing sarcoma datasets embedded in the cell cycle space, where the first (respectively second) dimension corresponds to the mean expression of genes associated with G1/S (respectively G2/M) phases of the cell cycle. Cell profiles typically revolve around a loop in this space, according to each cell's progression in its cell cycle (Aynaud et al., 2020). Integrating cells in the cell cycle space has several applications. It can be used to correct the loop domain of an average-quality dataset with respect to a high-quality reference dataset, or to infer the cell cycle state of cells in a semi-supervised fashion with label transfer methods if the reference dataset is labeled.

Here, we choose to integrate an Ewing sarcoma patient-derived xenograft (PDX) dataset of 1937 cells with a majority of non-proliferating cells (located in the bottom-left of a cell cycle plot), onto an Ewing sarcoma cell lines dataset (3752 cells) mainly composed of proliferating cells (Fig 3.1b). A good integration should result in mapping non-proliferating cells of PDX onto non-proliferating cells of CHLA9, while preserving the distribution of proliferating cells of PDX over the cell cycle loop.

As in the previous example, we have not been able to achieve satisfying convergence for all GW-based methods in this example (including SCOT). Fig. 6.13a presents integration results using unweighted optimal transport. Once again, we assess integration methods on two criteria: domain overlapping (left panes) and cell cycle phase matching (right panes). As we can see, unweighted OT integration passes the domain overlapping test but fails the phase matching one, with non-proliferating cells matched onto proliferating states. Weighted optimal transport integration (Fig. 6.13b) convincingly passed both tests.

Running times on this example were 5s for unweighted optimal transport and 100s for weighted optimal transport.

3.8 Integration results on balanced single-cell multi-omics datasets

Diagonal dataset integration is a challenging task that consists of computing a joint embedding of samples between two biological modalities, without any prior knowledge of cell types or labels. It does not only require a robust alignment technique, but also sound cross-representation projections to construct a meaningful common space for the integration to take place. In the multi-omics dataset we are interested in, performing a 19-component PCA on gene expression data, and reducing chromatin accessibility data to a 19-dimensional space was sufficient to recover close corresponding clusters in the joint PCA space. We then performed dataset integration in this PCA space embedding. This time, we were able to make all methods converge except for the unbalanced formulation of SCOT.

As both datasets have a clear 3-clusters structure after embedding in a 19-PC space (Fig. 3.1c) with a one-to-one mapping between cells and clusters, and all integration

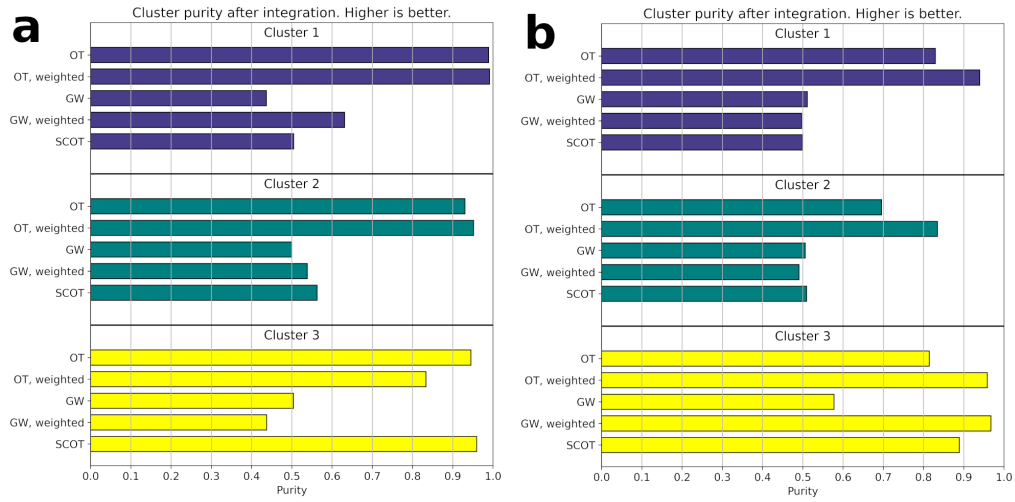

Figure 3.4: **Comparison of integration methods on multi-omics scSNAREseq datasets, assessed as cluster-wise purity after integration. The best methods should display high purity value for all three clusters.** (a) Clusters are balanced between datasets. (b) Clusters are unbalanced between datasets. *From (Fouché and Zinovyev, 2021).*

methods provide a good domain overlapping (Fig. 6.14), we decide to assess integration quality via cluster purity (as defined in Section 3.10.9) after concatenating reference and integrated datasets (Fig. 3.4a). All OT-based methods overperform in this test, while GW methods (including SCOT) struggle to preserve cluster purity properly. In particular, unweighted optimal transport integration achieves over 90% purity in all clusters, which is expected as we deal here with perfectly balanced data, so there is no real reason to apply reweighting. The poor results of GW-based methods may be due to the symmetrical structure of data, as GW does not penalize isometries between datasets.

Running times on this example were 1.7s for unweighted optimal transport, 4.7s for weighted optimal transport, 3.8s for unweighted GW, 12.9s for weighted GW, and 150s for balanced SCOT.

3.9 Integration results on unbalanced single-cell multi-omics datasets

We then decide to challenge all considered integration methods, by unbalancing them on purpose so that corresponding clusters do not match in proportion anymore. This situation can be representative of real-life applications: cell type imbalance is a typical issue when dealing with unmatched datasets. Once again, all methods perform well in terms of domain overlapping (Fig. 6.15) but only OT-based methods give satisfying purity results (Fig. 3.4b). This time, unweighted optimal transport struggles from the unbalance with 69% purity in cluster 2. On the other hand, weighted optimal transport is robust to these changes, with all clusters above 80% purity. All GW-based methods are severely outclassed in this example, with less that 50% purity for clusters 1 and 2 corresponding to random attribution.

Running times on this example were 1.8s for unweighted optimal transport, 3.3s for weighted optimal transport, 5.5s for unweighted GW, 11.8s for weighted GW, and 127s for balanced SCOT.

3.10 Materials and methods

3.10.1 Datasets

We use four pairs of datasets in various dimensions d to test the different integration methods: a pair of synthetic datasets ($d = 3$), a pair of single-cell datasets embedded in cell cycle space ($d = 2$) and one scSNAREseq dataset split in two parts, gene expression ($d > 10^4$) and chromatin accessibility (embedded in a $d = 19$ space). This last dataset is used in two versions: one complete and one unbalanced. Theoretically, dimensionality does not play a role when integrating datasets using OT as it only depends on weights and cost matrix, but in practice, defining a relevant cost between points in high dimensional spaces is challenging. All datasets are available in supplementary materials.

3.10.2 Synthetic datasets

We generate two one-dimensional datasets, non-identically distributed and unbalanced on purpose, that we embed in a 3D space along a spiral. Query spiral contains 500 points, and reference spiral 1000 points. Spirals are then randomly translated in space, and noise is added (Fig. 3.1a).

3.10.3 Ewing sarcoma single-cell datasets

scRNAseq datasets were gathered from (Aynaud et al., 2020) for Ewing sarcoma patient-derived xenografts (PDX), and from (Miller et al., 2020b) for Ewing sarcoma cell lines. All raw datasets were preprocessed using standard methods as follows. Cells with less than 200 genes expressed, as well as genes expressed in less than 3 cells were discarded. Then, cells with raw counts below 15,000 or above 50,000 or expressing more than 15% mitochondrial genes were taken out. Cell counts were then normalized to 10,000 counts per cell, before being log-transformed by the function $\log(1 + x)$. The 10,000 genes with higher variance were kept. All datasets were then smoothed by neighborhood averaging using ten closest neighbors, using 50 components of PCA for the nearest neighbors computation. The G1/S and G2/M scores for each cell were computed using Ewing sarcoma-specific signatures of cell cycle phases (Aynaud et al., 2020).

3.10.4 Multi-omics scSNAREseq dataset

Public chromatin accessibility and gene expression datasets were generated with scSNARE-seq (Chen et al., 2019a) technology, using a mixture of human cell lines (BJ, H1, K562, and GM12878) (Chen et al., 2019a). Every cell was analyzed in both of the assays, and is consequently present in both datasets. In other words, the cells were matched between two data modalities. Chromatin accessibility records were preprocessed using (Chen et al., 2019a) guidelines, including noise reduction followed by dimensionality reduction using the `cisTopic` (González-Blas et al., 2019) `R` package, resulting in a 1047×19 matrix. ScRNA-seq data was normalized to one count per cell for appropriate scaling in the latent space. The count matrix was log-normalized, then the 1000 top variable genes were kept. The matrix was then Z-score-normalized and gloabally divided by 100. Eventually, a 19-component PCA was carried out to match the chromatin accessibility dimension (Fig. 3.1c).

These two datasets have eventually been used to generate unbalanced scSNAREseq data, in order to make the integration more challenging. We selected at random a fraction of samples from each cluster: in the scATACseq data, we kept 80% of cells from cluster 0, 100% from cluster 1 and 60% from cluster 2. In the gene expression dataset, 100% of cells were kept from cluster 0, 60% from cluster 1 and 80% from cluster 2.

3.10.5 Optimal transport

OT problem between discrete distributions can be pictured as follows (Peyré et al., 2019). We are given a set of n "warehouses" and a set of m "factories", described with a cost matrix $\boldsymbol{C} \in (\mathbb{R}^+)^{n \times m}$ containing pairwise cost of transport between warehouses and factories: $\boldsymbol{C}_{ij}$ is the cost required to move one unit of goods from warehouse i to factory j.

Each warehouse contains a number of goods $\boldsymbol{v}_i \in \mathbb{R}^+$, and each factory requires a number of goods $\boldsymbol{w}_j \in \mathbb{R}^+$. We assume $\mathbf{1}_n^T \boldsymbol{v} = \mathbf{1}_m^T \boldsymbol{w} = 1$.

A transport plan between warehouses and factories is uniquely defined by a matrix $\boldsymbol{P} \in (\mathbb{R}^+)^{n \times m}$, where $\boldsymbol{P}_{ij}$ is the number of goods sent from warehouse i to factory j. A transport plan is said to be *valid* if $\boldsymbol{P}\mathbf{1}_m = \boldsymbol{v}$ and $(\boldsymbol{P}^T)\mathbf{1}_n = \boldsymbol{w}$, and we denote the set of transport plans with valid marginals $\mathbf{U}(\boldsymbol{v}, \boldsymbol{w})$. The total cost $\mathcal{C}$ of a transport plan $\boldsymbol{P} \in \mathbf{U}(\boldsymbol{v}, \boldsymbol{w})$ with respect to $\boldsymbol{C}$ is then defined as

$$\mathcal{C}_{\boldsymbol{C}}(\boldsymbol{P}) = \sum_{i,j}^{n,m} \boldsymbol{C}_{ij} \boldsymbol{P}_{ij} = \langle \boldsymbol{C}, \boldsymbol{P} \rangle_F \tag{3.7}$$

where $\langle . , . \rangle_F$ denotes the Froebenius inner product. A transport plan $\boldsymbol{P}^*$ is said to be *optimal* if it minimizes the cost defined in Eq. 3.7 over all valid plans, and its cost is called the *Wasserstein distance* between the query and reference distribution,

$$\mathcal{W}_{\boldsymbol{C}}(\boldsymbol{v}, \boldsymbol{w}) = \min_{\boldsymbol{P} \in \mathbf{U}(\boldsymbol{v}, \boldsymbol{w})} \langle \boldsymbol{C}, \boldsymbol{P} \rangle_F \tag{3.8}$$

If factories represent the *reference* distribution and warehouses the *query* one, we get an intuition behind OT-based dataset integration: a transport plan describes how to displace the whole mass from the query distribution onto the reference one. The OT plan intuitively favors a natural displacement, with well-preserved local topology, as the trajectory of masses will typically not cross. Nonetheless, the method is very prone to overfitting, as it can align any distribution onto any other. Empirical distributions with similar underlying manifolds (for instance, two point clouds where points are defined as a ring) but different local densities also tend to be incorrectly integrated, as shown in section 3.6.

The optimal transport solution can be approximated using an entropic regularizer, and computed efficiently with the help of Sinkhorn's algorithm (Cuturi, 2013; Peyré et al., 2019).

Optimal transport (OT) has already inspired a number of innovative tools in the single-cell field. It was first presented in (Schiebinger et al., 2019) with Waddington-OT, an OT-based computational framework dedicated to the analysis of cell fate. Several OT-based approaches have since been proposed to estimate cell-cell similarity based on transcriptomics profile (Huizing et al., 2021a,b; Bellazzi et al., 2021). OT has also been successfully applied to tackle the dataset integration problem, with novel tools like SCOT (Demetci et al., 2020) and Pamona (Cao et al., 2020b), or spatial inference from omics data in NovoSpaRc (Moriel et al., 2021). We use the latest SCOT version to date for our tests, v0.2.0.

3.10.6 Gromov-Wasserstein problem

In the general case, defining a cost matrix between two datasets may be quite complex, for instance when they are embedded in different data spaces. In these cases, the OT approach is difficult to set up. The Gromov-Wasserstein (GW) problem is a natural extension of OT that can overcome these limitations, at the cost of extra hypotheses and computation time. Let $X \in (\mathcal{X}, d_{\mathcal{X}})$ and $Y \in (\mathcal{Y}, d_{\mathcal{Y}})$ be two datasets embedded in two metric spaces, and $\boldsymbol{C}^{\boldsymbol{X}} \in \mathbb{R}^{n \times n}$ (resp. $\boldsymbol{C}^{\boldsymbol{Y}} \in \mathbb{R}^{m \times m}$) be two matrices containing pairwise distances between points in X (resp. Y).

For a transport plan $\boldsymbol{P}$, we define the transport cost with respect to $\boldsymbol{P}$ as in (Peyré et al., 2019),

$$\varepsilon_{\boldsymbol{C^X},\boldsymbol{C^Y}}(\boldsymbol{P}) = \sum_{i,i',j,j'}^{n,m} |\boldsymbol{C^X_{i,i'}} - \boldsymbol{C^Y_{j,j'}}|^2 \boldsymbol{P}_{i,j}\boldsymbol{P}_{i',j'}.$$

Finding a transport plan $\boldsymbol{P}$ so that $\boldsymbol{P}_{i,j}$ is large if and only if for all i', j', if $\boldsymbol{P}_{i',j'}$ is large then the distance between x_i and $x_{i'}$ is close to the distance between y_j and $y_{j'}$ makes this cost small. Given two histograms $\boldsymbol{v} \in \mathbb{R}^n$ and $\boldsymbol{w} \in \mathbb{R}^m$, the weighted GW distance between X and Y is then defined as

$$\mathcal{GW}_{\boldsymbol{C^X},\boldsymbol{C^Y}}(\boldsymbol{v},\boldsymbol{w}) = \min_{\boldsymbol{P}\in U(\boldsymbol{v},\boldsymbol{w})} \varepsilon_{\boldsymbol{C^X},\boldsymbol{C^Y}}(\boldsymbol{P}). \tag{3.9}$$

This problem is non-convex in this form, but can be rewritten as a quadratic assignment problem (Loiola et al., 2007). It is NP-hard in the general case, but admits an entropic regularization and can be solved quite efficiently,while tackling distributions in different spaces which makes it highly relevant for multi-omics integration. Nonetheless, its lack of sensivity relative to isometries between datasets can sometimes create incorrect results.

3.10.7 Unbalanced optimal transport

When applying optimal transport to real-life applications, the most common issue to deal with is associated with sampling biases, and in particular, class imbalance. For instance, in the single-cell field, we often observe two datasets with similar cell types but dissimilar relative proportions of these cell types between datasets. In the most extreme case, there are even cell types that only appear in a single dataset. This situation typically induces severe overfitting in dataset integration methods, causing cells of a type to be integrated into cells of another type, resulting in unusable altered data.

A typical way to approach this issue is to use an alternative optimal transport formulation, called unbalanced optimal transport, introduced in (Benamou, 2003). The idea is to relax marginal constraints on the transport plan by expressing them via penalties, given a divergence between probability distributions $\mathcal{D}$ (for instance Kullback-Leibler or Jensen-Shannon) and at the cost of extra regularization parameters h_1 and h_2,

$$\mathcal{W}^h_{\boldsymbol{C}}(\boldsymbol{v},\boldsymbol{w}) = \min_{\boldsymbol{P}\in(\mathbb{R}^+)^{n\times m}} \langle \boldsymbol{C},\boldsymbol{P}\rangle_F + h_1\mathcal{D}(\boldsymbol{v},\boldsymbol{P}\mathbf{1}_m) + h_2\mathcal{D}(\boldsymbol{w},\boldsymbol{P}^T\mathbf{1}_n) \tag{3.10}$$

This approach is notably implemented in the SCOT tool (Demetci et al., 2020), and has been shown to help deal with class imbalance in some cases. We propose an alternative to this regularization, aiming to correct cell type heterogeneity before optimal transport.

3.10.8 Gaussian kernel bandwidth selection

We choose a Gaussian kernel for the uniformization method described in Section 3.2. The method contains a single parameter σ called the *bandwidth*. For $x, y \in \mathbb{R}^n$,

$$\mathcal{K}_i(\boldsymbol{x}) = \frac{1}{\sigma_i\sqrt{2\pi}} \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x_i}\|_2^2}{2\sigma_i^2}\right). \tag{3.11}$$

Parametrizing this kernel over a given dataset is a nontrivial task, as an excessively small bandwidth won't allow any influence of a point over its neighborhood, while an excessively large one will blur data points' neighborhood. We follow the nearest neighbor-based bandwidth selection used in UMAP (Becht et al., 2019). The idea is to choose for each point $\boldsymbol{x_i}$ a bandwidth σ_i so that

$$\sum_{j=1}^{k} \exp\left(\frac{-\max(0,\|\boldsymbol{x_i}-\boldsymbol{x_{i_j}}\|_2-\rho_i)}{\sigma_i}\right) = \log_2(k), \tag{3.12}$$

where $\{\boldsymbol{x_{i_1}}, \dots, \boldsymbol{x_{i_k}}\}$ denote the k-nearest neighbors of $\boldsymbol{x_i}$ ordered by distance in increasing order and $\rho_i = \|\boldsymbol{x_i} - \boldsymbol{x_{i_1}}\|_2$. According to (Becht et al., 2019), using this bandwidth selection greatly improves the representation of high dimensional data.

We use the UMAP implementation to compute these values using $k = 15$ neighbors. We did not observe major differences when tuning this parameter between 10 and 50.

3.10.9 Assessing integration quality in scSNAREseq data

ScSNARE-seq data is clearly clustered in three main clusters of similar size (between 300 and 350 cells each), labeled beforehand. To assess integration quality, we run a clustering analysis after integration using the K-means algorithm ($k = 3$), and measure label purity in each of these clusters, defined as the frequency of the majority class in a given cluster, varying between 1/3 (mixed cluster) and 1 (pure cluster).

3.10.10 Assessing the computational time

All computation times have been recorded on a desktop computer running Arch Linux, equipped with a 12/24 cores Ryzen 9 3900x, 32GB of DDR4 RAM. Computation was not GPU-accelerated.

3.11 Discussion about this data integration approach

OT- and GW-based dataset integration methods were originally developed for image processing, but find applications in various domains, notably in single-cell with tools like SCOT or Pamona. We propose an original extension of this class of pipelines, tackling the issue of integrating datasets with similar effective support domains but different cell types or phenotypes distribution. These datasets are frequent in single-cell biology, especially when cells are gathered from different tissues or patients.

We defined an unsupervised procedure that automatically selects sample weights before OT- or GW-based integration, so that dense regions are associated with lower weights and vice-versa. The intuition behind this is to correct for cell type unbalance and prioritize dataset geometry. We demonstrate the effectiveness of this approach on four pairs of datasets of various dimensions. In particular, we demonstrate this approach's robustness in the case of unbalanced datasets, for a reasonable computational cost. We furthermore formulate a quantitative heuristic that parameterizes the procedure in an unsupervised fashion.

We also tried to demonstrate the importance of choosing between the OT and the GW transportation problem. As we showed, it has an important influence on final results and appears crucial for a successful dataset integration. OT is only usable when a relevant metric can be defined between samples of different datasets, but will penalize any kind of transformation between datasets, which can help in the case of symmetrical datasets (see scSNAREseq examples). On the other hand, GW solution is invariant with respect to any isometry but can be applied even when both datasets to integrate do not share the same data space, which makes it very useful in transcriptomics data where finding a relevant common gene space between datasets is often out of reach. It is also associated with an extra computational cost, and a more difficult problem to solve that can lead to convergence issues. In our examples, OT was consistently superior both in terms of speed and consistency compared to GW.

Nonetheless, the weighted OT integration pipeline does not solve all limitations of OT dataset integration. For instance, it still struggles with datasets in which at least one dataset contains specific cell types. In this case, the method will inevitably overcorrect, resulting in a fraction of mismatched samples. Using distance-based cost matrices also does not scale well with high dimensional data; therefore, designing a ground cost suitable

to high dimensional single-cell data is a crucial question – recently addressed in (Huizing et al., 2021a).

Another concern for all integration techniques is the question of outliers. In our case, outliers in the reference dataset can be an issue for the weighting procedure. Indeed, it is easy to show that using a variance-normalized distance matrix to define kernel density is not robust to very distant outliers. For now, outliers detection is expected to be a preprocessing step using distance thresholds, but the need to manually tune extra parameters is never a particularly satisfying solution.

Also, solving a quadratic problem in the probability simplex is challenging in very high dimension, though we did not find this to be an obstacle as our applications do not exceed medium-size datasets ($< 10^4$ cells). With such dataset sizes, standard interior point methods are efficient enough to necessitate reasonable computation time (from a second to a minute). Alternative formulations and strategies may exist though, to at least approximate the result more efficiently to scale to much bigger datasets (up to 10^6 cells). Finally, multi-omics integration (VI) with such methods still highly depends on the latent space construction, which is still an unsolved question, probably highly dependent on the application field and needing *ad hoc* constructions.

Dataset integration using OT and GW is a promising method that can yield high-quality results in a very competitive time in the context of single-cell data analysis. Nonetheless, we see the integration pipeline not to be trivial: several decisions must be taken, such as choosing between OT and GW, using uniform weights or reweighting prior to integration or defining relevant costs. These decisions highly influence the integration result, and are determinant for the success of downstream applications. In particular, if making all weights equal is a tempting (and easy) approach, we demonstrate it causes issues with unbalanced datasets that are usual when dealing with real-life single-cell data. Performing unweighted OT and GW integration can, in these cases, lead to severe overfitting and can be solved using reweighting techniques for a reasonable extra computational cost. We believe OT- and GW-based integration have great potential for the single-cell field, but need extra pre- and post-processing algorithms to support them; we are looking forward to seeing more approaches to do so. Finding a comprehensive formulation for integration quality assessment is also an important question that is still quite open, and will hopefully lead to various improvements in integration techniques.

Chapter 4

Modeling progression of single cell populations through the cell cycle as a sequence of switches

Adapted from (Zinovyev et al., 2022).

Progression through the cell cycle represents a complex dynamical process regulated at multiple levels, such as transcriptome and proteome. The major components of it have been characterized (Hunt, 1991; Hunt et al., 2011), and a complex molecular machinery has been revealed (Tyson, 1991). Nevertheless, many aspects of cell cycle functioning remain to be elucidated (Giotti et al., 2019).

Progression through the cell cycle can be seen as a trajectory in a multidimensional space of all possible cellular states, similar to other processes such as differentiation or aging. However, this trajectory is characterized by special properties because it represents a periodic process. From an oversimplified perspective, at the end of this trajectory, a cell splits into two daughter cells twice as small, where each daughter cell has a state identical to the initial state of its parent. This requirement imposes certain constraints on the geometry and underlying mechanisms of the cell cycle trajectory (CCT), which could be reproduced with a mathematical model.

The cell cycle process has been a subject of mathematical modeling for many decades (Sible and Tyson, 2007; Chen et al., 2004a; Ingolia and Murray, 2004). Most existing models focused on reproducing the regulatory logic at the level of protein expression, protein-protein interactions, and post-translational modifications. Multiple modeling formalisms have been used such as chemical kinetics (Tyson, 1991; Chen et al., 2004b), logical modeling (Fauré et al., 2006; Deritei et al., 2019), Petri nets (Kotani, 2002), or approaches based on tropical algebra (Noel et al., 2012; Radulescu et al., 2012). A hybrid approach, combining discrete, governed by Boolean dynamics, and continuous, governed by chemical kinetics, variables was suggested to model cell cycle (Singhania et al., 2011; Noël et al., 2013). The mathematical description of the cell cycle transcriptional dynamics has not yet been thoroughly addressed.

High-throughput omics measurements gave rise to many molecular studies to characterize each cell cycle phase regarding their associated molecular changes, i.e., sets of specifically expressed genes (Giotti et al., 2019; Dominguez et al., 2016). The appearance of single-cell technologies reinforced the interest towards the description of the molecular organization of the cell cycle for several reasons. First, it explicitly allows the visualization of the cell cycle trajectory without synchronizing individual cells, which can be problematic, especially *in vivo*. Then, recent single-cell transcriptomic and proteomic studies provide molecular description of progression through the cell cycle in a continuous fashion. Such representation attempts to delineate the cell cycle phase borders and also characterizes each cell for its precise progression position within each phase (Hsiao et al.,

2020; Mahdessian et al., 2021a; Liu et al., 2017; Leng et al., 2015).

A thorough understanding of cell cycle functioning is of utmost importance for cancer research, where deviation from normal cell cycle progression is expected. Several questions can be raised; among these, what is the regular pattern of the events comprising a cell cycle, and to what extent does it vary in normal physiology? What deviations from a normal cell cycle are characteristic for a tumor cell? What processes trigger these changes, and are they specific to a cancer type?

4.1 Background

Some mathematical models of the cell cycle try to tackle these questions. For example, agent-based or cellular automaton cell cycle models focus on the optimization of cancer drug delivery (Altinok et al., 2007), competition of fast and slow cell cycles within a tumor under treatment (Tzamali et al., 2020), or cell confluence and elongation of the G1 phase (Bernard et al., 2019). However, most of the existing models remain limited to describing the behavior of the cell cycle during tumorigenesis at full complexity, because of the current discrepancy between the nature of the available molecular data and the level of the details of these models. Thus, the most comprehensive data source currently available is at the level of transcriptomic changes in single-cells, while the existing modeling efforts focus on protein players. The data reveals the role of hundreds of genes and proteins in cell cycle dynamics, while the models include a tiny fraction of this number. Therefore, we believe that the development of mathematical models matching the scale and the nature of the abundant available data is still highly needed. In particular, even a simple mechanical model of cell cycle transcriptome dynamics, capturing its main features, is lacking in the field. Using dynamical variables representing relatively large lumps of genes (e.g., all genes involved in DNA replication) might be a useful coarse-grained approach to model cellular transcriptomes, which is one motivation of this study.

Single-cell studies provide a snapshot of actively proliferating cells along cell cycle trajectories and represent a unique opportunity to formulate the most general principles of cell cycle functioning. A recent study has introduced the principle of minimizing transcriptomic acceleration (Schwabe et al., 2020), which suggests that the transcriptomic cell cycle trajectory represents a spiral, or, after neglecting the relatively slow drift unrelated to cell cycle progression, a shape close to "a flat circle". This type of trajectory was indeed phenomenologically observed in the HeLa cell line profiled with scRNASeq technology, after deconvoluting the transcriptomic dynamics connected to the cell cycle from other sources of transcriptional heterogeneity. In particular, the absence of cell cycle-related transcriptional epochs was deduced from this model.

In the current study, we suggest an extended and different point of view on the properties of transcriptomic cell cycle trajectory, which, in our opinion, in some cases better matches its observed properties in various cellular systems when sufficiently good quality data can be collected. We propose a formal model of CCT as a sequence of epochs of growth during each of which the trajectory is approximately linear in the space of logarithmic coordinates. Therefore, CCT can be modeled as a piecewise linear trajectory in the space of logarithms of some extensive cell properties, followed by a shift at the vector with coordinates $-\log 2$ representing the cell division event. This model explicitly assumes the existence of well-defined transcriptional epochs in CCT.

Movement along a linear trajectory in the space of logarithms of the values of some cellular properties means that along the trajectory any two such properties x_i, x_j are connected through a power law dependence $x_i = \alpha x_j^{\beta}$, with α, β some constants. Such dependencies are known as allometric in many fields of biology (Pretzsch, 2020; Zhou et al., 2021; Packard, 2017; Holford and Anderson, 2017; White et al., 2019). Some approaches in mathematical chemistry and theoretical biology, dealing with systems in stable non-equilibrium, exploit the linear relations between chemical potentials which can be ex-

pressed as logarithms of species concentrations (Bauer, 1935; Gorban, 2018).

Particular cases of allometric dependencies are when all the quantities grow linearly with physical time, or when all the quantities follow exponential growth or decay $x_i = b_i \exp(a_i t)$. The model of movement along piecewise linear trajectories with an event of cell division represents the simplest scenario, easy to simulate and analyse theoretically. Nevertheless, the most important conclusions derived from this analysis will stay valid for the trajectories that do not deviate too much from linearity.

Using the model of piecewise linear growth with division, we formulate a fundamental statement about correspondence between the number of linear segments in the cell cycle trajectory m, which corresponds to a number of the most important states of the cell cycle-related transcriptional machinery, and its effective embedding dimension n. The first part of the statement, $m \geq n$, can be described as a strict theorem with formal proof, whereas the second part, $m \leq n$, can be formulated as a feasible hypothesis, that can be validated using available data. The correspondence $m = n$ suggests that the embedding dimensionality of the transcriptomic cell cycle trajectory is larger than 2, since the number of segments we can observe can be as high as 4 or 5. This allows us to state that the shape of the cell cycle trajectory is essentially not flat.

The type of models discussed here was partly introduced by (Shkolnik, 1989). Here, we significantly extend the previous effort and adapt it to the description of the cell cycle trajectory in single-cell datasets.

In order to connect the geometric properties of cell cycle trajectory to interpretable mechanistic parameters, we extended the model of piecewise linear growth in logarithmic coordinates to a simple kinetic model with rates depending on time as piecewise constant functions. In this case, some of the trajectory segments become nonlinear but remain smooth and do not deviate from linearity too far. Therefore, the suggested model is conceptually similar to previously suggested hybrid discrete-continuous models, but conceptualizes them, addresses the transcriptional dynamics and can be fit to multiple available scRNASeq datasets (Singhania et al., 2011; Noël et al., 2013).

The suggested cell cycle modeling framework and the representation of the cell cycle progression as a system of switches allows us to 1) determine which genes play the most important role in each transcriptional epoch, in a concrete system under study, 2) compare the genes related to the same transcriptional epoch between two biological systems or conditions, 3) predict the ratios between physical time durations of the transcriptional epochs, 4) predict the effect of shortening of certain transcriptional epochs on the shape of the cell cycle trajectory and transcriptional dynamics of the related groups of genes, and 5) predict the doubling time of proliferating cell populations from the length of the cell cycle trajectory observed in single-cell RNA-Seq data. The suggested framework can be exploited to study the cell cycle in various systems, from cell lines to tumors.

4.2 Methods and materials

4.2.1 Single-cell data used in this study

We made a screening of available single-cell sequencing of cancer cell lines in order to identify datasets with a sufficient number of good quality single-cell transcriptomic profiles, and in which the principal source of transcriptomic heterogeneity would be progression through the cell cycle. We identified publicly available RNA-Seq data on CHLA9 Ewing sarcoma cell line, produced with 10x Genomics sequencing technology (Miller et al., 2020a), which contained more than 4000 cells with a total number of unique molecular identifiers (UMIs) varying from 10000 to 50000 per cell, after applying the standard quality criteria and filtering cells containing a significant fraction (>20%) of reads in mitochondrial genes. For this dataset, we reanalyzed the raw sequencing data using Kallisto mapper (Bray et al., 2016), resulting in a loom file that could be used for obtaining the gene expression levels and for quantifying RNA velocity vectors (La Manno et al., 2018).

In addition, we used a recently published collection of 200 scRNASeq profiles of cancer cell lines from the Cancer Cell Line Encyclopedia (CCLE) collection (Kinker et al., 2020). We also analyzed several scRNASeq datasets by downloading them directly from Gene Expression Omnibus (GEO).

The estimation of cell line doubling times, when available, were obtained from the Cellosaurus database (Bairoch, 2018)

4.2.2 Definition of cell cycle genes

We systematically tested several existing definitions of cell cycle gene sets. We verified that our results remain qualitatively invariant even if the choice of cell cycle gene set can vary. In our experiments, we used the following cell cycle gene set definitions:

- Standard "Regev's set": markers of S- and G2/M cell cycle phases used in *scanpy* tutorials (Tirosh et al., 2016)
- Set of cell cycle genes annotated in Reactome pathway database (Jassal et al., 2020)
- Set of top-contributing genes, extracted from the application of independent component analysis (ICA) to the dataset under study, from those components whose top-contributing genes were strongly associated with the cell cycle. In particular, similar to our previous work (Aynaud et al., 2020), two independent components were significantly enriched with the markers of S- and G2/M cell cycle phases in all single-cell cell line datasets that we analyzed.

Cell cycle phase scores were computed as an average expression of marker genes for the corresponding cell cycle phase in log scale, which roughly corresponds to the geometric mean of the raw count measures.

4.2.3 Pooling reads from neighboring cells for compensating the technical drop-out

We found out that the cell cycle trajectories appear less noisy and more tractable by trajectory inference methods when the standard pooling approach was applied to the raw count data, using an initial estimate of cell-to-cell proximity. More precisely, we used the initial standard data normalization and dimensionality reduction in order to compute the distances between cells and construct the initial kNN graph, which was used to pool row reads from a cell and all its k-nearest neighbors. In our experiments, we used $k = 10$ and $n = 30$ components to reduce the data dimensionality during normalization. Pooled read counts were used for final normalization, but the initial total read counts per cell measure were kept for visualization and further analysis.

4.2.4 Cell cycle trajectory-based single-cell data normalization

The total number of reads in a cell is a strongly variable signal in proliferating cell populations, strongly correlated with cell cycle progression: we observed that cells with the lowest total number of reads are typically just after the mitosis mark, while cells in the state preceding mitosis are often associated to a high total number of counts. By itself, it is an extensive value such that it should be divided (approximately) by half in the process of cell division. In our modeling approach, we needed a description of the cell state in terms of extensive values of gene expression levels measured. They would also be divided approximately by two on average after the moment of cell division. Therefore, the widely used global library size normalization did not suit our purposes, since after global library size normalization, cell division does not lead to halving the total number of reads.

At the same time we observed that without any library size normalization, the cells presumably located at similar stages of cell cycle progression could be characterized by

a wide range of total number of reads, probably caused by technical variability factors. Therefore, library size normalization was required but not at the global cell population level. We hypothesized that the total number of reads should increase in the course of cell cycle progression on average such that the cells characterized by similar value of pseudotime along the cell cycle trajectory could be normalized to the same local library size. As usual, this poses a chicken-or-egg problem because for reconstructing the cell cycle trajectory one needs normalized data, and for normalization of the library size one needs a reconstructed trajectory. This problem is similar to those approaches which use normalization locally conditioned on clusters in single-cell datasets (Azizi et al., 2018).

We used a simplified two-stage approach for library size normalization which preserved both the geometric structure of CCT and the trend of increasing the total number of reads along CCT.

1. The row count data have been normalized to the global median number of counts and ln(x+1)-transformed, using standard functions of scanpy. 10,000 most variable genes have been selected; the dimensionality was reduced to 30 by PCA. In the reduced space, a kNN graph has been computed using the standard Euclidean distance for k=10. This graph was used for pooling reads from neighbor cells, as described above.

2. For such initially normalized dataset, we computed closed cell cycle trajectory in the subspace of cell cycle genes, by fitting a principal closed curve, using the Python implementation of ElPiGraph (Albergante et al., 2020). The data points were partitioned according to the proximity to the nodes of the EPC.

3. In each partition, we analyzed the distribution of the total number of reads across cells. We corrected cell-to-node assignment by splitting an anomalously wide partition between two neighboring partitions. The anomalously wide partition corresponded to the moment of cell division since it contained both cells at the very end of the cell cycle progression with the largest number of reads and cells just after cell division event containing the minimal number of reads. Splitting this distribution allowed us to distinguish cells before and after the cell division into distinct partitions.

4. The median total number of counts in each corrected partition was computed. The median values of the total number of reads in the cells of each partition have been smoothed by univariate spline or a piecewise-linear function of pseudotime, taking into account the cyclic boundaries of the trajectory.

5. Each cell's library size was normalized to the smoothed local median value of the total number of reads.

6. The newly normalized pseudocount data matrix passed through the same pre-processing as described in 1), namely a) Pooling reads from neighbor cells using the kNN graph obtained with trajectory-based normalized data, b) ln(x+1) transformation, selecting most variable 10000 genes.

The cell cycle trajectory-based normalization procedure is illustrated in the Jupyter notebook at `https://github.com/auranic/CellCycleTrajectory_SegmentModel`, which can be easily reused for other cell lines.

