UNIVERSITY OF CALIFORNIA,
IRVINE

Integrating Single-Cell Transcriptomics Data with Spatial Imaging Data

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Mathematics

by

Floyd Maseda

Dissertation Committee:
Professor Qing Nie, Chair
Professor German Enciso
Professor Long Chen

2021

DEDICATION

Dedicated to my wife, HaeRee Lee, for her unwavering support.

Dedicated to my parents, Jeanna Rutledge and Floyd Maseda, Sr., for instilling in me the importance of education from a young age.

Dedicated to my wife's parents, Mija Kim and DaeHee Lee, for supporting us both.

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

ACKNOWLEDGMENTS

I would like first and foremost to thank my advisor and committee chair, Dr. Qing Nie, for their invaluable support and mentorship throughout my time at UCI. Their insightful feedback and support, even at times when I may not have deserved it, have been instrumental to my success as a PhD student and have provided a firm footing for a future career. I would further like to thank committee members Dr. German Enciso and Dr. Long Chen for their insightful comments and suggestions related to this dissertation, as well as Dr. Xiaohui Xie and Dr. Maksim Plikus for serving on my advancement committee.

Another invaluable aide along my PhD journey has been Dr. Zixuan Cang, who took me under their wing and provided a plethora of useful suggestions and ideas, many of which contribute to the foundation of this dissertation. Without their help, my graduation would not have been possible. The same can be said of Dr. Chris Rackauckas and Dr. Seth Figueroa, who provided assistance and insight during my time in the Nie Lab.

Many other professors have provided support for me throughout my career in higher education, beginning with Dr. Khin Maung Maung, Dr. James Lambers, Dr. Larry Mead, and Dr. Chris Winstead at the University of Southern Mississippi. Dr. Maung in particular was instrumental to my success as an undergraduate researcher and has continued to be one of my most trusted advisors throughout my education. Dr. Lambers has also been an indispensable mentor while I was at USM, as well as during my (difficult) transition from physics to mathematics at UCI. During my master's degree at the University of Alabama, Dr. Ben Harms, Dr. Allen Stern, and Dr. Conor Henderson were instrumental to my success, providing knowledge and guidance that allowed me to continue reaching higher to achieve my goals. Finally, at the University of California, I would like to acknowledge the teaching skills of Dr. Umut Isik, without whom I would not have even made it through the first year of study in the math program. During this difficult transitionary time, Dr. Patrick Guidotti also served as an irreplaceable advisor in both academia and on a personal level.

In addition, there have been many friends and colleagues who have provided academic and moral support fundamental to my completion of this degree. Above all, I would like to thank my wife HaeRee Lee, to whom this dissertation is dedicated, for staying by my side and providing encouragement during this challenging journey; I look forward to spending the rest of our lives together and cannot wait to see what the future has in store. I further would like to thank Dr. Seulip Lee and SeongHee Jeong for their continued friendship during our time together at UCI and beyond. Other treasured friends who I would like to recognize include Dr. Lara Clemens, Jada Cruce, Sarah Bogen, Dr. Sean Horan, and Zach Drumbor for their companionship and collaboration during our frequent late night study sessions.

Finally, I would like to thank the National Institutes of Health, the National Science Foundation, and the Simons Foundation for providing the funding necessary to complete this dissertation. I would also like to thank the University of Southern Mississippi Honors College and the Ronald E. McNair Post-Baccalaureate Achievement Program for providing funding, guidance, and mentorship earlier in my educational career.

VITA

Floyd Maseda

EDUCATION

Doctor of Philosophy in Mathematics **2021**
University of California, Irvine *Irvine, California*

Master of Science in Mathematics **2017**
University of California, Irvine *Irvine, California*

Master of Science in Physics **2014**
University of Alabama *Tuscaloosa, Alabama*

Bachelor of Science in Physics **2012**
University of Southern Mississippi *Hattiesburg, Mississippi*

RESEARCH AND TEACHING EXPERIENCE

Graduate Research Assistant **2017–2021**
University of California, Irvine *Irvine, California*

Teaching Assistant **2015–2021**
University of California, Irvine *Irvine, California*

Teaching Assistant **2012–2014**
University of Alabama *Tuscaloosa, Alabama*

Undergraduate Researcher **2011–2012**
University of Southern Mississippi *Hattiesburg, Mississippi*

PUBLICATIONS

Maseda F, Cang Z and Nie Q (2021) DEEPsc: A Deep Learning-Based Map Connecting Single-Cell Transcriptomics and Spatial Imaging Data. *Front. Genet.* 12:636743.

SOFTWARE

DEEPsc `https://github.com/fmaseda/DEEPsc`
System-adaptive, deep learning-based method to impute spatial information onto a scRNA-seq dataset from a given spatial reference atlas.

ABSTRACT OF THE DISSERTATION

Integrating Single-Cell Transcriptomics Data with Spatial Imaging Data

By

Floyd Maseda

Doctor of Philosophy in Mathematics

University of California, Irvine, 2021

Professor Qing Nie, Chair

The advent of sophisticated single-cell RNA sequencing (scRNA-seq) techniques now allows investigation of the transcriptomic landscape of tens of thousands of genes across tissues at the resolution of individual cells. However, scRNA-seq necessitates dissociation of the sample, thereby destroying any spatial context which can be crucial to the understanding of cellular development and dynamics. The loss of spatial information in scRNA-seq data can be partially mitigated by referring to known spatial expression patterns of a small subset of genes, termed a "spatial reference atlas." Several recent computational methods have been developed to impute spatial data onto existing scRNA-seq datasets to achieve individual-cell resolution while retaining the spatial arrangement. In Chapter 2, we discuss a novel deep learning-based, system-adaptive method (DEEPsc) of integrating non-spatial scRNA-seq data with spatial imaging data. DEEPsc and other mapping methods rely on a high quality reference atlas which must be compiled from raw images into a useable form. In Chapter 3, we introduce AtlasGeneratorOT, a novel software suite which uses techniques in optimal transport theory to more fully automate the creation of a spatial reference atlas for use with DEEPsc and other integration methods. In Chapter 4, we extend AtlasGeneratorOT with additional capabilities for three-dimensional biological systems imaged in serial slices, allowing for alignment of and interpolation between slices to provide a more cohesive, comprehensive atlas than previously available.

Chapter 1

Introduction

Although all cells of a biological system have access to the same genetic blueprint, the function and fate of each cell is influenced by many other factors, which can activate or suppress the expression of genes in cells of different types at different locations within the system. The study of how and why certain genes are expressed in certain contexts and not in others is known as *transcriptomics*.

Originally coined as a term in 1996 by Charles Auffray,[5] the field of transcriptomics developed rapidly throughout the late 1990s and early 2000s. With the advent of sophisticated single-cell RNA sequencing (scRNA-seq) techniques in the late 2000s and 2010s,[6] it is now possible to investigate the transcriptomic landscape of tens of thousands of genes across tissues at the resolution of individual cells.[7, 8] However, a drawback to scRNA-seq methods is the necessity of dissociating the cells in the biological sample under study, thereby destroying any spatial context which can be crucial to the understanding of cellular development and dynamics.[9]

A common task related to scRNA-seq datasets is to perform analysis such as unsupervised clustering of cells and identifying marker genes with known spatial expression associated with

each cell cluster.[10, 11] Several existing methods further attempt to impute a pseudospatial or pseudotemporal axis onto the data;[12, 13, 14, 15] however, little related to physical space is immediately discernible from scRNA-seq data alone.