4.2.5 Computing the cell cycle trajectory and quantifying pseudotime

We used the ElPiGraph Python package to fit Elastic Principal Curves (EPC) or Closed EPC (principal circles) to single-cell data distributions (Albergante et al., 2020). ElPiGraph was applied in the data space defined by the set of 10,000 most variable genes or by

the cell cycle-related genes, after dimensionality reduction by PCA (first 30 principal components were retained). In order to compute open EPC with q nodes, first a closed curve was fit with $q/2$ nodes, then a node with the least number of data points projected onto it was removed from the principal graph, and this configuration was used as an initialization to compute the elastic principal graph without branching and having q nodes.

The pseudotime s_i for a data point x_i was computed as a continuous geodesic distance measured from the root node to the projection of x_i onto the principal curve, quantified in the units of the number of edges. Therefore, the value of the pseudotime was in the range $[0, q-1]$, where q is the number of nodes. The root of the principal curve was chosen as one of its ends, such that the value of the initial total number of reads would increase as a function of pseudotime.

4.2.6 Curvature analysis of the cell cycle trajectory

In order to compute the Riemannian curvature of the principal curve defined by the position of its nodes in the multi-dimensional space $y_i \in R^n, i = 1...q$, the node coordinates were first represented as n functions of the natural parameter (pseudotime) s, $y_i^k = y_i^k(s_i), i = 1...q, k = 1...n$. The value s_i for each node was taken as the number of edges of the EPC connecting the node i to the root node. Each set of numbers $y_i^k(s_i), i = 1...q$ was interpolated by a cubic univariate spline $y^k(s)$. In each node i of the curve, the curvature was evaluated as $R_i = \sum_{k=1}^{n} \left(\frac{d^2 y^k(s)}{ds^2} |_{s=s_i} \right)^2$.

4.2.7 Estimating the effective dimensionality of a set of vectors

In order to estimate the effective dimensionality of CCT, we used the *scikit-dimension* Python package (Bac et al., 2021). We used linear estimators of global intrinsic dimensionality, based on application of PCA and various approaches to select the significant number of eigenvalues from the scree plot.

In order to compute the effective rank of a rectangular matrix, we looked at the distribution of its singular values, and selected such a number of them that the ratio between the largest and the smallest number would not exceed 10, such that the reduced matrix is well-conditioned.

4.3 Example of a cell cycle trajectory extracted from single-cell data

The current study is motivated by the observation that after appropriate pre-processing of single-cell RNA-Seq data (see Methods), one can observe the cell cycle trajectory (Figure 4.1) which can be approximated by a piecewise linear curve, with a gap between the beginning and the end of the trajectory corresponding to the cell division moment.

Here we use the example of Ewing sarcoma cell line CHLA9 sequenced at single-cell level using the Chromium 10x technology (Miller et al., 2020a). The distinguishing feature of this dataset was that it contained a significant number of proliferating cells with single-cell transcriptomes of good quality (more than 4000 cells with the total number of Unique Molecular Identifiers (UMIs) between 10000 and 50000). Also, the proliferation signal in this dataset seems to explain the largest fraction of transcriptomic heterogeneity, since in the plane of the first two principal components one can clearly observe the cyclic trajectory. In other cell line single-cell datasets, the proliferative signal can be masked by other sources of transcriptomic heterogeneity, requiring special procedures of data treatment to reveal it (Aynaud et al., 2020) (Liang et al., 2020; Schwabe et al., 2020).

The scRNA-Seq data have been normalized in order to preserve the pattern of dynamics of the total number of counts (UMIs) along the CCT, see Methods section. The normalized gene expression levels are represented at the logarithmic scale, following the standard

practice. The multi-dimensional distribution of single-cell transcriptomic profiles projected into the space of the first 30 principal components has been approximated by a principal curve (see Methods). The curvature of the principal curve has been estimated using the standard formulas of differential geometry, which revealed the existence of curvature peaks, and reflecting the rapid turning points of the trajectory. We hypothesized that these turning points correspond to the large-scale changes in the transcriptional programs of the cell cycle process. The pattern of momentary velocities of the transcriptomic changes, estimated with RNA velocity, was compatible with this hypothesis (Figure 4.1,A).

The pseudo-temporal dynamics of the known cell cycle-related genes confirmed that the trajectory curvature peaks delineate biologically meaningful transcriptional epochs. The epoch 0-A-B can be understood as an early G1 phase of the cell cycle, B-C as significantly overlapping with late G1- and S-phases, and C-D as overlapping with S- and G2- phases. The epoch D-E can presumably reflect the relatively short M phase (mitosis). Analysis of pseudotemporal gene expression dynamics inferred for this cell cycle trajectory shows that known cell cycle genes such as different cyclin types or E2F transcription factors have behaviour compatible with our interpretation (Figure 4.1,C). We denote the identified transcriptional epochs as T1, T1s, T2s and Tm.

The switches between transcriptional epochs should not be confused with the action of cell cycle checkpoints that delineate cell cycle phases. The connection between the known molecular checkpoint mechanisms involving mainly protein-protein interactions and post-translational protein modifications and the transcriptional epochs might not be trivial or direct: partly, due to the delay between the gene and protein expression, and partly due to different parameters and constraints on the transcriptional and protein-protein interaction dynamics.

We can clearly observe the existence of the restriction point at the level of single-cell transcriptome. In our notations, it belongs to the A-B segment of the cell cycle trajectory shown in Figure 4.1,A,right. This transcriptional epoch separates post-mitotic (denoted as T1) and pre-replication parts of G1 phase, which corresponds to the classical definition of the R-point (e.g., from (Zetterberg et al., 1995)). Interestingly, in Figure 4.1,A,right, one can observe that RNA velocity vectors reflect cells exiting from cell cycle and re-entering the cell cycle in the epoch between A and B turning points. Just after this transcriptional epoch, the expression of E2F transcription factors and Cyclin E start to increase as expected (Figure 4.1,C).

We can also observe how, during each particular epoch, the components of a specific checkpoint mechanism are transcriptionally produced 'just in time'. For example, components of the G1 DNA damage checkpoint (e.g., CDC25A, CDKN1A) are produced during the T1s epoch of the cell cycle trajectory where the S phase starts, the components of G2 DNA damage checkpoints (e.g., CDC25B, CDC25C, CHEK2) are produced in the late part of the C-D epoch (T2s), and spindle checkpoint components (e.g., CDC20) are transcriptionally abundant during the mitosis-related epoch D-E (Tm) and after the cell division in T1 (Figure 4.1,C). In this sense, the transcriptional dynamics prepare the correct ground for a proper succession of post-transcriptional events but the exact borders of the transcriptional epochs do not have to match the precise checkpoint timing.

Remarkably, within each of the identified transcriptional cell cycle epochs, the global dynamics of the transcriptome remain close to linear in the logarithmic scale. This allows us to suggest a simple model which can, for example, represent the collective dynamics of the genes related to the S-phase and G2/M phases (see below).

4.4 Model of cell cycle as a trajectory of allometric growth with switches and divisions

Based on the observations of the properties of the cell cycle trajectory in several scRNASeq datasets, we hypothesized that it can be recapitulated by a formal model of linear growth

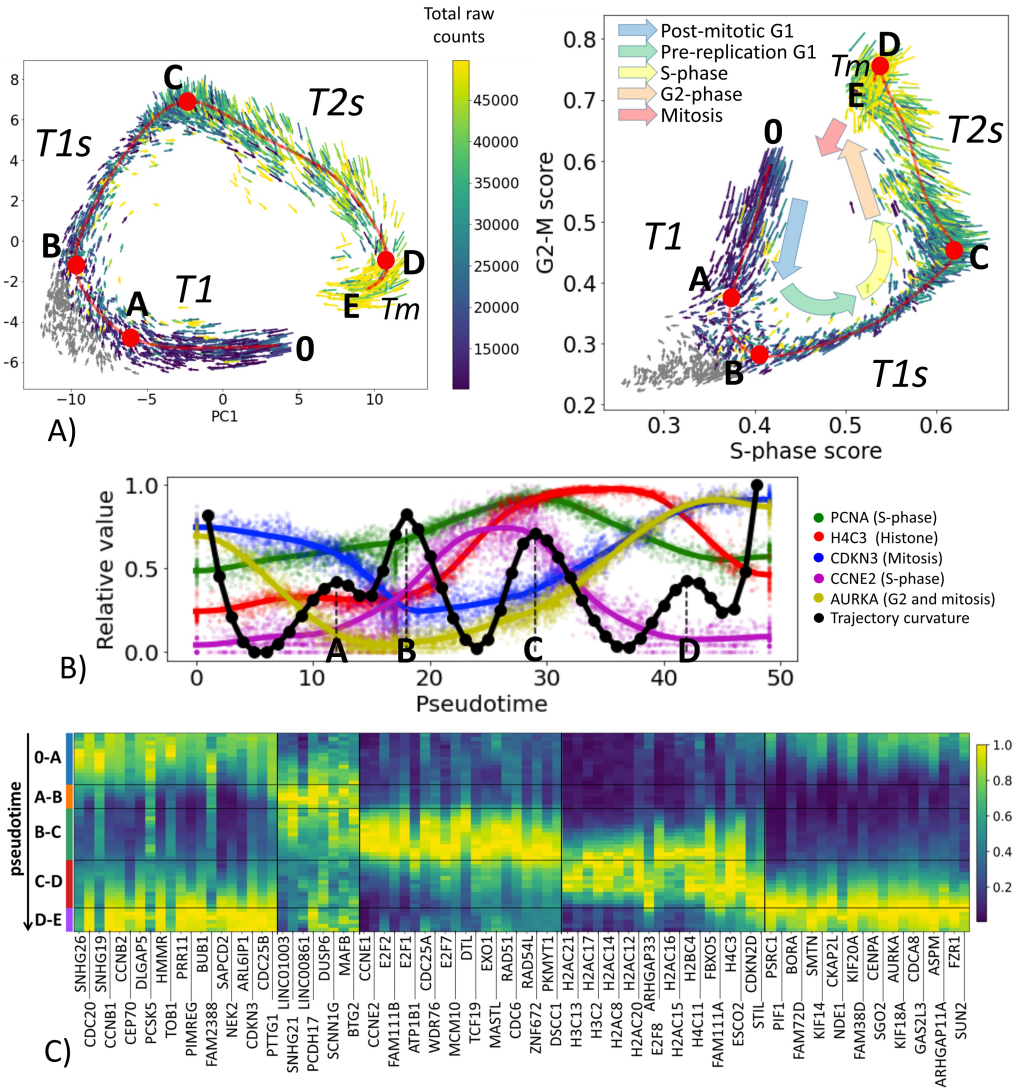

Figure 4.1: **Cell cycle trajectory (CCT) of CHLA9 Ewing sarcoma cell line in the single-cell transcriptomic space.** (a) Each cell is represented by an arrow reflecting the momentary direction and the speed of transcriptomic changes, estimated with RNA velocity. Two projections are shown, in the first two principal components and in the plane of S-phase and G2-M scores. The color of the arrows signifies either the total amount of RNA counts in the single-cell profile (blue to yellow scale) or the cells in non-proliferative state (shown in grey). Red line shows an approximation of the cell cycle trajectory with a principal curve computed with ElPiGraph, directly in the 30-dimensional space of the first principal components of the dataset. Several particular positions along the trajectory (A,B,C,D) mark either the peaks of the Riemannian curvature of the principal curve (also shown in B) panel) or the beginning (0) and the end (E) of the trajectory. (b) Pseudotemporal transcriptomic dynamics of several cell cycle-related genes along CCT, shown relatively to the maximum value units. The pseudotime range is from 0 to 49, corresponding to the number of nodes in the approximation of the principal curve (50 nodes). In black, an estimation of the Riemannian curvature of the principal curve is shown, with peaks indicated by letters (A,B,C,D). (c) Pseudotemporal dynamics of genes whose expression is relatively high in one of the transcriptional epochs (trajectory segment) compared to other epochs. For each epoch the genes are ranked accordingly to the fold change of the mean expression of the gene in the epoch and outside the epoch. Only the genes having relatively large total variance across all cells are shown, and only top 20 genes maximum are shown per epoch for readability. *From (Zinovyev et al., 2022).*

in logarithmic coordinates with switches and a cell division event. The suggested model is hybrid in nature, similar to some previously published models (Singhania et al., 2011; Noël et al., 2013). Namely, we distinguish the extrinsic observable cell state, characterized by continuous variables, and the intrinsic hidden cell state, characterized by discrete variables. The intrinsic state of a cell determines the parameters of the extrinsic dynamic process as in (Singhania et al., 2011).

Let the extrinsic state of a proliferating cell be determined by n substances quantified by their amounts, not their concentrations. Instead of their natural units (such as RNA counts), let us use the logarithms of these amounts. The cell is represented as an n-dimensional vector, and all possible combinations of these vector components define the cell configuration space. For our model, it is important that the considered n quantities are extensive measures, not intensive ones. Extensiveness here means that the total amount of a substance is a sum of the amounts found in different parts of a cell. A division (for two almost equal) daughter cells is formalized as a shift by the vector with all components equal $-\log 2$ in this space. A relevant example of extensive quantity is the total amount of RNA molecules present in a cell, or the amount of any specific subset of RNA molecules, i.e., representing mRNAs of the genes involved in a particular process (such as mitosis or S-phase).

We assume that there exists a finite discrete set of intrinsic cell states. In each of these states, the cell follows a linear trajectory in the extrinsic and continuous cell state space. This trajectory extends until the cell meets a condition, where a switch into another intrinsic state of the cell happens, which changes the direction of the trajectory. For simplicity, we assume that the conditions of a switch can be described by a linear function. The cell movement continues until a particular condition is met in which the cell division event is triggered leading to the aforementioned translation of the vector representing the extrinsic cell state.

Let us introduce some mathematical notations and consider a deterministic automaton A whose complete state is represented by a pair (x, s), where $x \in R^n$ is a vector in n-dimensional continuous space (extrinsic state), and $s \in S$ is an integer number from a finite set $S = \{S_1, .., S_m\}$ (intrinsic state). In the rest of the study, we will call x a position of A and s an intrinsic state of A. We will denote the automaton A in position x and in the intrinsic state s as $A(x|s)$.

Each intrinsic state S_k is parameterized by a vector $a_k \in R^n, k = 1..m$ and by a linear manifold D_k of dimensionality $n-1$ embedded in R^n (hyperplane), which we will call "the cell division hyperplane". D_k can be undefined, in this case, we denote $D_k = null$.

Let us also introduce a set of p functions $G = \{g_1, ..., g_p\}, g_i : S \to S$, which we will call switches. Each switch g_i is a map which converts an intrinsic state $s_j \in S$ into another intrinsic state $s_r \in S$. Each switch g_i is parametrized by a hyperplane L_i existing in R^n and inducing the switch function g_i each time the trajectory of the automaton intersects L_i (see Figure 4.2,A).

Finally, we introduce the cell division event ϕ which is a map between two states of A, such that $\phi((x, s)) \to (x + d, s_d)$, where $d \in R^{n-}$ is a vector with negative components, and $s_d \in S$ is one of the possible intrinsic states of A.

We will characterize any hyperplane here by a linear functional $f(x|b, c) = b + < c, x >$, $b \in R, c \in R^n$, where $<, >$ denotes the standard scalar product between two vectors. Using such a functional, for any pair of vectors $x_i, x_j \in R^n$ we can determine if the linear segment connecting x_i and x_j intersects the hyperplane or not. If the segment intersects the hyperplane then $f(x_i)f(x_j) < 0$, and if it does not intersect then $f(x_i)f(x_j) > 0$. $f(x_i)f(x_j) = 0$ is satisfied only in a non-general position when either x_i or x_j is located exactly on the hyperplane.

The update rules for the automaton A are described as follows. The automaton is in some initial position x_0 and the intrinsic state s_0. It starts to move along the linear trajectory described by the equation $x = x_0 + a_0 t$, where a_0 is the vector of movement

associated with the state s_0. This movement continues unless one of the two events happens. In the first case, A reaches the corresponding cell division plane D_0 (in case D_0 is not null). Then, the cell division event is triggered, $A(x|s) \to A(x+d|s_d)$. In the second case, x reaches a switch hyperplane L_j and then a switch of the intrinsic state of A happens without changing its position, $A(x|s_0) \to A(x|g_j(s_0))$. The movement continues along a new trajectory, corresponding to the new cell state, following the same rules: either the trajectory hits the cell division hyperplane or any of the switch planes.

To summarize, the automaton A is characterized by its position and the intrinsic state, see Figure 4.2,A. The asymptotic (in the infinite time limit) temporal dynamics of A is parameterized by a set of cell division planes $D = D_i, i = 1...k$, a set of switch functions $G = \{g_i\}, i = 1...p$, the corresponding switch hyperplanes $L = \{L_i\}, i = 1...p$, and the parameters of the cell division event (namely, the translation vector d and the state after cell division s_d).

It is convenient to encode the state s as a binary sequence of length r representing the on-off states of r triggers. In this case, a switch can be thought of as changing only one particular trigger from on to off or vice versa. In many situations, this makes the description of switch functions $g : S \to S$ quite natural as explained below. Also, the state of the trigger might not be strictly binary but characterized by several discrete positions, for example $\{0, 1, 2\}$, just as it is the case in modeling multi-level discrete dynamics, where each discrete variable can take a value from a pre-defined finite set of levels.

The exact asymptotic trajectory of the automaton A can, in principle, depend on the initial position x_0 and the initial intrinsic state s_0 of A.

4.5 Simple example of dynamics with switches and cell division events

In the above-described switch-like dynamics, one can find examples of relatively complex behaviors even for simple model settings (Figure 4.2,B-E). As an illustration, we modeled a simple dividing automaton characterized by a position vector x with only two coordinates x_1, x_2. The automaton intrinsic state s encoded by only one binary trigger, so the automaton can be in two states $s = 0$ and $s = 1$, characterized by two vectors of movement a_0 and a_1, respectively. In order to be able to modify the trigger in both directions, we have to introduce two switch hyperplanes $L^{(+)}$ and $L^{(-)}$ with corresponding switch functions $g^{(+)} = 1$ (switch trigger on) and $g^{(-)} = 0$ (switch trigger off). Note that in this case the switch functions are constant, i.e., they map any state (which can be either 0 or 1) to a particular state. Let us also assume that the division event changes the automaton position but does not change its intrinsic state.

In this simple toy example, by slightly varying parameters of the switching hyperplanes and the movement vectors, one can observe several interesting scenarios. Firstly, we observe that the automaton can approach and stay on a limit cycle trajectory, or it can diverge, meaning that one of the coordinates of the vector x goes to infinity or zero (Figure 4.2,B-C). Convergence or divergence to a limit cycle depends on the initial intrinsic state and the initial position of the automaton on the birth hyperplane.

In a more complex scenario, the switching dynamics trajectory can be characterized by two limit cycles that can be achieved from different initial intrinsic states and positions (Figure 4.2,D).

By varying the positions of the switching hyperplanes in this toy example, one can observe the effect of non-trivial sensitivity to the initial conditions (Figure 4.2,E). In this case, the birth hyperplane can be split into a sequence of alternating intervals of equal length such that starting from one interval, the dynamics finally converges to the limit cycle, and starting from another interval, the dynamics diverges to infinity.

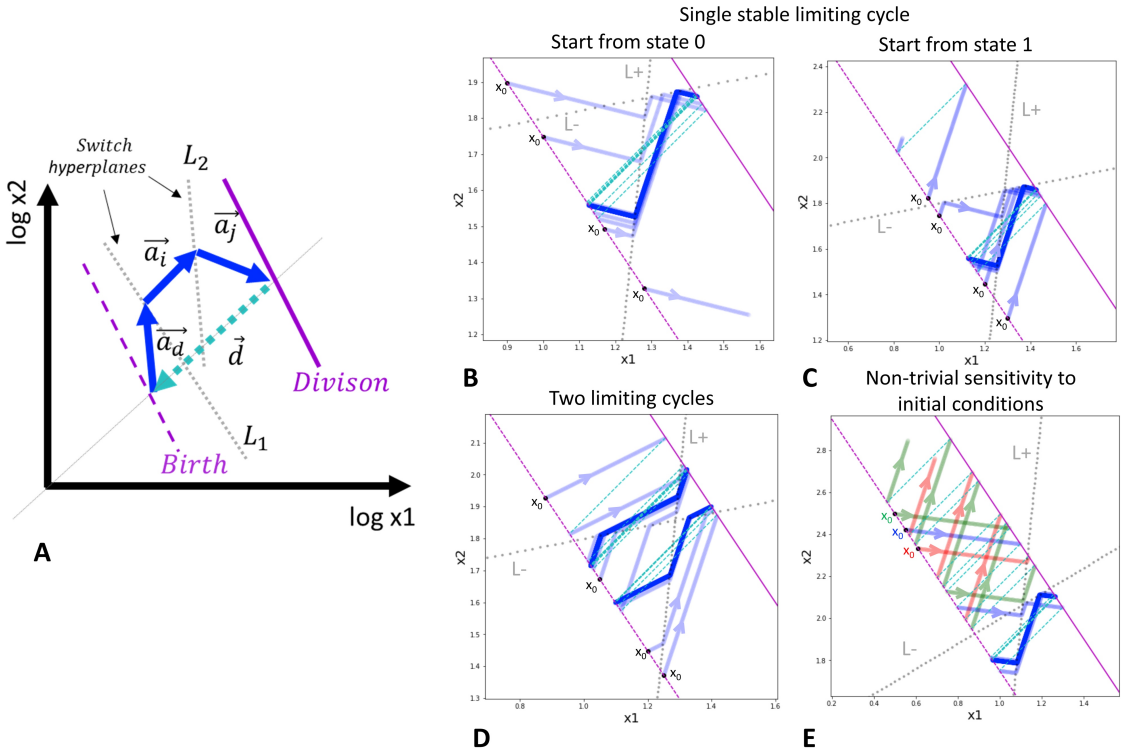

Figure 4.2: **General schema of switch-like dynamics and application to a toy model with a single trigger.** (a) Schematic two-dimensional example of a limiting trajectory with division. The division hyperplane D is shown in purple, solid line. The birth hyperplane B is obtained from D by translation at vector d, shown in cyan (the most natural is to assume all the components of d to be $-\log 2$). Two switch hyperplanes L_1 and L_2 are shown by dotted grey lines. The limiting cycling trajectory is represented by blue arrows. (b) and (c) Example of single limiting cycle in the switching dynamics. Depending on the initial state of the automaton and the initial position, the trajectory enters into the limit cycle or degenerates (goes to infinity). For the same parameters, four initial conditions are shown. The trajectory is plotted with semi-transparent blue color such that the intense blue line designates the trajectory cycling multiple times on top of itself. (d) Example of existence of two limit cycles. Depending on the initial state and position, the automaton ends up in one of the two possible limit cycles. (e) Example of non-trivial dependence of the switching dynamics on the initial position of the automaton. The trajectories drawn by different colors from three closely located initial positions are shown, with two leading to degenerated dynamics and one located in between the first two, leading to the limit cycle. In B)-E) panels, the initial position of the automaton is always shown at the birth hyperplane B (shown by dashed purple line), therefore, it is characterized by a single number. *From (Zinovyev et al., 2022).*

4.6 Two-dimensional model of cell cycle progression, fitted to the single-cell transcriptomic data

Let us denote the aggregate signal related to the activation of genes associated with the S-phase of the cell cycle program as S, and the signal related to the activity of genes in G2 and M phases as M. Therefore, we will characterize the position of the automaton by a vector (x_S, x_M), just as it is presented in Figure 4.1,A, right panel. Let us denote the position of the turning points in the trajectory as $(x_S^{(i)}, x_M^{(i)})$, where $i \in \{0, A, B, C, D, E\}$.

We will encode the state of the system by the levels of two triggers, one associated with the S signal and another associated with the M signal. The three levels are denoted as a set $\{2 = synthesis, 1 = decay, 0 = degradation\}$. Intuitively, these levels correspond to the state of active transcription of the corresponding set of transcripts ('synthesis'), absence of active transcription in which the transcripts are passively degraded according to some base rate ('decay'), and the process of active degradation when the transcripts are degraded more rapidly than the base rate ('degradation'). The state of the system is thus encoded by a pair of 3-level variables $i, j \in \{0, 1, 2\}$. The 2D vectors of linear movement a_{ij} are encoded by six rates $k_j^v, i \in \{0, 1, 2\}, v \in \{s, m\}$, such that $a_{ij} = (k_i^s, k_j^m)$. Following the intuition behind the introduced trigger levels, we assume constraints $k_2^v > 0, k_1^v < 0, k_0^v < 0, k_0^v < k_1^v < k_2^v$.

Let us introduce 3 switches. The first switch g_1 turns on the synthesis of both variables, i.e. $g_1 : (\bullet, \bullet) \to (2, 2)$, where $\bullet$ designates any level of the trigger. The second switch turns off the synthesis of genes in S-phase: $g_2 : (2, \bullet) \to (1, \bullet)$. The third switch turns off all the transcription, $g_2 : (\bullet, \bullet) \to (1, 1)$. We assume that the division is possible only in the state $(1, 1)$ with transcription switched off, and that after the division event, the cell enters into the state of active degradation of the cell cycle genes $(0, 0)$.

The three introduced switches will be characterized by the corresponding switching hyperplanes. The first switch is triggered when the sum of the collective aggregated levels of expression of the genes involved in S and G2/M phases reaches some minimum c_{min}, therefore, the linear functional associated with the first switch hyperplane is $f_1(x_s, x_m) = x_m + x_s - c_{min}$. The second switch is triggered whenever the collective aggregated level of expression of S phase-associated genes reaches some maximum value S_{max}, therefore, the linear functional associated with the second switch hyperplane is $f_2(x_S, x_M) = x_S - S_{max}$. Finally, the third switch is triggered when the collective aggregated level of expression of G2/M phase-associated genes reaches some maximum value M_{max}, therefore, the linear functional associated with the third switch hyperplane is $f_3(x_S, x_M) = x_M - M_{max}$.

In the end, the cell division event is triggered when the collective aggregated level of expression of G2/M phase-associated genes crosses some threshold M_e, therefore, the linear functional associated with the division event is $f_d(x_S, x_M) = M_e - x_M$.

Let us define the number of parameters in this simple switching model. Three introduced switches are characterized by 4 parameters $c_{min}, S_{max}, M_{max}, M_e$. There exist 6 rates k_i^v characterizing the movement vectors in the $9 = 3^2$ possible states, corresponding to all possible combinations of trigger levels. However, qualitatively, the dynamics in each automaton state is determined only by the direction of the corresponding vector and not its amplitude: therefore, one parameter per state visited is needed during the progression through the cell cycle. Under certain constraints on the rates formulated above, and also on the switch parameters (namely, $c_{min} < S_{max}, M_{max}$, $M_e < M_{max}$), the suggested model is constructed such that along the cell cycle trajectory only 4 states will be visited in a predefined order: $(0, 0) \to (2, 2) \to (1, 2) \to (1, 1)$. Therefore, the total number of parameters equals 8.

Knowing the position of four characteristic points along the cell cycle trajectory, namely $(x_S^{(B)}, x_M^{(B)})$, $(x_S^{(C)}, x_M^{(C)})$, $(x_S^{(D)}, x_M^{(D)})$, $(x_S^{(E)}, x_M^{(E)})$, it is possible to completely parameterize the automaton. The starting and the ending point of the cell cycle trajectory must be connected by the relation $(x_S^{(0)}, x_M^{(0)}) = (x_S^{(E)}, x_M^{(E)}) + d$, where d is the vector with

components $(-\log_{10} 2, -\log_{10} 2)$.

Therefore, we put $S_{max} = x_S^{(C)}, M_{max} = x_M^{(D)}, M_e = x_M^{(E)}$. Instead of using directly the B point, we will use the position of the non-proliferating cell with the maximum sum of the coordinates in the S, M plane, and we designate it as $x_S^{B'}, x_M^{B'}$ (other choices are also possible). Then $c_{min} = x_S^{(B')} + x_M^{(B')}$. Then we define rates:

$$k_2^v = \frac{x_v^{(C)} - x_v^{(B')}}{||x^C - x^{B'}||}, k_1^S = \frac{x_S^{(C)} - x_S^{(D)}}{||x^C - x^D||}, k_1^M = \frac{x_M^{(E)} - x_M^{(D)}}{||x^{(E)} - x^{(D)}||}, k_0^v = \frac{x_v^{(B')} - x_v^{(0)}}{||x^{(0)} - x^{B'}||}$$

The resulting steady state cell cycle trajectory is shown in Figure 4.3.

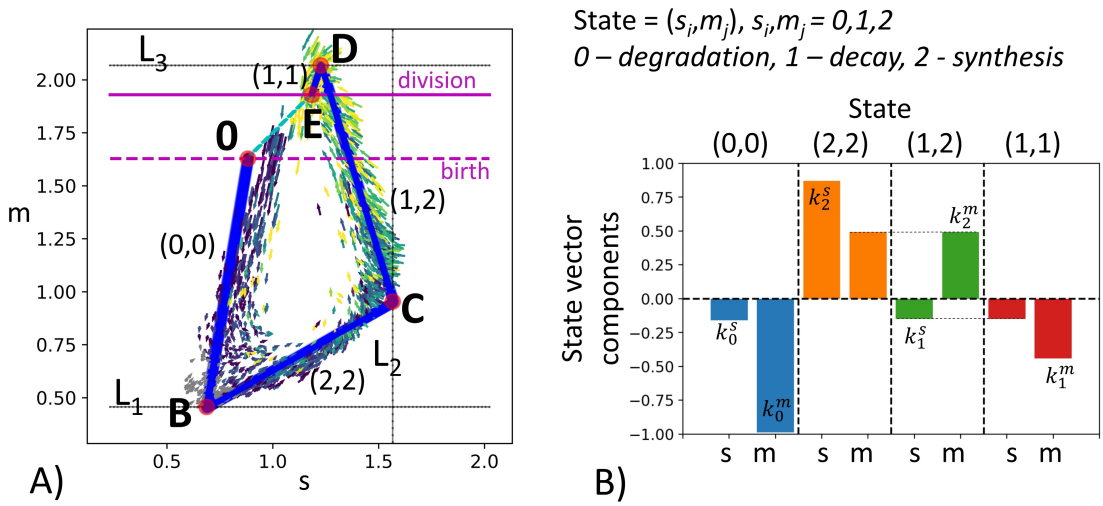

Figure 4.3: **Modeling transcriptomic cell cycle trajectory by an allometric growth with switches.** (a) Piecewise linear cell cycle trajectory fit to the single-cell RNASeq data (cell cycle trajectory, shown in Figure 4.1,A,right). The model contains three switching planes L_1, L_2, L_3, and is characterized by 4 states. The states are encoded with two triggers, each possessing three possible levels 0,1,2, the biological meaning of which is specified in B). (b) The growth vectors associated with each state are encoded by rates k_i^S, k_j^M, such that the components of the growth vectors equal (k_i^S, k_j^S), where i and j are the levels of the corresponding triggers. *From (Zinovyev et al., 2022).*

We denote the linear segments of the trajectory shown in Figure 4.3 as T1, Ts, T2, Tm, assuming that they have significant overlap with G1, S, G2 and M phases correspondingly.

The suggested model describes 2D dynamics of the signals S, M which are empirically shown to explain most of the variance of all cell cycle genes in scRNASeq data (see below). However, higher-dimensional generalization of the suggested model is always possible. Also, in the model, we simplified the observed dynamics in Figure 4.1,A, left which seems to contain 5 segments, with an additional curvature peak in point A. The segment A-B seems to contain non-proliferating cells, and might correspond to the transcriptional epoch most similar to the quiescent cell state, when the active degradation of the mitotic transcripts is completely finalized. The existence of this epoch is less pronounced in the S, M projection (Figure 4.1,A,right), therefore we merged segments 0-A and A-B' as the first order approximation.

4.7 Connection between the effective embedding dimensionality of cell cycle trajectory and the number of intrinsic states

The introduced cell cycle modeling framework is a simple and empirical model, lacking mechanistic details. Its main advantage is the possibility of analytical treatment of the

most general geometrical cell cycle trajectory properties. In this section, we use this framework to prove a theorem connecting the number of the intrinsic states of the cell cycle trajectory and its intrinsic dimensionality.

This geometry is embedded into a space of omics measurements, whose dimensionality might be very high (e.g., expression of thousands of genes). However, we can assume that the intrinsic dimensionality (ID) of CCT is much smaller and that the extrinsic state of the cell progressing through the cell cycle can be characterized by n extensive variables, where n is relatively small. We will refer to n as CCT embedding dimensionality. Empirically, it can be estimated by studying the snapshot of dividing single-cells profiled with a particular technology, and computing its global intrinsic dimensionality (ID), provided that other non cell cycle-related sources of heterogeneity could be dismissed in measurements. Estimating ID can be done using one of the many existing methods for ID estimation (Bac and Zinovyev, 2020; Albergante et al., 2019; Bac et al., 2021).

Let us establish the expected relation between n and the number of intrinsic states m of the automaton approximating CCT. We intend to claim that theoretically n should match m under some natural assumptions.

We first state that m cannot be smaller than n. In the theory of allometric growth with switches this statement has a character of strict theorem (see below), $m \geq n$. Secondly, we state that n is expected to be at least equal to m. Both statements are based on argumentation using "general position" statements. However, the former one is strictly necessary, while the latter one represents a feasible hypothesis.

Theorem on the number of intrinsic cell cycle states. *The number of segments m in the cell cycle trajectory modeled by the automaton with switches and linear growth in logarithmic coordinates is not less than the cell cycle trajectory intrinsic dimensionality n, or $m \geq n$.*

Proof. Let us consider the CCT dynamics in its n coordinates each of which represents an extensive variable. The variable extensiveness means, in particular, that its value, after the cell division moment, is divided by two. In logarithmic scale the cell division corresponds to the shift by vector $d \in R^n$ with n coordinates each of which equals $-\log 2$. Each intrinsic state is associated with a growth vector $a_i \in R^n, i = 1..m$. All non-negative linear combinations of a_i form a convex cone $Q = \{\sum_{i=1}^{m} \lambda_i a_i\}, \lambda_i \geq 0$. If $m < n$ then the set of vectors $\{d, \{a_i, i = 1..m\}\}$ is almost always linear independent and $-d \notin Q$. Hence, $-d$ is linearly separable from Q, according to the standard separability theorems. Linear separability of a point from a convex cone can be expressed as that for any non-zero $x \in Q$ we can find a linear function $l()$ such that $l(d) = 0$ and $l(x) > 0$. This makes the periodic cell cycle model impossible, because the function $l(x)$ increases along any growth direction, since for any i and $\lambda > 0$ we have $l(x + \lambda a_i) = l(x) + \lambda l(a_i) > l(x)$, and after cell division $l()$ does not change since $l(x + d) = l(x) + l(d) = l(x)$. Therefore, the necessary condition of existence of stable cell cycle trajectory is $m \geq n$, when the set of vectors $\{d, \{a_i, i = 1..m\}\}$ is linearly dependent, and also such choice of a_i that $-d \in Q$. Only in this case one can satisfy the cyclic condition $\sum_{i}^{m} \lambda_i a_i + d = 0$ in general position of vectors $\{d, \{a_i, i = 1..m\}\}$.

In simple words, this means that if $m < n$ then in a general position, each cell division (shift by d) moves a cell state out of the subspace defined by the growth vectors. The only way to make the trajectory stay in this subspace is to make the cell division vector d belong to this subspace that can be guaranteed only if $m \geq n$ (see Figure 4.4). The condition $m \geq n$ is necessary but not sufficient for a model to converge to a limit cycle. For example, in Figure 4.7, $m = n = 2$ (the theorem condition is satisfied) but the limit cycle in the model can be achieved only from some initial conditions and for some choice of vectors a_0, a_1.

Note that the proven Theorem is more general than the model of allometric growth with switches itself since it does not assume any particular shape of the switching surfaces L_k: they can be linear or nonlinear. Another generality consists in that the vector d can

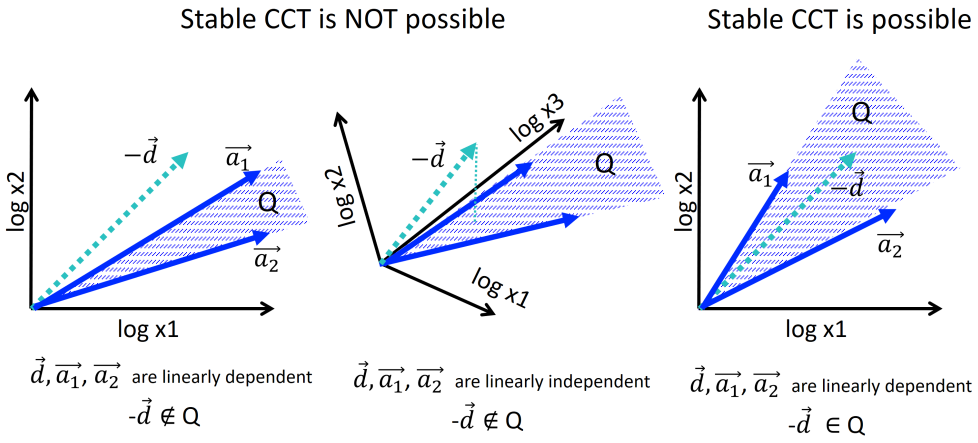

Figure 4.4: **Condition of existence of stable cell cycle trajectory in the model of allometric growth with switches.** For illustration, only two growth vectors a_1, a_2 are considered, and 2D or 3D embedding space. Stable piecewise linear trajectory is possible only if the negative of the cell division vector $-d$ belongs to the convex cone $Q = \sum_i^m \lambda_i a_i, \lambda_i \geq 0$. Only in this case, the cyclic equality $\sum_i^m \lambda_i a_i + d = 0$ is possible. In general position, the condition can be met only when $m \geq n$, where n is the dimensionality of the trajectory space (see text for the formal proof).

have any non-zero coordinates, not necessarily equal to $-\log 2$.

Examples in Figures 4.2,4.3 shows the case $n = 2, m > n$. The cell cycle trajectory modeled in Figure 4.3 contains $m = 4$ segments in 2D, which makes the vectors $a_i \in R^2, i = 1...4$ linearly dependent, and, of course, $d \in R^2$. The cell cycle model based on allometric growth is not contradictory in this case.

Now let us formulate our second statement. We can recall that vectors a_i are confined to the n-dimensional intrinsic subspace of CCT by projection from the multi-dimensional ambient space of all elementary measurements. The choice of n depends on our estimate of the CCT intrinsic dimensionality. However, movement along vectors a_i can be also seen in the complete space with thousands of coordinates. In this space, for sufficiently small m, any m vectors will almost always be linearly independent. Only projection into smaller than m-dimensional space will guarantee that these vectors are linearly dependent. This makes us hypothesize: if m segments are observed in CCT piecewise linear approximation in any linear projection then the most natural choice for n is at least m, i.e. $n \geq m$. Combining the two statements ($m \geq n$ and $n \geq m$) allows us to state that the correspondence $m = n$ is the most natural expectation for a cell cycle trajectory.

We explicitly verified this correspondence for the trajectory shown in Figure 4.1. The curvature analysis suggests the existence of 5 segments for the cell cycle trajectory reconstructed in the subspace of 30 first principal components of the complete dataset. However, some of these components might correspond to the variance not related to the progression through the cell cycle. In order to diminish the possible role of this variance, we considered a reduced version of the dataset confined to cell cycle-related genes only. We estimated the global intrinsic dimensionality, using six different linear ID estimators from scikit-dimension Python package (Bac et al., 2021), and it varied from 2 to 7, with average value 4.0. The scree plot shows existence of two dominant eigenvalues explaining 83% of total variance, indicating that the trajectory is relatively flat and located close to a 2D linear manifold. However, the residual variance demonstrated visible patterns related to transcriptional epochs in at least the first four principal components (Figure 4.5). The distribution of projections on the first four principal components well separated some transcriptional epochs (Figure 4.5,diagonal). Also, projections in higher dimensions high-

lighted the existence of sharp turning points between the segments which were less clear in the 2D projection on the first two principal components.

In addition, we split the data points into 5 classes according to projection on 5 segments of the principal curve (0-A, A-B, B-C, C-D, D-E), each of which is approximately linear. For each of this class, we computed the unity vector corresponding to the direction of the first principal component in the space of cell cycle genes with 198 dimensions. Afterwards, we estimated the effective rank of the matrix composed of 5 vectors representing the directions of the transcriptional epochs in the multi-dimensional space (see Methods), and it appeared to be 4, which indicates to that at least 4 out of 5 vectors determining the trajectory segments can be considered linearly independent.

As a result, we concluded that the embedding dimensionality for the transcriptomic cell cycle trajectory can be estimated as close to four. Therefore, restricting the trajectory to the plane of aggregate collective expressions of genes associated with S phase and G2/M phase (which roughly corresponds to the first two principal components) is a useful but incomplete approximation of CCT dynamics. Our reasoning suggests searching for additional biologically meaningful and statistically independent scores describing the progression through the cell cycle. The concrete gene expression dynamics shown in Figure 4.1,B provides a hint in this direction, but a careful and complete investigation of this question should be a subject of a separate study. As an additional argument, we can mention that some mathematical cell cycle models based on a fit to real data are four-dimensional (Singhania et al., 2011).

4.8 Extending the modeling formalism to piecewise smooth trajectories: simple kinetic model of cell cycle at transcriptomic level

The piecewise-linear model of automaton with switches described in the previous sections is phenomenological and lacks any notion of physical time and connection to the underlying kinetics of the lumped expression of genes involved in S phase and G2/M phases. A simple way to make it more concrete but still analytically tractable consists in introducing explicit processes of synthesis and degradation of the corresponding quantities, with kinetic rates changing in time. The simplest form of such dependence is piecewise-constant, with changes in the value of kinetic rates corresponding to the observed switches between transcriptional epochs of cell cycle progression.

Assuming the same epochs of cell cycle progression as above, and the same notations for variables (S, M, lumped expression of genes involved in S and G2/M phases correspondingly), their dynamics can be expressed as:

$$\begin{cases} \frac{dS}{dt} = k_t^S(t) - k_d^S(t)S \\ \frac{dM}{dt} = k_t^M(t) - k_d^M(t)M \end{cases} . \tag{4.1}$$

These equations must be accompanied by circular boundary conditions

$$\begin{cases} S(T) = S_f S(0) \\ M(T) = M_f M(0) \end{cases} , \tag{4.2}$$

where $S_f, M_f > 1$ are some numbers describing the drop of the lumped cell cycle variables after the moment of cell division. The most natural choice for them is $S_f, M_f = 2$, as before: however, here we prefer not to fix these parameters and rather fit them from the actually observed trajectory.

There exist several reasons for which S_f and M_f might appear in the range $1 \leq S_f, M_f \leq 2$ and not be equal. The most important of them is the technical biases introduced by sampling a limited amount of RNA, in the process of single-cell transcriptome

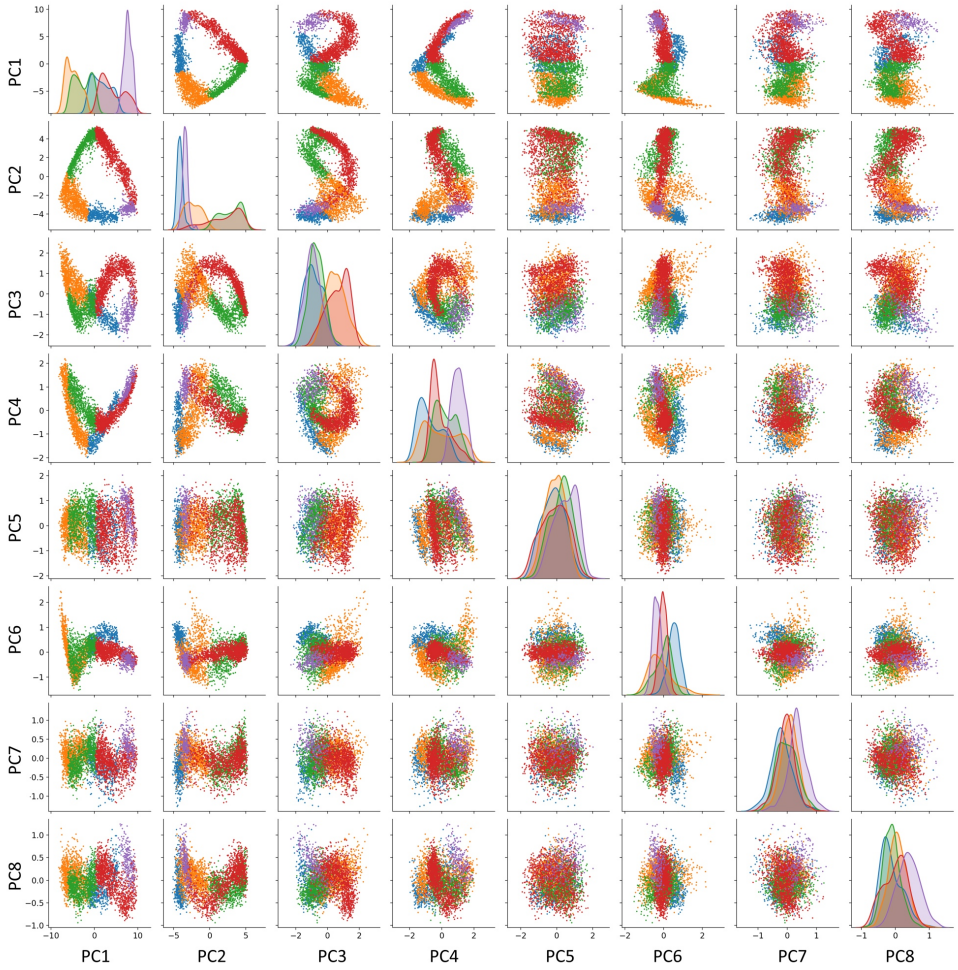

Figure 4.5: **Visualizing the transcriptomic cell cycle trajectory of CHLA9 cell line in projections on the first 8 principal components, computed in the subspace of known cell cycle genes.** The data points are partitioned according to the segmentation of the CCT into 5 transcriptomic epochs, also shown in Figure 4.1, 0-A (blue), A-B (orange), B-C (green), C-D (red), D-E (purple). *From (Zinovyev et al., 2022).*

sequencing. It can lead to the situation when after cell division, the amount of RNA decreases non-uniformly between molecular processes. In particular, in all our experiments, we do observe the total amount of RNA reads does not decrease exactly by 2.0 and is rather close to 1.7-1.8. The decrease of the individual gene expression after cell division in terms of the number of reads, forms a bell-shaped distribution around this value with standard deviation close to 0.2.

The equations (4.1) with piecewise-constant in time kinetic rates and the boundary conditions (4.2) can be solved analytically for arbitrary number of levels in the piecewise-constant functions $k_t(t), k_d(t)$. The resulting dynamics in the plane $\log S(t), \log M(t)$ represents a cell cycle trajectory parameterized by physical time, which consists of piecewise-smooth segments of three types. If a segment is characterized by $k_t^S(t) = k_t^M(t) = 0$ then the corresponding segment is linear in the logarithmic coordinates (since the underlying dynamics is exponentially decaying). If a segment is characterized by $k_d^S(t) = k_d^M(t) = 0$ then the corresponding segment is also linear in both logarithmic and initial coordinates. For a segment where at least one degradation k_d^* and one production kinetic rate k_t^* are positive, the dynamics follows a nonlinear curve in the logarithmic space, which remains monotonous (each of the coordinates does not change the derivative sign). The nonlinearity of the segment becomes important when one of the variables is in a stage exponentially increasing or decreasing, while the other is in a linear or close to saturation stage. Otherwise, the segment remains close to a line in logarithmic coordinates.

In order to choose the number of constant levels of the kinetic rates, we studied the averaged RNA velocity values along the cell cycle as a function of pseudotime (see Figure 4.6,A,B). For the S variable, we decided to keep only one non-zero level of $k_t^S(t)$ during the transcriptional epoch Ts, and two levels of $k_d^S(t)$, one for the exit from mitosis epoch and one for the rest of the dynamics. The choice was similar for M variable, but we took into account that a boost of expression of the lumped $G2/M$ genes is visible in the beginning of the transcriptional epoch T2s, just after switching off the S phase genes. During mitosis we assumed that all production rates are zero, corresponding to the lack of transcription in the M phase. The resulting choice of levels for the kinetic rates is shown in Figure 4.6,C.

The advantage of the proposed simple model of cell cycle trajectory is that it is fully analytically tractable and its parameters can be uniquely fit to the cell cycle trajectory observed in single-cell data, given some biologically meaningful constraints. Thus, assuming that the duration of mitosis is by order of magnitude faster than the T1s epoch, for CHLA9 cell line one estimates the ratio between transcriptional epochs T2s and T1s close to 1.0 and the value of transcriptional boost of G2/M genes in T2s epoch close to 2.5-fold (Figure 4.6,C). The determined values of all other parameters can be found in the Jupyter notebook at `https://github.com/auranic/CellCycleTrajectory_SegmentModel`.

4.9 Fitting parameters of the simple kinetic cell cycle model

Using the choice of levels for piecewise constant kinetic rates shown in Figure 4.6,C, we could derive the dependence of the initial state of the cell cycle from the kinetic rates and the durations of four transcriptional epochs T_1, T_{1s}, T_{2s}, T_m:

$$\begin{cases} S(0) = \dfrac{k_t^S}{k_d^{S,2}} \dfrac{e^{k_d^{S,2} T_{1s}} - 1}{S_f e^{k_d^{S,2}(T_{1s}+T_{2s}+T_m)} - e^{-k_d^S T_1}} \\ M(0) = \dfrac{k_t^M}{k_d^{M,2}} \dfrac{p \cdot e^{k_d^{M,2}(T_{1s}+T_{2s})} - (p-1) e^{k_d^{M,2} T_{1s}} - 1}{M_f e^{k_d^{M,2}(T_{1s}+T_{2s}+T_m)} - e^{-k_d^M T_1}}. \end{cases} \tag{4.3}$$

Starting from the initial point of the trajectory $S(0), F(0)$ it is possible to analytically write down the coordinates of all other borders of the transcriptional epochs:

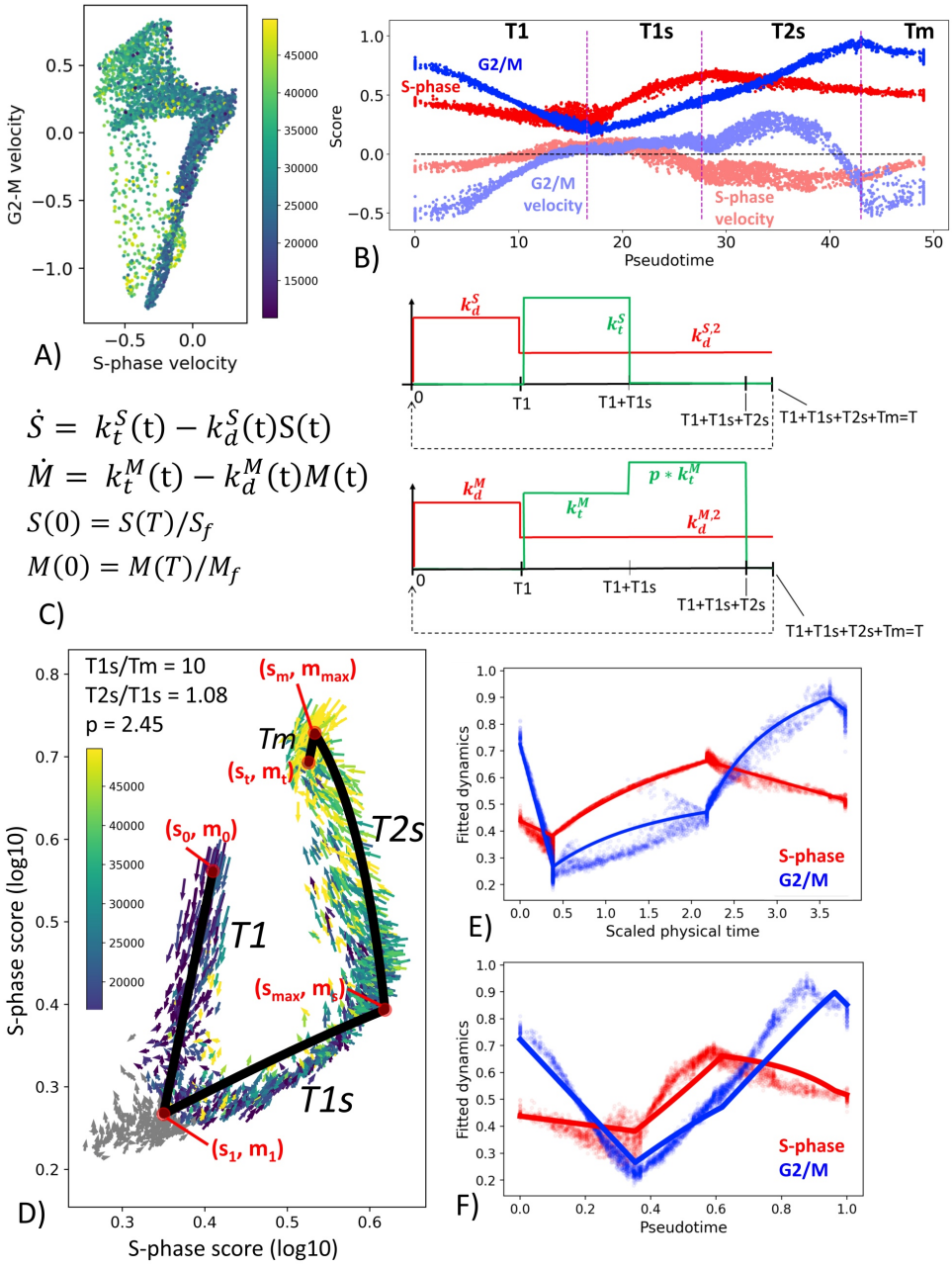

Figure 4.6: **Simple kinetic model of cell cycle transcriptome dynamics.** (a) Mean RNA velocity values for S-phase and G2/M genes. (b) Pseudotemporal dynamics of S-phase and G2/M scores (shown with more intense color) and mean RNA velocity values (shown with semi-transparent color). (c) Description of the simple kinetic model of cell cycle transcriptome. Model equations are shown on the left and the changes in the values of kinetic rates (degradation, in red, and synthesis, in green). (d) Result of fitting the model dynamics to cell cycle transcriptome dynamics observed in CHLA9 cell line. (e), (f) Inferred physical time and pseudotemporal dynamics of cell cycle transcriptome in CHLA9 cell line. *From (Zinovyev et al., 2022).*

$$\begin{cases} S(T_1) = S(0)e^{-k_d^S T_1} \\ M(T_1) = M(0)e^{-k_d^M T_1} \\ S(T_1+T_{1s}) = \frac{k_t^S}{k_d^{S,2}}\left(1-(1-\frac{k_d^{S,2}}{k_t^S}S(T_1))e^{-k_d^{S,2}T_{1s}}\right) \\ M(T_1+T_{1s}) = \frac{k_t^M}{k_d^{M,2}}\left(1-(1-\frac{k_d^{M,2}}{k_t^M}*M(T_1))e^{-k_d^{M,s}T_{1s}}\right) \\ S(T_1+T_{1s}+T_{2s}) = S(T_1+T_{1s})e^{-k_d^{S,2}T_{2s}} \\ M(T_1+T_{1s}+T_{2s}) = \frac{p\cdot k_t^M}{k_d^{M,2}}\left(1-(1-\frac{k_d^{M,2}}{p\cdot k_t^M}*M(T_1+T_{1s}))e^{-k_d^{M,2}T_{2s}}\right) \\ S(T) = S(T_1+T_{1s}+T_{2s})e^{-k_d^{S,2}T_m} \\ M(T) = M(T_1+T_{1s}+T_{2s})e^{-k_d^{M,2}T_m}, \end{cases} \tag{4.4}$$

where $T = T_1 + T_{1s} + T_{2s} + T_m$ is the full duration of the cell cycle. One can estimate the position of these points from the analysis of observed cell cycle trajectory curvature ((s_0, m_0), (s_1, m_1), (s_{max}, m_s), (s_m, m_{max}), (s_t, m_t), shown by red points in Figure 4.6,D)) by requiring that the model trajectory should pass as close as possible to them. This defines an optimization problem which can be easily solved numerically by iterations, using the simplest fixed-point algorithm. The details of parameter fitting are provided in the Jupyter notebook at https://github.com/auranic/CellCycleTrajectory_SegmentModel.

We note that this optimization does not allow us to determine all the model parameters uniquely, since they enter in the aforementioned optimization functional as certain combinations (as simple rational functions), namely, $\frac{k_t^S}{k_d^{S,2}}$, $\frac{k_t^M}{k_d^{M,2}}$, $k_d^S T_1$, $k_d^M T_1$, $k_d^{S,2} T_{1s}$, $k_d^{M,2} T_{1s}$, $k_d^{S,2} T_{2s}$, $k_d^{M,2} T_{2s}$, $k_d^{S,2} T_m$, $k_d^{M,2} T_m$. Two other parameters M_f, S_f define the observed cell division vector in (4.2). One extra parameter p denotes transcriptional production acceleration of $G2/M$ genes during the transcriptional epoch T2s compared to the transcriptional epoch T1s (Figure 4.6,C). Not all these quantities are independent, some of them are connected through nonlinear relations:

$$\frac{k_d^{S,2}\cdot T_{2s}}{k_d^{S,2}\cdot T_{1s}} = \frac{k_d^{M,2}\cdot T_{2s}}{k_d^{M,2}\cdot T_{1s}}, \frac{k_d^{S,2}\cdot T_m}{k_d^{S,2}\cdot T_{1s}} = \frac{k_d^{M,2}\cdot T_m}{k_d^{M,2}\cdot T_{1s}}, \tag{4.5}$$

which overall gives 11 independent combinations of parameters provided 10 measurable coordinates of cell trajectory turning points in Figure 4.6,D.

Altogether, this means that 1) one needs to introduce at least one additional constraint in order to make the trajectory reconstruction unique and 2) physical time of the epochs T_1, T_{1s}, T_{2s}, T_m can not be uniquely computed from the cell cycle trajectory observed in the plane of S-, G2/M-phase scores. From the analysis of equations (4.5) it follows that the model can be uniquely parameterized if one will constrain one of the three quantities $p, \frac{T_{2s}}{T_{1s}}, \frac{T_m}{T_{1s}}$. Finally, it is convenient to fix the durations T_1, T_{1s} to some arbitrary values which allows to determine parameters k_d^S, k_d^M and the ratios $\frac{T_{2s}}{T_{1s}}, \frac{T_m}{T_{1s}}$.

In our numerical experiments, we fixed the values of T_1 and T_{1s} to their corresponding pseudotemporal durations (as the corresponding fractions of the total length of the cell cycle trajectory). We also fixed the ratio $\frac{T_m}{T_{1s}} = 10$, assuming that the mitosis must be fast in physical time compared to the transcriptional epoch including activating the expression of the genes involved in the S-phase.

4.10 Simulating cell cycle trajectories with various durations of temporal transcriptional epochs

After fitting the kinetic parameters for an observable in the S-phase vs G2/M score plane cell cycle trajectory, one can perturb the parameters and investigate how the trajectory geometry depends on them.

In real life scRNASeq datasets, we observe that CCT geometry can appear very different in various biological systems. When projecting onto the plane of standard scores of S-phase and G2/M phase genes, scRNASeq datasets might not always reveal the circular nature of CCT. In some cases, the circular structure is not at all detectable via this projection, (Figure 4.7), and the two scores might be connected via a strong positive or negative correlation. Also, in some systems we observed co-existence of several CCT shapes, like it is the case in the U2OS cell line dataset (GSE146773). The univariate histograms of two score distributions might be characterized by bi- or uni-modal character.

Quite strikingly, we were able to reproduce these patterns qualitatively by fitting the kinetic parameters to the CHLA9 scRNASeq dataset, and then by manipulating the durations of $T1$, $T1s$ and $T2s$ transcriptional epochs and producing computer-simulated trajectory examples. Thus, significant reduction in the duration of both $T1$ and $T1s$ epochs led to the negative correlation pattern between S-phase and G2/M scores. This could be interpreted as drastic reduction of the G1 cell cycle phase. In real life datasets, such pattern has been observed in human embryonic stem cells (dataset GSE85917).

If both $T1$ and $T2s$ were shortened then this led to the increase of the positive correlation between two scores, (Figure 4.7). This pattern was indeed observed in human bone marrow and human neural epithelial stem cell-related single-cell datasets (GSE99095 and GSE81475).

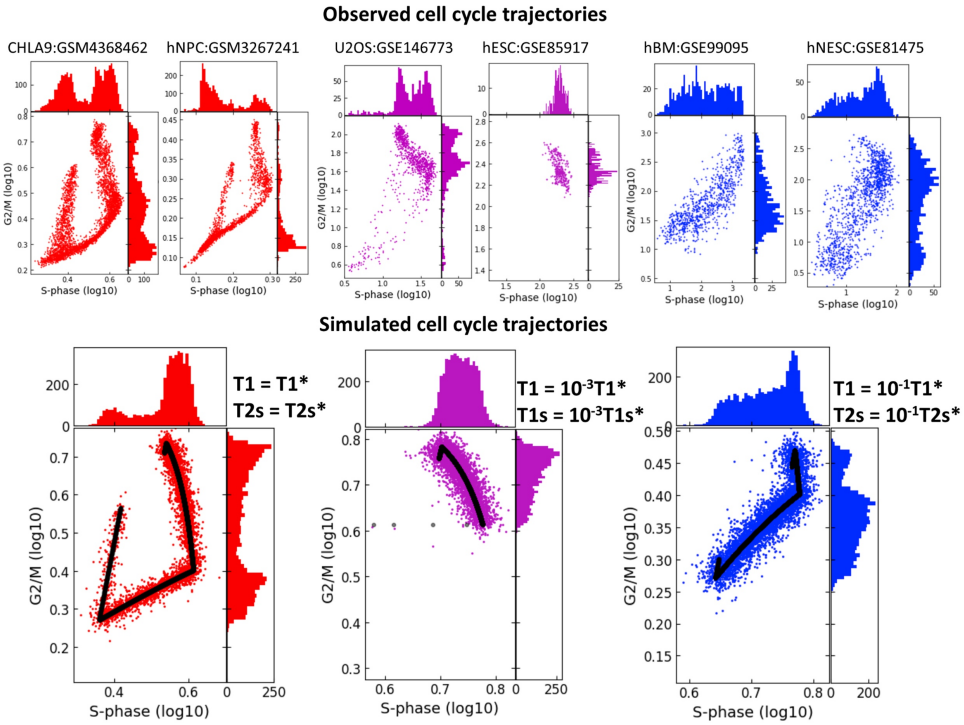

Figure 4.7: **Studying the effect of shortening the durations of transcriptional epochs T1 and T1s or T1 and T2s on the geometry of cell cycle trajectory projected onto the S-phase and G2/M-phase scores plane.** The simulated trajectories (in the lower part of the figure) are produced by taking the parameters of the CHLA9 fit of model dynamics (red plot) and changing the durations of T1 and T1s epochs (violet plot) or the durations of T1 and T2s epochs (blue plot). Each simulation shows the trajectory (black line) sampled with Laplacian noise added, with score distribution histograms shown at the plot margins. The upper part of the plot shows six real-life cell cycle trajectories observed in different systems, with GEO identifiers indicated. In each plot title either cell line name is provided, or hNPC means human neural precursor cells, hESC - human embryonic stem cell, hBM - human bone marrow, hNESC - human neural epithelial stem cell. *From (Zinovyev et al., 2022).*

4.11 Predicting cell line doubling time from the geometrical properties of cell cycle trajectory

The developed simple kinetic model leads to a simple prediction which can be validated: *the total length of the transcriptomic cell cycle trajectory must diminish in rapidly dividing cells*. This can be interpreted as a consequence of the fact that in a rapid proliferation process, during the post-mitotic G1 phase (T1 transcriptional epoch), there is not enough time to degrade all mitotic transcripts produced before the cell division moment, so they are reused in the consequent cell cycle phases, shortening the subsequent G1 phase.

We verified this prediction in a relatively large collection of cell line scRNASeq datasets. Using the data from Cellosaurus database, we identified those few ones for which the cell line doubling time has been estimated, and for which the number of available good quality single-cell profiles exceeded 300.

We used the total length of the principal circle fit in the 2D plane of the scaled to maximum equals one cell cycle phase scores, as a proxy to quantify the level of CCT contraction (see Methods). This measure was correlated with cell line doubling time in hours. Two cell lines CHLA10 and SCC25 appeared to be strong outliers from otherwise significant positive regression line (Pearson correlation 0.931, p-value=10^{-5}) (Figure 4.8). When this regression line was used as a predictor, CHLA10 cell line was predicted to have doubling time around 64 hours (instead of determined by database search of around 32 hours) and for SCC25 around 78 instead of 50 hours. It is known that cell line doubling time can vary depending on the growth conditions, so we hypothesized that this variability could explain the appearance of two outliers. If two of them were kept in the regression calculation, it remained significant but less strong (Pearson correlation 0.67, p-value=0.01).

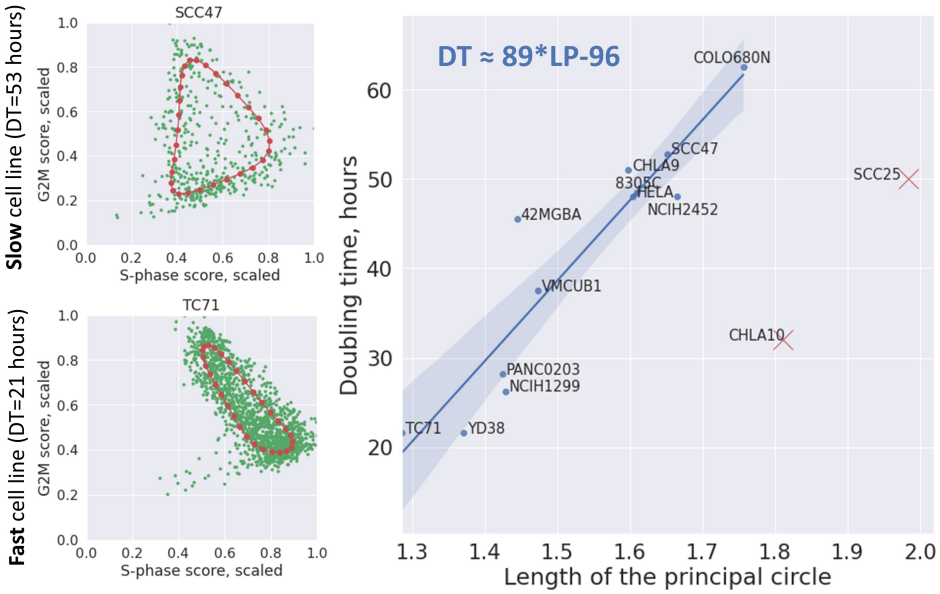

Figure 4.8: **Dependence of cell line doubling time (DT) on the length of the principal circle (LP) approximating the cell cycle trajectory in the 2D plane of scaled (divided by the maximum value) S-phase and G2M scores.** On the left two examples of principal circles are shown in red, and cells in green. On the right the linear regression line with confidence intervals is shown connecting the length of the principal circle with cell line doubling time (Pearson correlation 0.931, p-value=10^{-5}). The regression formula is shown on the plot in top left corner. Two cell lines indicated by red crosses were eliminated from the regression as evident outliers. *From (Zinovyev et al., 2022).*

4.12 Discussion

This chapter provided a framework for analyzing the cell cycle trajectories using single-cell omics measurements such as scRNASeq data. Unlike the previously suggested model of the trajectory as a flat circle, we provide arguments that at least in some conditions the piecewise-linear in logarithmic coordinates approximation appears to fit the single-cell transcriptomic data and to be biologically tractable. In particular, it allows us to delineate transcriptional epochs of cell cycle at which the corresponding segment of the trajectory remains close to linear in logarithmic coordinates which corresponds to locally allometric changes of the transcriptome.

We suggest two modeling formalisms to recapitulate the cell cycle transcriptomic dynamics as a sequence of switches. The first one is purely phenomenological and describes the dynamics as a change of states of a hidden automaton, leading to the switches of parameters of allometric growth, followed by a shift representing the cell division event. The advantage of this formalism is that it allows us to treat most general properties of cell cycle trajectory geometry.

In particular, we could prove a fundamental theorem on the number of intrinsic cell cycle states, which connects the number of linear segments in the trajectory and the embedding dimensionality of the cell cycle trajectory. The nature of this theorem, relying on "general position"-type arguments, is reminiscent of the well-known results imposing constraints on the number of the system's internal states and the effective dimensionality of its environment, in several fields of science. For example, the Gause's law of competitive exclusion and its generalizations states that the number of competing species is limited by the effective number of resources, characterizing the environment (Gauze, 1934; Gorban, 2007). The famous Gibbs' phase rule in thermodynamics connects the effective number of the intensive variables with the number of components and phases in a system at thermodynamic equilibrium (Gibbs, 1961; Alper, 1999). All these results are also similar in terms of practical difficulties related to determining the effective system's dimensionality.

From the physico-chemical point of view, the effective dimensionality is the number of the substances "lumps" in the cell cycle kinetics. Lumping-analysis produces a partition of all chemical species into a few groups and then considers these groups ("lumps") as independent entities (Wei and Kuo, 1969). "Amounts" of these lumps are the combinations of the amounts of the chemical species (Li and Rabitz, 1989, 1990). The theorem on the number of intrinsic cell cycle states that the number of lumps n does not exceed the number of the internal states of the cell cycle transcription machinery. This means that kinetics allows reduction of the huge-dimensional space of all components to $n \leq m$ number of aggregated lumps.

The second modeling formalism that we suggested connects the geometric properties of the cell cycle trajectory to the underlying transcriptional kinetics and physical time. It uses the simplest chemical kinetics equations with kinetic rates represented as piecewise-constant functions of time. We show that the suggested model is fully analytically tractable and, under some biologically transparent assumptions, allows unique determination of its independent parameter combinations. This type of modeling allowed us to explicitly study the relation between pseudotime and physical time.

The precise connection between physical time and pseudotime (geometric time) in the cell cycle is worth studying in more detail since this is the central question in the dynamic phenotyping approach in general (Golovenkin et al., 2020). Some of these relations can be potentially quantified from exploring the variations of point density along the inferred trajectories (Chen et al., 2019a). Related to this, one can expect non-trivial phenomena in studying the cell cycle trajectory, such as effects of partial cell population synchronization under assumption of equal cell cycle durations in individual cells. This effect can lead to the appearance of density peaks in the reconstructed cell cycle trajectories that cannot be explained by nonlinear relation between physical time and pseudotime (Gorban, 2007).

As one of the applications of the suggested modeling formalism, we performed several

numerical experiments on changing the durations of the transcriptional epochs overlapping with G1 or G2 cell cycle phases. We observed that these parameters might have a drastic effect on the shape of the CCT geometry and the form of the univariate variable distributions. This model prediction can be qualitatively confirmed by observing CCT properties of several in vitro and in vivo systems. The effect of CCT shrinkage might be relevant in characterizing the cell cycle properties in various conditions: for example, when one can manipulate the activity of an oncogene (Aynaud et al., 2020). We show that the CCT geometry can be predictive to estimate the cell line doubling time which can be a proxy of cell cycle duration.

The relation between transcriptomic dynamics and the established definitions of cell cycle phases and cell cycle checkpoints has been discussed and even quantified using standard molecular biology techniques (Giotti et al., 2019; Hsiao et al., 2020). In this study, we deliberately leave open the question on defining the exact cell cycle phase borders from the transcriptomic CCT geometry. We found that this relation can not be the exact match: one of the reasons for this is delayed production of proteins, and dependence of the cell cycle progression from post-translational protein modifications. The transcriptomic dynamics is relatively slow, and activation of protein synthesis is switched on in advance, leaving time for producing enough proteins needed at a certain stage of the cell cycle molecular program. Same is true for the process of degradation of RNAs involved in cell cycle: a cell needs enough time after mitosis to degrade all cell cycle-related transcripts.

The suggested formalism is not limited to transcriptomic data. It looks promising to analyze the geometrical properties of cell cycle trajectory measured in unsynchronized cell populations profiled at various levels of molecular description, including epigenetics and protein expression, when the datasets of sufficient volume and quality will become available.

A more mechanistic description of the cell cycle has been already proposed in the context of yeast or mammalian cells (Tyson, 1991; Novák and Tyson, 2004). The mathematical models can be based on chemical kinetics or on discrete or hybrid frameworks (Singhania et al., 2011; Noël et al., 2013), but in all cases, the difficulty when constructing these models is to select the genes that can capture the main features of the cell cycle and the different events that allow the switch from one phase to another. We anticipate that the type of analyses presented here could orient the choice of these genes and inform on their dynamics.

Chapter 5

Uncovering Ewing sarcoma cell processes using inducible cell lines

Ewing sarcoma is an aggressive pediatric bone and soft tissue tumor, characterized by a high genomic stability and a specific oncogene involved in most Ewing sarcoma cases, EF1, which makes it a very relevant tumor type to study. We described its main features in the introduction of this Ph.D. thesis, and this chapter will be focused on presenting the methods we carried out to study the gene expression of Ewing sarcoma cells, and the results we obtained about their dynamics and their heterogeneity.