The loss of spatial information in scRNA-seq data can be partially mitigated by referring to spatial staining data.[16, 17] Several recent computational methods have been developed to impute spatial data onto existing scRNA-seq datasets through analyzing known spatial expression patterns of a small subset of genes, termed a "spatial reference atlas."[1, 2, 18] Another promising solution is the emerging field of spatial transcriptomics, which has led to the development of methods that obtain *in situ* spatial expression patterns of multiple genes simultaneously.[19, 20, 21, 22, 23, 24]

Compared to scRNA-seq, current spatial techniques often cover fewer cells or genes or with a suboptimal resolution and depth. It is therefore a trending theme to combine the strengths of both methods to achieve a high coverage and individual-cell resolution while retaining or recovering the spatial arrangement.[9, 11] Many existing spatial integration methods rely on predefined algorithms for computing a correspondence score between cells in a scRNA-seq dataset and locations in a given biological system.[1, 2, 18, 25, 26] Other existing methods are more broadly focused on integration of datasets in general, spatial data being only one among many inputs.[27, 28, 29, 30, 31]. Since the spatial characteristics of different biological systems could be significantly different, we aim to develop a system-adaptive integration method specifically designed for imputing spatial information onto scRNA-seq data.

Another trending area of research is that of machine learning and deep learning, particularly into the application of various well-established ML/DL techniques to biological data. Many ML/DL-based methods have been developed for the task of transferring high-level information such as cell types between datasets by formulating a supervised learning problem with the high-level information being the target.[32, 33, 34, 35, 36, 37, 38, 39] Other forays of machine learning into the realm of biology include advances in the field of metric learning,

specifically the determination of a pseudometric between different modalities of data,[40] as well as increasing interest in applying deep learning to metric learning.[41, 42]

In Chapter 2, we develop a system-adaptive deep learning-based method (DEEPsc) for imputing spatial data onto scRNA-seq data. The training of a DEEPsc network can be regarded as a general metric learning task,[43] wherein we learn a nonlinear metric between cells in the scRNA-seq dataset and positions in a spatial reference atlas. In addition, we develop a comprehensive measure, which was previously lacking, for evaluating how well a given method maps scRNA-seq data to known spatial origins, called a performance score. Using this score on four biological systems, including one (the murine follicle[12]) for which we generated a novel reference atlas, we show that DEEPsc maintains a comparable accuracy to four existing methods while achieving a better balance between precision and robustness.

In order for DEEPsc or any other reference atlas-based method of integrating spatial data with scRNA-seq data to perform adequately, the reference atlas should be of a high quality. Many such reference atlases exist of varying spatial resolutions numbers of genes (cf. Table 1 in [18]), and more are being created constantly as fast, high-resolution imaging technology becomes more ubiquitous.[19, 20, 21, 22, 23, 24] However, the creation of a novel reference atlas from a collection of images is a nontrivial task which is often performed manually. Besides being slow and tedious, a manual creation process may potentially introduce unintentional biases into the atlas, which may influence the training of a DEEPsc network and therefore the spatial mapping of future scRNA-seq datasets.

There have been many recent advancements in the field of optimal transport, particularly in the area of graph matching and cross-domain alignment,[44, 45] as well as advances directly involving scRNA-seq data itself.[46, 47, 48] In Chapter 3, we introduce a novel automation platform, AtlasGeneratorOT, which, with minimal user input, allows for the creation of a reference atlas from a collection of images using techniques based in optimal transport theory. We introduce a novel method of accelerating the computation of a coupling matrix for various

optimal transport formulations in the context of matching point clouds and produce novel reference atlases of fifteen serial slices of the murine neural crest using an imaging dataset provided by Soldatov, et al.[4]

Like the murine neural crest dataset, it is often the case that a three-dimensional biological system is imaged in multiple two-dimensional slices which may then need to be recombined to form a coherent three-dimensional structure. We expand on the capabilities of AtlasGeneratorOT in Chapter 4, describing a process by which disparate two-dimensional reference atlases can be aligned into a common geometry, producing in the process a detailed three-dimensional atlas of the murine neural crest. We further introduce an interpolation method, also based in optimal transport theory, which allows for continuous interpolation between serial slices of a biological system, thus allowing for the creation of a high-resolution, three-dimensional reference atlas of a biological system, even when only a few serial slices are provided.

Taken together, the techniques introduced in this dissertation can drastically improve and optimize the workflow of creating an unbiased, high-resolution reference atlas from a collection of images of a small number of genes, and using that reference atlas to determine the spatial origin of cells in an scRNA-seq dataset. This improved workflow can facilitate many advances in the understanding of cellular development and dynamics.[9]

Chapter 2

DEEPsc: A Deep Learning-Based Map Connecting Single-Cell Transcriptomics and Spatial Imaging Data

This chapter is a reprint of the material as it appears in Frontiers in Genetics.[49] The co-authors listed in this publication directed and supervised research which forms the basis for this chapter.

2.1 Background

While cells of a biological system have access to the same genetic blueprint, they navigate through different developmental paths toward various cell fates. These diverse fate programs of cells are controlled by their own states, interactions with spatially neighboring cells, and

other environmental cues.[50] To decipher the processes of cell fate acquisitions, observations of the transcriptomics with single-cell resolution in spatial context are desired. The advent of sophisticated single-cell RNA sequencing (scRNA-seq) techniques now allows investigation of the transcriptomic landscape of tens of thousands of genes across tissues at the resolution of individual cells.[7, 8] However, a drawback to scRNA-seq methods is the necessity of dissociating the sample in question, thereby destroying any spatial context which can be crucial to the understanding of cellular development and dynamics.[9] In current common workflows of scRNA-seq data analysis, unsupervised clustering of cells is carried out, followed by identifying marker genes associated with each cell cluster.[10] While the list of marker genes for each cell cluster can be screened for genes associated with known spatial regions to estimate the spatial origin of the cluster, the spatial arrangement of individual cells remains unclear.[10, 11] Several existing methods attempt to impute a pseudospatial or pseudotemporal axis onto the data;[12, 13, 14, 15] however, little related to physical space is immediately discernible from scRNA-seq data alone.

The loss of spatial information in scRNA-seq data can be partially mitigated by referring to spatial staining data.[16, 17] Another promising solution is the emerging spatial transcriptomics methods such as osmFISH[19], MERFISH[20], seqFISH[21], seqFISH+[22], STARmap[23], and Slide-seq[24] that obtain *in situ* spatial expression patterns. Compared to scRNA-seq, current spatial techniques often cover fewer cells or genes or with a suboptimal resolution and depth. It is therefore a trending theme to combine the strengths of both methods to achieve a high coverage and individual-cell resolution while retaining the spatial arrangement.[11, 9] Due to these differences among the scRNA-seq and spatial techniques, and biological systems, it is challenging to derive a generally applicable computation method to integrate the two kinds of data.