In this project, we studied inducible Ewing sarcoma cell lines in which EF1 can be suppressed, and we used these to follow the effects of removing and reintroducing the oncogene in the Ewing sarcoma cells. The idea is to observe which cell processes are perturbated at the gene expression level in the absence of EF1, and to follow the recovery dynamics of these biological signals when EF1 is no longer suppressed. We are also interested in seeing whether some changes in gene expression that happen after EF1 suppression are irreversible, or if all cells can recover their full initial malignant phenotype. We also wanted to know if biological signals identified in our last study published in (Aynaud et al., 2020) could be found in those new datasets in an unsupervised fashion, and if new Ewing sarcoma cell processes can be identified. The reason for thinking this could be the case is the fact we new use a more recent scRNA-seq technology, 10X Genomics, than those that were used during Aynaud's study, SmartSeq, and Chromium. Using 10X should therefore yield datasets with a greater number of cells and more UMIs, which may allow us to detect more subtle biological signals that could not be observed in the previous study.

We will first present the experimental setup used to cultivate and gather data from these Ewing sarcoma inducible cell lines, as well as preliminary data analyses we carried out (quality control and data exploration). We will then discuss how we identified Ewing sarcoma biological processes at the transcriptional level in these datasets, and provide a comparison with the ones identified in the previous study. We will finally dive into these new transcriptional signatures, and observe how EF1 levels influence the activity of other Ewing sarcoma biological processes.

5.1 Inducible Ewing sarcoma cell lines, a time-resolved study

5.1.1 Experimental setup

We use ASP14 Ewing sarcoma cell lines that contain a genetic construct that expresses small hairpin RNAs (shRNAs) anti-EF1 that suppress the effect of EF1, the main Ewing sarcoma oncogene (Fig. 5.1a). The promoter of this genetic construct contains a TET system that can be used to control the expression of these shRNAs. By default, anti-EF1

shRNAs are not expressed, meaning EWSR1-FLI1 mRNAs are normally expressed and translated into proteins. Introducing doxycycline in the growth medium frees the promoter (Fig. 5.1b), which triggers the expression of the shRNAs that causes the suppression of EF1 by degrading EWSR1-FLI1 mRNAs. Removing doxycycline from the medium restores the initial condition.

The idea of this time-resolved experiment is to follow gene expression at the transcriptome level, and at different points during the induction of EF1 (Fig. 5.1c), to observe how EF1 affects various cell processes at the gene expression level; all wet lab experiments have been carried out by Lou Carlier during her Master internship (École Normale Supérieure Paris-Saclay) and Karine Laud-Duval (Institut Curie, U830). ASP14 cells have been unfrozen, and scRNA-seq acquisition has been performed at day 0 and day 2 in non-doxycycline conditions (with full EF1 effect). In parallel, another ASP14 culture was grown for a week in the presence of doxycycline, and has been sampled at days 2, 3, 4, and 7 for scRNA-seq data acquisition. Doxycycline was removed from the growth medium at day 7, and new scRNA-seq datasets were gathered at days 8, 9, 11, 14, 17, and 21. This allowed us to study both an inhibition phase (days 2, 3, 4, and 7), a rescue phase (days 8, 9, 11, 14, 17, and 21), and a control condition (days 0 and 2 from the first condition). In the end, we ended up with 12 scRNA-seq datasets to be analyzed.

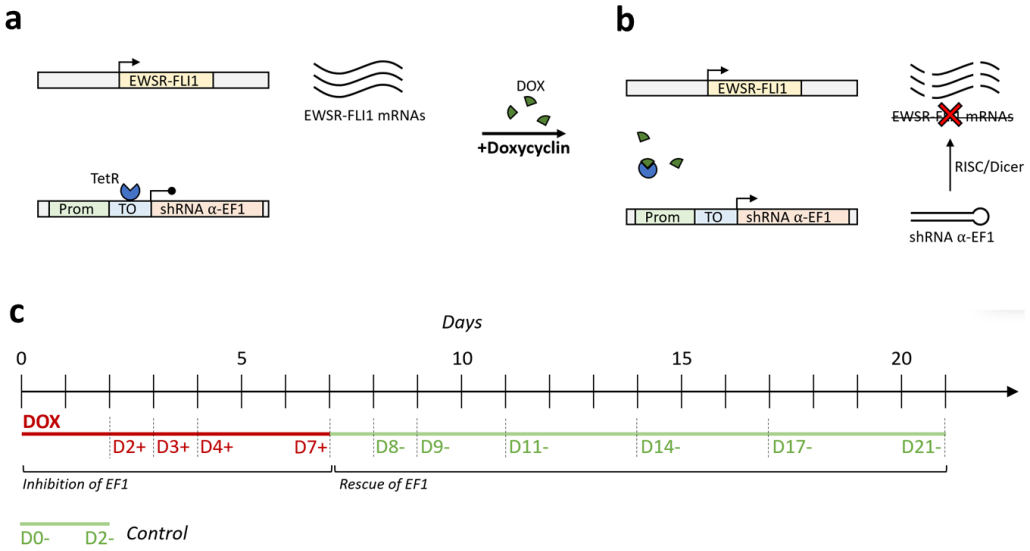

Figure 5.1: **Inducible cell lines setup. (a)** In the absence of doxycyclin, EF1 mRNAs are transcribed. **(b)** In the presence of doxycyclin, EF1 shRNAs are expressed and trigger the degradation of EF1 mRNAs. **(c)** Time course of the experiment, between day 0 and day 7 EF1 is inactivated by doxycyclin, doxycyclin is removed at day 7, and induction of EF1 is followed between days 7 and 21. RNA-seq assays are conducted at days 0 and 2 without doxycyclin (control), and at days 2, 3, 4, 7, 8, 9, 11, 14, 17 and 21.

5.1.2 Quality control and data exploration

We processed all 12 datasets using typical scRNA-seq preprocessing steps. We first filtered out all cells with only a few genes expressed (<200) and all genes expressed in less than three cells. We also removed all cells with more than 10,000 gene counts, as we suspected they corresponded to doublets, as well as cells with more than 10% of mitochondrial genes that may be undergoing apoptosis. All thresholds have been chosen *ad hoc* based on visual inspection of statistical plots. Gene counts per cell were then normalized to 10,000, and the 10,000 most variable genes were kept in each dataset. We finally performed a denoising

step where each cell was pooled toward the average of its five nearest neighbors.

Depending on the dataset, this first step filtered a varying proportion of cells (Fig. 5.2a), between 210 cells at day 17 and 2867 cells at day 4; interestingly, days during which doxycycline was present in the growth medium coincide with days with the highest number of cells. We also visualized all the datasets together by applying dimensionality reduction algorithms in the space of their common genes. According to the PCA plot (Fig. 5.2b), datasets seem to be localized per day; in particular, we see day 0 and day 21 datasets colocalize, and datasets are positioned inbetween in order, in the couterclockwise direction. This suggests that cells at day 21 have similarities with cells at day 0 and 2, meaning a part of the malignant gene expression profile may have been recovered. An examination of the UMAP plot which leverages a non-linear unsupervised dimensionality reduction algorithm (Fig. 5.2c) reveals that cells are also clustered per day in this space, but day 0 and day 2 do not colocalize with day 21. Instead we see three big clusters, one corresponding to day 0 and day 2 (pre-inhibition), one corresponding to the doxycycline condition (inhibition), and one containing the activation days (day 8 to day 21). Interestingly, day 7 does not cluster with the other inhibition days despite cells also being inhibited at this time point. In order to get a better understanding of this experiment, we will discuss and interpret these datasets in the next sections through the prism of cell processes and transcriptional signatures.

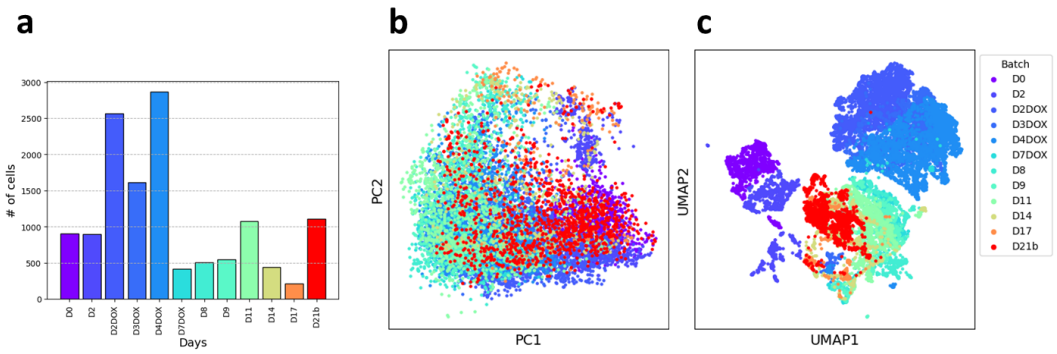

Figure 5.2: New inducible cell lines datasets presentation. **(a)** Number of cells selected after preprocessing within each batch. **(b)** PCA representation of datasets concatenated in common genes space. **(c)** UMAP representation of datasets concatenated in common genes space.

5.2 Identifying Ewing sarcoma transcriptional signatures

Ewing sarcomas are complex biological systems composed of cells that carry out various biological processes. For instance, some cells undergo cell cycle or metabolize glucose, others show apopotosis signals... The important subtlety to account for is that these cell processes are not mutually exclusive, meaning a single cell can undergo several processes at the same time (but not necessarily at equal intensity). For this reason, there is a crucial need for deconvolution methods that would facilitate the discovery of the biological processes carried out within a cell mixture, as well as identifying, for each individual cell, which cell processes this cell is undergoing. This type of methodology allows scientists to get an insight into the heterogeity of processes happening within tumors, which is a precious information to better understand their development.

In this section, we will present how Independent Component Analysis (ICA) can be applied as a deconvolution method to identify the various cell processes happening within a population of cells measured through a single-cell RNA-seq assay. We will present two applications of this methodology, one that we carried out in the past on various Ewing

sarcoma datasets, and a more recent application to inducible Ewing sarcoma cell lines. By modulating the oncogene expression, we were able to observe how cancer-related cell processes slow down when the oncogene activity is removed, and we could follow the orchestration of the come-back of these phenotypes once the oncogene is reintroduced. Also, ICA being a linear deconvolution method, it naturally yields individual gene contribution to each of the identified processes which can be used as a powerful interaction discovery tool.

5.2.1 Inducible Ewing sarcoma cell lines setup

As discussed in the introduction of this thesis, we applied ICA in the past to single-cell RNA-seq datasets (Aynaud et al., 2020). It allowed us to identify a set of transcriptional signatures related to various Ewing sarcoma biological processes, such as expression of EF1 targets, cell cycle-related signatures, glucose metabolism, hypoxia and extracellular matrix organization. Even though all these signatures represent relevant Ewing sarcoma phenotypes, they are clearly skewed towards "EF1-high"-related signals. Furthermore, they were identified based on the Fluidigm-C1 scRNA-seq technology, which is far behind today's standards in terms of sequencing depth and number of cells.

We therefore decided to carry out a new experiment using the inducible ASP14 cell lines using the 10X Genomics technology, in order to answer several questions. First of all, we wanted to ensure these transcriptional signatures are robust and can be redisovered in a new independent dataset. If this is the case, we also wanted to see if a higher resolution scRNA-seq assay could identify more non-EF1-high transcriptional programs. The experiment was scheduled over 21 days as follows, and was carried out by K. Laud-Duval and L. Carlier (U830):

- Cells were sequenced at day 0 (D0) and D2 in the absence of doxycyclin, thus allowing the effect of EF1. These time-points represent the *pre-inhibition* condition.
- In another cell culture, doxycyclin was added at D0. Cells were sequenced at D2, D3, D4 and D7. These time points represent the *inhibition* condition.
- Doxycyclin was removed from the medium at D7. Cells were sequenced at D8, D9, D11, D14, D17 and D21. These time points represent the *induction* condition, and we expect the EF1-high phenotype to progressively come back throughout these days.

This time-resolved experiment allowed us to obtain a set of 12 scRNA-seq datasets: D0, D2, D2DOX, D3DOX, D4DOX, D7DOX, D8, D9, D11, D14, D17 and D21. These datasets were preprocessed using the *scanpy* (Wolf et al., 2018) Python package following state-of-the-art guidelines for preprocessing scRNA-seq data. We first filtered out cells with too few gene counts, cells with too many (doublets), as well as apoptotic cells characterized by a large proportion of mitochondrial genes expressed. We used in this first step thresholds empirically determined for each dataset. We then independently normalized total counts at 10,000 in each cell, and applied the $\log(1 + x)$ function to each value of the resulting matrix. We then selected for each dataset the 10,000 most variable genes, in order to improve downstream analyses both in terms of efficiency and quality. We finally pooled each cell to the average of its 5 nearest neighbors, to mitigate biological noise and smooth the dataset.

5.2.2 Indentifying consensual independent components

One of the goal of this project was to identify new Ewing sarcoma transcriptional signatures, if possible using cells from all datasets in the analysis. The main problem of this approach is the various batch effects that exist between the different batches: scRNA-seq

analyses are indeed carried out at different time points, under different biological conditions. For this reason, it was clear that concatenating all datasets into a single one then applying ICA to this would highlight these batch effects. In this case we were nontheless able to rediscover a few independent components, such as the ones associated with cell cycle (G1/S and G2/M), as well as the component that we interpreted as EF1 targets expression. In order to obtain more subtle signals, we therefore decided to carry out a different approach.

The idea was to first compute the independent components in each dataset separately, and then use this result to define "consensual" independent components that would be reproducible in many datasets. We therefore used a mutual nearest neighbors search approach between independent components that can be summarized as follows. Given N scRNA-seq datasets $\mathbf{X}_i$ ($i \leq N$), we first computed a set of $d \geq 0$ independent components in each of them ($V_i = \{\mathbf{v}_{i1}, \ldots, \mathbf{v}_{id}\}$ for $\mathbf{X}_i$). Then, for each pair of component sets V_i and V_j, components in V_i are matched to the components in V_j following a mutual nearest neighbor (MNN, or Reciprocal Best Hit, RBH) approach. For an integer neighborhood size k, two components $\mathbf{v}_1 \in V_1$ and $\mathbf{v}_2 \in V_2$ are MNN if $\mathbf{v}_1$ belongs to the k-nearest neighbors of $\mathbf{v}_2$ among the components of V_1, and reciprocally. Doing so over all pairs of component sets yields a large network of components, where edges represent the MNN hits.

We carried out this analysis using the *stabilized-ica* package (Captier et al., 2022), computing independent components from each day separately. In order to also capture signals that may not strongly vary within one dataset such as oncogene activity, we added also three meta-days to the analysis: *INHIBITION* which groups D2DOX, D3DOX, D4DOX and D7DOX, *ACTIVATION* which groups D8, D9, D11, D14, D17 and D12, and *ALL* which groups all datasets. We then conducted the MNN-based graph construction which yielded the network displayed on Fig. .... We then used the Cytoscape tool Shannon et al. (2003) to refine the graph layout in order to better visualize its structure Fig .... Interestingly, many pseudo-cliques (i.e. clusters of nodes with strong levels of interconnectivity) can be observed with nodes that represent similar components coming from different datasets. In the following section, we describe how we can turn these component cliques into transcriptional signatures.

5.2.3 From independent components to transcriptional signatures

We opted for a simple rank-based selection scheme to obtain a representative gene set of each pseudo-clique. We ranked gene scores within each component $\mathbf{v}_i$, selected the top 25 genes $G_i = \{g_{i_1}, \ldots, g_{i_{25}}\}$. We finally defined the set G of consensus genes for the component as $G = \bigcup_{i=1}^{N} G_i$, so that any gene that is highly ranked within at least one component is included in the final set. The value 25 was chosen after some testing as it allowed to have consensus gene sets of reasonable size (around 100 genes), but we are aware that tuning this parameter can drastically change the result. Once this step has been completed, we end up with a gene set for each pseudo-clique.

In order to interpret these gene sets, one must go through a *gene enrichment* process that consists in screening the existing databases for cell processes or pathways involving genes of these sets. We opted for the ToppGene tool (Chen et al., 2009), which allows the user to input a list of genes and searches diverse types of databases (molecular functions, biological processes, cellular components, mouse and human phenotypes, pathways, diseases, literature...) for these genes. ToppGene then yields a complete report that details the entities found in the database that match the set of genes with additional information such as the number of genes matched as well as the Bonferroni p-value of the hit (available both without and with Bonferroni correction). We performed here an empirical selection, only keeping gene sets with convincing Bonferroni p-values ($< 10^{-10}$) and process specificity.

Once gene sets have been enriched, we manually curate them by investigating for

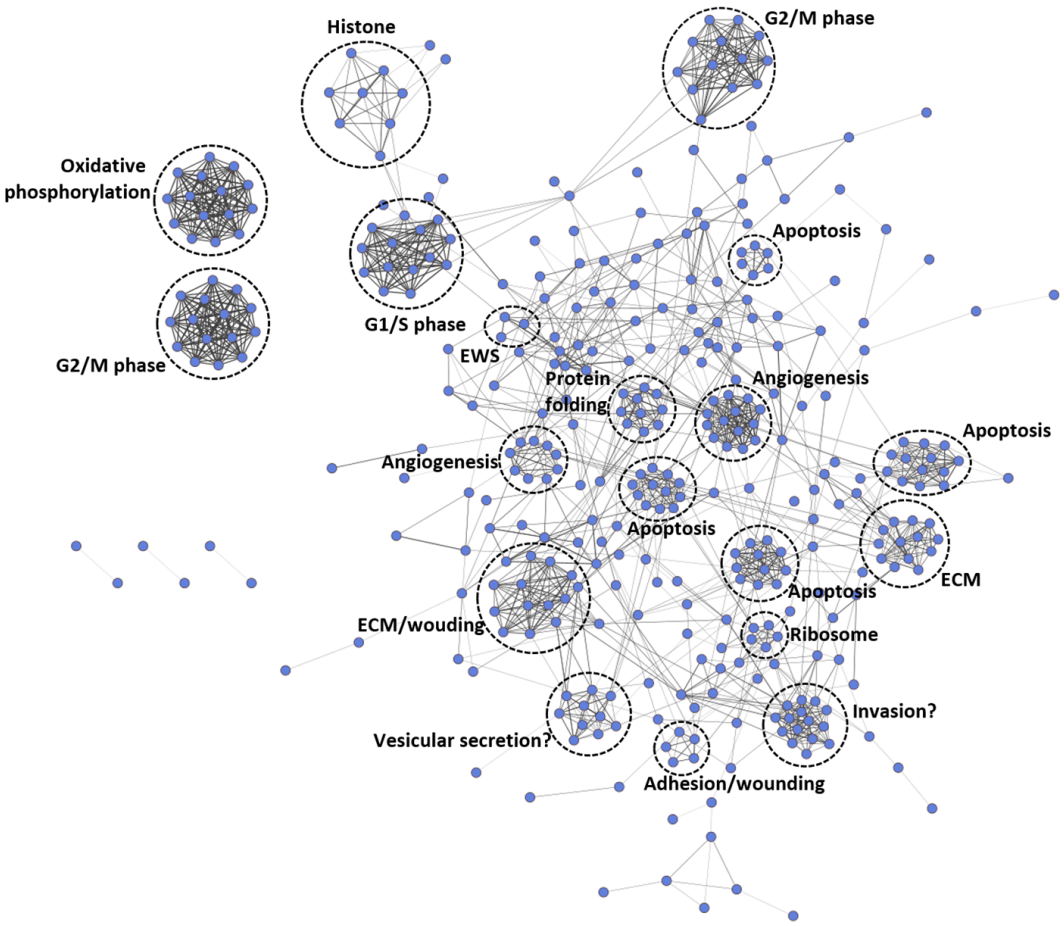

Figure 5.3: **Full graph of Ewing sarcoma independent components computed from the transcriptional data. Each node represents one independent component from a given day of the induction. Pseudo-cliques have been annotated based on their most significant and consensual contributing genes.**

each gene set the different hits, as well as evaluating their relevance. We ended up by keeping 16 pseudo-cliques whose we were able to relate the genes set to a diversity one of several biological processes including EF1 activity, cell cycle, chromatin state, cellular organization, morphogenesis, stress and glucose metabolism. The cliques we chose not to conserve either appeared to contain mixed signals that we were not able to properly interpret. For this reason, we decided not to include them in further analyses. In the next section, we will cover in depth the nature of these 16 signatures.

5.3 New Ewing sarcoma cell processes have been identified

In the past we identified, using another ICA-based approach, a set of gene signatures that we could link to several Ewing sarcoma set processes (Aynaud et al., 2020); we ended up confidently characterizing 7 independent components corresponding to EF1 targets expression, G1/S cell cycle phase, G2/M cell cycle phase, two components related to redox reactions, mRNA splicing and a metabolic signal containing glycolysis and hypoxia factors. We plotted for every pair of signals the x/y plot of all cells (Fig. 5.5), and we can already see that some old signatures have been independently rediscovered (for instance, G1/S and G2/M cell cycle components that have an almost perfect correlation compared to those reported in the Aynaud study). On the other hand, many signals appear to be new, such as the two cell cycle-related signatures "G2/M Exit" and "Histone". In the remainder of this section, we will investigate these signals more closely.

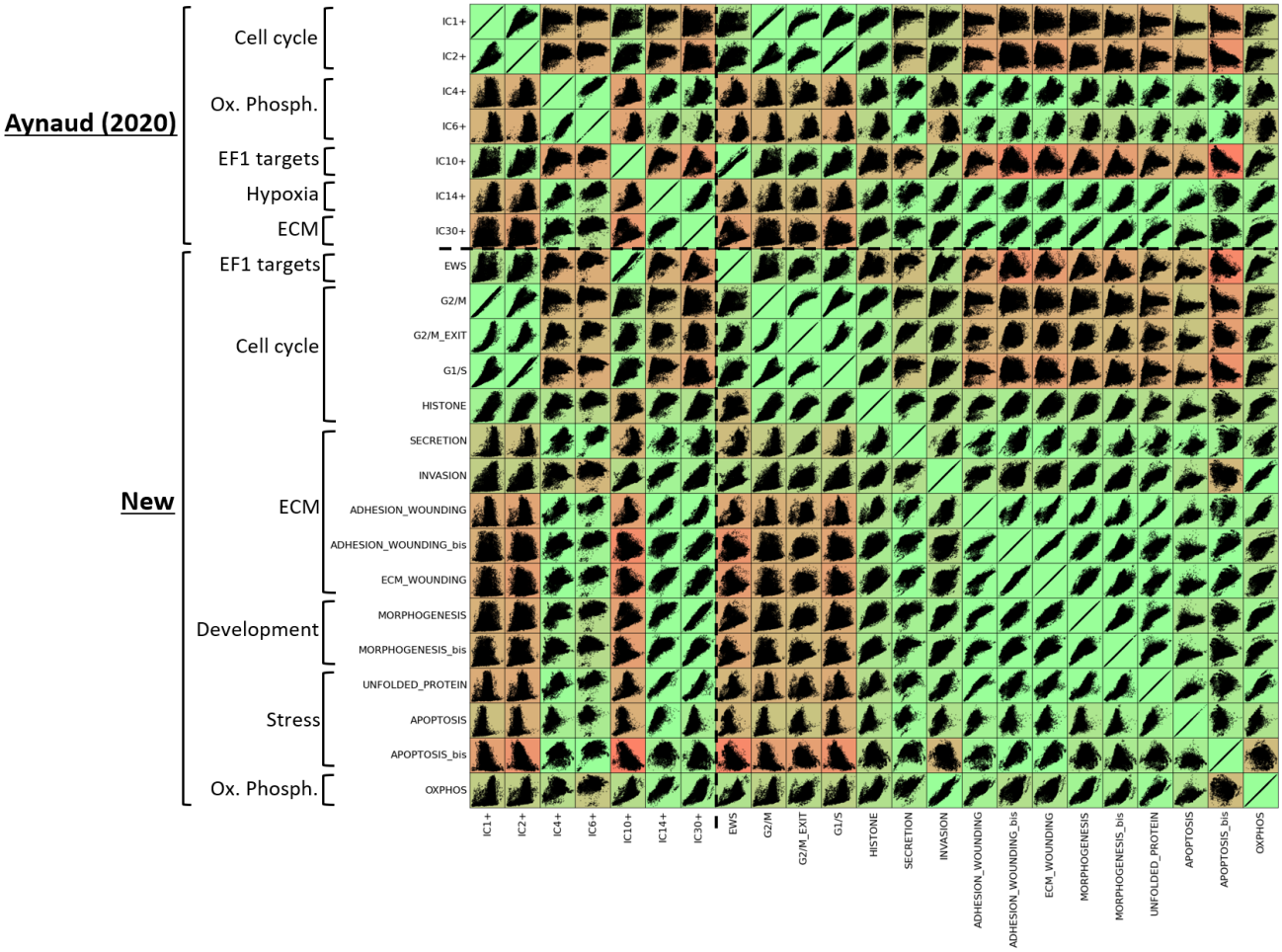

Figure 5.4: **Correlation plots between all pairs of transcriptional signals (old and new). Plot color indicates the correlation coefficient, red for -1 and green for +1.**

5.3.1 New signatures cover various Ewing sarcoma cell processes

We discovered in this study 16 consensus transcriptional signatures from the study of the inducible cell lines datasets, signatures that cover various aspects of the Ewing sarcoma's biology: cell proliferation, metabolism, EF1 targets, stress, development, structure... Complete signatures can be retrieved at the following address: File link

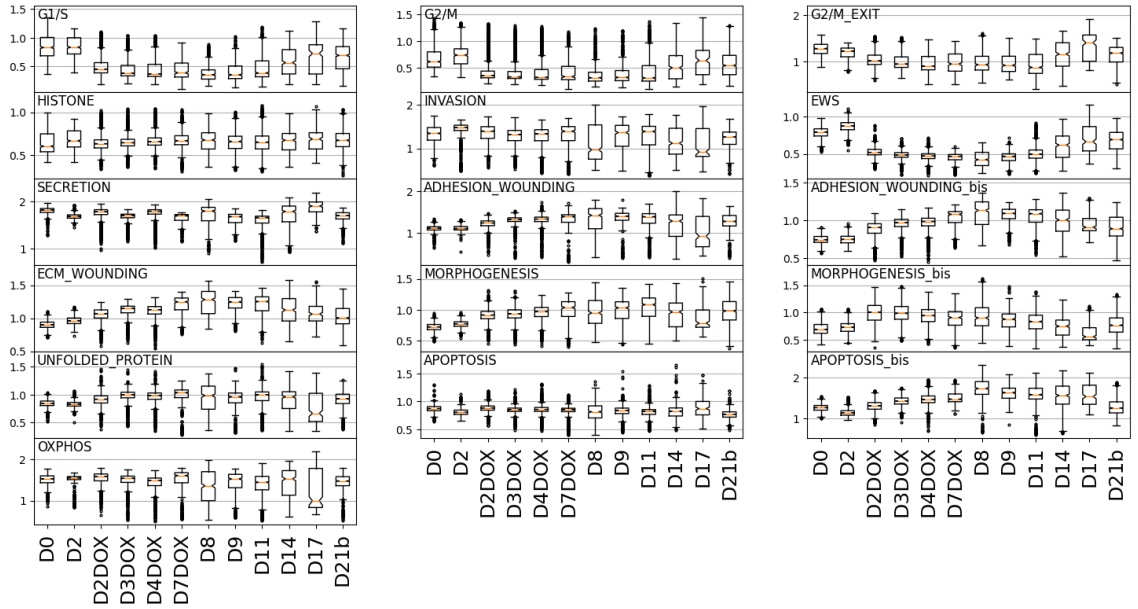

Figure 5.5: **Expression dynamics of the 16 Ewing sarcoma transcriptional signals identified throughout the induction.**

EF1 targets

Let us discuss first the central transcriptional signature, *IC-EF1*, that has been identified in our study and contains many genes whose expression is directly correlated with the expected EF1 levels: genes from this signature tend to be highly expressed in the initial condition, before doxycyclin is added into the medium, they then drop in expression in presence of doxycyclin which inhibits the effect of EF1, and slowly increase once doxycyclin is removed and EF1 levels go up again. This behavior has been validated by analyzing EF1 levels by Western blot, confirming IC-EF1 is high when the EF1 protein is present and vice-versa. Interestingly, the ICs cluster related to this signature only consists of 3 ICs that come from the three meta datasets, *ACTIVATION*, *INHIBITION* and *ALL*.

In particular, 29 genes in this IC-EF1 signatures were already flagged as Ewing sarcoma-related genes according to the database DisGeNET (Piñero et al., 2020): in alphabetical order, ABHD6, ADRB3, CALCB, CAV1, CCND1, CDH11, CEBPB, EPHA3, FAS, FCGRT, GJA1, GLG1, HES1, IGF1, JAK1, LINGO1, MCL1, MMP1, MYC, NKX2-2, NR0B1, PRKCB, RAMP1, SOX2, STEAP1, TCF4, TNNT1, TWIST1, and WT1. We also find 35 genes from the IC-EF1 signatures in common with the prioritized EF1 targets identified in (Aynaud et al., 2020): in alphabetical order, ABHD6, ATP1A1, CADPS2, CAPRIN1, CAV1, CCND1, CDH8, CES1, CLDN1, COL21A1, CTTNBP2, DLGAP1, HES1, HMCN1, IGF1, KCNE4, KDSR, LIPI, LRRC4C, MAN2A1, MYC, NPTXR, NTNG1, PCDH7, PCSK2, PRKCB, RBM11, SLAIN1, SLC26A2, SLCO5A1, TNFAIP6, TRPM4, TSPAN13, UGT3A2, and ZC3H13. These matchings comfort us in flagging this consensus IC as being related to Ewing sarcoma.

This gene set will be of utmost importance in our analyses as it can be used for us as a proxy for EF1 activity, with several advantages: first, it is challenging to read EWSR1-FLI1 mRNAs in a single-cell assay as they cannot be distinguished from the wild-type

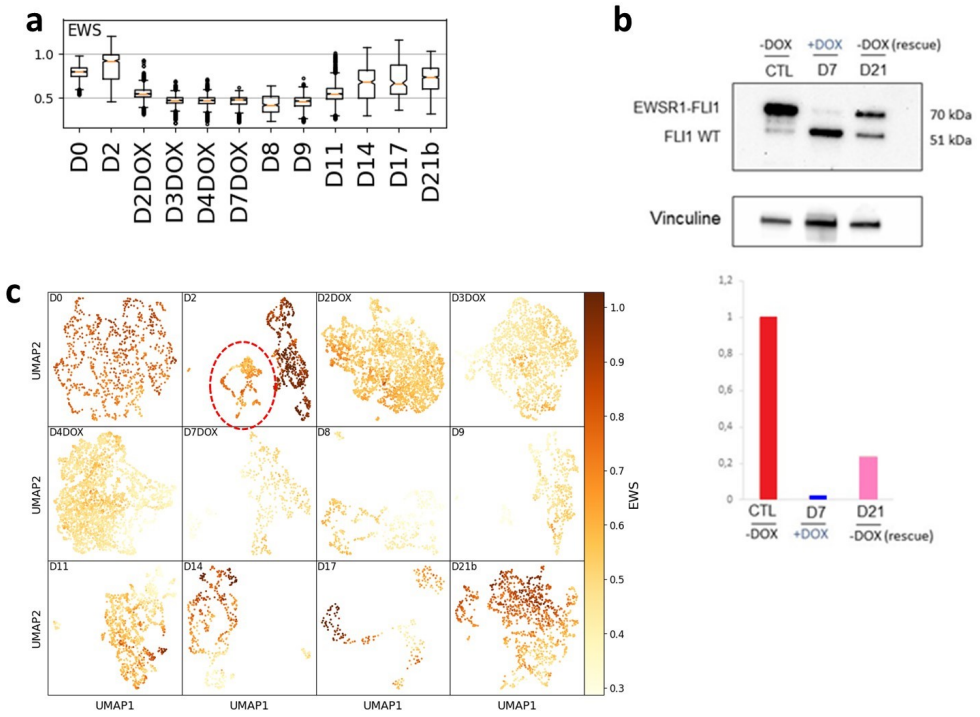

Figure 5.6: **Dynamics of the transcriptional signal identified as EF1-related throughout the induction, *IC-EF1*.** (a) Distribution of the signal among cells throughout the induction. (b) Western blot of the EWSR1-FL1 protein at day 0, day 7 (doxycycline condition) and day 21 (top). Relative quantitative estimation of the EWSR1-FL1 presence according to this Western blot (bottom). (c) UMAP representations of cells from each day independently, colored by IC-EF1; day 2 seems to contain a subpopulation of cells that present lesser expression of IC-EF1.

FLI1 ones. Furthermore, following the expression of a single gene tends to be more noisy than averaging counts over an entire gene signature. For this reason, in the following sections, we will use IC-EF1 as a readily accessible measure approximating the presence of EF1 mRNAs.

As expected, we observed a noticeable decrease in IC-EF1 levels upon introducing doxycycline in the growth medium, which suggests that EWSR1-FLI1 was correctly suppressed (Fig. 5.6a). It is followed by a steady increase of IC-EF1 levels between days 9 and 17 after doxycycline has been removed after day 7. These EWSR1-FLI1 dynamics have been confirmed by Western blot, carried out by Lou Carlier (ENS Paris-Saclay, U830), shown in Fig. 5.6b. Closer day-per-day inspection, notably using the UMAP dimensionality reduction algorithm Becht et al. (2019), allows us to take a closer look into dataset heterogeneity (Fig. 5.6c). If IC-EF1 signals are quite homogeneously high on day 1, we observe a cell subpopulation on day 2 with a lesser IC-EF1 signal that we could not explain. DOX days present noticeably decreased IC-EF1 signal, which we can see progressively reappear starting from day 11 (3 days after removal of doxycycline). Interestingly, day 21 stays heterogeneous, with approximately half of the cells that have recovered a pre-DOX IC-EF1 signal. It is still unclear if EF1-low cells at day 21 would have been able to shift to an EF1-high profile if the experiment had been conducted over a longer period or if they were locked into an EF1-low phenotype. Further investigation will be required to answer this question.

Proliferation signals

Four cell cycle-related Ewing sarcoma transcriptional signatures have been identified in our new study, while we reported only two during the previous project (Aynaud et al., 2020), one related to genes associated with phases G1 and S, and another related to G2 and M-phase genes. We identified with high confidence these two signals within our ICs network, each of which corresponding to a clique containing one IC from each dataset. In addition, we characterize here two new cell cycle-related cell processes: after enrichment, one corresponds to genes involved in the mitosis process, and the other contains a number of histones and other chromatin organization factors.

The two first gene signatures, related to the G1 and S phases for the first one and to the G2 and M phases for the second one, contain many well-known proteins that play a role in the cell cycle machinery or regulation. We can for instance mention PCNA, genes from the ORC family, various cyclin and CDKs, or centromeric proteins. We will investigate later in this manuscript the interplay between IC-EF1 signal and these two cell processes.

We identified a third consensual cell cycle signature associated with G2/M-related genes. Upon closer inspection, many genes in this signature are involved in cell processes such as mitosis, cytoskeleton organization, and chromatin organization. We also noticed a nonlinear positive correlation with the G2/M component. For these reasons, we interpreted this biological signature as cellular processes involved in mitosis resolution and labeled it "G2/M exit". The last cell cycle signature we identified contains many genes involved in chromosome organization such as centromeric proteins, histones and other nucleosome packaging factors such as ASF1B.

These four cell cycle components can be observed throughout the induction in Fig. 5.7. We see in these plots quite typical cell cycle signal profiles, with a clear cell cycle loop during the early days and day 21. Interestingly, we still observe strong cell cycle signals at days 2, 3, and 4 while EF1 is suppressed, and lesser signals at days 7, 8, and 9. We also observe a well-resolved cell cycle loop in the early days, that degrades over days once EF1 is suppressed to the point it is barely visible on day 7. Once doxycycline is removed from the growth medium and EF1 suppression stops, the cell cycle loop reappears progressively until it retrieves its initial shape at day 21. Also, the dataset at day 21 is to take with a grain of salt, as it contains a heterogeneous cellular population mixing EF1-low and EF1-high profiles. This suggests that in the absence of EF1, genes associated with the G1/S

and G2/M phases of the cell cycle are less expressed, which may lead to lower proliferation levels. We also observe a complete recovery of the cell cycle signal in cells that recover the EF1-high phenotype once doxycycline is removed, which suggests the proliferation phenotype could be recovered.

5.3.2 Factors not directly related to the high expression of EF1

Many new transcriptional signals have also been highlighted during this study, notably some connected to non-EF1-high cell phenotypes. We identified in particular clear varying signals during the induction, and we will focus on three categories: stress signals, structural signals, and metabolic signals.

Stress signals

First of all, we identified three stress-associated transcriptional signals (Fig. 5.4, Fig. 5.8): two signatures related to apoptosis, and one related to protein folding and unfolded proteins signals. Interestingly, one apoptosis-related signature (APOPTOSIS BIS) seems to be strongly anticorrelated with the EF1 targets signature (Fig. 5.8b), which suggests this component may correspond to a cellular stress associated with the depletion of EF1 – correlations are best observed at day 21 which contains a mixed population of EF1-low and EF1-high cells. We notably see that this stress component is mostly low at days 0 and 2, then increases as doxycycline is added to the growth medium to peak around day 8, then more and more cells retrieve low levels of this stress signature as the suppression of EF1 is removed. The levels of the two other stress-related signals, APOPTOSIS and UNFOLDED PROTEINS does not seem to be strongly affected by the abundance of EF1.