Several recent computational methods have been developed to impute spatial data onto existing scRNA-seq datasets through analyzing known spatial expression patterns of a small

subset of genes, termed a "spatial reference atlas." Seminal methods were developed independently by Achim et al.[18] and Satija et al.[1] and were applied to the *Platynereis dumerilii* brain and zebrafish embryo, respectively, using binarized reference atlases derived from *in situ* hybridization (ISH) images. DistMap, another method that uses a binarized ISH-based reference atlas, was developed by Karaiskos et al.[2] and applied to the *Drosophila* embryo. Achim et al.[18] use an empirical correspondence score between each cell-location pair based on the specificity ratio of genes. Satija et al.[1] (Seurat v1) fits a bimodal mixture model to the scRNA-seq data and then projects cells to their spatial origins using a probabilistic score. DistMap applies Matthew's correlation coefficients to the binarized spatial imaging and scRNA-seq data to assign a cell-location score.[2] Several methods have also been developed which use spatial reference atlases directly measuring the RNA counts that are comparable to scRNA-seq data without binarization.[25, 26] More recently, computational methods have been developed for imputing gene expression in spatial data,[51], transferring cell type label from scRNA-seq data to spatial data,[52, 53, 54] *de novo* spatial placement of single cells,[55] and inferring spatial distances between single cells.[47]

In addition to the methods designed specifically for integrating spatial data and scRNA-seq data, other computational methods have been developed recently for general data integration. Such methods focus on the general task of integrating RNA sequencing datasets obtained from the same biological system through different technologies, *in situ* data being one possibility among many, into one large dataset offering a more complete description of the system under study. These methods include newer versions of Seurat[27, 28], LIGER[29], Harmony[30], and Scanorama[31] which are mainly based on correlation analyses and matrix factorizations. Another more specific task is to transfer high-level information such as cell types between datasets. Many machine learning- and deep learning-based methods have been developed for this task by formulating a supervised learning problem with the high-level information being the target.[32, 33, 34, 35, 36, 37, 38, 39]

Since the spatial characteristics of different biological systems could be significantly different, we aim to develop a system-adaptive method specifically designed for imputing spatial information onto scRNA-seq data. To this end, unlike other spatial integration methods that use predefined algorithms for computing scores, we learn a specialized correspondence score between cells and locations for a given biological system. This can then be regarded as a general metric learning task.[43] In addition to linear methods that learn a pseudometric,[40] there has been increasing interest in applying deep learning to metric learning.[41, 42] These methods are mostly designed for cases where the pair of data points to be compared are in the same space. Though the common genes from the spatial data and scRNA-seq data are used here, directly treating them as in the same space may cause inaccuracy due to differences in the original datasets such as scaling and noise.

Here we develop a system-adaptive deep learning-based method (DEEPsc) for imputing spatial data onto scRNA-seq data. A DEEPsc network accepts a low-dimensional feature vector corresponding to a single position in the spatial reference atlas along with a corresponding feature vector of the gene expression of a single cell and returns a likelihood the input cell originated from the input position. The network is trained and validated using positions in the spatial reference atlas as simulated scRNA-seq data. The network is also validated through the task of predicting the scRNA-seq data from the spatial reference atlas or the other way around. In addition, we implemented several strong baseline methods using different norms and linear metric learning for benchmark comparison. We further develop a comprehensive measure, which was previously lacking, for evaluating how well a given method maps scRNA-seq data to known spatial origins, called a performance score. This score contains three components that measure the accuracy, precision, and robustness of a method, respectively. Using this score on four biological systems, we show that DEEPsc maintains a comparable accuracy to four existing methods while achieving a better balance between precision and robustness.

2.2 Results

2.2.1 A deep-learning based method to connect scRNA-seq datasets and spatial imaging data

Given any spatial reference atlas consisting of binary or continuous gene expression levels for a biological system on a set of locations with known spatial coordinates, and a scRNA-seq dataset consisting of binary or continuous gene expression levels for the same biological system, we introduce a **D**eep-learning based **E**nvironment for the **E**xtraction of **P**ositional information from **sc**RNA-seq data (DEEPsc) to impute the spatial information onto the scRNA-seq data.

In DEEPsc, we first select a common set of genes from the reference atlas and scRNA-seq data, then perform dimensionality reduction via principal component analysis (PCA) on the reduced reference atlas to shorten training time (Figure 2.1A). The scRNA-seq data is then projected into the same PCA space on which we learn a metric for comparison between cells and spatial positions. The DEEPsc network accepts a concatenated feature vector for a single cell and a single position and returns a likelihood the input cell originated from the input position. The network contains two fully connected hidden layers with N nodes each, where N is the number of principal components kept from PCA, and a single node in the output layer. Sigmoid activation functions are applied to each node, including the output node, so that the resulting output is in $[0, 1]$ and can be interpreted as a likelihood that the input cell originated from the input spatial position. To train the DEEPsc network, we use the spatial position feature vectors as simulated scRNA-seq data for comparison (Figure 2.1B). Each simulated cell is compared pairwise with every position in the spatial reference atlas; if the simulated cell is an exact match to a given position, the target output is 1 (a high likelihood of origin), and if the simulated cell and chosen position are not an exact

match, the target output is 0 (a low likelihood of origin). Training is terminated when the error on a randomly chosen validation set is no longer improving.

After training the DEEPsc network, a feature vector associated with an actual cell from the scRNA-seq data is fed in as input and compared to each position in the reference atlas individually. We display the results as a heatmap on the schematic diagram of the biological system, choosing the spatial position with the largest likelihood of origin according to DEEPsc as the determined origin of the cell. This process is repeated for each cell in the scRNA-seq dataset to assign spatial origins of all cells (Figure 2.1C).

2.2.2 Quantifying spatial mapping performance

Each of the highlighted methods to impute spatial data onto scRNA-seq data, including DEEPsc, can be essentially boiled down to the following: For some tissue with a well-defined standard spatial structure, given known binary or continuous expression levels of G genes at each of P spatial locations (the reference atlas), calculate a correspondence score, S, of how similar each of C cells in an scRNA-seq dataset is to each of the P positions in the atlas. That is, define a function, $S : [0,1]^G \times [0,1]^G \to [0,1]$, such that $S(c_i, p_j)$; $i = 1, 2, \ldots, C$; $j = 1, 2, \ldots, P$ describes the likelihood that cell c_i originated from position p_j, based on the similarity of the expression vectors of the cell and position.

To quantify how well a given method performs for a given spatial reference atlas, we use the reference atlas itself as simulated single cell data; that is, we generate a simulated scRNA-seq dataset with $C = P$ cells, each an exact copy of a reference atlas position. This allows us to treat the simulated scRNA-seq data as having a known spatial origin, against which we can compare the output of each method. We define a system-adaptive, comprehensive performance score, consisting of three penalty terms: accuracy, which determines whether or not the known spatial origin was given a high likelihood of origin; precision, which determines

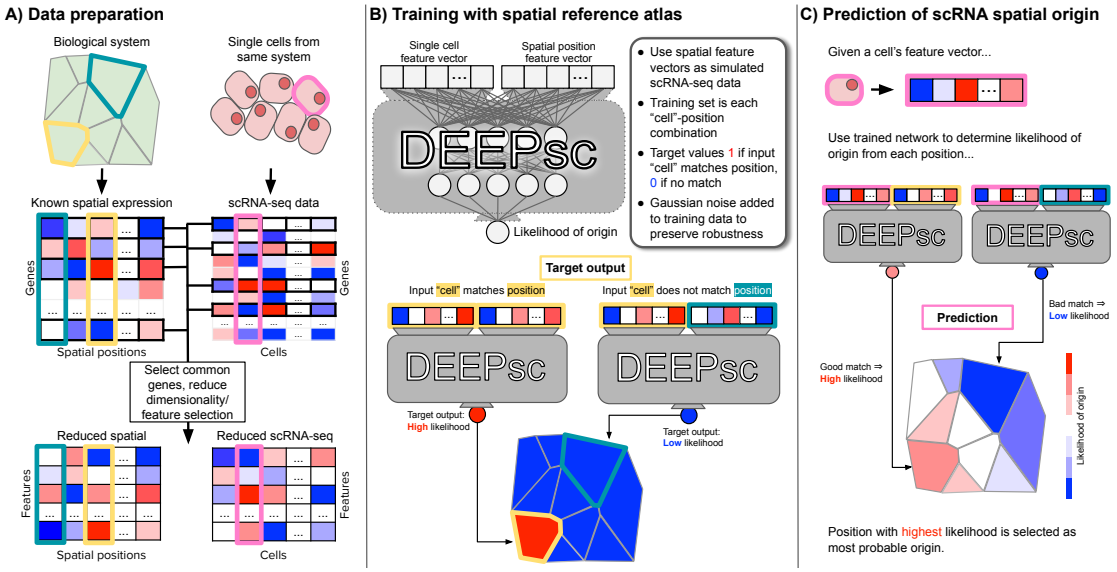

Figure 2.1: The general workflow of training and implementing DEEPsc. **(A)** Given a spatial reference atlas of gene expression levels for some biological system and a scRNA-seq dataset, genes common to both are selected, and dimensionality of the data is reduced (e.g., by PCA, UMAP). Each spatial position in the reference atlas and each cell in the scRNA-seq dataset is associated with a feature vector in the reduced space. **(B)** The DEEPsc architecture takes as input the feature vectors of one single cell and one spatial position, returning a likelihood between 0 (low likelihood) and 1 (high likelihood) that the cell originated from the spatial position. A DEEPsc network is trained using the spatial position feature vectors as simulated scRNA-seq data. The target output is a 1 (high likelihood of origin) if the simulated input cell matches the input position, and 0 (low likelihood of origin) if they do not match. **(C)** Once the DEEPsc network is sufficiently trained, a feature vector associated with a cell in the scRNA-seq dataset can be fed into the network with each spatial position individually. The resulting likelihoods are displayed as a heatmap depicting the likelihood of origin of the cell from each position. The position with the highest likelihood is chosen as the origin of the cell. This process is repeated for each cell in the scRNA-seq dataset.

whether or not other locations were given low likelihoods of origin; and robustness, which determines how sensitive a mapping method is to random noise in the input data. Each penalty term is represented by a number in $[0, 1]$, with 0 being no penalty and 1 being a worst-case scenario. The performance score is defined as $E = \frac{1}{P}\sum_{i=1}^{P} E_i$, where

$$E_i = 1 - \frac{1}{3}\left(\underbrace{1 - S_{i,i}}_{\text{Accuracy}} + \underbrace{\left|\frac{1 - \sum_{j=1}^{P} S_{i,j}}{P - 1}\right|}_{\text{Precision}} + \underbrace{(1 - \sigma^*)^4}_{\text{Robustness}}\right), \tag{2.1}$$

$S_{i,j} = S(c_i, p_j)$ is the correspondence score of cell c_i to position p_j, and E_i is interpreted as the error in the mapping of cell c_i. The quantity σ^* in the robustness term is calculated by determining the accuracy and precision penalty terms with no Gaussian noise added to the input data, then calculating the same two penalties with various levels of Gaussian noise with standard deviation $\sigma \in [0, 1]$. The quantity σ^* is defined to be the level of Gaussian noise required to raise the mean of the accuracy and precision penalties by 0.1 from their values with no added noise, or $\sigma^* = 1$, whichever is smallest. The exponent of four in the robustness term was chosen empirically such that the robustness term does not dominate the performance score, keeping in mind that expression levels are normalized to $[0, 1]$ before calculating the correspondence scores, so e.g., $\sigma^* = 0.5$ means a method requires noise on the order of half of the expression levels to raise the precision and accuracy penalties by 0.1. The performance score has a range of $[0, 1]$, where a performance score of $E = 1$ represents an ideal mapping that maps a cell to its known location with high confidence, to all other locations with low confidence, and does so in a manner robust to noise. An illustration of each term in the performance score is shown in Figure 2.2.

This performance score is limited by the fact that it relies on ground truth knowledge of the spatial origin of a single cell/spot to determine the performance of a given mapping method. However, this ground truth knowledge is not available for actual scRNA-seq data.

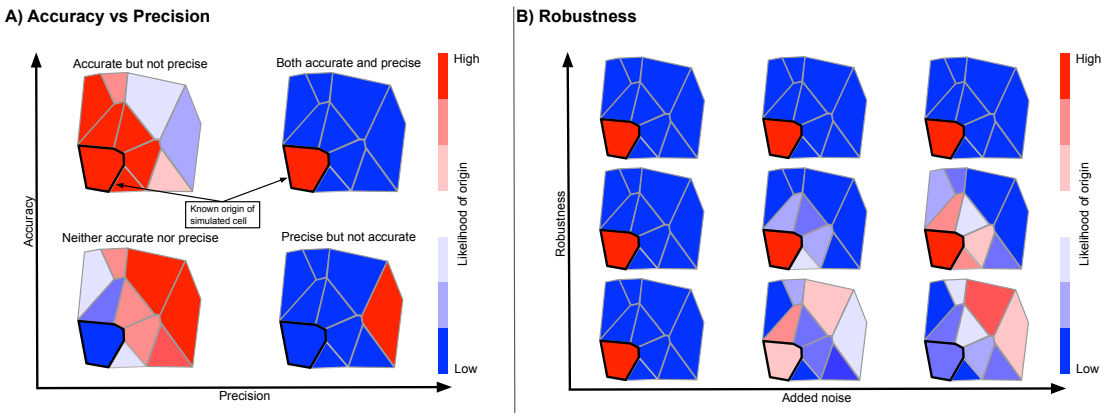

Figure 2.2: Explanation of the terms constituting the performance score. In each hypothetical mapping heatmap, the known location of the input cell is highlighted in black. **(A)** The accuracy score measures whether or not the known location receives a high likelihood; the precision score measures whether or not other locations receive low likelihoods. **(B)** The robustness score measures how much the accuracy and precision scores change if random noise is added to the input cell. A mapping method which is accurate, precise, and robust is given a high performance score while a mapping method that lacks in any or all of the three areas is given a lower performance score.

To more directly quantify the mapping performance on actual scRNA-seq datasets, we use a measure of predictive reproducibility, obtained from a k-fold cross validation scheme, in which we randomly split the common genes in the reference atlas and scRNA-seq data into k folds and calculate the correspondence score for each method using all but one fold. The correspondence scores are then used as coefficients in a weighted sum to predict the value of the dropped-out genes in each fold for each cell (scRNA-seq predictive reproducibility) or each spatial position (atlas predictive reproducibility) and determine the error in the predicted expression level. The predicted expression of gene k in cell c_i is computed as $\hat{c}_i^{(k)} = \sum_{j=1}^{P} S_{i,j}^{(k)} p_j^{(k} / \sum_{j=1}^{P} S_{i,j}^{(k)}$, and the predicted expression of gene k in position p_j is computed as $\hat{p}_j^{(k)} = \sum_{i=1}^{C} S_{i,j}^{(k)} c_i^{(k} / \sum_{i=1}^{C} S_{i,j}^{(k)}$ where $S_{i,j}^{(k)}$ is the correspondence score between cell c_i and position p_j with genes in folds not containing gene k and $c_i^{(k)}$ and $p_j^{(k)}$ are the known expression values of gene k from the scRNA-seq and the spatial atlas data, respectively. To accommodate the sparsity of data, we compute the predictive reproducibility scores separately for cells or positions with zero expression values and with positive expression

values. For example, we measure the predictive reproducibility for the task of reproducing gene k in scRNA-seq data on cells with zero expression using $R^{(k)}_{sc\_zero} = 1 - \sum_{i \in I^{(k)}_{sc\_zero}} |\hat{c}^{(k)}_i - c^{(k)}_i|/|I^{(k)}_{sc\_zero}|$ where $I^{(k)}_{sc\_zero} = \{i : c^{(k)}_i = 0\}$. Taking the average over all common genes results in a single score R_{sc_zero}, and in the same manner, we define $R_{sc\_nonzero}$, R_{atlas_zero}, and $R_{atlas\_nonzero}$. When producing predictive reproducibility scores, we use the same k-fold split across all methods to ensure a fair comparison.