Structural and invasion signals

We also characterized four consensus independent components containing genes related to cellular structure, adhesion, motility and extracellular matrix configuration (Fig. 5.9). The first signature, INVASION (Fig. 5.9a), contains genes related to tumoral cell invasion such as ZEB1, ZEB2 SNAI1, SNAI2, Twist1 or Notch1, and its expression seems to be correlated with the abundance of EF1. The three other signatures (ADHESION WOUNDING, ADHESION WOUNDING BIS and ECM WOUNDING) (Fig. 5.9b-d) contain many genes related to cell-cell adhesion, extracellular matrix and response to wounding, and appear to be anti-correlated with EF1 targets expression, meaning genes from these signatures tend to be downregulated in the presence of EF1. For these reasons, we believe these four signatures can help monitor Ewing sarcoma cell processes related to EMT, loss of cell adhesion and more generally metastasis.

Morphogenesis and angiogenesis development signals

We could finally confidently associate two consensual components (MORPHOGENESIS and MORPHOGENESIS BIS) to the expression of development genes, and in particular some factors involved in angiogenesis (Fig. 5.10). These two components have very similar patterns, and appear to be highly correlated (Fig. 5.5). Interestingly, genes contained in these two morphogenesis-related signatures appear to be at higher expression levels when EF1 is suppressed.

5.4 Transcriptional programs of Ewing sarcoma: a summary and what comes next

Ewing sarcoma is a complex disease characterized by highly heterogeneous tumors. The usage of inducible Ewing sarcoma cell lines in which the abundance of EF1, the Ewing sarcoma oncogene, can be finely monitored in conjunction with single-cell RNA-seq

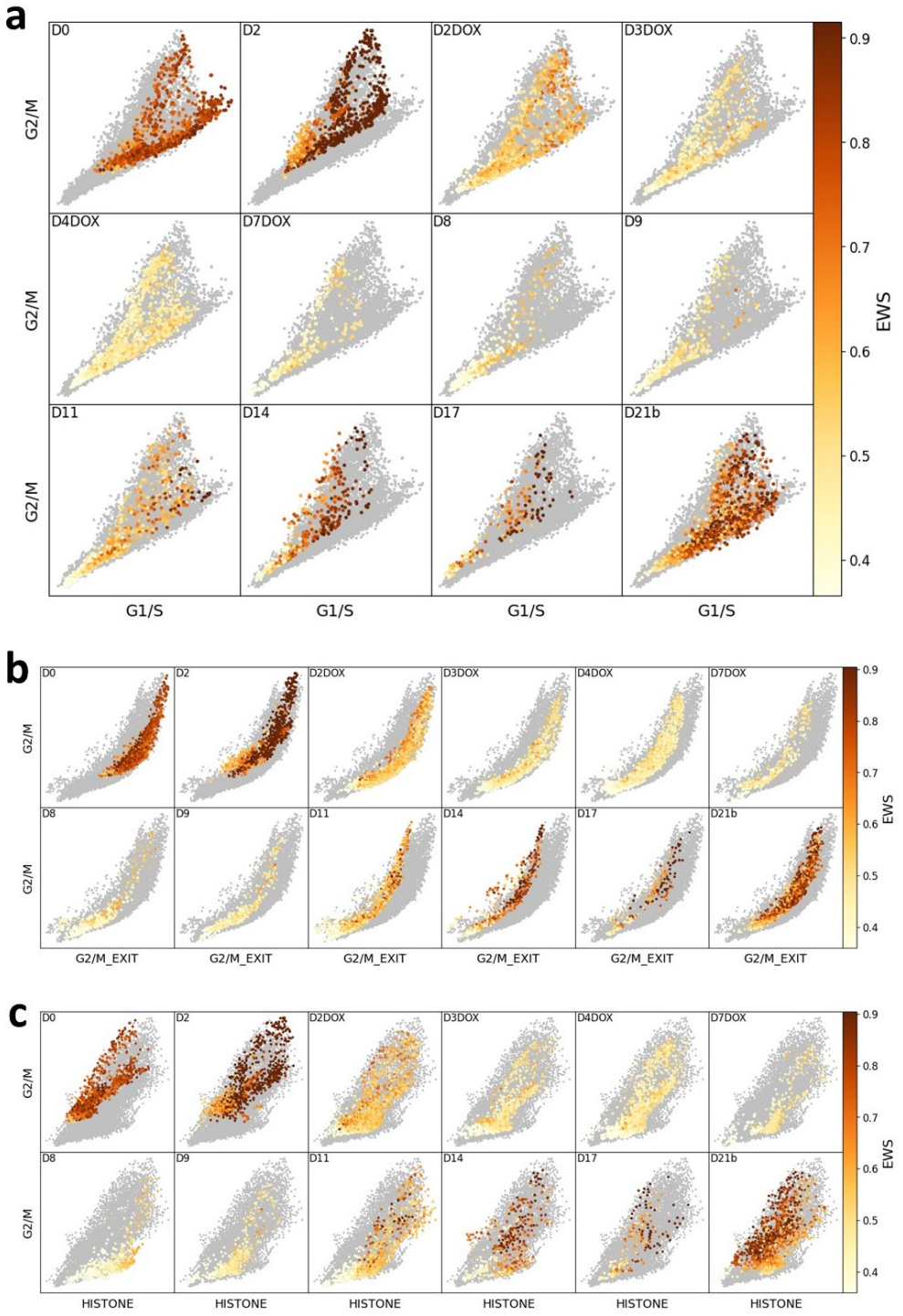

Figure 5.7: **2D Ewing sarcoma cell cycle plots, day by day. Each dot represents a cell, and each cell's coordinates correspond to the average expression of genes contained in a pair of cell cycle signatures. Gray dots represent cells from other days, for reference.** (a) Representing cells in the space of G1/S genes versus G2/M genes makes cell cycle loops appear, especially during the early days. (b) G2/M genes versus "G2/M exit" genes (genes involved in mitosis). (c) G2/M genes versus "histones" signature (this signature also contains other chromatin remodeling factors).

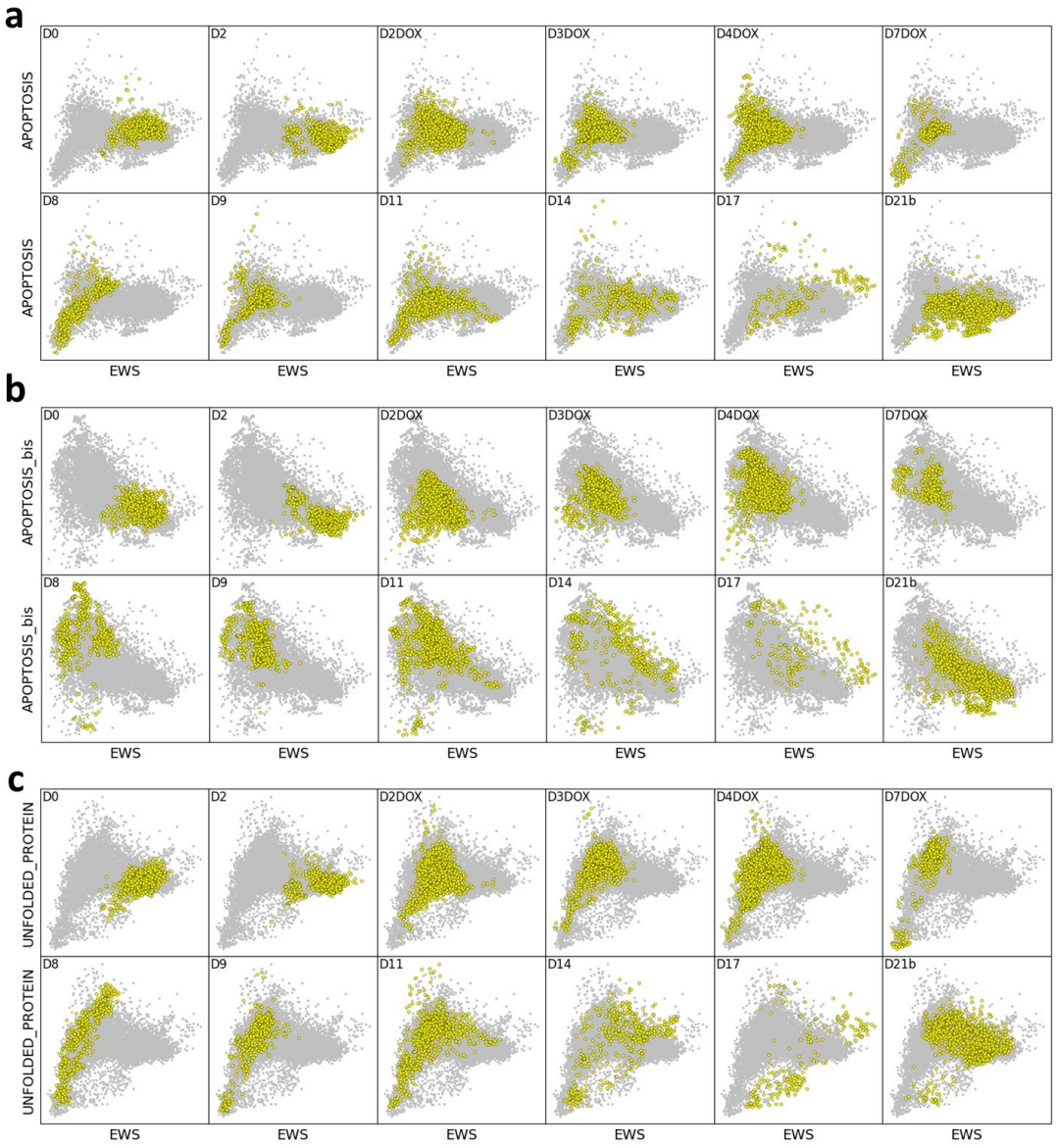

Figure 5.8: **2D Ewing sarcoma gene expression plots, day by day. Each dot represents a cell, the x-axis corresponds to the average expression of the genes contained in the newly identified EF1 targets signature, the y-axis corresponds to the average expression of the genes contained in each stress signature. Gray dots represent cells from other days, for reference.** (a) EF1-independent apoptosis signal. (b) EF1-dependent apoptosis signal. (c) Unfolded proteins signal.

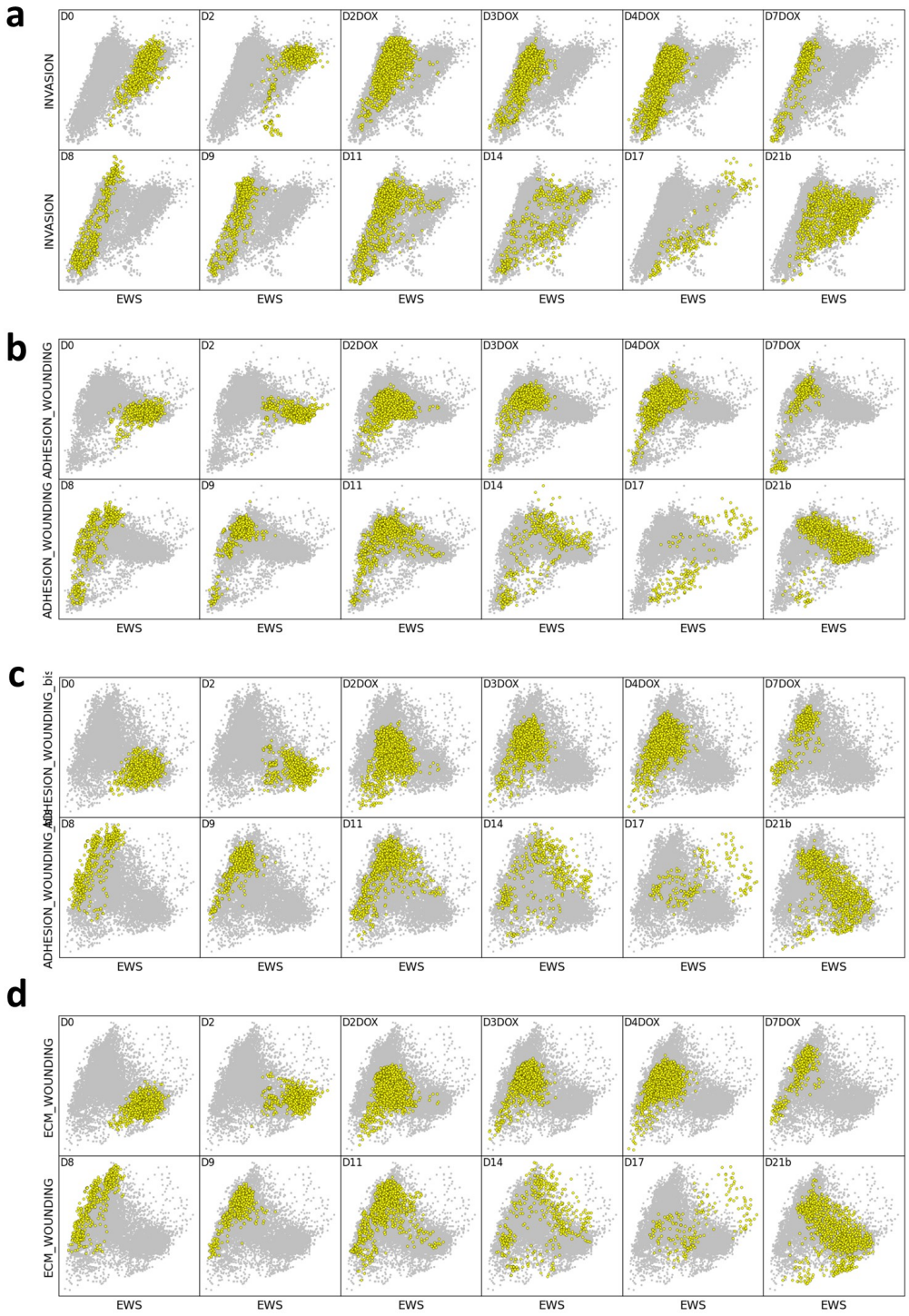

Figure 5.9: **2D Ewing sarcoma gene expression plots, day by day. Each dot represents a cell, the x-axis corresponds to the average expression of the genes contained in the newly identified EF1 targets signature, the y-axis corresponds to the average expression of the genes contained in each structure-related signature. Gray dots represent cells from other days, for reference.** (a) Invasion-like signal. (b) Adhesion and response to wounding genes signal (component 1). (c) Adhesion and response to wounding genes signal (component 2). (d) Extracellular matrix and response to wounding signal.

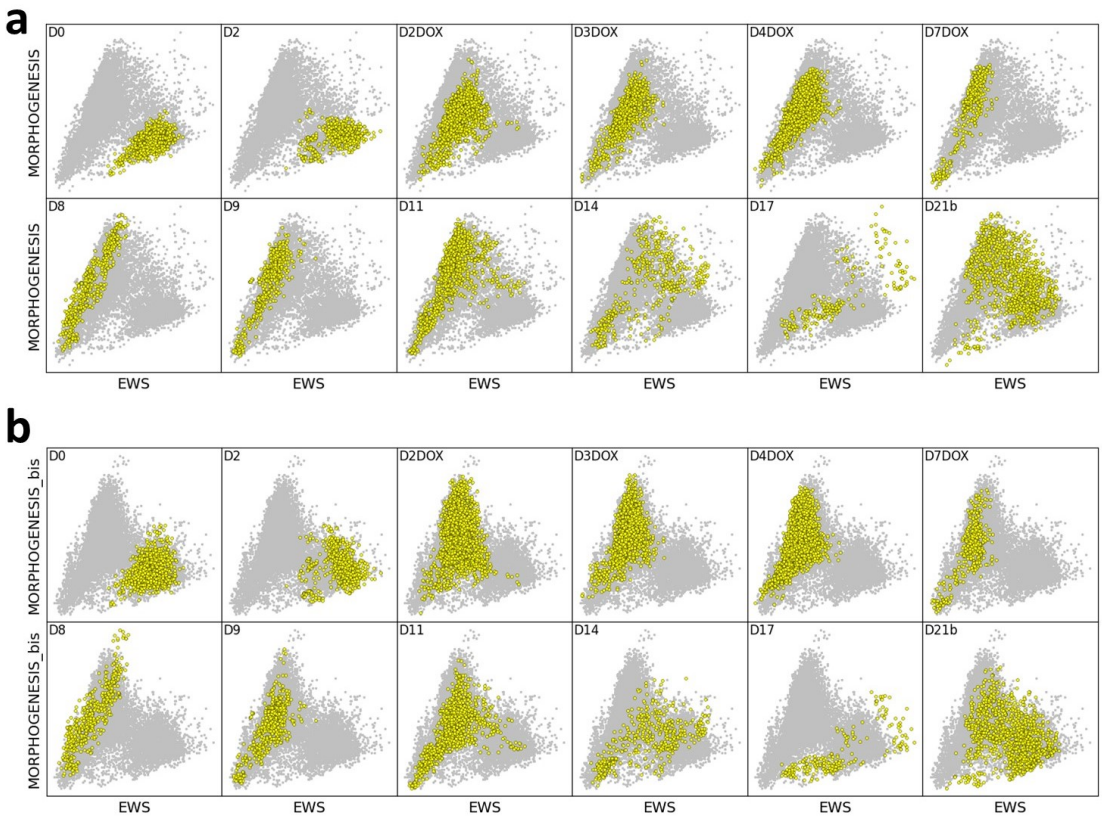

Figure 5.10: **2D Ewing sarcoma gene expression plots, day by day. Each dot represents a cell, the x-axis corresponds to the average expression of the genes contained in the newly identified EF1 targets signature, the y-axis corresponds to the average expression of the genes contained in each development-related signature. Gray dots represent cells from other days, for reference.** (a) First morphogenesis-related signal. (a) Second morphogenesis-related signal.

data analysis allowed us to get an insight into the transcriptional heterogeneity of Ewing sarcoma cells.

We were able to both confirm a part of the results of the previous study reported in (Aynaud et al., 2020), and iterate on them by identifying many new Ewing transcriptional programs, especially those that are not related to the EF1-high condition. Furthermore, by applying consensus ICA as a deconvolution method, we could identify transcriptional signatures shared amongst many datasets, increasing our approach's robustness. First of all, we found in an unsupervised manner a consensual gene set containing a major gene overlap with the IC10 signature reported in (Aynaud et al., 2020) that we could link to the expression of EF1 target genes. We also identified four cell cycle-related components: G1/S genes, G2/M genes, end of G2, and a last component containing histone genes as well as other factors related to chromatin conformation – while only the first two were reported in the previous study; we also discussed how the expression of these cell cycle factors is modulated in the presence of EF1. We then uncovered many components associated to cell processes downregulated when EF1 is abundant, such as apoptosis, cellular stress associated to unfolded proteins, or cell-cell and cell-matrix adhesion factors. Finally, we highlighted one component containing genes involved in tumoral invasion, EMT, and tumor metastasis, that appears to be upregulated in the EF1-high conditions. We believe that identifying these gene signatures could facilitate the tracking of their associated cellular processes, and may improve our capabilities of understanding the tumoral heterogeneity of Ewing sarcomas.

There is also the question of the possible phenotype shift irreversibility after the EF1 suppression. Indeed, we observed that unlike day 0, where most cells expressed high levels of the EF1 signature, a large fraction of cells on day 21 still appear to be EF1-low profiles. This observation raises the following question: Would these EF1-low cells acquire an EF1-high profile if we waited longer, or are these cells stuck within an EF1-low state due to the EF1 depletion? Answering this problem will be a puzzling task, as even if we conducted the experiment for longer, it would have been difficult to distinguish between shifts from EF1-low to EF1-high, and natural selection that would be caused by EF1-high cells that have a competitive advantage due to their more active proliferation. A possibility to explore could be to use molecular tags in order to follow cell lineages, and identify EF1-high cells whose mother cell was EF1-low. In all cases, I think it will be difficult to conclude on this point with the data currently available.

Chapter 6

Discussion and conclusion

In this thesis, we reported our works on three main research axes: single-cell data analysis, development of data integration methods for single-cell data, and study of the cell cycle at the transcriptional level. In this concluding chapter, we will briefly return to these three topics, and discuss our contributions to them as well as future challenges that are still to be overcome.

6.1 The future of data integration

Data integration consists of distinct challenges depending on the anchoring that exists between datasets, and each facet of DI requires distinct tools that leverage various algorithmic strategies. For instance, metric-based methods excel at solving HI tasks, whereas linear matrix analysis methods excel at solving VI tasks. Machine learning paradigms with high abstraction levels, such as manifold alignment methods and deep neural networks, are excellent assets for dealing with DI and MI problems, the latter also performing well at HI and VI tasks. Overall, VI methods are pretty good at solving the task, HI methods are capable of dealing with small to moderate batch effects but still struggle to mitigate significant batch effects such as inter-species data, and DI/MI problems are arguably still unsolved in the general case.

We introduced the *transmorph* framework that articulates computational blocks to conceive HI pipelines in an attempt to unify many types of data integration methods, in particular those that rely on two successive computation steps: a matching step, that identifies similar cells across datasets, and an embedding step that converts this matching into a joint representation. This flexible framework is entirely open source, and can therefore be extended with new algorithms or features in order to increase its expressiveness.

It is essential to note that there are important pitfalls to data integration that must not be overlooked. The primary issue that can be encountered is named *overcorrection* and describes an undesirable event where a data integration method incorrectly aligns cells that do not share the same biological type or state. This typically happens when batch effects are too strong, when a dataset contains specific cell types, when cell type distribution is highly imbalanced, or when there is little anchoring between batches. Overcorrection can be difficult to detect when there is no easy access to cell labels and is a critical issue that hinders every subsequent analysis step. Indeed, it can lead to cells belonging to the same cluster without sharing critical biological properties such as cell type or states. Other issues are worth noting even though they are not exclusive to the data integration task, such as the difficulty in differentiating between true zeros and missing values in RNA-seq datasets or the fact that different modalities are often expressed using different data types (e.g., binary or integer data) which may be difficult to handle jointly within mathematical frameworks. Finally, data integration tools based on abstract machine learning paradigms such as deep autoencoders often come at the cost of a decrease in model interpretability

which is an important downside for any health-related application. However, many efforts are made to overcome this issue (Svensson et al., 2020; Treppner et al., 2022), and we expect to see many more in the years to come.

To conclude, years of algorithmic and computational advances made it possible to solve most HI and VI problems with satisfying performance, with only the most complicated instances still being problematic (e.g., HI of many batches with strong batch effects). Solving DI and MI is the next computational challenge. The most promising approaches that have been developed to tackle it are based on deep learning models, particularly deep autoencoders. It has been shown that purely unsupervised DI may not be a well-posed problem and could suffer fundamental flaws (Xu and McCord, 2022), which greatly incentivizes using knowledge-driven tools that allow the user to include external information to enhance models with functional information that link features across modalities. Finally, apart from developing new tools, there is also an urgent need to enrich the data integration ecosystem with organizing frameworks, standardized benchmarks, datasets, and quality assessment metrics.

6.2 Studying the cell cycle in the gene expression space: future challenges

We studied the notion of the cell cycle in the transcriptional space in depth through these two projects, and attempted to tackle two challenging questions: first, we provided a new way to model the cell cycle progression under the prism of gene expression as a piecewise-linear trajectory in the transcriptional space. We also show how horizontal data integration and notably optimal transport can be leveraged to accurately match cycling cells between two scRNA-seq datasets – either using the *transmorph* framework or with the alternative kernel methodology. I would like to end this chapter by mentioning that there is still a lot to achieve to improve our understanding of the cell cycle from a transcriptomic point view, and by providing a few examples of the current and future challenges in this topic.

First of all, a challenging question discussed in (Chervov and Zinovyev, 2022) is to demonstrate the existence of different cell cycle profiles that correspond to different trajectory geometries in the cell cycle space. In this theory, different trajectory shapes in the cell cycle gene expression space suggest different cell cycle speeds, in particular according to Chervov, at least two profiles can be identified: "normal" cell cycle where cells revolve around the typical triangular trajectory versus "fast" cell cycle, where most cells follow a more linear anticorrelated trajectory (Fig. 6.1). Interestingly, some cell populations even display a mixture of these two profiles, with a fraction of cells following the "standard" cell cycle trajectory and another fraction following the "fast" trajectory. Identifying molecular markers of the "standard" and "fast" cell cycles would give a way to safely deconvolute such datasets by separating the two intertwined trajectories.

Another fundamental question I could not answer is the following: Do all cells whose transcriptome revolves around the cell cycle loop actually proliferate? It may seem counterintuitive to think that cells could express high levels of cell cycle genes even if they do not undergo cell division, but the examination of some datasets makes me think it could be the case. For instance, in chapter 5, we will study Ewing sarcoma datasets in which the oncogene EF1 has been almost completely suppressed, yet we observe many of those cells with high cell cycle gene expression, which is even more puzzling given the fact that we do not observe many dividing cells under the microscope in these conditions. For this reason, the transcriptional cell cycle loop may be a condition *sine qua non* for cell division to occur, but may not be sufficient.

Finally, even though horizontal data integration algorithms now provide sound ways of aligning single-cell datasets in the cell cycle space – notably thanks to optimal transport, some caveats persist in order to fully exploit these algorithms for automatic cell cycle phase annotation. The most prominent one is the accurate delineating of cell cycle phases

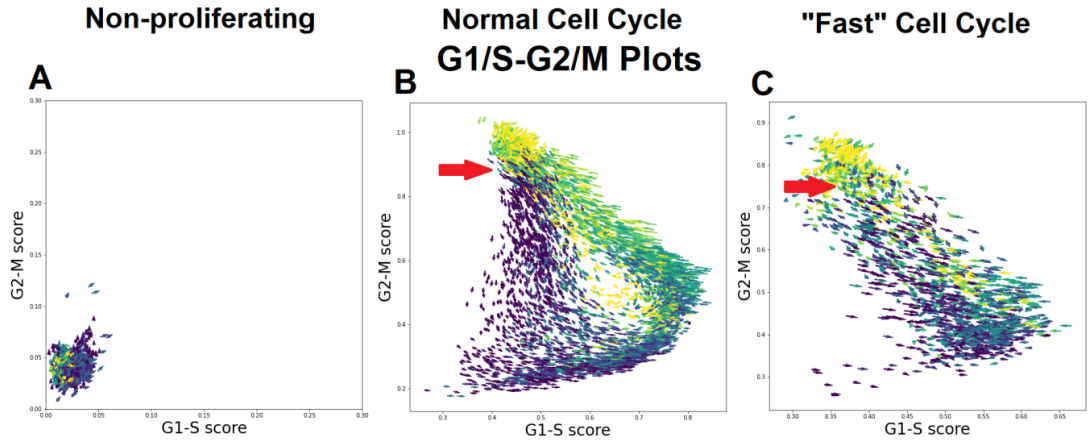

Figure 6.1: **According to Chervov, there exists several trajectory geometries in the cell cycle gene expression space that correspond to different proliferation speeds. Cells are colored by number of counts, red arrows indicate to the point of mitosis.** (a) Non-proliferating cell populations. (b) Proliferating cell populations. (c) Fast proliferating cell populations. *Taken from (Chervov and Zinovyev, 2022).*

along the cell cycle trajectories, in particular the junction between the G1 phase and S phases and the junction between the G2 and M phases. Indeed, these two pairs of phases appear to be close in terms of expressed genes as a major function of G1 (respectively G2) is the synthesis of proteins involved in the S (respectively M) phase. This makes accurate automatic labeling challenging, as the boundaries within these two pairs of phases appear to be fuzzy.

I would like to close this section by stating that I do not expect scRNA-seq to be informative enough to answer these three important problems confidently. In my opinion, information contained within other modalities, such as proteomics, chromatin conformation, and microscopy, will be key to better understanding the cell cycle at the molecular level. Unfortunately, multimodal data acquisition and integration is still challenging and costly today, but I am optimistic that the next few years will provide us with new exciting insights into the cell cycle molecular machinery, and that multimodal approaches will help us get a full picture of this complex system.

6.3 Single-cell data is a powerful asset in computational biology to decipher complex heterogeneous cell processes

Many complex biological systems, such as tumors, the brain, or the immune system, are highly heterogeneous in terms of cell types and states. Deciphering the biological processes happening in such systems is very enticing, but also highly challenging because of the very high heterogeneity of cells, as well as the high number of interactions happening between them. For this reason, single-cell assays have proven to be invaluable in analyzing such biological systems. In my projects, I focused on studying single-cell RNA-seq data, as a cell's transcriptome provides relevant insight into many processes undergone or about to be undergone by the cell. Nonetheless, it is important to remember that RNA-seq is just one biological modality among others, and that it is difficult to get the full picture of a cell's biology by just looking at its genetic expression. For this reason, studying complex biological systems can be facilitated by using other biological modalities, at the single-cell level or not, such as chromatin conformation, proteomics, genomics, or spatial transcriptomics. This also brings the need for not only horizontal data integration methods, but also vertical and diagonal integration algorithms that can combine the information present

throughout the different biological modalities, in order to get a better understanding of the underlying biological processes and cell-cell interactions.

We also showed the importance of using single-cell data when studying Ewing sarcoma: all the chapter 5 could not have been written if we had used bulk RNA-seq instead of single-cell RNA-seq assays. Even though differences between cells are more subtle than in other biological systems such as the immune system, where the gene expression is so different between cell types that clear clusters are identifiable in the transcriptional space, differences between transcriptional programs in Ewing sarcoma cells still exist, and are very relevant to understand the cell processes they carry out. For instance, we would not have been able to understand the heterogeneity of Ewing sarcoma cells at the end of the induction without having access to single-cell data: both bulk RNA-seq analyses and other assays, such as Western blot yielded an average profile between EF1-low and EF1-high cells, that finally did not correspond to any of the two subpopulations. Here, we were able to conclude on the coexistence of two Ewing sarcoma cell subpopulations, one composed of cells that have fully recovered their initial malignant state and cell processes, and another with cells that have not. This observation yields relevant biological questions, such as the possible irreversibility of the malignant to non-malignant phenotype shift upon depletion of EF1, and may lead to future studies.

For this reason, we would advocate for using single-cell assays when studying Ewing sarcoma cells whenever possible, despite the additional time, effort, and cost it takes to carry out such experiments. Single-cell assays not only provide finer insights into the biology of cells contained in a biological system, but also help avoid formulating erroneous conclusions and hypotheses when studying such heterogeneous cell populations. We are very excited to see the future of these projects leveraging the analysis of Ewing sarcoma inducible cell lines at the single-cell level, notably by exploring not only the RNA-seq modality but also others such as chromatin accessibility and proteomics. We also think that unsupervised deconvolution methods, such as ICA or NMF, will continue to prove useful to deconvolute gene expression data in tumors and other data modalities, possibly in other biological systems.

Published prior to the Ph.D.

[Aynaud1] Marie-Ming Aynaud, Olivier Mirabeau, Nadege Gruel, Sandrine Grossetête, Valentina Boeva, Simon Durand, Didier Surdez, Olivier Saulnier, Sakina Zaïdi, Svetlana Gribkova, et al. Transcriptional programs define intratumoral heterogeneity of ewing sarcoma at single-cell resolution. *Cell reports*, 30(6):1767–1779, 2020.

Publications

[Personal1] Aziz Fouché and Andrei Zinovyev. Unsupervised weights selection for optimal transport based dataset integration. *bioRxiv*, pages 2021–05, 2021.

[Personal2] Aziz Fouché and Andrei Zinovyev. Omics data integration in computational biology viewed through the prism of machine learning paradigms. *Frontiers in Bioinformatics*, 3, 2023.

[Personal3] Aziz Fouché, Loïc Chadoutaud, Olivier Delattre, and Andrei Zinovyev. Transmorph: a unifying computational framework for modular single-cell rna-seq data integration. *NAR Genomics and Bioinformatics*, 5(3):lqad069, 2023.

[Personal4] Evgeny M Mirkes, Jonathan Bac, Aziz Fouché, Sergey V Stasenko, Andrei Zinovyev, and Alexander N Gorban. Domain adaptation principal component analysis: base linear method for learning with out-of-distribution data. *Entropy*, 25(1):33, 2022.

[Personal5] Andrei Zinovyev, Michail Sadovsky, Laurence Calzone, Aziz Fouché, Clarice S Groeneveld, Alexander Chervov, Emmanuel Barillot, and Alexander N Gorban. Modeling progression of single cell populations through the cell cycle as a sequence of switches. *Frontiers in Molecular Biosciences*, 8:1340, 2022.

Bibliography

Tamim Abdelaal, Soufiane Mourragui, Ahmed Mahfouz, and Marcel JT Reinders. Spage: spatial gene enhancement using scrna-seq. *Nucleic acids research*, 48(18):e107–e107, 2020.

ACS. Types of cancer that develop in children, 2023. URL https://www.cancer.org/cancer/cancer-in-children/types-of-childhood-cancers.html.

Andrew C Adey. Integration of single-cell genomics datasets. *Cell*, 177(7):1677–1679, 2019.

Ruedi Aebersold and Matthias Mann. Mass spectrometry-based proteomics. *Nature*, 422(6928): 198–207, 2003.

Akshay Agrawal, Alnur Ali, Stephen Boyd, et al. Minimum-distortion embedding. *Foundations and Trends® in Machine Learning*, 14(3):211–378, 2021.

Luca Albergante, Jonathan Bac, and Andrei Zinovyev. Estimating the effective dimension of large biological datasets using Fisher separability analysis. In *Proceedings of the International Joint Conference on Neural Networks*, volume 2019-July, 2019. ISBN 9781728119854. doi: 10.1109/IJCNN.2019.8852450.

Luca Albergante, Evgeny Mirkes, Jonathan Bac, Huidong Chen, Alexis Martin, Louis Faure, Emmanuel Barillot, Luca Pinello, Alexander Gorban, and Andrei Zinovyev. Robust and scalable learning of complex intrinsic dataset geometry via ElPiGraph. *Entropy*, 22(3), 2020. ISSN 10994300. doi: 10.3390/e22030296.

Joseph S. Alper. The Gibbs Phase Rule Revisited: Interrelationships between Components and Phases. *Journal of Chemical Education*, 76(11):1567–1569, 1999. ISSN 00219584. doi: 10.1021/ed076p1567. URL https://pubs.acs.org/doi/abs/10.1021/ed076p1567.

Atilla Altinok, Francis Lévi, and Albert Goldbeter. A cell cycle automaton model for probing circadian patterns of anticancer drug delivery. *undefined*, 59(9-10):1036–1053, aug 2007. ISSN 0169409X. doi: 10.1016/J.ADDR.2006.09.022.

Jason Altschuler, Jonathan Niles-Weed, and Philippe Rigollet. Near-linear time approximation algorithms for optimal transport via sinkhorn iteration. *Advances in neural information processing systems*, 30, 2017.

Matthew Amodio and Smita Krishnaswamy. MAGAN: Aligning biological manifolds. *arXiv preprint arXiv:1803.00385*, 2018.

Ali Anaissi, Seid Miad Zandavi, Basem Suleiman, Widad Alyassine, Ali Braytee, and Fatemeh Vafaee. A benchmark of pre-processing effect on single cell rna sequencing integration methods. *ResearchSquare*, 2022.

Simon Anders, Paul Theodor Pyl, and Wolfgang Huber. Htseq—a python framework to work with high-throughput sequencing data. *bioinformatics*, 31(2):166–169, 2015.

Ilias Angelidis, Lukas M Simon, Isis E Fernandez, Maximilian Strunz, Christoph H Mayr, Flavia R Greiffo, George Tsitsiridis, Meshal Ansari, Elisabeth Graf, Tim-Matthias Strom, et al. An atlas of the aging lung mapped by single cell transcriptomics and deep tissue proteomics. *Nature communications*, 10(1):1–17, 2019.

Christof Angermueller, Stephen J Clark, Heather J Lee, Iain C Macaulay, Mabel J Teng, Tim Xiaoming Hu, Felix Krueger, Sébastien A Smallwood, Chris P Ponting, Thierry Voet, et al. Parallel single-cell sequencing links transcriptional and epigenetic heterogeneity. *Nature methods*, 13(3): 229–232, 2016.

Ricard Argelaguet, Britta Velten, Damien Arnol, Sascha Dietrich, Thorsten Zenz, John C Marioni, Florian Buettner, Wolfgang Huber, and Oliver Stegle. Multi-omics factor analysis—a framework for unsupervised integration of multi-omics data sets. *Molecular systems biology*, 14(6):e8124, 2018.

Ricard Argelaguet, Damien Arnol, Danila Bredikhin, Yonatan Deloro, Britta Velten, John C. Marioni, and Oliver Stegle. MOFA+: a statistical framework for comprehensive integration of multi-modal single-cell data. *Genome Biology*, 21(1):111, May 2020. ISSN 1474-760X. doi: 10.1186/s13059-020-02015-1. URL https://doi.org/10.1186/s13059-020-02015-1.

Ricard Argelaguet, Anna SE Cuomo, Oliver Stegle, and John C Marioni. Computational principles and challenges in single-cell data integration. *Nature biotechnology*, 39(10):1202–1215, 2021.