2.2.3 Comparisons of multiple methods using simulated scRNA-seq data

Using the performance score, we benchmarked the methods developed by Achim et al.[18] and Satija et al.[1] (Seurat v1), Karaiskos et al.[2] (DistMap), and Peng et al.[25] together with our DEEPsc method and applied them to four different biological systems: the zebrafish embryo[1], the *Drosophila* embryo[2], the murine hair follicle[12], and the murine frontal cortex, downloaded from the 10x Genomics Spatial Gene Expression Datasets. The reference atlas for the zebrafish embryo contains the binarized expression of 47 genes on 64 spatial bins that assemble half of the hemisphere of the 6hpf embryo.[1] The *Drosophila* embryo reference atlas contains 84 genes on 3,039 spatial positions.[2] The spatial reference atlas generated with the Visium technology[56] for the murine frontal cortex contains 32,285 genes on 961 spatial positions (a subset presenting the frontal cortex from the original data), from which we kept 2755 genes from the 3,000 most variable genes in spatial data that are also present in scRNA-seq data. Segmenting a standard diagram of the follicle into 233 spatial positions and using FISH imaging of eight genes identified as spatially localized,[12] we manually defined a continuous reference atlas for the follicle (section 2.5). For mapping methods requiring a binary reference atlas, we defined a cutoff expression of 0.2 to be considered on in the follicle reference atlas. We further implemented several baseline methods for benchmark comparisons, including several methods using predefined metrics where the correspondence

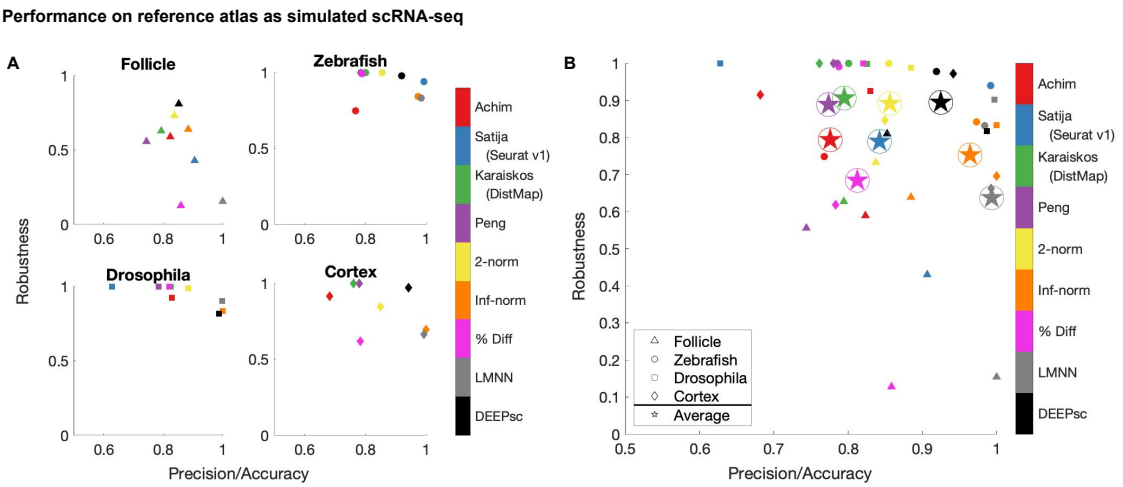

Figure 2.3: Summary of the robustness, precision, and accuracy scores of the implemented methods on four different biological systems **(A)**, as well as the simple average across all four **(B)**. These scores are each defined to be one minus the corresponding penalty term in the performance score, so that a higher score is better. Since most methods have near perfect accuracy scores, the x-axis shows a mean of the precision and accuracy scores. The y-axis shows the robustness scores for each method. Due to memory constraints, we were unable to run Seurat v1 on the cortex dataset.

score S is defined to be the 2-norm, infinity norm, or mean percent difference in the space of common genes between the input cell and spatial position. We also implemented a large margin nearest neighbor (LMNN) method that learns a linear metric (section 2.5). Figure 2.3 shows a scatter plot of the penalty terms constituting the performance score of each implemented method on each of the four biological systems, as well as the average for each method across all four systems. Table 2.1 includes the numerical values for each penalty term, as well as the calculated performance score for each method. Figure 2.4 includes example heatmaps of simulated cells for each of the biological systems. The penalty terms for the individual locations are shown in Figure 2.5.

The majority of methods were able to project the simulated scRNA-seq cells to their known spatial origins with high accuracy. Specifically, Seurat v1 and DistMap achieve high performance scores in the zebrafish embryo and *Drosophila* embryo datasets that they were originally applied to, respectively. Designed to be a system-adaptive method, DEEPsc has

the best average performance score across the four datasets (Table 2.1). Moreover, while some methods are stronger in terms robustness or precision, DEEPsc attains a balance between robustness and precision (Figure 2.3). This balance is especially important when investigating the impact of cellular spatial neighborhood on cell fate acquisition. It is desired to narrow down the inferred spatial neighborhood (precision) and at the same time reduce the sensitivity to noise (robustness). The high precision and robustness of DEEPsc is consistently observed across all locations in the dataset (Figure 2.5). Finally, it is worth mentioning that DEEPsc has a significantly higher robustness in the follicle dataset which has the smallest number of genes and is the noisiest among the four datasets.

Method (Author)	**Accuracy Term**	**Precision Term**	**Robustness Term**	**Performance Score**
Follicle				
(Achim)	0.0043	0.3484	0.4116	0.7452
Seurat v1 (Satija)	0.0795	0.1076	0.5704	0.7475
DistMap (Karaiskos)	0.0043	0.4076	0.3723	0.7386
(Peng)	**0.0000**	0.5118	0.4439	0.6814
2-norm (baseline)	**0.0000**	0.3255	0.2686	0.8020
Inf-norm (baseline)	0.0005	0.2299	0.3613	0.8028
% difference (baseline)	**0.0000**	0.2829	0.8722	0.6150
LMNN (baseline)	**0.0000**	**0.0002**	0.8455	0.7181
DEEPsc (ours)	0.0272	0.2684	**0.1904**	**0.8380**
Zebrafish				
(Achim)	**0.0000**	0.4645	0.2516	0.7613
Seurat v1 (Satija)	**0.0000**	**0.0156**	0.0604	**0.9747**
DistMap (Karaiskos)	**0.0000**	0.3989	**0.0000**	0.8670
(Peng)	**0.0000**	0.4296	**0.0000**	0.8568
2-norm (baseline)	**0.0000**	0.2902	0.0003	0.9302
Inf-norm (baseline)	**0.0000**	0.0536	0.1588	0.9292
% difference (baseline)	**0.0000**	0.4249	0.0095	0.8552
LMNN (baseline)	**0.0000**	0.0315	0.1689	0.9332
DEEPsc (ours)	0.0339	0.1281	0.0230	0.9383
Drosophila				

(Achim)	**0.0000**	0.3407	0.0759	0.8611
Seurat v1 (Satija)	0.6605	0.0848	**0.0000**	0.7516
DistMap (Karaiskos)	**0.0000**	0.3496	0.0024	0.8827
(Peng)	**0.0000**	0.4313	**0.0000**	0.8562
2-norm (baseline)	**0.0000**	0.2310	0.0130	0.9186
Inf-norm (baseline)	**0.0000**	**0.0006**	0.1671	0.9441
% difference (baseline)	**0.0000**	0.3597	0.0013	0.8797
LMNN (baseline)	**0.0000**	0.0052	0.0987	**0.9653**
DEEPsc (ours)	0.0087	0.0179	0.1827	0.9303
Cortex				
(Achim)	**0.0000**	0.6357	0.0859	0.7594
Seurat v1 (Satija)	—	—	—	—
DistMap (Karaiskos)	**0.0000**	0.4778	**0.0000**	0.8407
(Peng)	**0.0000**	0.4400	**0.0000**	0.8533
2-norm (baseline)	**0.0000**	0.3008	0.1546	0.8482
Inf-norm (baseline)	**0.0000**	**0.0006**	0.3042	0.8984
% difference (baseline)	**0.0000**	0.4332	0.3817	0.7284
LMNN (baseline)	**0.0000**	0.0143	0.3376	0.8827
DEEPsc (ours)	**0.0000**	0.1167	0.0289	**0.9515**
Average				
(Achim)	0.0011	0.4473	0.2063	0.7818
Seurat v1 (Satija)	0.1850	0.0693	0.2103	0.8246
DistMap (Karaiskos)	0.0011	0.4085	**0.0937**	0.8323
(Peng)	**0.0000**	0.4532	0.1110	0.8119
2-norm (baseline)	**0.0000**	0.2869	0.1091	0.8748
Inf-norm (baseline)	0.0001	0.0712	0.2479	0.8936
% difference (baseline)	**0.0000**	0.3752	0.3162	0.7696
LMNN (baseline)	**0.0000**	**0.0128**	0.3627	0.8748
DEEPsc (ours)	0.0175	0.1328	0.1063	**0.9145**

Table 2.1: Numerical values of each of the three constituent terms of the performance score, as determined from simulated scRNA-seq data for each biological system, as well as the average across all systems. For each term, a value closer to zero signifies lower error. For the performance score, a value closer to one indicates a better performing method. The best method for each term is highlighted in red for each system.

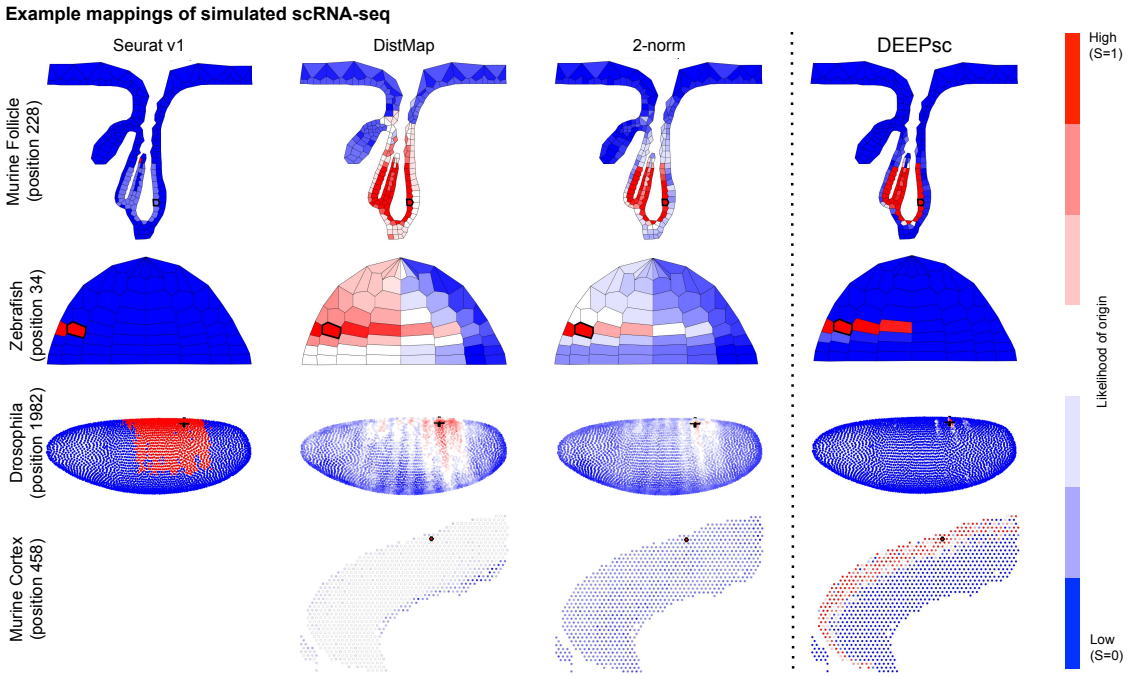

Figure 2.4: Example mappings of simulated single cells produced by various existing methods on four different biological systems, with DEEPsc mappings for comparison. The simulated input cell for the murine follicle system corresponds to position 228. For the Zebrafish system (for which Seurat was introduced), the simulated input cell corresponds to position 34. For *Drosophila* (for which DistMap was introduced), the simulated input cell corresponds to position 1982. For the murine frontal cortex, the simulated input cell corresponds to position 458. Each known position is highlighted in black in each of the heatmaps.

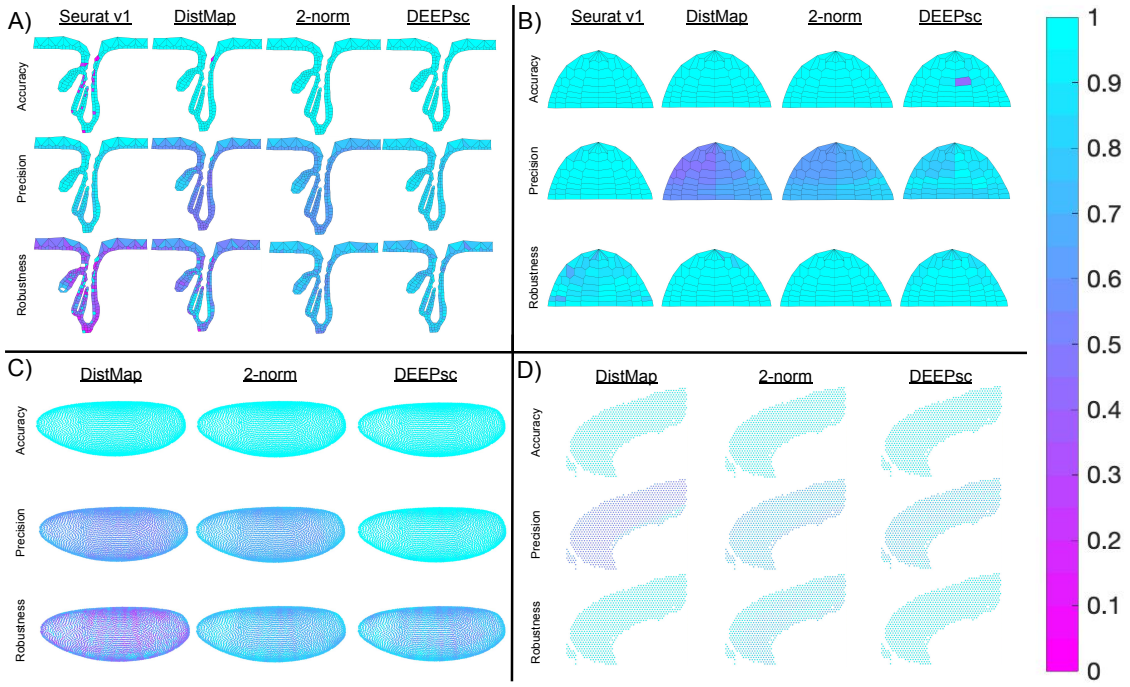

Figure 2.5: Heatmap representation of the various components of the performance score on a per position basis in **(A)** the follicle system, **(B)** the Zebrafish, **(C)** the *Drosophila* embryo, and **(D)** the murine frontal cortex. We were unable to run Seurat v1 on the *Drosophila* embryo and cortex data due to memory constraints. The penalty terms for each simulated cell, including robustness, were computed individually and plotted as a heatmap.