Tal Ashuach, Mariano I Gabitto, Michael I Jordan, and Nir Yosef. Multivi: deep generative model for the integration of multi-modal data. *bioRxiv*, pages 2021–08, 2021.

Elham Azizi, Ambrose J. Carr, George Plitas, Andrew E. Cornish, Catherine Konopacki, Sandhya Prabhakaran, Juozas Nainys, Kenmin Wu, Vaidotas Kiseliovas, Manu Setty, Kristy Choi, Rachel M. Fromme, Phuong Dao, Peter T. McKenney, Ruby C. Wasti, Krishna Kadaveru, Linas Mazutis, Alexander Y. Rudensky, and Dana Pe'er. Single-Cell Map of Diverse Immune Phenotypes in the Breast Tumor Microenvironment. *Cell*, 174(5):1293–1308.e36, aug 2018. ISSN 10974172. doi: 10.1016/j.cell.2018.05.060. URL https://pubmed.ncbi.nlm.nih.gov/29961579/.

Jonathan Bac and Andrei Zinovyev. Lizard Brain: Tackling Locally Low-Dimensional Yet Globally Complex Organization of Multi-Dimensional Datasets. *Frontiers in Neurorobotics*, 13, jan 2020. ISSN 1662-5218. doi: 10.3389/fnbot.2019.00110. URL https://www.frontiersin.org/article/10.3389/fnbot.2019.00110/full.

Jonathan Bac, Evgeny M Mirkes, Alexander N Gorban, Ivan Tyukin, and Andrei Zinovyev. Scikit-dimension: a python package for intrinsic dimension estimation. *Entropy*, 23(10):1368, 2021.

Amos Bairoch. The Cellosaurus, a Cell-Line Knowledge Resource. *J Biomol Tech*, 25(2):25–38, nov 2018. doi: 10.1038/s41588-020-00726-6. URL https://pubs.acs.org/doi/abs/10.1038/s41588-020-00726-6.

Nikolas Barkas, Viktor Petukhov, Daria Nikolaeva, Yaroslav Lozinsky, Samuel Demharter, Konstantin Khodosevich, and Peter V Kharchenko. Joint analysis of heterogeneous single-cell rna-seq dataset collections. *Nature methods*, 16(8):695–698, 2019.

Elnaz Barshan, Ali Ghodsi, Zohreh Azimifar, and Mansoor Zolghadri Jahromi. Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds. *Pattern Recognition*, 44(7):1357–1371, 2011.

E Bauer. *Theoretical Biology*. Budapest: Akademiai Kiado, 1935.

Etienne Becht, Leland McInnes, John Healy, Charles-Antoine Dutertre, Immanuel WH Kwok, Lai Guan Ng, Florent Ginhoux, and Evan W Newell. Dimensionality reduction for visualizing single-cell data using umap. *Nature biotechnology*, 37(1):38–44, 2019.

Riccardo Bellazzi, Andrea Codegoni, Stefano Gualandi, Giovanna Nicora, and Eleonora Vercesi. The gene mover's distance: Single-cell similarity via optimal transport. *arXiv preprint arXiv:2102.01218*, 2021.

Jean-David Benamou. Numerical resolution of an "unbalanced" mass transport problem. *ESAIM: Mathematical Modelling and Numerical Analysis*, 37(5):851–868, 2003.

David Bernard, Odile Mondesert, Aurélie Gomes, Yves Duthen, Valérie Lobjois, Sylvain Cussat-Blanc, and Bernard Ducommun. A checkpoint-oriented cell cycle simulation model. *Cell Cycle*, 18(8):795–808, apr 2019. ISSN 15514005. doi: 10.1080/15384101.2019.1591125/SUPPL_FILE/KCCY_A_1591125_SM1721.ZIP. URL https://www.tandfonline.com/doi/abs/10.1080/15384101.2019.1591125.

Tommaso Biancalani, Gabriele Scalia, Lorenzo Buffoni, Raghav Avasthi, Ziqing Lu, Aman Sanger, Neriman Tokcan, Charles R Vanderburg, Åsa Segerstolpe, Meng Zhang, et al. Deep learning and alignment of spatially resolved single-cell transcriptomes with tangram. *Nature methods*, 18 (11):1352–1362, 2021.

Arnaud Blomme, Gaetan Van Simaeys, Gilles Doumont, Brunella Costanza, Justine Bellier, Yukihiro Otaka, Félicie Sherer, Pierre Lovinfosse, Sébastien Boutry, Ana Perez Palacios, et al. Murine stroma adopts a human-like metabolic phenotype in the pdx model of colorectal cancer and liver metastases. *Oncogene*, 37(9):1237–1250, 2018.

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008 (10):P10008, 2008.

Immanuel M Bomze. On standard quadratic optimization problems. *Journal of Global Optimization*, 13(4):369–387, 1998.

Jean-François Brasme, Martin Chalumeau, Odile Oberlin, Dominique Valteau-Couanet, and Nathalie Gaspar. Time to diagnosis of ewing tumors in children and adolescents is not associated with metastasis or survival: a prospective multicenter study of 436 patients. *Journal of Clinical Oncology*, 32(18):1935–1940, 2014.

Nicolas L. Bray, Harold Pimentel, Páll Melsted, and Lior Pachter. Near-optimal probabilistic RNA-seq quantification. *Nature Biotechnology*, 34(5):525–527, may 2016. ISSN 15461696. doi: 10.1038/nbt.3519. URL http://www.nature.com/.

Jean-Philippe Brunet, Pablo Tamayo, Todd R Golub, and Jill P Mesirov. Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the national academy of sciences*, 101(12):4164–4169, 2004.

Jason D Buenrostro, Beijing Wu, Howard Y Chang, and William J Greenleaf. Atac-seq: a method for assaying chromatin accessibility genome-wide. *Current protocols in molecular biology*, 109 (1):21–29, 2015a.

Jason D Buenrostro, Beijing Wu, Ulrike M Litzenburger, Dave Ruff, Michael L Gonzales, Michael P Snyder, Howard Y Chang, and William J Greenleaf. Single-cell chromatin accessibility reveals principles of regulatory variation. *Nature*, 523(7561):486–490, 2015b.

Andrew Butler, Paul Hoffman, Peter Smibert, Efthymia Papalexi, and Rahul Satija. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature biotechnology*, 36(5):411–420, 2018.

Jordi Camps, Floriane Noël, Robin Liechti, Lucile Massenet-Regad, Sidwell Rigade, Lou Götz, Caroline Hoffmann, Elise Amblard, Melissa Saichi, Mahmoud M Ibrahim, et al. Meta-analysis of human cancer single-cell rna-seq datasets using the immucan database. *Cancer Research*, 83 (3):363–373, 2023.

Laura Cantini, Ulykbek Kairov, Aurélien de Reyniès, Emmanuel Barillot, François Radvanyi, and Andrei Zinovyev. Assessing reproducibility of matrix factorization methods in independent transcriptomes. *Bioinformatics*, 35(21):4307–4313, 04 2019. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz225. URL https://doi.org/10.1093/bioinformatics/btz225.

Laura Cantini, Pooya Zakeri, Celine Hernandez, Aurelien Naldi, Denis Thieffry, Elisabeth Remy, and Anaïs Baudot. Benchmarking joint multi-omics dimensionality reduction approaches for the study of cancer. *Nature communications*, 12(1):124, 2021.

Kai Cao, Xiangqi Bai, Yiguang Hong, and Lin Wan. Unsupervised topological alignment for single-cell multi-omics integration. *Bioinformatics*, 36(Supplement_1):i48–i56, 2020a.

Kai Cao, Yiguang Hong, and Lin Wan. Manifold alignment for heterogeneous single-cell multi-omics data integration using pamona. *bioRxiv*, 2020b.

Kai Cao, Qiyu Gong, Yiguang Hong, and Lin Wan. A unified computational framework for single-cell data integration with optimal transport. *Nature Communications*, 13(1):7419, 2022a.

Kai Cao, Yiguang Hong, and Lin Wan. Manifold alignment for heterogeneous single-cell multi-omics data integration using pamona. *Bioinformatics*, 38(1):211–219, 2022b.

Zhi-Jie Cao and Ge Gao. Multi-omics single-cell data integration and regulatory inference with graph-linked embedding. *Nature Biotechnology*, 40(10):1458–1466, 2022.

Nicolas Captier, Jane Merlevede, Askhat Molkenov, Ainur Seisenova, Altynbek Zhubanchaliyev, Petr V Nazarov, Emmanuel Barillot, Ulykbek Kairov, and Andrei Zinovyev. BIODICA: a computational environment for Independent Component Analysis of omics data. *Bioinformatics*, 38(10):2963–2964, 04 2022. ISSN 1367-4803. doi: 10.1093/bioinformatics/btac204. URL https://doi.org/10.1093/bioinformatics/btac204.

Huidong Chen, Luca Albergante, Jonathan Y Hsu, Caleb A Lareau, Giosuè Lo Bosco, Jihong Guan, Shuigeng Zhou, Alexander N Gorban, Daniel E Bauer, Martin J Aryee, et al. Single-cell trajectories reconstruction, exploration and mapping of omics data with stream. *Nature communications*, 10(1):1903, 2019a.

Jing Chen, Eric E Bardes, Bruce J Aronow, and Anil G Jegga. Toppgene suite for gene list enrichment analysis and candidate gene prioritization. *Nucleic acids research*, 37(suppl_2): W305–W311, 2009.

Katherine C. Chen, Laurence Calzone, Attila Csikasz-Nagy, Frederick R. Cross, Bela Novak, and John J. Tyson. Integrative analysis of cell cycle control in budding yeast. *Molecular Biology of the Cell*, 15(8):3841–3862, aug 2004a. ISSN 10591524. doi: 10.1091/mbc.E03-11-0794. URL www.molbiolcell.org/cgi/doi/10.1091/mbc.E03-11-0794.

Katherine C Chen, Laurence Calzone, Attila Csikasz-Nagy, Frederick R Cross, Bela Novak, and John J Tyson. Integrative analysis of cell cycle control in budding yeast. *Molecular biology of the cell*, 15(8):3841–3862, 2004b.

Song Chen, Blue B Lake, and Kun Zhang. High-throughput sequencing of the transcriptome and chromatin accessibility in the same cell. *Nature biotechnology*, 37(12):1452–1457, 2019b.

Yu-Pei Chen, Jian-Hua Yin, Wen-Fei Li, Han-Jie Li, Dong-Ping Chen, Cui-Juan Zhang, Jia-Wei Lv, Ya-Qin Wang, Xiao-Min Li, Jun-Yan Li, et al. Single-cell transcriptomics reveals regulators underlying immune cell diversity and immune subtypes associated with prognosis in nasopharyngeal carcinoma. *Cell research*, 30(11):1024–1042, 2020.

Lih Feng Cheow, Elise T Courtois, Yuliana Tan, Ramya Viswanathan, Qiaorui Xing, Rui Zhen Tan, Daniel SW Tan, Paul Robson, Yuin-Han Loh, Stephen R Quake, et al. Single-cell multimodal profiling reveals cellular epigenetic heterogeneity. *Nature methods*, 13(10):833–836, 2016.

Alexander Chervov and Andrei Zinovyev. Computational challenges of cell cycle analysis using single cell transcriptomics. *arXiv preprint arXiv:2208.05229*, 2022.

Ana Conesa and Stephan Beck. Making multi-omics data accessible to researchers. *Scientific data*, 6(1):251, 2019.

National Research Council et al. *Mapping and sequencing the human genome*. National Academies Press, 1988.

Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

Francis H Crick. On protein synthesis. In *Symp Soc Exp Biol*, volume 12, page 8, 1958.

Anna SE Cuomo, Daniel D Seaton, Davis J McCarthy, Iker Martinez, Marc Jan Bonder, Jose Garcia-Bernardo, Shradha Amatya, Pedro Madrigal, Abigail Isaacson, Florian Buettner, et al. Single-cell rna-sequencing of differentiating ips cells reveals dynamic genetic effects on gene expression. *Nature communications*, 11(1):810, 2020.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26:2292–2300, 2013.

Olivier Delattre, Jessica Zucman, Béatrice Plougastel, Chantal Desmaze, Thomas Melot, Martine Peter, Heinrich Kovar, Isabelle Joubert, Pieter De Jong, Guy Rouleau, et al. Gene fusion with an ets dna-binding domain caused by chromosome translocation in human tumours. *Nature*, 359 (6391):162–165, 1992.

Pinar Demetci, Rebecca Santorella, Bjorn Sandstede, William Stafford Noble, and Ritambhara Singh. Gromov-Wasserstein optimal transport to align single-cell multi-omics data. *BioRxiv*, 2020.

Pinar Demetci, Rebecca Santorella, Björn Sandstede, William Stafford Noble, and Ritambhara Singh. Scot: Single-cell multi-omics alignment with optimal transport. *Journal of Computational Biology*, 29(1):3–18, 2022.

Dávid Deritei, Jordan Rozum, Erzsébet Ravasz Regan, and Réka Albert. A feedback loop of conditionally stable circuits drives the cell cycle from checkpoint to checkpoint. *Scientific Reports 2019 9:1*, 9(1):1–19, nov 2019. ISSN 2045-2322. doi: 10.1038/s41598-019-52725-1. URL https://www.nature.com/articles/s41598-019-52725-1.

Daniel Dominguez, Yi Hsuan Tsai, Nicholas Gomez, Deepak Kumar Jha, Ian Davis, and Zefeng Wang. A high-resolution transcriptome map of cell cycle reveals novel connections between periodic genes and cancer. *Cell Research*, 26(8):946–962, aug 2016. ISSN 17487838. doi: 10.1038/cr.2016.84. URL www.cell-research.com.

Wei Dong, Charikar Moses, and Kai Li. Efficient k-nearest neighbor graph construction for generic similarity measures. In *Proceedings of the 20th international conference on World wide web*, pages 577–586, 2011.

Jinzhuang Dou, Shaoheng Liang, Vakul Mohanty, Xuesen Cheng, Sangbae Kim, Jongsu Choi, Yumei Li, Katayoun Rezvani, Rui Chen, and Ken Chen. Unbiased integration of single cell multi-omics data. *biorxiv*, pages 2020–12, 2020.

Zhana Duren, Xi Chen, Mahdi Zamanighomi, Wanwen Zeng, Ansuman T Satpathy, Howard Y Chang, Yong Wang, and Wing Hung Wong. Integrative analysis of single-cell genomics data by coupled nonnegative matrix factorizations. *Proceedings of the National Academy of Sciences*, 115(30):7723–7728, 2018.

James Eberwine, Hermes Yeh, Kevin Miyashiro, Yanxiang Cao, Suresh Nair, Richard Finnell, Martha Zettel, and Paul Coleman. Analysis of gene expression in single live neurons. *Proceedings of the National Academy of Sciences*, 89(7):3010–3014, 1992.

Gökcen Eraslan, Lukas M Simon, Maria Mircea, Nikola S Mueller, and Fabian J Theis. Single-cell rna-seq denoising using a deep count autoencoder. *Nature communications*, 10(1):390, 2019.

Mitsuhiro Eto, Wataru Hirota, Shigeto Seno, and Hideo Matsuda. Asymmetric integration of single-cell transcriptomic data using latent dirichlet allocation and procrustes analysis. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 2129–2135. IEEE, 2018.

James Ewing. Diffuse endothelioma of bone. *CA: A cancer journal for clinicians*, 22(2):95–98, 1972.

Adrien Fauré, Aurélien Naldi, Claudine Chaouiya, and Denis Thieffry. Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle. In *Bioinformatics*, volume 22(14), pages 124–131. Oxford Academic, jul 2006. doi: 10.1093/bioinformatics/btl210. URL https://academic.oup.com/bioinformatics/article/22/14/e124/227890.

Sira Ferradans, Nicolas Papadakis, Julien Rabin, Gabriel Peyré, and Jean-François Aujol. Regularized discrete optimal transport. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 428–439. Springer, 2013.

Evelyn Fix and Joseph Lawson Hodges. Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review/Revue Internationale de Statistique*, 57 (3):238–247, 1989.

Rémi Flamary and Nicolas Courty. POT Python optimal transport library, 2017. URL https://pythonot.github.io/.

Oscar Franzén, Li-Ming Gan, and Johan LM Björkegren. Panglaodb: a web server for exploration of mouse and human single-cell rna sequencing data. *Database*, 2019, 2019.

JosephF Fraumeni and AndrewG Glass. Rarity of ewing's sarcoma among us negro children. *The Lancet*, 295(7642):366–367, 1970.

Nathalie Gaspar, Douglas S Hawkins, Uta Dirksen, Ian J Lewis, Stefano Ferrari, Marie-Cecile Le Deley, Heinrich Kovar, Robert Grimer, Jeremy Whelan, Line Claude, et al. Ewing sarcoma: current management and future approaches through collaboration. *J Clin Oncol*, 33(27):3036–3046, 2015.

Georgii Frantsevich Gauze. *The struggle for existence*. Baltimore,The Williams I& Wilkins company, 1934. URL https://www.biodiversitylibrary.org/item/23409. https://www.biodiversitylibrary.org/bibliography/4489.

Adam Gayoso, Romain Lopez, Galen Xing, Pierre Boyeau, Valeh Valiollah Pour Amiri, Justin Hong, Katherine Wu, Michael Jayasuriya, Edouard Mehlman, Maxime Langevin, Yining Liu, Jules Samaran, Gabriel Misrachi, Achille Nazaret, Oscar Clivio, Chenling Xu, Tal Ashuach, Mariano Gabitto, Mohammad Lotfollahi, Valentine Svensson, Eduardo da Veiga Beltrame, Vitalii Kleshchevnikov, Carlos Talavera-López, Lior Pachter, Fabian J. Theis, Aaron Streets, Michael I. Jordan, Jeffrey Regier, and Nir Yosef. A python library for probabilistic analysis of single-cell omics data. *Nature Biotechnology*, Feb 2022. ISSN 1546-1696. doi: 10.1038/s41587-021-01206-w. URL https://doi.org/10.1038/s41587-021-01206-w.

Shila Ghazanfar, Carolina Guibentif, and John C Marioni. Stabmap: Mosaic single cell data integration using non-overlapping features. *bioRxiv*, pages 2022–02, 2022.

J.W. Gibbs. *The scientific papers. Vol. 1, Thermodynamics*. Dover, New York, 1961.

Marc Gillespie, Bijay Jassal, Ralf Stephan, Marija Milacic, Karen Rothfels, Andrea Senff-Ribeiro, Johannes Griss, Cristoffer Sevilla, Lisa Matthews, Chuqiao Gong, et al. The reactome pathway knowledgebase 2022. *Nucleic acids research*, 50(D1):D687–D692, 2022.

Bruno Giotti, Sz Hau Chen, Mark W. Barnett, Tim Regan, Tony Ly, Stefan Wiemann, David A. Hume, and Tom C. Freeman. Assembly of a parts list of the human mitotic cell cycle machinery. *Journal of Molecular Cell Biology*, 11(8):703–718, feb 2019. ISSN 17594685. doi: 10.1093/jmcb/mjy063. URL /pmc/articles/PMC6788831//pmc/articles/PMC6788831/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC6788831/.

Sergey E. Golovenkin, Jonathan Bac, Alexander Chervov, Evgeny M. Mirkes, Yuliya V. Orlova, Emmanuel Barillot, Alexander N. Gorban, and Andrei Zinovyev. Trajectories, bifurcations, and pseudo-time in large clinical datasets: Applications to myocardial infarction and diabetes data. *GigaScience*, 9(11):1–20, nov 2020. ISSN 2047217X. doi: 10.1093/gigascience/giaa128. URL http://orcid.org/0000-0002-9517-7284.

Boying Gong, Yun Zhou, and Elizabeth Purdom. Cobolt: integrative analysis of multimodal single-cell sequencing data. *Genome biology*, 22(1):1–21, 2021.

Carmen Bravo González-Blas, Liesbeth Minnoye, Dafni Papasokrati, Sara Aibar, Gert Hulselmans, Valerie Christiaens, Kristofer Davie, Jasper Wouters, and Stein Aerts. cisTopic: cis-regulatory topic modeling on single-cell ATAC-seq data. *Nature methods*, 16(5):397–400, 2019.

A. N. Gorban. Model reduction in chemical dynamics: slow invariant manifolds, singular perturbations, thermodynamic estimates, and analysis of reaction graph, sep 2018. ISSN 22113398.

Alexander N Gorban. Selection theorem for systems with inheritance. *Mathematical Modelling of Natural Phenomena*, 2(4):1–45, 2007.

Alexander N Gorban, Neil R Sumner, and Andrei Yu Zinovyev. Topological grammars for data approximation. *Applied Mathematics Letters*, 20(4):382–386, 2007.

Federico Gossi, Pushpak Pati, Panagiotis Chouvardas, Adriano Luca Martinelli, Marianna Kruithof-de Julio, and Maria Anna Rapsomaniki. Matching single cells across modalities with contrastive learning and optimal transport. *Briefings in bioinformatics*, 24(3):bbad130, 2023.

John C Gower. Generalized procrustes analysis. *Psychometrika*, 40:33–51, 1975.

Thomas GP Grünewald, Florencia Cidre-Aranaz, Didier Surdez, Eleni M Tomazou, Enrique de Álava, Heinrich Kovar, Poul H Sorensen, Olivier Delattre, and Uta Dirksen. Ewing sarcoma. *Nature reviews Disease primers*, 4(1):5, 2018.

Hongshan Guo, Ping Zhu, Xinglong Wu, Xianlong Li, Lu Wen, and Fuchou Tang. Single-cell methylome landscapes of mouse embryonic stem cells and early embryos analyzed using reduced representation bisulfite sequencing. *Genome research*, 23(12):2126–2135, 2013.

Christoph Hafemeister and Rahul Satija. Normalization and variance stabilization of single-cell rna-seq data using regularized negative binomial regression. *Genome biology*, 20(1):296, 2019.

Laleh Haghverdi, Aaron TL Lun, Michael D Morgan, and John C Marioni. Batch effects in single-cell rna-sequencing data are corrected by matching mutual nearest neighbors. *Nature biotechnology*, 36(5):421–427, 2018.

Douglas Hanahan. Hallmarks of cancer: new dimensions. *Cancer discovery*, 12(1):31–46, 2022.

Douglas Hanahan and Robert A Weinberg. The hallmarks of cancer. *cell*, 100(1):57–70, 2000.

Douglas Hanahan and Robert A Weinberg. Hallmarks of cancer: the next generation. *cell*, 144(5): 646–674, 2011.

Yuhan Hao, Stephanie Hao, Erica Andersen-Nissen, William M Mauck III, Shiwei Zheng, Andrew Butler, Maddie J Lee, Aaron J Wilk, Charlotte Darby, Michael Zager, et al. Integrated analysis of multimodal single-cell data. *Cell*, 184(13):3573–3587, 2021.

Brian Hie, Bryan Bryson, and Bonnie Berger. Efficient integration of heterogeneous single-cell transcriptomes using scanorama. *Nature biotechnology*, 37(6):685–691, 2019.

Nick HG Holford and Brian J Anderson. Allometric size: the scientific theory and extension to normal fat mass. *European Journal of Pharmaceutical Sciences*, 109:S59–S64, 2017.

Leroy Hood and Lee Rowen. The human genome project: big science transforms biology and medicine. *Genome medicine*, 5:1–8, 2013.

Harold Hotelling. Relations between two sets of variates. *Breakthroughs in statistics: methodology and distribution*, pages 162–190, 1992.

Chiaowen Joyce Hsiao, Po Yuan Tung, John D. Blischak, Jonathan E. Burnett, Kenneth A. Barr, Kushal K. Dey, Matthew Stephens, and Yoav Gilad. Characterizing and inferring quantitative cell cycle phase in single-cell RNA-seq data analysis. *Genome Research*, 30(4):611–621, apr 2020. ISSN 15495469. doi: 10.1101/gr.247759.118. URL http://www.genome.org/cgi/doi/10.1101/gr.247759.118.

Geert-Jan Huizing, Laura Cantini, and Gabriel Peyré. Unsupervised ground metric learning using wasserstein eigenvectors, 2021a.

Geert-Jan Huizing, Gabriel Peyré, and Laura Cantini. Optimal transport improves cell-cell similarity inference in single-cell omics data. *bioRxiv*, 2021b.

T. Hunt. Cell biology. Cell cycle gets more cyclins. *Nature*, 350(6318):462–463, April 1991. ISSN 0028-0836. doi: 10.1038/350462a0.

Tim Hunt, Kim Nasmyth, and Béla Novák. The cell cycle. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 366(1584):3494–3497, December 2011. ISSN 1471-2970. doi: 10.1098/rstb.2011.0274.

A. Hyvarinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999. doi: 10.1109/72.761722.

IGR. Sarcome d'ewing, 2023. URL https://www.gustaveroussy.fr/fr/sarcome-ewing.

Nicholas T Ingolia and Andrew W Murray. The ups and downs of modeling the cell cycle, 2004. ISSN 09609822.

Bijay Jassal, Lisa Matthews, Guilherme Viteri, Chuqiao Gong, Pascual Lorente, Antonio Fabregat, Konstantinos Sidiropoulos, Justin Cook, Marc Gillespie, Robin Haw, Fred Loney, Bruce May, Marija Milacic, Karen Rothfels, Cristoffer Sevilla, Veronica Shamovsky, Solomon Shorser, Thawfeek Varusai, Joel Weiser, Guanming Wu, Lincoln Stein, Henning Hermjakob, and Peter D'Eustachio. The reactome pathway knowledgebase. *Nucleic Acids Research*, 48(D1):D498–D503, jan 2020. ISSN 13624962. doi: 10.1093/nar/gkz1031. URL https://pubmed.ncbi.nlm.nih.gov/31691815/.

Muhammad U Jawad, Michael C Cheung, Elijah S Min, Michaela M Schneiderbauer, Leonidas G Koniaris, and Sean P Scully. Ewing sarcoma demonstrates racial disparities in incidence-related and sex-related differences in outcome: an analysis of 1631 cases from the seer database, 1973-2005. *Cancer: Interdisciplinary International Journal of the American Cancer Society*, 115(15): 3526–3536, 2009.

RobertD Jensen and RobertM Drake. Rarity of ewing's tumour in negroes. *The Lancet*, 295(7650): 777, 1970.

Suoqin Jin, Lihua Zhang, and Qing Nie. scai: an unsupervised approach for the integrative analysis of parallel single-cell transcriptomic and epigenomic profiles. *Genome biology*, 21:1–19, 2020.

Nelson Johansen and Gerald Quon. scalign: a tool for alignment, integration, and rare cell identification from scrna-seq data. *Genome biology*, 20(1):1–21, 2019.

W Evan Johnson, Cheng Li, and Ariel Rabinovic. Adjusting batch effects in microarray expression data using empirical bayes methods. *Biostatistics*, 8(1):118–127, 2007.

Michael J Joyce, David C Harmon, Henry J Mankin, Herman D Suit, Alan L Schiller, and John T Truman. Ewing's sarcoma in female siblings: A clinical report and review of the literature. *Cancer*, 53(9):1959–1962, 1984.

Christian Jutten and Jeanny Herault. Blind separation of sources, part i: An adaptive algorithm based on neuromimetic architecture. *Signal processing*, 24(1):1–10, 1991.

Ulykbek Kairov, Laura Cantini, Alessandro Greco, Askhat Molkenov, Urszula Czerwinska, Emmanuel Barillot, and Andrei Zinovyev. Determining the optimal number of independent components for reproducible transcriptomic data analysis. *BMC Genomics*, 18(1):712, Sep 2017. ISSN 1471-2164. doi: 10.1186/s12864-017-4112-9. URL https://doi.org/10.1186/s12864-017-4112-9.

Minoru Kanehisa, Susumu Goto, Yoko Sato, Miho Furumichi, and Mao Tanabe. Kegg for integration and interpretation of large-scale molecular data sets. *Nucleic acids research*, 40(D1): D109–D114, 2012.

Peter V Kharchenko, Lev Silberstein, and David T Scadden. Bayesian approach to single-cell differential expression analysis. *Nature methods*, 11(7):740–742, 2014.

Gabriela S Kinker, Alissa C Greenwald, Rotem Tal, Zhanna Orlova, Michael S Cuoco, James M McFarland, Allison Warren, Christopher Rodman, Jennifer A Roth, Samantha A Bender, Bhavna Kumar, James W Rocco, Pedro ACM Fernandes, Christopher C Mader, Hadas Keren-Shaul, Alexander Plotnikov, Haim Barr, Aviad Tsherniak, Orit Rozenblatt-Rosen, Valery Krizhanovsky, Sidharth V Puram, Aviv Regev, and Itay Tirosh. Pan-cancer single cell RNA-seq uncovers recurring programs of cellular heterogeneity. *Nature Genetics*, 52(11), nov 2020. doi: 10.1038/s41588-020-00726-6. URL https://pubs.acs.org/doi/abs/10.1038/s41588-020-00726-6.

Vladimir Yu Kiselev, Andrew Yiu, and Martin Hemberg. scmap: projection of single-cell rna-seq data across data sets. *Nature methods*, 15(5):359–362, 2018.

Hiroaki Kitano. Computational systems biology. *Nature*, 420(6912):206–210, 2002.

Allon M Klein, Linas Mazutis, Ilke Akartuna, Naren Tallapragada, Adrian Veres, Victor Li, Leonid Peshkin, David A Weitz, and Marc W Kirschner. Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell*, 161(5):1187–1201, 2015.

Ilya Korsunsky, Nghia Millard, Jean Fan, Kamil Slowikowski, Fan Zhang, Kevin Wei, Yuriy Baglaenko, Michael Brenner, Po-ru Loh, and Soumya Raychaudhuri. Fast, sensitive and accurate integration of single-cell data with harmony. *Nature methods*, 16(12):1289–1296, 2019.

Shuji Kotani. A Computational Model of Mammalian Cell Cycle Using Petri Nets. *Genome Informatics*, 460:459–460, 2002. ISSN 0919-9454. doi: 10.11234/GI1990.13.459. URL http://www.genomicobject.net/.

Konstantina Kourou, Konstantinos P Exarchos, Costas Papaloukas, Prodromos Sakaloglou, Themis Exarchos, and Dimitrios I Fotiadis. Applied machine learning in cancer research: A systematic review for patient diagnosis, classification and prognosis. *Computational and Structural Biotechnology Journal*, 19:5546–5555, 2021.

April R Kriebel and Joshua D Welch. Uinmf performs mosaic integration of single-cell multi-omic datasets using nonnegative matrix factorization. *Nature communications*, 13(1):780, 2022.

Kazuki Kurimoto, Yukihiro Yabuta, Yasuhide Ohinata, Yukiko Ono, Kenichiro D Uno, Rikuhiro G Yamada, Hiroki R Ueda, and Mitinori Saitou. An improved single-cell cdna amplification method for efficient high-density oligonucleotide microarray analysis. *Nucleic acids research*, 34(5):e42–e42, 2006.

Gioele La Manno, Ruslan Soldatov, Amit Zeisel, Emelie Braun, Hannah Hochgerner, Viktor Petukhov, Katja Lidschreiber, Maria E. Kastriti, Peter Lönnerberg, Alessandro Furlan, Jean Fan, Lars E. Borm, Zehua Liu, David van Bruggen, Jimin Guo, Xiaoling He, Roger Barker, Erik Sundström, Gonçalo Castelo-Branco, Patrick Cramer, Igor Adameyko, Sten Linnarsson, and Peter V. Kharchenko. RNA velocity of single cells. *Nature*, 560(7719):494–498, aug 2018. ISSN 14764687. doi: 10.1038/s41586-018-0414-6. URL https://doi.org/10.1038/s41586-018-0414-6.

Gioele La Manno, Ruslan Soldatov, Amit Zeisel, Emelie Braun, Hannah Hochgerner, Viktor Petukhov, Katja Lidschreiber, Maria E Kastriti, Peter Lönnerberg, Alessandro Furlan, et al. Rna velocity of single cells. *Nature*, 560(7719):494–498, 2018.

David Lähnemann, Johannes Köster, Ewa Szczurek, Davis J McCarthy, Stephanie C Hicks, Mark D Robinson, Catalina A Vallejos, Kieran R Campbell, Niko Beerenwinkel, Ahmed Mahfouz, et al. Eleven grand challenges in single-cell data science. *Genome biology*, 21(1):1–35, 2020.

Yunxin Lai, Xinru Wei, Shouheng Lin, Le Qin, Lin Cheng, and Peng Li. Current status and perspectives of patient-derived xenograft models in cancer research. *Journal of hematology & oncology*, 10:1–14, 2017.

Christopher Lance, Malte D Luecken, Daniel B Burkhardt, Robrecht Cannoodt, Pia Rautenstrauch, Anna Christine Laddach, Aidyn Ubingazhibov, Zhi-Jie Cao, Kaiwen Deng, Sumeer Khan, et al. Multimodal single cell data integration challenge: results and lessons learned. *bioRxiv*, 2022.

William H Lawton and Edward A Sylvestre. Self modeling curve resolution. *Technometrics*, 13 (3):617–633, 1971.

Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

Ning Leng, Li Fang Chu, Chris Barry, Yuan Li, Jeea Choi, Xiaomao Li, Peng Jiang, Ron M. Stewart, James A. Thomson, and Christina Kendziorski. Oscope identifies oscillatory genes in unsynchronized single-cell RNA-seq experiments. *Nature Methods*, 12(10):947–950, sep 2015. ISSN 15487105. doi: 10.1038/nmeth.3549. URL https://www.nature.com/articles/nmeth.3549.

Bin Li, Wen Zhang, Chuang Guo, Hao Xu, Longfei Li, Minghao Fang, Yinlei Hu, Xinye Zhang, Xinfeng Yao, Meifang Tang, et al. Benchmarking spatial and single-cell transcriptomics integration methods for transcript distribution prediction and cell type deconvolution. *Nature Methods*, pages 1–9, 2022.

Genyuan Li and Herschel Rabitz. A general analysis of exact lumping in chemical kinetics. *Chemical Engineering Science*, 44(6):1413–1430, jan 1989. ISSN 00092509. doi: 10.1016/0009-2509(89)85014-6.

Genyuan Li and Herschel Rabitz. A general analysis of approximate lumping in chemical kinetics. *Chemical Engineering Science*, 45(4):977–1002, jan 1990. ISSN 00092509. doi: 10.1016/0009-2509(90)85020-E.

Xiangjie Li, Kui Wang, Yafei Lyu, Huize Pan, Jingxiao Zhang, Dwight Stambolian, Katalin Susztak, Muredach P Reilly, Gang Hu, and Mingyao Li. Deep learning enables accurate clustering with batch effect removal in single-cell rna-seq analysis. *Nature communications*, 11(1):2338, 2020.

Shaoheng Liang, Fang Wang, Jincheng Han, and Ken Chen. Latent periodic process inference from single-cell RNA-seq data. *Nature Communications*, 11(1441), march 2020. doi: 10.1038/s41467-020-15295-9. URL https://pubs.acs.org/doi/abs/10.1038/s41467-020-15295-9.

Wolfram Liebermeister. Linear modes of gene expression determined by independent component analysis. *Bioinformatics*, 18(1):51–60, 2002.

Matthias Liero, Alexander Mielke, and Giuseppe Savaré. Optimal entropy-transport problems and a new hellinger–kantorovich distance between positive measures. *Inventiones mathematicae*, 211(3):969–1117, 2018.

Yingxin Lin, Tung-Yu Wu, Sheng Wan, Jean YH Yang, Wing H Wong, and YX Rachel Wang. scjoint integrates atlas-scale single-cell rna-seq and atac-seq data with transfer learning. *Nature biotechnology*, 40(5):703–710, 2022.

Jie Liu, Yuanhao Huang, Ritambhara Singh, Jean-Philippe Vert, and William Stafford Noble. Jointly embedding multiple single-cell omics measurements. In *Algorithms in bioinformatics:... International Workshop, WABI..., proceedings. WABI (Workshop)*, volume 143. NIH Public Access, 2019.

Zehua Liu, Huazhe Lou, Kaikun Xie, Hao Wang, Ning Chen, Oscar M. Aparicio, Michael Q. Zhang, Rui Jiang, and Ting Chen. Reconstructing cell cycle pseudo time-series via single-cell transcriptome data. *Nature Communications*, 8(1):1–9, dec 2017. ISSN 20411723. doi: 10.1038/s41467-017-00039-z. URL www.nature.com/naturecommunications.

Eric F Lock, Katherine A Hoadley, James Stephen Marron, and Andrew B Nobel. Joint and individual variation explained (jive) for integrated analysis of multiple data types. *The annals of applied statistics*, 7(1):523, 2013.

Harvey Lodish, Arnold Berk, Chris A Kaiser, Chris Kaiser, Monty Krieger, Matthew P Scott, Anthony Bretscher, Hidde Ploegh, Paul Matsudaira, et al. *Molecular cell biology*. Macmillan, 2008.