2.2.4 Applications to real scRNA-seq datasets

We now map actual scRNA-seq data for each system and calculate the predictive reproducibility for each method (Table 2.2 and Figure 2.6). For the follicle, the scRNA-seq data contains 1,422 cells with 26,024 genes measured containing the eight genes in the spatial atlas.[12] For the *Drosophila* embryo, we used the scRNA-seq dataset with 1,297 cells and 8,924 genes among which all the 84 spatial genes are present.[2] For the Zebrafish embryo, there are 1,152 cells and 11,978 genes in the scRNA-seq dataset with all the 47 spatial genes included.[1] For the murine frontal cortex, we used the scRNA-seq dataset provided by the Allen Institute,[3] generated with SMART-Seq2, which contains 14,249 cells and 34,617 genes, from which a set of 2,755 genes were found to be present in the top 3,000 highly variable genes in spatial atlas. These four datasets cover different situations. The follicle

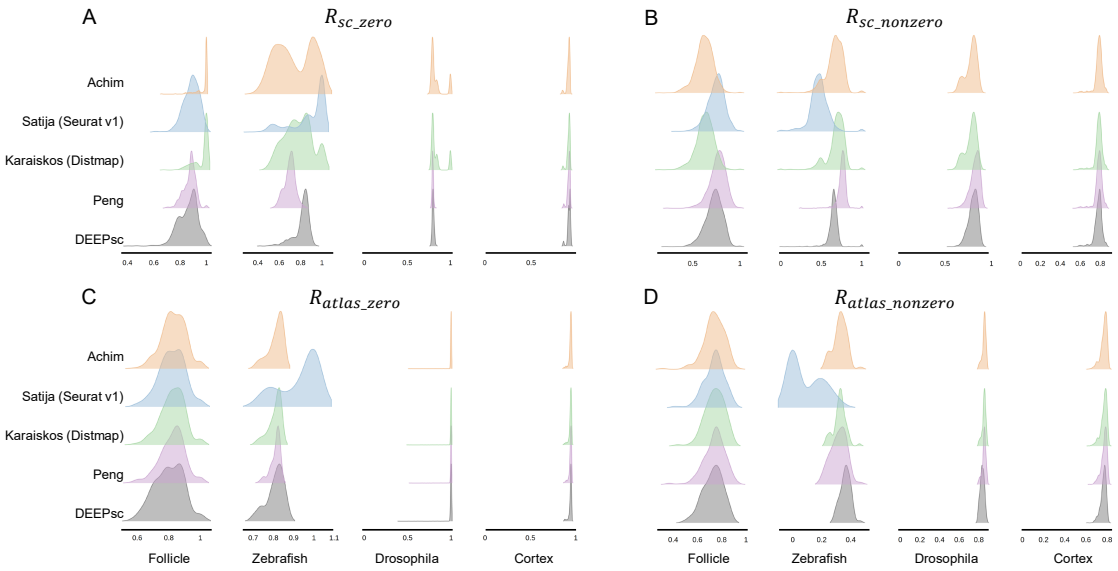

Figure 2.6: Ridgeline plots of the zero **(A)** and nonzero **(B)** scRNA-seq predictive reproducibility of individual cells in the scRNA-seq datasets and zero **(C)** and nonzero **(D)** atlas predictive reproducibility of individual positions in the spatial atlas for the four studied systems. We were unable to run Seurat v1 on the *Drosophila* embryo and cortex data due to memory constraints.

data contains a moderate number of locations, and the cells form well-defined layered structures such that there could be long and thin spatial regions that contain the same cells. The zebrafish embryo spatial data has a suboptimal resolution such that each spatial location consists of multiple cells. This data helps to evaluate the methods in treating coarse spatial atlases. The *Drosophila* embryo data contains rich spatial characteristics. There is a well-defined global ventral-dorsal/anterior-posterior coordinate system. Locally, there is also a striped pattern in the lateral side of the embryo. The frontal cortex data examines spatial gene expression at the transcriptomics level, and functions as a demonstration that DEEPsc is able to maintain a high performance on high-dimensional datasets.

Method (Author)	**Follicle**	**Zebrafish**	**Drosophila**	**Cortex**	**Average**
R_{sc_zero}					
(Achim)	**0.8772**	0.5537	0.7798	0.8019	0.7531
Seurat v1 (Satija)	0.8335	0.6842	—	—	0.7589
DistMap (Karaiskos)	0.8404	0.6641	0.7850	0.8055	0.7738
(Peng)	0.8219	0.6375	0.7859	0.8092	0.7636

2-norm (baseline)	0.8017	0.6973	0.7874	0.8114	0.7745
Inf-norm (baseline)	0.8641	0.6180	0.7807	0.8141	0.7692
% difference (baseline)	0.8357	0.5657	0.7790	0.8079	0.7471
LMNN (baseline)	0.8254	0.6795	0.7917	0.8120	0.7772
DEEPsc (ours)	0.8344	**0.7335**	**0.7961**	**0.8165**	**0.7951**
$\mathbf{R_{sc\_nonzero}}$					
(Achim)	0.7495	0.7698	0.8126	0.6693	0.7503
Seurat v1 (Satija)	0.7640	0.6975	—	—	0.7308
DistMap (Karaiskos)	0.7705	0.7619	0.8103	0.6685	0.7528
(Peng)	0.7801	0.7663	0.8114	0.6680	0.7565
2-norm (baseline)	**0.7891**	0.7386	0.8083	0.6667	0.7507
Inf-norm (baseline)	0.7496	0.7636	**0.8128**	**0.6695**	0.7489
% difference (baseline)	0.7740	**0.7721**	0.8115	0.6690	**0.7567**
LMNN (baseline)	0.7730	0.7477	0.8117	0.6643	0.7492
DEEPsc (ours)	0.7352	0.7026	0.8080	0.6691	0.7287
$\mathbf{R_{atlas\_zero}}$					
(Achim)	0.7680	0.9042	0.9264	0.8360	0.8587
Seurat v1 (Satija)	0.7681	0.9088	—	—	0.8385
DistMap (Karaiskos)	0.7674	0.9005	0.9259	0.8374	0.8578
(Peng)	0.7707	0.9006	0.9267	0.8406	0.8597
2-norm (baseline)	0.7681	0.9003	0.9278	0.8411	0.8593
Inf-norm (baseline)	0.7623	0.9050	0.9259	0.8343	0.8569
% difference (baseline)	0.7714	0.9035	0.9261	0.8438	**0.8612**
LMNN (baseline)	0.7677	0.8937	0.9289	**0.8359**	0.8566
DEEPsc (ours)	**0.7881**	**0.9148**	0.9257	0.8415	0.8675
$\mathbf{R_{atlas\_nonzero}}$					
(Achim)	0.7598	0.6658	0.8523	0.5124	0.6976
Seurat v1 (Satija)	0.7570	0.6776	—	—	0.7173
DistMap (Karaiskos)	0.7584	0.6709	0.8527	0.5127	0.6987
(Peng)	0.7570	0.6682	0.8530	0.5135	**0.6979**
2-norm (baseline)	0.7582	0.6755	0.8530	0.5135	0.7001
Inf-norm (baseline)	0.7583	0.6745	0.8534	0.5130	0.6998
% difference (baseline)	0.7573	0.6669	0.8524	0.5134	0.6975
LMNN (baseline)	0.7573	0.6764	0.8564	**0.5129**	0.7008
DEEPsc (ours)	**0.7724**	**0.7079**	0.8527	0.5125	0.7114

Table 2.2: Predictive reproducibility of each method for real scRNA-seq data. A value closer to one signifies higher predictive reproducibility. A missing entry signifies that the relevant method was not able to run successfully on the given dataset.