Eliane Maria Loiola, Nair Maria Maia de Abreu, Paulo Oswaldo Boaventura-Netto, Peter Hahn, and Tania Querido. A survey for the quadratic assignment problem. *European journal of operational research*, 176(2):657–690, 2007.

Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature methods*, 15(12):1053–1058, 2018.

Martin Loza, Shunsuke Teraguchi, Daron M Standley, and Diego Diez. Unbiased integration of single cell transcriptome replicates. *NAR Genomics and Bioinformatics*, 4(1):lqac022, 2022.

Malte D Luecken, Maren Büttner, Kridsadakorn Chaichoompu, Anna Danese, Marta Interlandi, Michaela F Müller, Daniel C Strobl, Luke Zappia, Martin Dugas, Maria Colomé-Tatché, et al. Benchmarking atlas-level data integration in single-cell genomics. *Nature methods*, 19(1):41–50, 2022.

Allen W Lynch, Christina V Theodoris, Henry W Long, Myles Brown, X Shirley Liu, and Clifford A Meyer. Mira: Joint regulatory modeling of multimodal expression and chromatin accessibility in single cells. *Nature Methods*, 19(9):1097–1108, 2022.

Vince Lyzinski, Donniell E Fishkind, Marcelo Fiori, Joshua T Vogelstein, Carey E Priebe, and Guillermo Sapiro. Graph matching: Relax at your own risk. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):60–73, 2015.

Evan Z Macosko, Anindita Basu, Rahul Satija, James Nemesh, Karthik Shekhar, Melissa Goldman, Itay Tirosh, Allison R Bialas, Nolan Kamitaki, Emily M Martersteck, et al. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell*, 161(5):1202–1214, 2015.

Diana Mahdessian, Anthony J. Cesnik, Christian Gnann, Frida Danielsson, Lovisa Stenström, Muhammad Arif, Cheng Zhang, Trang Le, Fredric Johansson, Rutger Shutten, Anna Bäckström, Ulrika Axelsson, Peter Thul, Nathan H. Cho, Oana Carja, Mathias Uhlén, Adil Mardinoglu, Charlotte Stadler, Cecilia Lindskog, Burcu Ayoglu, Manuel D. Leonetti, Fredrik Pontén, Devin P. Sullivan, and Emma Lundberg. Spatiotemporal dissection of the cell cycle with single-cell proteogenomics. *Nature*, 590(7847):649–654, feb 2021a. ISSN 14764687. doi: 10.1038/s41586-021-03232-9. URL https://doi.org/10.1038/s41586-021-03232-9.

Diana Mahdessian, Anthony J Cesnik, Christian Gnann, Frida Danielsson, Lovisa Stenström, Muhammad Arif, Cheng Zhang, Trang Le, Fredric Johansson, Rutger Shutten, et al. Spatiotemporal dissection of the cell cycle with single-cell proteogenomics. *Nature*, 590(7847):649–654, 2021b.

Henry E. Miller, Aparna Gorthi, Nicklas Bassani, Liesl A. Lawrence, Brian S. Iskra, and Alexander J.R. Bishop. Reconstruction of ewing sarcoma developmental context from mass-scale transcriptomics reveals characteristics of ewsr1-fli1 permissibility. *Cancers*, 12(4):948, apr 2020a. ISSN 20726694. doi: 10.3390/cancers12040948. URL www.mdpi.com/journal/cancers.

Henry E Miller, Aparna Gorthi, Nicklas Bassani, Liesl A Lawrence, Brian S Iskra, and Alexander JR Bishop. Reconstruction of Ewing sarcoma developmental context from mass-scale transcriptomics reveals characteristics of EWSR1-FLI1 permissibility. *Cancers*, 12(4), 2020b.

Kodai Minoura, Ko Abe, Hyunha Nam, Hiroyoshi Nishikawa, and Teppei Shimamura. A mixture-of-experts deep generative model for integrated analysis of single-cell multiomics data. *Cell reports methods*, 1(5):100071, 2021.

Noa Moriel, Enes Senel, Nir Friedman, Nikolaus Rajewsky, Nikos Karaiskos, and Mor Nitzan. Novosparc: flexible spatial reconstruction of single-cell gene expression with optimal transport. *Nature Protocols*, pages 1–24, 2021.

NCI. Cancer in children and adolescents, 2023. URL https://www.cancer.gov/types/childhood-cancers/child-adolescent-cancers-fact-sheet.

Vincent Noel, Dima Grigoriev, Sergei Vakulenko, and Ovidiu Radulescu. Tropical geometries and dynamics of biochemical networks application to hybrid cell cycle models. *Electronic Notes in Theoretical Computer Science*, 284:75–91, 2012.

Vincent Noël, Sergey Vakulenko, and Ovidiu Radulescu. A hybrid mammalian cell cycle model. *Electronic Proceedings in Theoretical Computer Science, EPTCS*, 125:68–83, sep 2013. doi: 10.4204/EPTCS.125.5. URL http://arxiv.org/abs/1309.0870http://dx.doi.org/10.4204/EPTCS.125.5.

Béla Novák and John J. Tyson. A model for restriction point control of the mammalian cell cycle. *Journal of Theoretical Biology*, 230(4):563–579, 2004. ISSN 0022-5193. doi: https://doi.org/10.1016/j.jtbi.2004.04.039. URL https://www.sciencedirect.com/science/article/pii/S0022519304002449.

Gary C Packard. The essential role for graphs in allometric analysis. *Biological Journal of the Linnean Society*, 120(2):468–473, 2017.

Liron Pantanowitz, Paul N Valenstein, Andrew J Evans, Keith J Kaplan, John D Pfeifer, David C Wilbur, Laura C Collins, and Terence J Colgan. Review of the current state of whole slide imaging in pathology. *Journal of pathology informatics*, 2(1):36, 2011.

Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport with applications to data science. *Foundations and Trends in Machine Learning*, 11(5-6):355–607, 2019.

Janet Piñero, Juan Manuel Ramírez-Anguita, Josep Saüch-Pitarch, Francesco Ronzano, Emilio Centeno, Ferran Sanz, and Laura I Furlong. The disgenet knowledge platform for disease genomics: 2019 update. *Nucleic acids research*, 48(D1):D845–D855, 2020.

Krzysztof Polański, Matthew D Young, Zhichao Miao, Kerstin B Meyer, Sarah A Teichmann, and Jong-Eun Park. Bbknn: fast batch alignment of single cell transcriptomes. *Bioinformatics*, 36 (3):964–965, 2020.

Hans Pretzsch. The course of tree growth. theory and reality. *Forest Ecology and Management*, 478:118508, 2020.

Ovidiu Radulescu, Alexander N Gorban, Andrei Zinovyev, and Vincent Noel. Reduction of dynamical biochemical reactions networks in computational biology. *Frontiers in genetics*, 3:131, 2012.

Pauline Rochefort, Antoine Italiano, Valérie Laurence, Nicolas Penel, Audrey Lardy-Cleaud, Olivier Mir, Christine Chevreau, François Bertucci, Emmanuelle Bompas, Loic Chaigneau, et al. A retrospective multicentric study of ewing sarcoma family of tumors in patients older than 50: management and outcome. *Scientific reports*, 7(1):17917, 2017.

Rahul Satija, Jeffrey A Farrell, David Gennert, Alexander F Schier, and Aviv Regev. Spatial reconstruction of single-cell gene expression data. *Nature biotechnology*, 33(5):495–502, 2015.

SCC. Traitements du sarcome d'ewing localisé chez l'enfant, 2023. URL https://cancer.ca/fr/cancer-information/cancer-types/bone-childhood/treatment/ewing-sarcoma.

Nicholas Schaum, Jim Karkanias, Norma F Neff, Andrew P May, Stephen R Quake, Tony Wyss-Coray, Spyros Darmanis, Joshua Batson, Olga Botvinnik, Michelle B Chen, et al. Single-cell transcriptomics of 20 mouse organs creates a tabula muris: The tabula muris consortium. *Nature*, 562(7727):367, 2018.

Geoffrey Schiebinger, Jian Shu, Marcin Tabaka, Brian Cleary, Vidya Subramanian, Aryeh Solomon, Joshua Gould, Siyan Liu, Stacie Lin, Peter Berube, et al. Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*, 176(4):928–943, 2019.

Daniel Schwabe, Sara Formichetti, Jan Philipp Junker, Martin Falcke, and Nikolaus Rajewsky. The transcriptome dynamics of single cells during the cell cycle. *Molecular Systems Biology*, 16(11):e9946, nov 2020. ISSN 1744-4292. doi: 10.15252/msb.20209946. URL https://www.embopress.org/doi/full/10.15252/msb.20209946https://www.embopress.org/doi/abs/10.15252/msb.20209946.

Human Genome Sequencing. Finishing the euchromatic sequence of the human genome. *Nature*, 431(7011):931–45, 2004.

Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

Paul Shannon, Andrew Markiel, Owen Ozier, Nitin S Baliga, Jonathan T Wang, Daniel Ramage, Nada Amin, Benno Schwikowski, and Trey Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome research*, 13(11):2498–2504, 2003.

E. M. Shkolnik. Dynamic theory of cell cycle. In *Dynamics of chemical and biological systems [in Russian]*, pages 159–190. Nauka plc. (Siberian branch, 1989.

Jill C. Sible and John J. Tyson. Mathematical modeling as a tool for investigating cell cycle control networks. *Methods*, 41(2):238–247, feb 2007. ISSN 10462023. doi: 10.1016/j.ymeth.2006.08.003.

Nikola Simidjievski, Cristian Bodnar, Ifrah Tariq, Paul Scherer, Helena Andres Terre, Zohreh Shams, Mateja Jamnik, and Pietro Liò. Variational autoencoders for cancer data integration: design principles and computational practice. *Frontiers in genetics*, 10:1205, 2019.

Amrit Singh, Casey P Shannon, Benoît Gautier, Florian Rohart, Michaël Vacher, Scott J Tebbutt, and Kim-Anh Lê Cao. Diablo: an integrative approach for identifying key molecular drivers from multi-omics assays. *Bioinformatics*, 35(17):3055–3062, 2019.

Rajat Singhania, R. Michael Sramkoski, James W. Jacobberger, and John J. Tyson. A Hybrid Model of Mammalian Cell Cycle Regulation. *PLOS Computational Biology*, 7(2):e1001077, feb 2011. ISSN 1553-7358. doi: 10.1371/JOURNAL.PCBI.1001077. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1001077.

Nicolas Sompairac, Petr V Nazarov, Urszula Czerwinska, Laura Cantini, Anne Biton, Askhat Molkenov, Zhaxybay Zhumadilov, Emmanuel Barillot, Francois Radvanyi, Alexander Gorban, et al. Independent component analysis for unraveling the complexity of cancer omics datasets. *International Journal of molecular sciences*, 20(18):4414, 2019.

Patrik L Ståhl, Fredrik Salmén, Sanja Vickovic, Anna Lundmark, José Fernández Navarro, Jens Magnusson, Stefania Giacomello, Michaela Asp, Jakub O Westholm, Mikael Huss, et al. Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. *Science*, 353(6294):78–82, 2016.

Stefan G Stark, Joanna Ficek, Francesco Locatello, Ximena Bonilla, Stéphane Chevrier, Franziska Singer, Gunnar Rätsch, and Kjong-Van Lehmann. Scim: universal single-cell matching with unpaired feature sets. *Bioinformatics*, 36(Supplement_2):i919–i927, 2020.

CJ Stein and GA Colditz. Modifiable risk factors for cancer. *British journal of cancer*, 90(2): 299–303, 2004.

B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd. OSQP: an operator splitting solver for quadratic programs. *Mathematical Programming Computation*, 12(4):637–672, 2020. doi: 10.1007/s12532-020-00179-2. URL https://doi.org/10.1007/s12532-020-00179-2.

Marlon Stoeckius, Christoph Hafemeister, William Stephenson, Brian Houck-Loomis, Pratip K Chattopadhyay, Harold Swerdlow, Rahul Satija, and Peter Smibert. Simultaneous epitope and transcriptome measurement in single cells. *Nature methods*, 14(9):865–868, 2017.

Tim Stuart and Rahul Satija. Integrative single-cell analysis. *Nature reviews genetics*, 20(5): 257–272, 2019.

Tim Stuart, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M Mauck III, Yuhan Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. Comprehensive integration of single-cell data. *Cell*, 177(7):1888–1902, 2019.

Reiichi Sugihara, Yuki Kato, Tomoya Mori, and Yukio Kawahara. Alignment of single-cell trajectory trees with capital. *Nature Communications*, 13(1):5972, 2022.

Duanchen Sun, Xiangnan Guan, Amy E Moran, Ling-Yun Wu, David Z Qian, Pepper Schedin, Mu-Shui Dai, Alexey V Danilov, Joshi J Alumkal, Andrew C Adey, et al. Identifying phenotype-associated subpopulations by integrating bulk and single-cell sequencing data. *Nature biotechnology*, 40(4):527–538, 2022.

Valentine Svensson, Adam Gayoso, Nir Yosef, and Lior Pachter. Interpretable factor models of single-cell rna-seq via variational autoencoders. *Bioinformatics*, 36(11):3418–3421, 2020.

Tabula Sapiens Consortium, Robert C Jones, Jim Karkanias, Mark A Krasnow, Angela Oliveira Pisco, Stephen R Quake, Julia Salzman, Nir Yosef, Bryan Bulthaup, Phillip Brown, et al. The tabula sapiens: A multiple-organ, single-cell transcriptomic atlas of humans. *Science*, 376(6594): eabl4896, 2022.

Arthur Tenenhaus and Michel Tenenhaus. Regularized generalized canonical correlation analysis. *Psychometrika*, 76:257–284, 2011.

Arthur Tenenhaus, Cathy Philippe, Vincent Guillemot, Kim-Anh Le Cao, Jacques Grill, and Vincent Frouin. Variable selection for generalized canonical correlation analysis. *Biostatistics*, 15(3):569–583, 2014.

Raoul Tibes, YiHua Qiu, Yiling Lu, Bryan Hennessy, Michael Andreeff, Gordon B Mills, and Steven M Kornblau. Reverse phase protein array: validation of a novel proteomic technology and utility for analysis of primary leukemia specimens and hematopoietic stem cells. *Molecular cancer therapeutics*, 5(10):2512–2521, 2006.

Itay Tirosh, Benjamin Izar, Sanjay M Prakadan, Marc H Wadsworth, Daniel Treacy, John J Trombetta, Asaf Rotem, Christopher Rodman, Christine Lian, George Murphy, et al. Dissecting the multicellular ecosystem of metastatic melanoma by single-cell rna-seq. *Science*, 352(6282): 189–196, 2016.

Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):1–12, 2019.

Hoa Thi Nhu Tran, Kok Siong Ang, Marion Chevrier, Xiaomeng Zhang, Nicole Yee Shin Lee, Michelle Goh, and Jinmiao Chen. A benchmark of batch-effect correction methods for single-cell rna sequencing data. *Genome biology*, 21:1–32, 2020.

Cole Trapnell, Lior Pachter, and Steven L Salzberg. Tophat: discovering splice junctions with rna-seq. *Bioinformatics*, 25(9):1105–1111, 2009.

Martin Treppner, Harald Binder, and Moritz Hess. Interpretable generative deep learning: an illustration with single cell gene expression data. *Human Genetics*, pages 1–18, 2022.

Trung Ngo Trong, Juha Mehtonen, Gerardo González, Roger Kramer, Ville Hautamäki, and Merja Heinäniemi. Semisupervised generative autoencoder for single-cell data. *Journal of Computational Biology*, 27(8):1190–1203, 2020.

J. J. Tyson. Modeling the cell division cycle: cdc2 and cyclin interactions. *Proceedings of the National Academy of Sciences of the United States of America*, 88(16):7328–7332, aug 1991. ISSN 00278424. doi: 10.1073/pnas.88.16.7328. URL https://www.pnas.org/content/88/16/7328https://www.pnas.org/content/88/16/7328.abstract.

Eleftheria Tzamali, Georgios Tzedakis, and Vangelis Sakkalis. Modeling How Heterogeneity in Cell Cycle Length Affects Cancer Cell Growth Dynamics in Response to Treatment. *Frontiers in Oncology*, 10:1552, sep 2020. ISSN 2234943X. doi: 10.3389/FONC.2020.01552/BIBTEX.

Monique GP Van Der Wijst, Harm Brugge, Dylan H De Vries, Patrick Deelen, Morris A Swertz, LifeLines Cohort Study, BIOS Consortium, and Lude Franke. Single-cell rna sequencing identifies celltype-specific cis-eqtls and co-expression qtls. *Nature genetics*, 50(4):493–497, 2018.

David Van Dijk, Roshan Sharma, Juozas Nainys, Kristina Yim, Pooja Kathail, Ambrose J Carr, Cassandra Burdziak, Kevin R Moon, Christine L Chaffer, Diwakar Pattabiraman, et al. Recovering gene interactions from single-cell data using data diffusion. *Cell*, 174(3):716–729, 2018.

Imre Vastrik, Peter D'Eustachio, Esther Schmidt, Geeta Joshi-Tope, Gopal Gopinath, David Croft, Bernard de Bono, Marc Gillespie, Bijay Jassal, Suzanna Lewis, et al. Reactome: a knowledge base of biologic pathways and processes. *Genome biology*, 8:1–13, 2007.

Titouan Vayer, Laetitia Chapel, Rémi Flamary, Romain Tavenard, and Nicolas Courty. Optimal Transport for structured data with application on graphs. *36th International Conference on Machine Learning, ICML 2019*, 2019-June:10940–10949, 2019.

Mai-Anh T Vu, Tülay Adalı, Demba Ba, György Buzsáki, David Carlson, Katherine Heller, Conor Liston, Cynthia Rudin, Vikaas S Sohal, Alik S Widge, et al. A shared vision for machine learning in neuroscience. *Journal of Neuroscience*, 38(7):1601–1607, 2018.

Dongfang Wang and Jin Gu. Vasc: dimension reduction and visualization of single-cell rna-seq data by deep variational autoencoder. *Genomics, proteomics & bioinformatics*, 16(5):320–331, 2018.

Zhong Wang, Mark Gerstein, and Michael Snyder. Rna-seq: a revolutionary tool for transcriptomics. *Nature reviews genetics*, 10(1):57–63, 2009.

James Wei and James C.W. Kuo. A lumping analysis in monomolecular reaction systems: Analysis of the Exactly Lumpable System. *Industrial and Engineering Chemistry Fundamentals*, 8(1): 114–123, feb 1969. ISSN 01964313. doi: 10.1021/i160029a019. URL https://pubs.acs.org/doi/abs/10.1021/i160029a019.

J Weinstein, E Collisson, GB Mills, KR Mills Shaw, BA Ozenberger, K Ellrott, I Shmulevich, C Sander, and Stuart JM. The cancer genome atlas pan-cancer analysis project. *Nature genetics*, 45(10):1113–1120, 2013.

Joshua D Welch, Alexander J Hartemink, and Jan F Prins. Matcher: manifold alignment reveals correspondence between single cell transcriptome and epigenome dynamics. *Genome biology*, 18 (1):1–19, 2017.

Joshua D Welch, Velina Kozareva, Ashley Ferreira, Charles Vanderburg, Carly Martin, and Evan Z Macosko. Single-cell multi-omic integration compares and contrasts features of brain cell identity. *Cell*, 177(7):1873–1887, 2019.

Markus R Wenk. The emerging field of lipidomics. *Nature reviews Drug discovery*, 4(7):594–610, 2005.

Reiner Westermeier and Rita Marouga. Protein detection methods in proteomics research. *Bioscience reports*, 25(1-2):19–32, 2005.

Craig R White, Dustin J Marshall, Lesley A Alton, Pieter A Arnold, Julian E Beaman, Candice L Bywater, Catriona Condon, Taryn S Crispin, Aidan Janetzki, Elia Pirtle, et al. The origin and maintenance of metabolic allometry in animals. *Nature Ecology & Evolution*, 3(4):598–603, 2019.

WHO. Latest global cancer data: Cancer burden rises to 18.1 million new cases and 9.6 million cancer deaths in 2018. *Press Relase n°263*, 2018.

Björn Widhe and Torulf Widhe. Initial symptoms and clinical features in osteosarcoma and ewing sarcoma. *JBJS*, 82(5):667, 2000.

Alan Geoffrey Wilson. The use of entropy maximising models, in the theory of trip distribution, mode split and route split. *Journal of transport economics and policy*, pages 108–126, 1969.

F Alexander Wolf, Philipp Angerer, and Fabian J Theis. Scanpy: large-scale single-cell gene expression data analysis. *Genome biology*, 19(1):1–5, 2018.

Jennifer Worch, Jobin Cyrus, Robert Goldsby, Katherine K Matthay, John Neuhaus, and Steven G DuBois. Racial differences in the incidence of mesenchymal tumors associated with ewsr1 translocation. *Cancer epidemiology, biomarkers & prevention*, 20(3):449–453, 2011.

Yang Xu and Rachel Patton McCord. Diagonal integration of multimodal single-cell data: potential pitfalls and paths forward. *Nature Communications*, 13(1):3505, 2022.

Yang Xu, Edmon Begoli, and Rachel Patton McCord. scican: single-cell chromatin accessibility and gene expression data integration via cycle-consistent adversarial network. *npj Systems Biology and Applications*, 8(1):33, 2022a.

Yang Xu, Priyojit Das, and Rachel Patton McCord. Smile: mutual information learning for integration of single-cell omics data. *Bioinformatics*, 38(2):476–486, 2022b.

A Zetterberg, O Larsson, and KG Wiman. What is the restriction point? *Curr Opin Cell Biol*, 7, 1995.

Ran Zhang, Laetitia Meng-Papaxanthos, Jean-philippe Vert, and William Stafford Noble. Multimodal single-cell translation and alignment with semi-supervised learning. *Journal of Computational Biology*, 29(11):1198–1212, 2022a.

Ziqi Zhang, Chengkai Yang, and Xiuwei Zhang. scdart: integrating unmatched scrna-seq and scatac-seq data and learning cross-modality relationship simultaneously. *Genome Biology*, 23 (1):139, 2022b.

Xiaolu Zhou, Mingxia Yang, Zelin Liu, Peng Li, Binggeng Xie, and Changhui Peng. Dynamic allometric scaling of tree biomass and size. *Nature Plants*, 7(1):42–49, 2021.

Yan Zhou, Dong Yang, Qingcheng Yang, Xiaobin Lv, Wentao Huang, Zhenhua Zhou, Yaling Wang, Zhichang Zhang, Ting Yuan, Xiaomin Ding, et al. Single-cell rna landscape of intratumoral heterogeneity and immunosuppressive microenvironment in advanced osteosarcoma. *Nature communications*, 11(1):1–17, 2020.

Andrei Zinovyev, Ulykbek Kairov, Tatyana Karpenyuk, and Erlan Ramanculov. Blind source separation methods for deconvolution of complex signals in cancer biology. *Biochemical and biophysical research communications*, 430(3):1182–1187, 2013.

Appendix

Supplementary figures

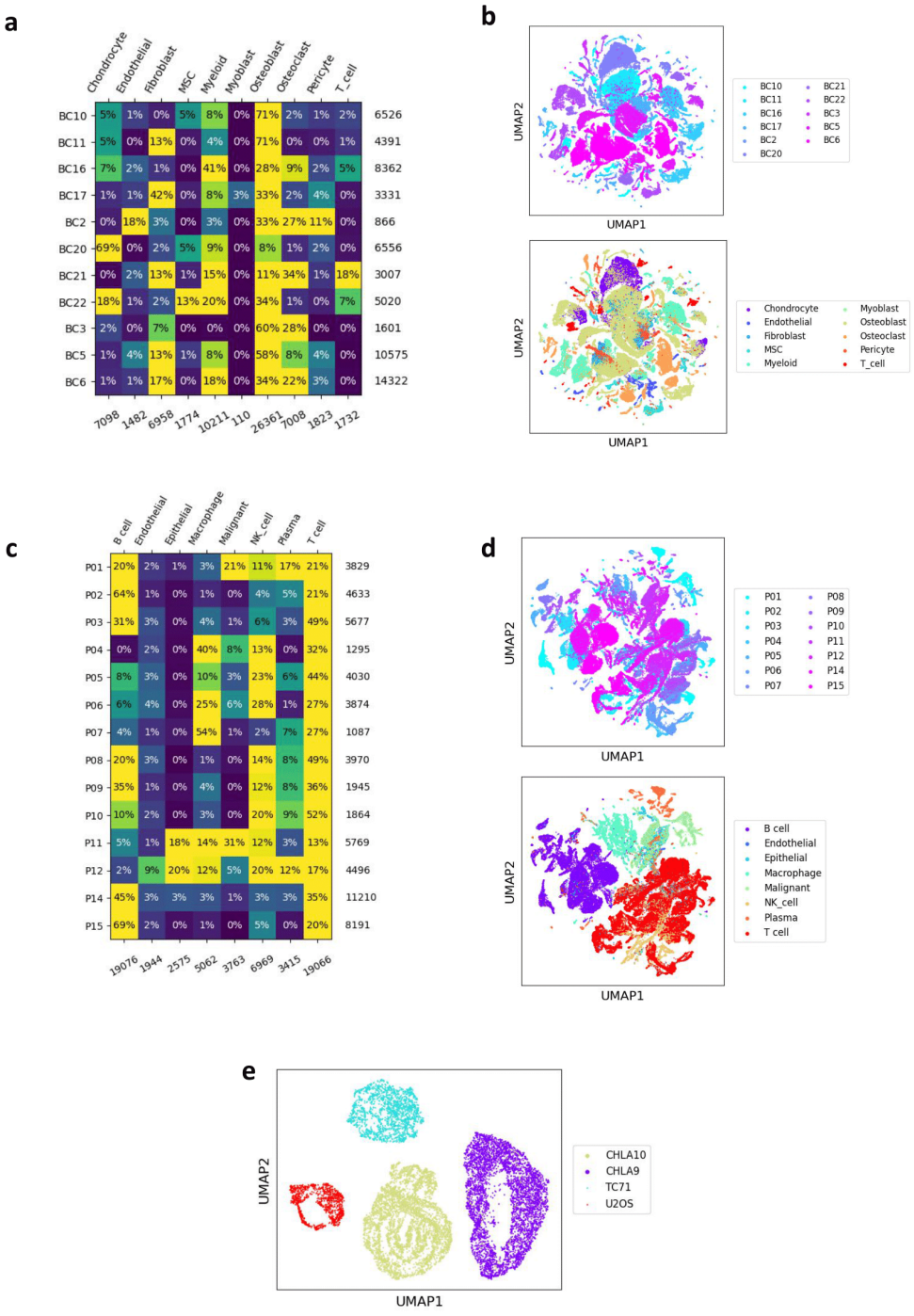

Figure 6.2: **Presentation of the datasets used in applications.** (a) Cell type distribution within the osteosarcoma datasets used for joint space integration. (b) UMAP representation of the osteosarcoma datasets within their common genes space, colored by dataset (top) and by author-provided cell type (bottom). (c) Cell type distribution within the nasopharyngeal carcinoma datasets used for gene space integration. (d) UMAP representation of the nasopharyngeal carcinoma datasets within their common genes space, colored by dataset (top) and by author-provided cell type (bottom). (e) UMAP representation of the osteosarcoma and Ewing sarcoma datasets used for cell cycle label transfer. *From (Fouché et al., 2023).*

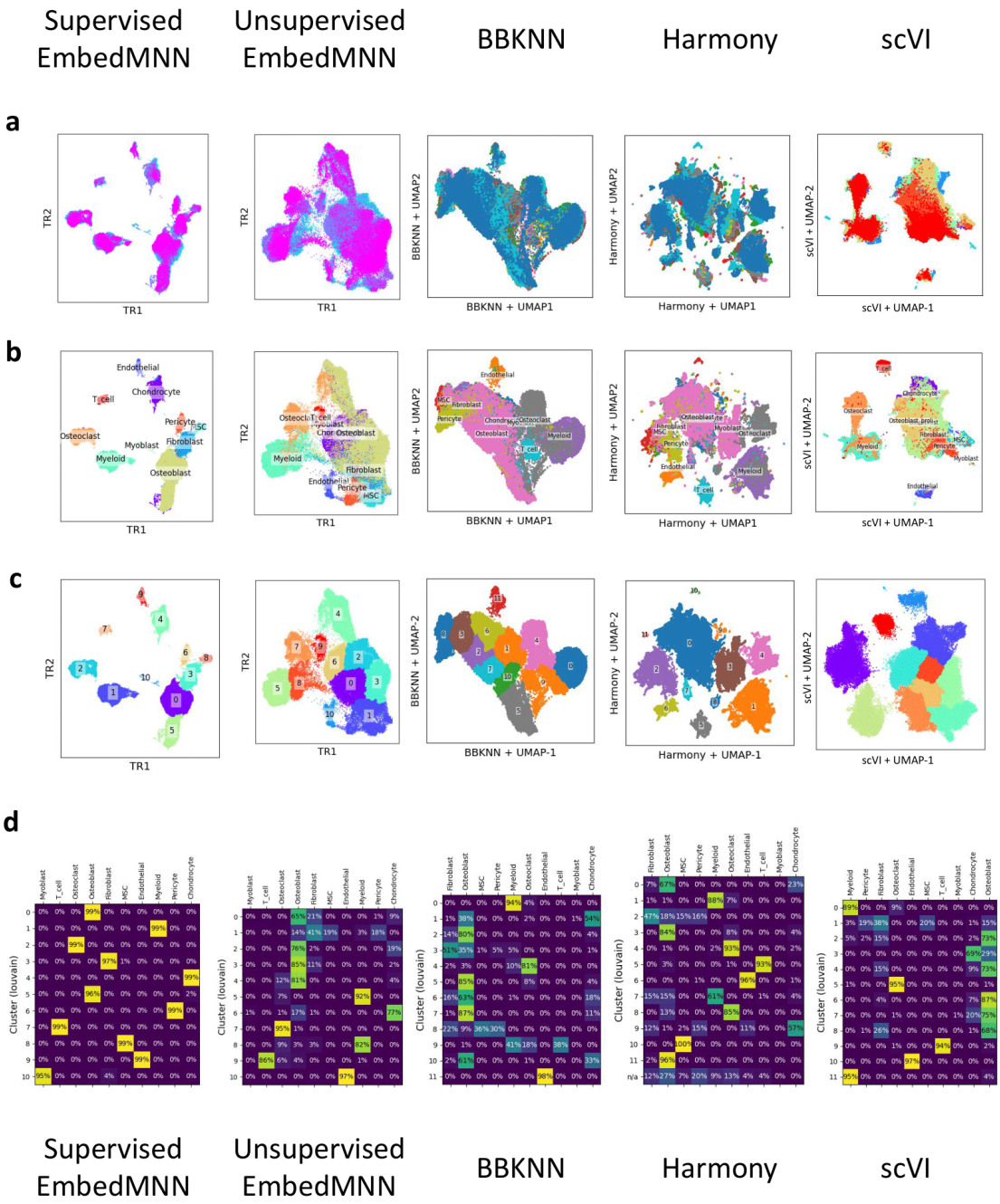

Figure 6.3: **Comparing the different joint analysis methods on osteosarcoma datasets.** (a) 2D embeddings of integration results, colored by dataset. (b) 2D embeddings of integration results, colored by author-provided cell type. (c) 2D embeddings of integration results, colored by Leiden clustering. (d) Cluster purity (percentage of cells belonging to cluster i with cell type j) *From (Fouché et al., 2023).*

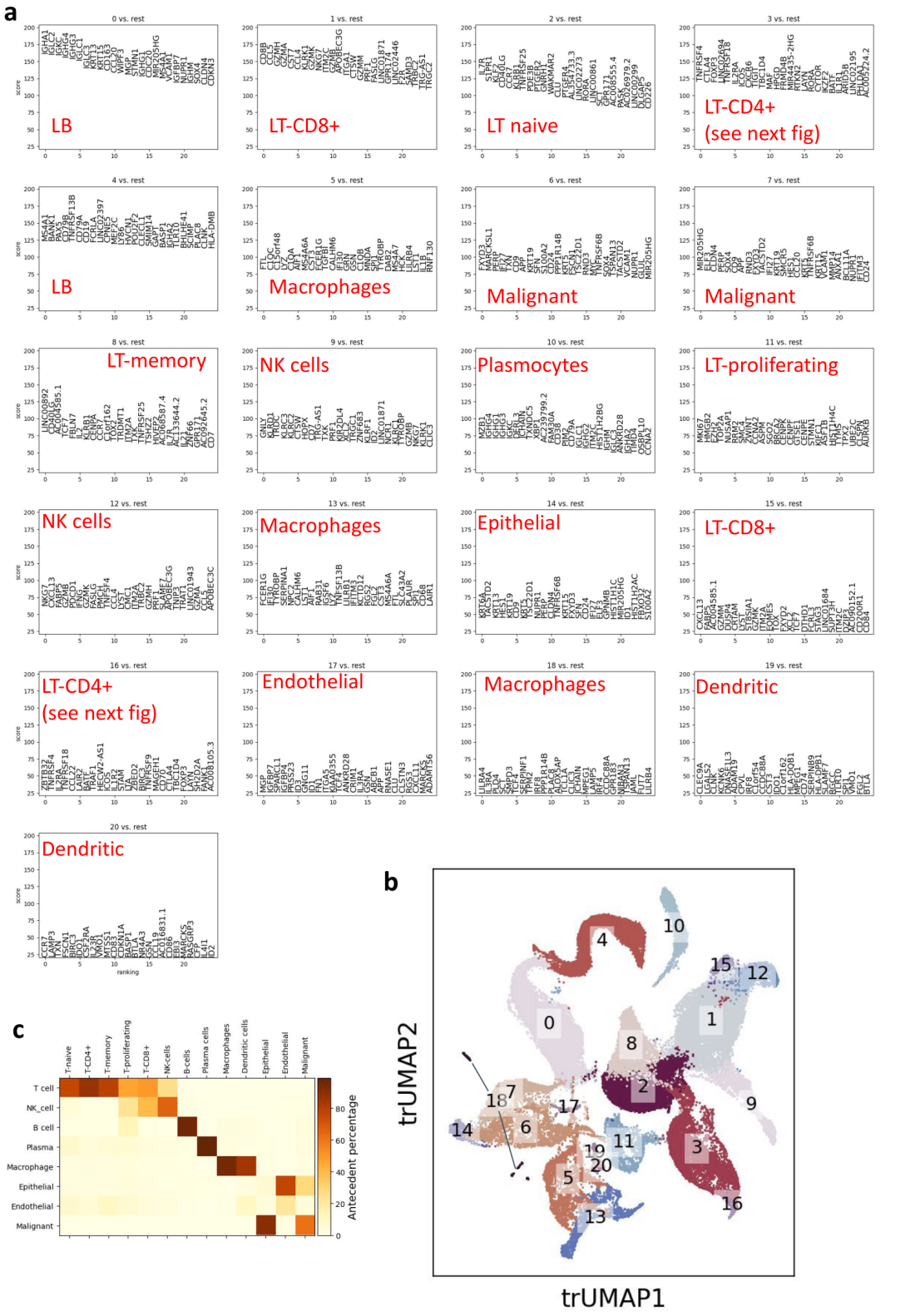

Figure 6.4: **Using gene space integration to improve dataset annotations.** (a) Most specific genes associated to each cluster according to a Wilcoxon rank-sum test. The proposed annotation is written in red. (b) Leiden clusters computed after gene space integration. (c) Annotation transfer rates indicate among cells with the new annotation, what fraction of them was of each previous label. *From (Fouché et al., 2023).*

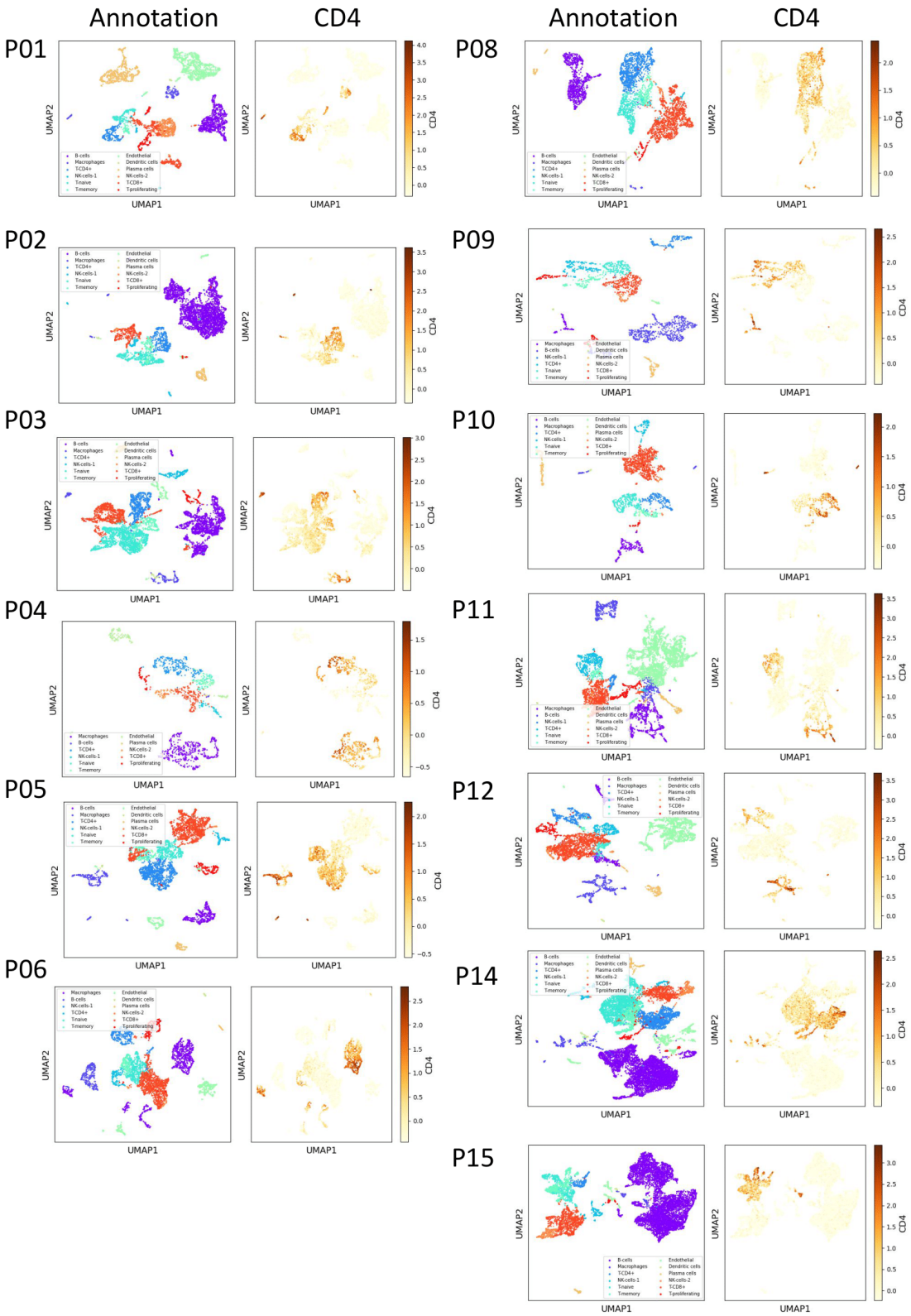

Figure 6.5: **Expression of CD4 co-localizes with the LT-CD4+ cluster** (CD4 gene is missing from the P07 dataset, therefore it is missing from the common gene space into which datasets are integrated). *From (Fouché et al., 2023).*

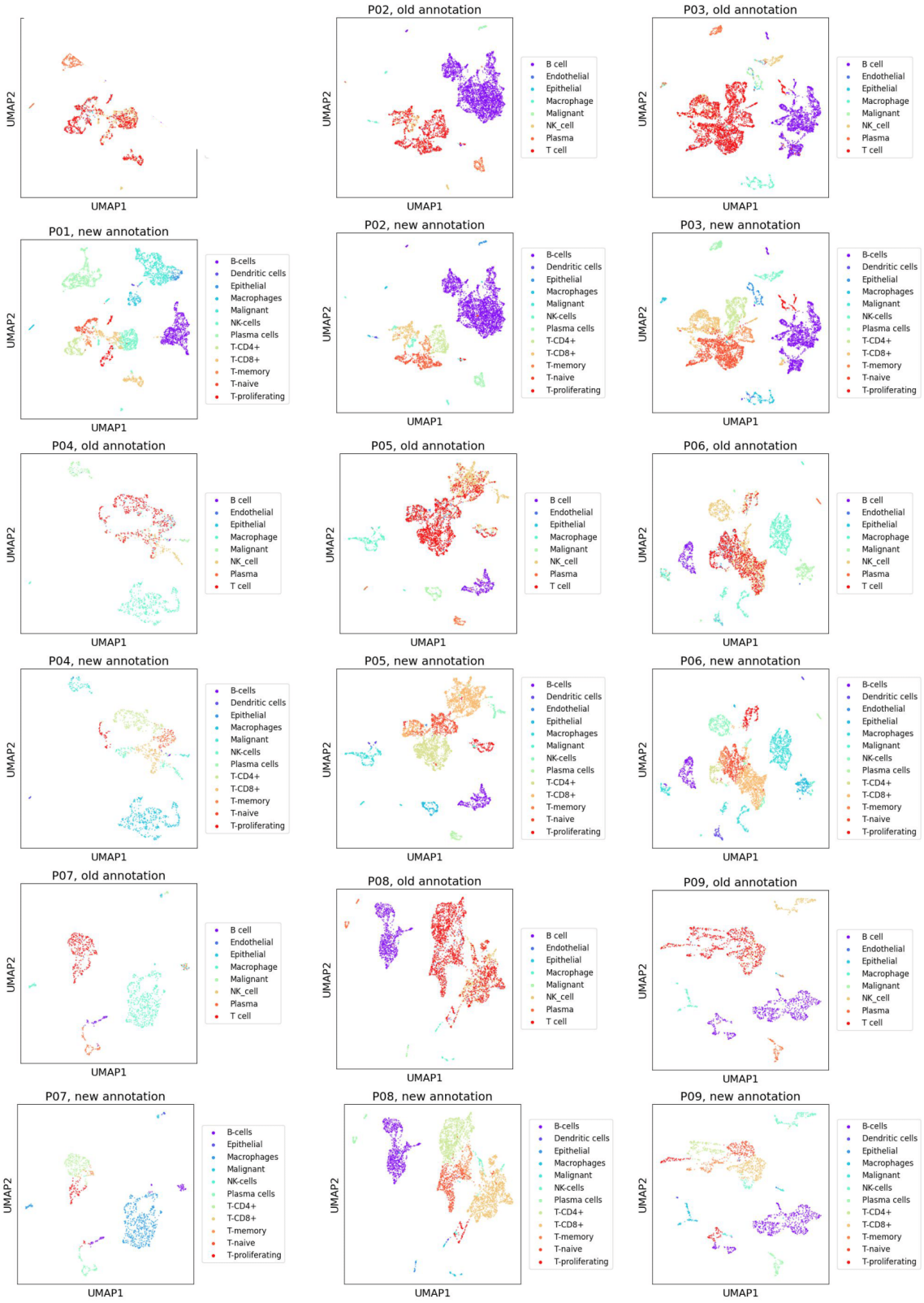

Figure 6.6: **UMAP representation of each nasopharyngeal carcinoma dataset, with old and new cell type annotations (P01 -> P09).** *From (Fouché et al., 2023).*

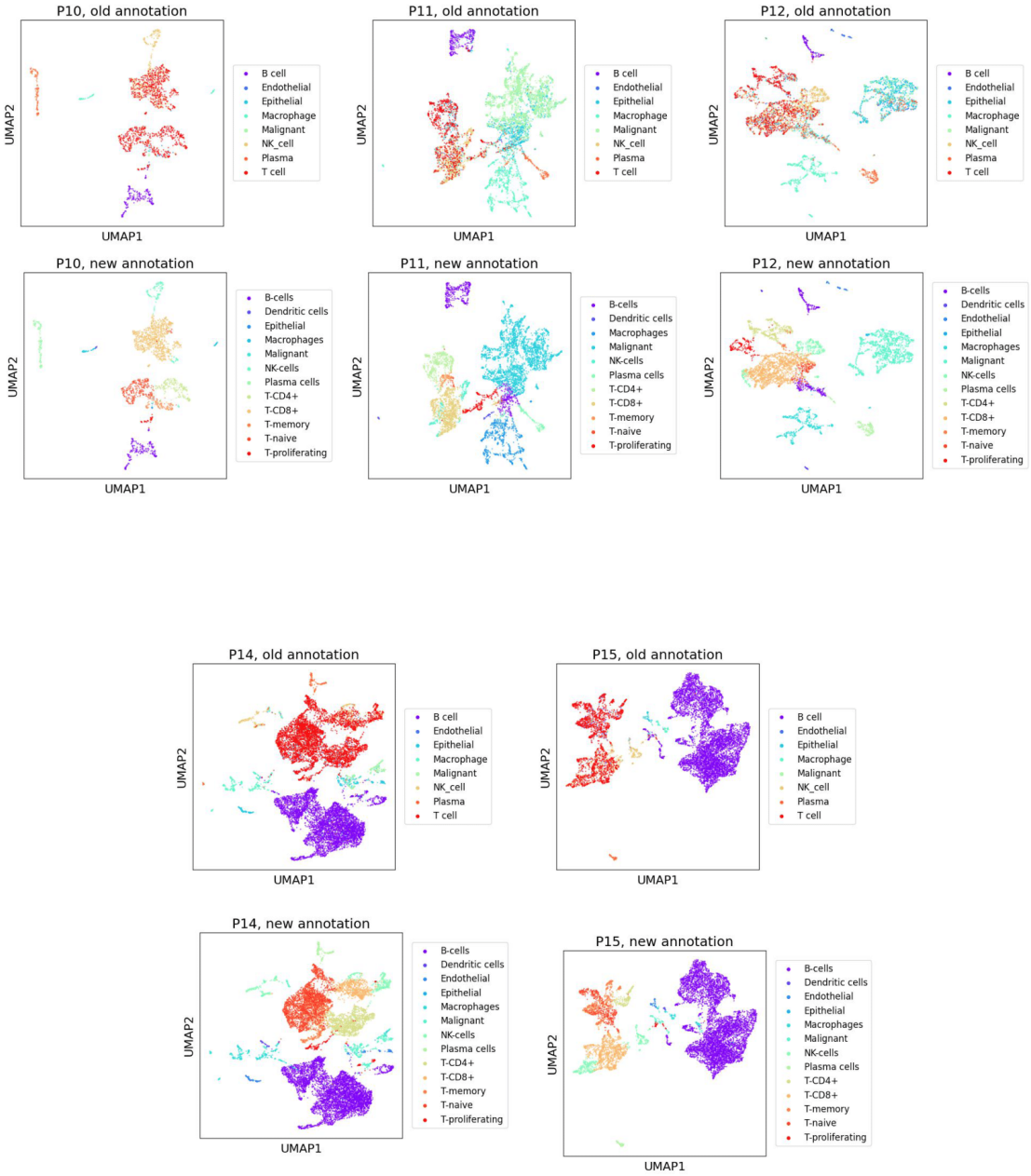

Figure 6.7: **SFigure 5B / UMAP representation of each nasopharyngeal carcinoma dataset, with old and new cell type annotations (P01 -> P09).** *From (Fouché et al., 2023).*

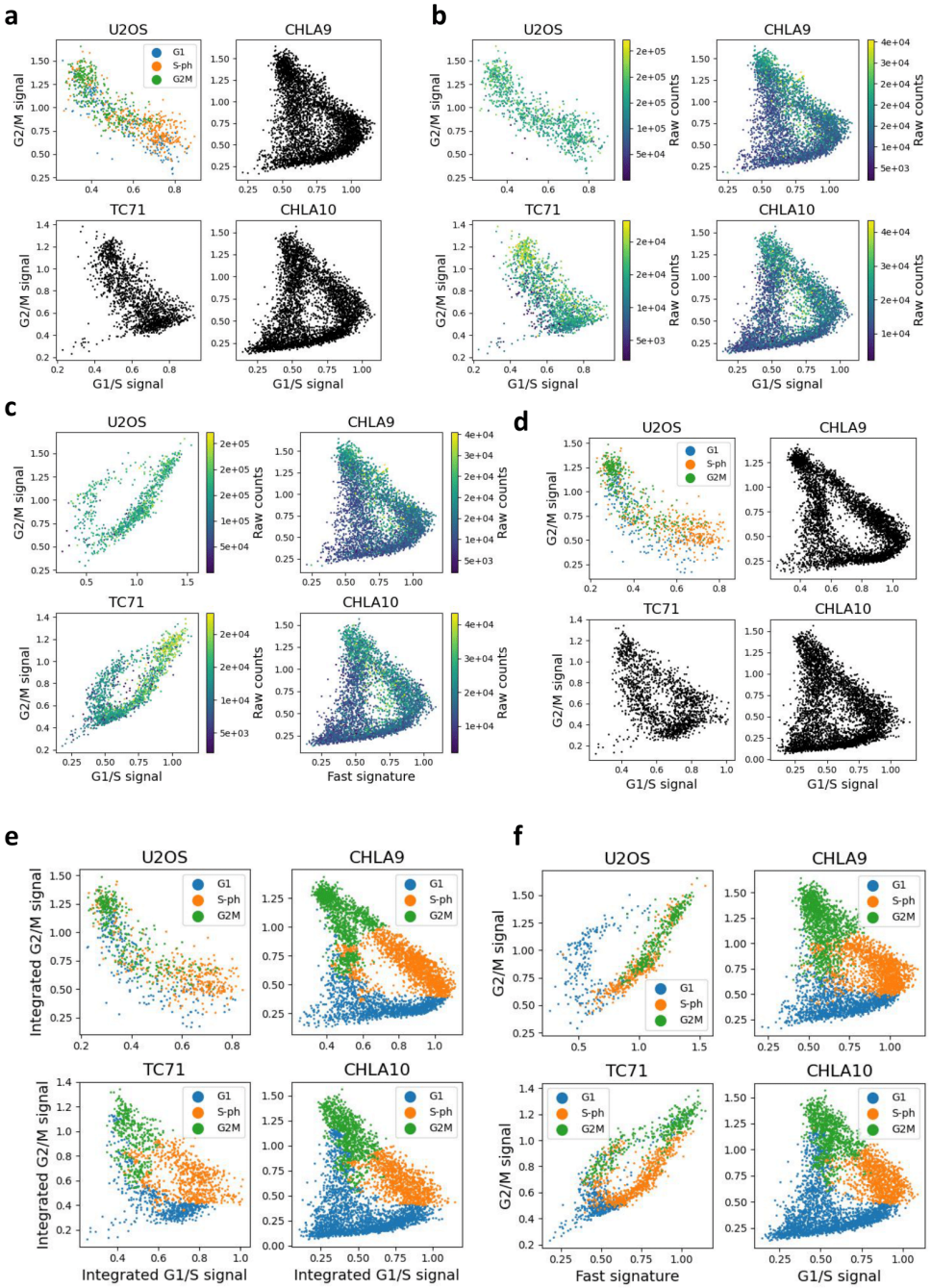

Figure 6.8: **MNN-based integration fails at the task of transferring cell cycle phase annotations across datasets.** (a) Representing each dataset in the G1/S signal versus G2/M signal basis. In this representation, it is difficult to see the cell cycle loop of fast-cycling datasets (U2OS and TC71). (b) Representing each dataset in the G1/S signal versus G2/M signal basis, colored by raw counts number. In this representation, it is difficult to identify the mitosis moment of fast-cycling datasets (U2OS and TC71). (c) Visualizing the cell cycle loop of each dataset, approximate positions of cell cycle phases are annotated. All datasets are colored according to individual cells' number of counts, which helps to see the point of mitosis (see the main figure for annotations). (d) Cells position in cell cycle space after MNN-based integration (CHLA10 is used as a reference during the integration). (e) MNN-based cell labeling, in cell cycle space after MNN-based integration. (f) MNN-based cell labeling, using an improved fast signature for fast cycling datasets to visualize the cell cycle loop. We observe here many annotation weaknesses. *From (Fouché et al., 2023).*

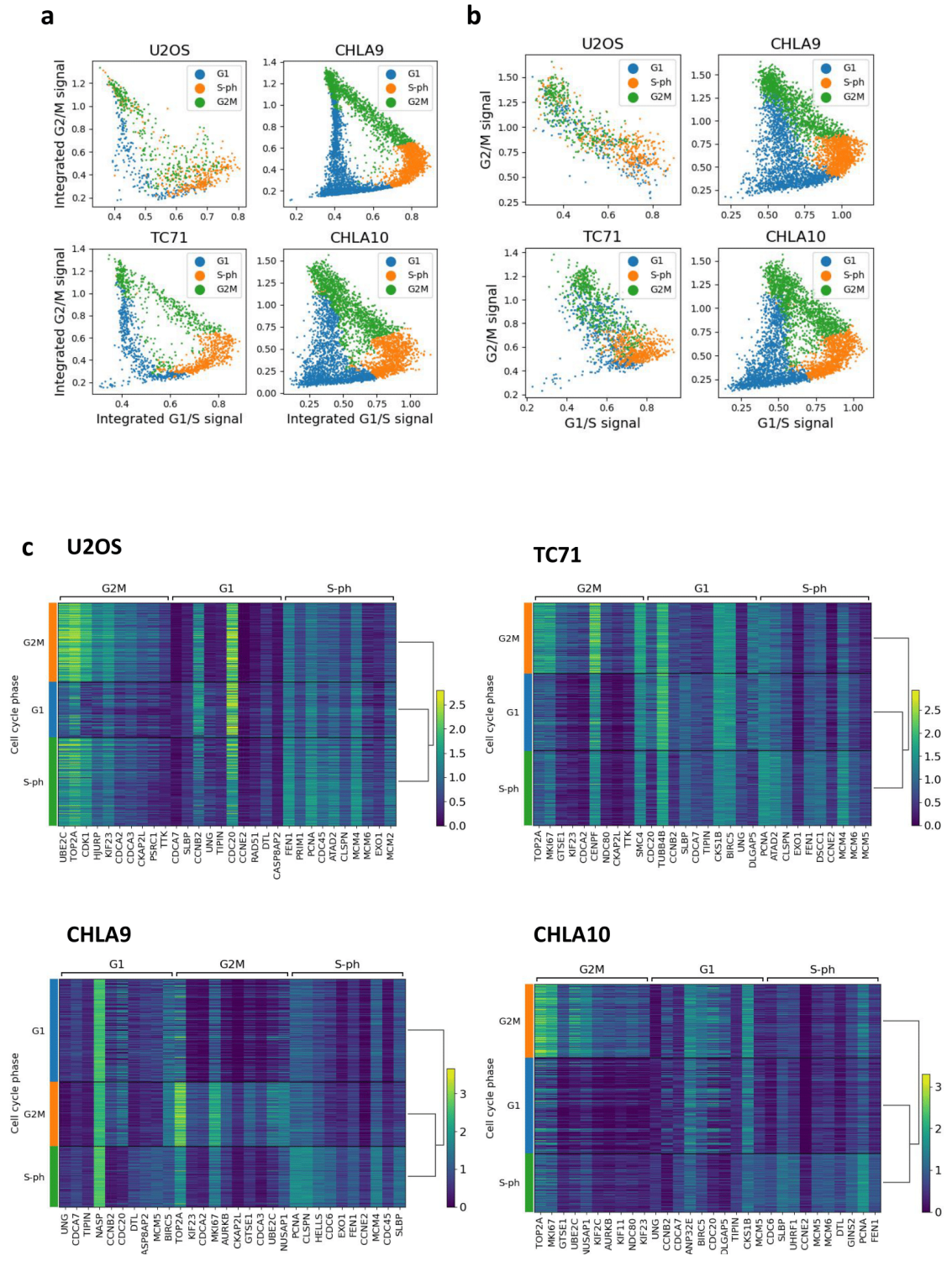

Figure 6.9: **Optimal transport-based integration allows for sound cell cycle phase labeling of osteosarcoma and Ewing sarcoma datasets.** (a) Transport-based cell labeling, in cell cycle space after Transport-based integration. (b) Transport-based cell labeling, within the initial G1/S signal versus G2/M signal basis. (c) Differential gene expression was performed using a Wilcoxon rank-sum test, showing the most specific genes associated with cells of each label. *From (Fouché et al., 2023).*

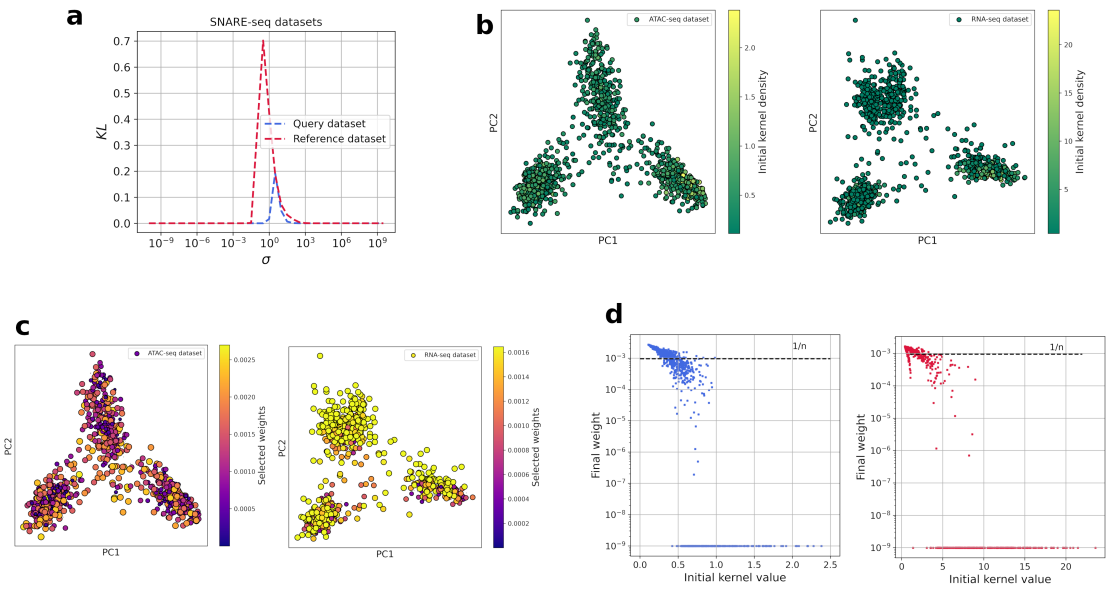

Figure 6.10: **Bandwidth and weightening selection of balanced scSNAREseq datasets.** (a) Bandwidth choice over scSNAREseq datasets. Right: Query dataset. Left: Reference dataset. (b) Point-wise Gaussian kernel density of each dataset before weights selection. Right: Query dataset. Left: Reference dataset. (c) Point-wise weights selection, illustrated by color and dot area. Right: Query dataset. Left: Reference dataset. (d) Relationship between initial point-wise Gaussian kernel density and selected weight. Right: Query dataset. Left: Reference dataset. *From (Fouché and Zinovyev, 2021).*

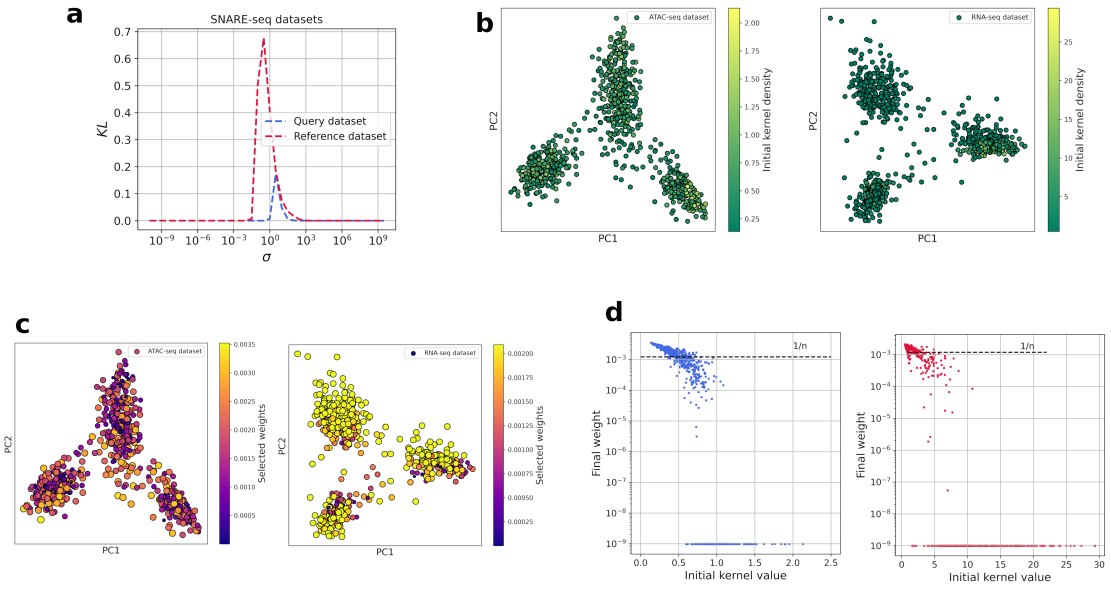

Figure 6.11: **Bandwidth and weightening selection of unbalanced scSNAREseq datasets.** (a) Bandwidth choice over scSNAREseq datasets. Right: Query dataset. Left: Reference dataset. (b) Point-wise Gaussian kernel density of each dataset before weights selection. Right: Query dataset. Left: Reference dataset. (c) Point-wise weights selection, illustrated by color and dot area. Right: Query dataset. Left: Reference dataset. (d) Relationship between initial point-wise Gaussian kernel density and selected weight. Right: Query dataset. Left: Reference dataset. *From (Fouché and Zinovyev, 2021).*

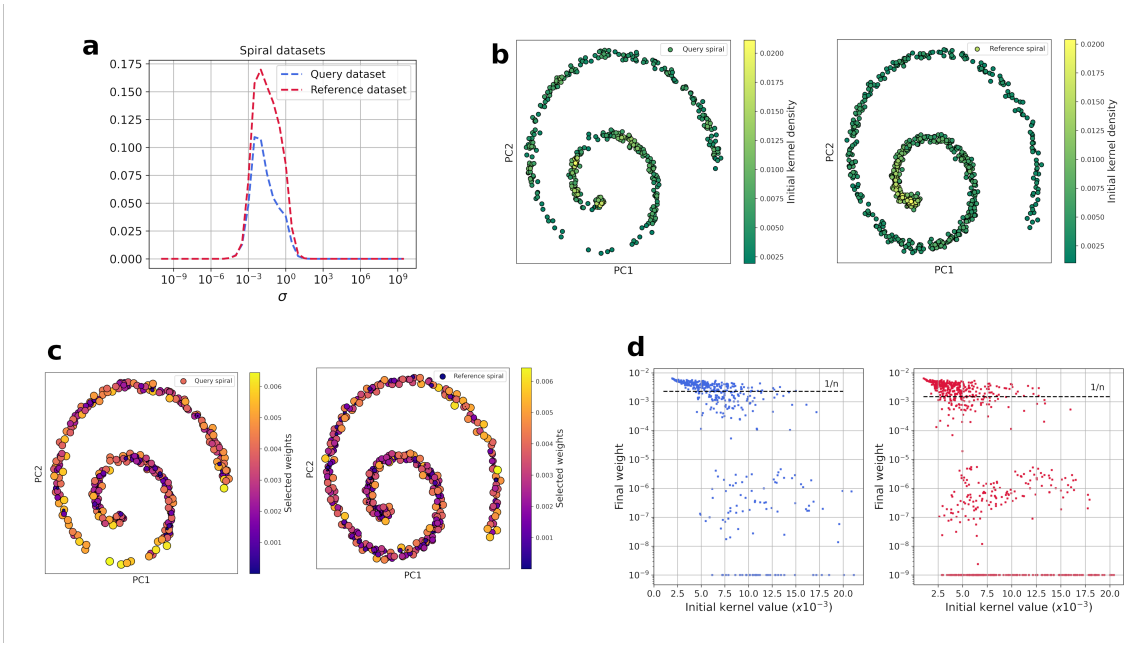

Figure 6.12: **Bandwidth and weightening selection of spirals datasets.** (a) Bandwidth choice over scSNAREseq datasets. Right: Query dataset. Left: Reference dataset. (b) Point-wise Gaussian kernel density of each dataset before weights selection. Right: Query dataset. Left: Reference dataset. (c) Point-wise weights selection, illustrated by color and dot area. Right: Query dataset. Left: Reference dataset. (d) Relationship between initial point-wise Gaussian kernel density and selected weight. Right: Query dataset. Left: Reference dataset. *From (Fouché and Zinovyev, 2021).*

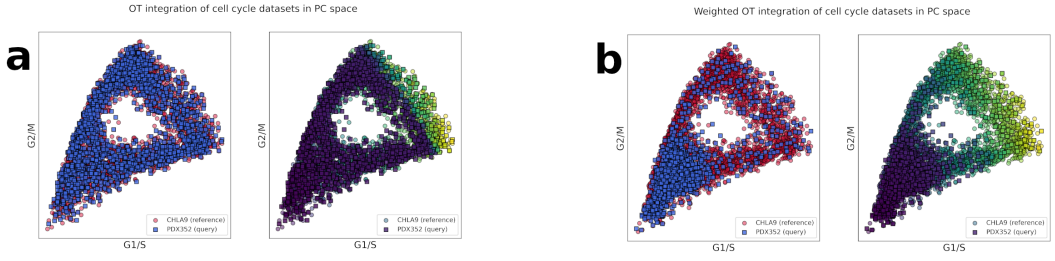

Figure 6.13: **Comparison of OT-based integration methods on Ewing sarcoma datasets embedded in cell cycle space. Left subpanes: colored by original dataset. Right subpanes: colored by initial position in the spiral, integration should preserve gradient.** (a) Unweighted optimal transport-based integration. (b) Weighted optimal transport-based integration. *From (Fouché and Zinovyev, 2021).*

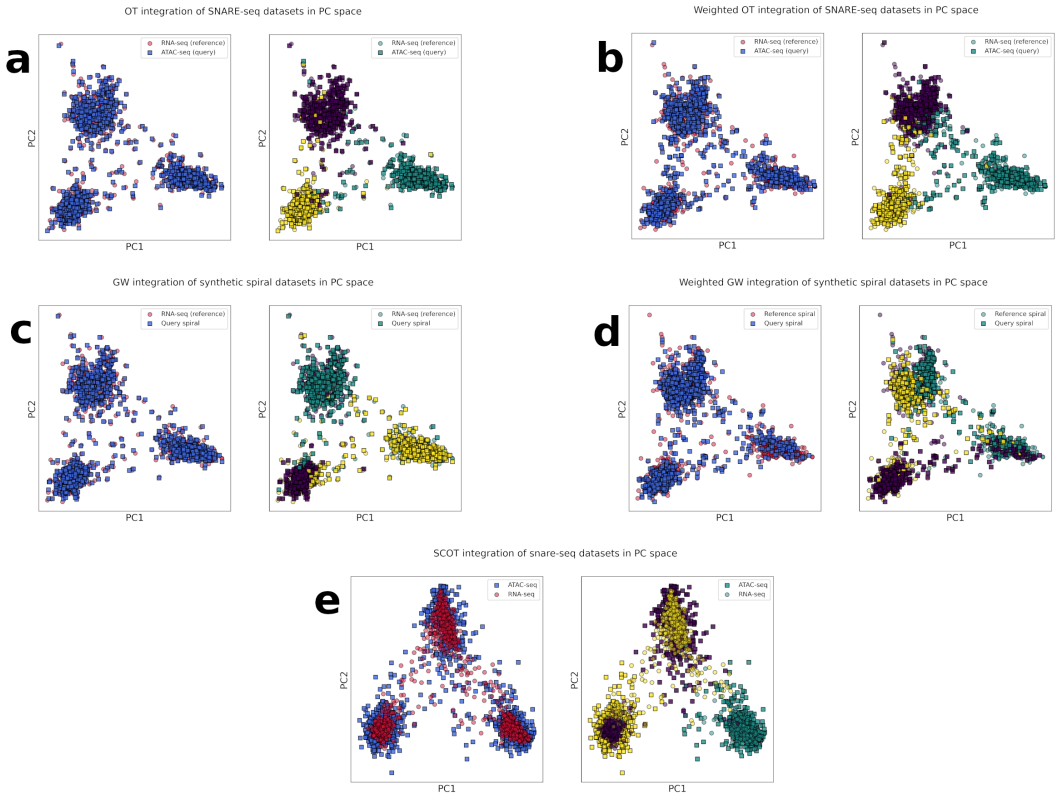

Figure 6.14: **Comparison of integration methods on balanced scSNAREseq datasets. Left subpanes: colored by original dataset. Right subpanes: colored by initial position in the spiral, integration should preserve gradient.** (a) Unweighted optimal transport-based integration. (b) Weighted optimal transport-based integration. (c) Unweighted GW-based transport-based integration. (d) Weighted GW-based transport-based integration. (e) Balanced SCOT integration. *From (Fouché and Zinovyev, 2021).*

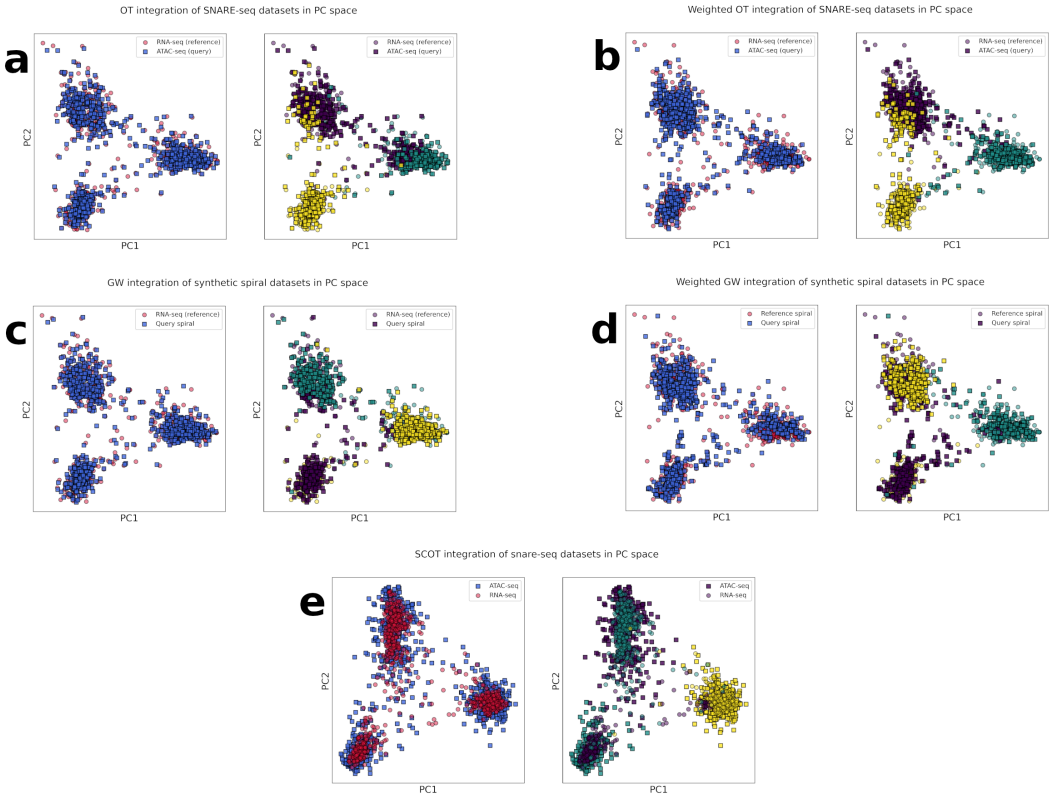

Figure 6.15: **Comparison of integration methods on unbalanced scSNAREseq datasets. Left subpanes: colored by original dataset. Right subpanes: colored by initial position in the spiral, integration should preserve gradient.** (a) Unweighted optimal transport-based integration. (b) Weighted optimal transport-based integration. (c) Unweighted GW-based transport-based integration. (d) Weighted GW-based transport-based integration. (e) Balanced SCOT integration. *From (Fouché and Zinovyev, 2021).*

RÉSUMÉ

Les tissus biologiques peuvent aujourd'hui être profilés à l’échelle de la cellule unique en utilisant le séquençage de l'ARN en *single-cell*, ce qui produit de grands jeux de données décrivant le paysage transcriptionnel des cellules individuelles. En nous appuyant sur des algorithmes d'apprentissage automatique, nous avons développé des méthodes automatisées pour faciliter l'intégration, l'analyse et l'interprétation de ces jeux de données, dont nous montrons des applications médicales dans l'étude des sarcomes d'Ewing, un type de cancer pédiatrique des os. En particulier, nous proposons un nouveau *framework* informatique hautement modulaire pour l'intégration des données, appelé *transmorph*, qui permet à l'utilisateur de construire et de comparer des pipelines d'intégration de données.

MOTS CLÉS

single-cell RNA-seq, apprentissage automatique, optimisation, transport optimal, intégration de données, apprentissage non-supervisé, déconvolution, sarcome d'Ewing, biologie computationnelle, cycle cellulaire,

ABSTRACT

Biological tissues can nowadays be profiled at the single-cell level using single-cell RNA sequencing, which yields large datasets describing the transcriptional landscape of the individual cells. Leveraging machine learning algorithms, we developed automated methods to facilitate the integration, the analysis and the interpretation of such datasets, finding medical applications in the study of Ewing sarcomas, a pediatric bone cancer type. In particular, we propose a new, highly modular computational framework for data integration called *transmorph* that allows the user to build and benchmark data integration pipelines.

KEYWORDS

single-cell RNA-seq, machine learning, optimization, optimal transport, data integration, unsupervised learning, deconvolution, Ewing sarcoma, computational biology, cell cycle, tumor heterogeneity