For the baseline models, we linearly normalized each gene in the log-normalized scRNA-seq dataset onto the interval $[0, 1]$. Continuous spatial atlases with expression values in the $[0, 1]$ range were used for the follicle, *Drosophila* embryo, and murine frontal cortex systems, the latter two having been linearly normalized to $[0, 1]$ in the same fashion as the scRNA-seq data. Since a continuous spatial atlas for Zebrafish embryo is lacking, we applied a spatial convolution to the binary atlas and added a small amount of Gaussian noise to simulate a continuous atlas. The 2-norm, Inf-norm, percent difference, and LMNN baseline models are then applied to the vectors of the commonly expressed genes in the spatial atlas and scRNA-seq data. For DEEPsc, we first applied a PCA reduction to the spatial atlas, and then applied the same linear transformation to the normalized expression values of the common genes in the scRNA-seq data. The feature vectors for the locations in the spatial atlas and the cells in the scRNA-seq data in the PCA space were then fed to the neural network. For the four existing methods, we followed the procedure as described in the associated original publications, scaling the resulting correspondence scores to $[0, 1]$ for direct comparison with baseline methods. For all the methods, we compute the predictive reproducibility by iterating over all common genes, attempting to reconstruct the expression of one gene using the k-fold cross validation scheme described in the previous section. We used $k = 4$ for the follicle and *Drosophila* embryo dataset, and $k = 5$ for the zebrafish embryo and cortex dataset.

DEEPsc has a comparable accuracy compared to other methods, and it also has a consistent performance across different systems (Table 2.2 and Figure 2.6). This consistent performance further demonstrates the system-adaptive advantage of DEEPsc and the benefit of using adaptive metrics over predefined ones. We also notice that similar to the simulated case, DEEPsc also achieves a balance between precision and robustness in the case of real

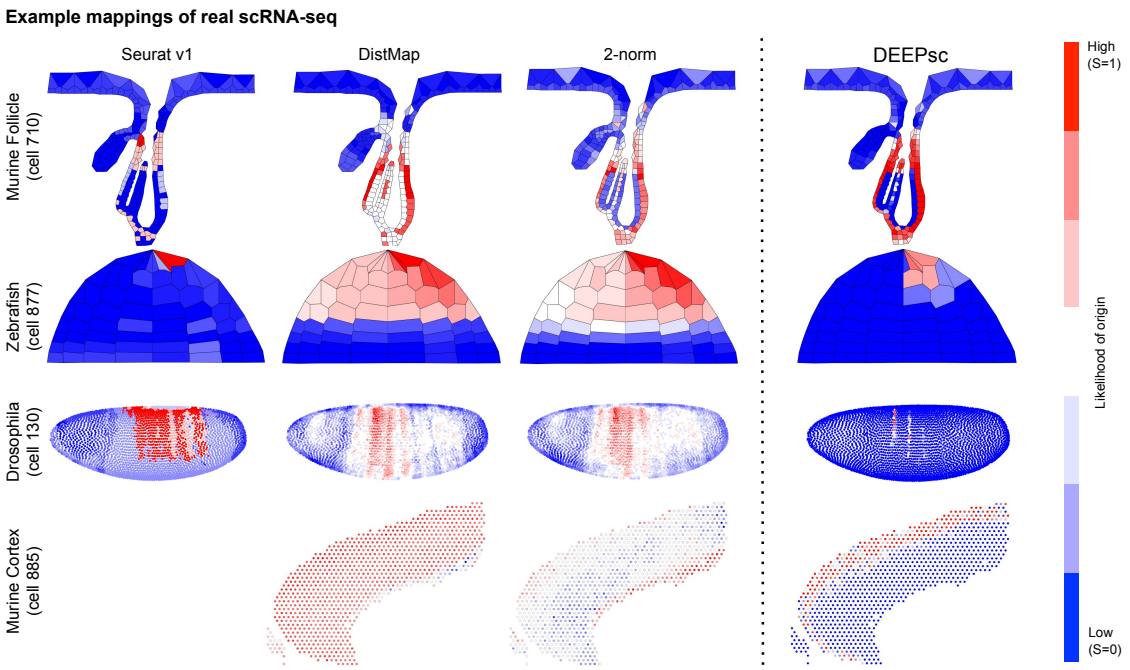

Figure 2.7: Example mappings of real single cells produced by various existing methods on four different biological systems, with DEEPsc mappings for comparison. The input cell for the murine follicle system is cell 710 from the Joost dataset. For the Zebrafish system (for which Seurat v1 was introduced), the input cell is cell 877 from the scRNA-seq dataset.[1] For *Drosophila* (for which DistMap was introduced), the input cell is cell 130 from the scRNA-seq dataset.[2] For the murine frontal cortex, the input cell is cell 885 from the Allen reference dataset.[3]

scRNA-seq data. For example, while it exhibits high precision by mapping the example cell to a specific local spot in the Zebrafish embryo or a local strip in *Drosophila* embryo, it also robustly maps a cell to the entire outer bulge of the follicle instead of only part of it (Figure 2.7). The high precision ensures that we can resolve the heterogeneity in the spatial environment and further relate them to the heterogeneity in cell fates. The high robustness prevents the identification of false correlations. Overall, DEEPsc achieves a high predictive reproducibility across all cells in the scRNA-seq dataset.

2.2.5 Comparison of dimensionality reduction methods

Dimension reduction is a crucial initial step of DEEPsc. A dimension reduction method that can be trained on one dataset and deterministically applied to another is needed due to the separated training and predicting steps. Here, we explore two different representative dimension reduction methods in the linear and nonlinear categories, PCA and Uniform Manifold Approximation and Projection (UMAP).[57] To compare these two methods, we trained several networks with varying amounts of added noise on the reference atlases of the four studied biological systems (Figure 2.8). We compared PCA (8 principal components), UMAP30 (n_components = 8, n_neighbors = 30), and UMAP5 (n_components = 8, n_neighbors = 5). While on the follicle system all three reduction methods performed virtually identically, on all three other systems PCA outperformed the other reduction methods by achieving a higher robustness score while maintaining similar accuracy.

2.3 Discussion

We have developed the DEEPsc framework, which trains a deep neural network using the known expression levels of a small subset of genes in a spatial context, then imputes that spatial information onto a non-spatial scRNA-seq dataset. Instead of using a predefined metric, DEEPsc finds a metric adaptive to data. This framework is system-adaptive and designed to be robust to noise. DEEPsc consistently performs at or above the level of several existing methods across all four biological systems studied herein, including systems for which existing methods were originally developed (Figure 2.3 and Tables 2.1, 2.2), based on our comprehensive performance measure. While DEEPsc achieves comparable accuracy and precision to other methods, it is significantly more robust to noise.

The source of DEEPsc's ability to perform well across multiple biological systems is likely

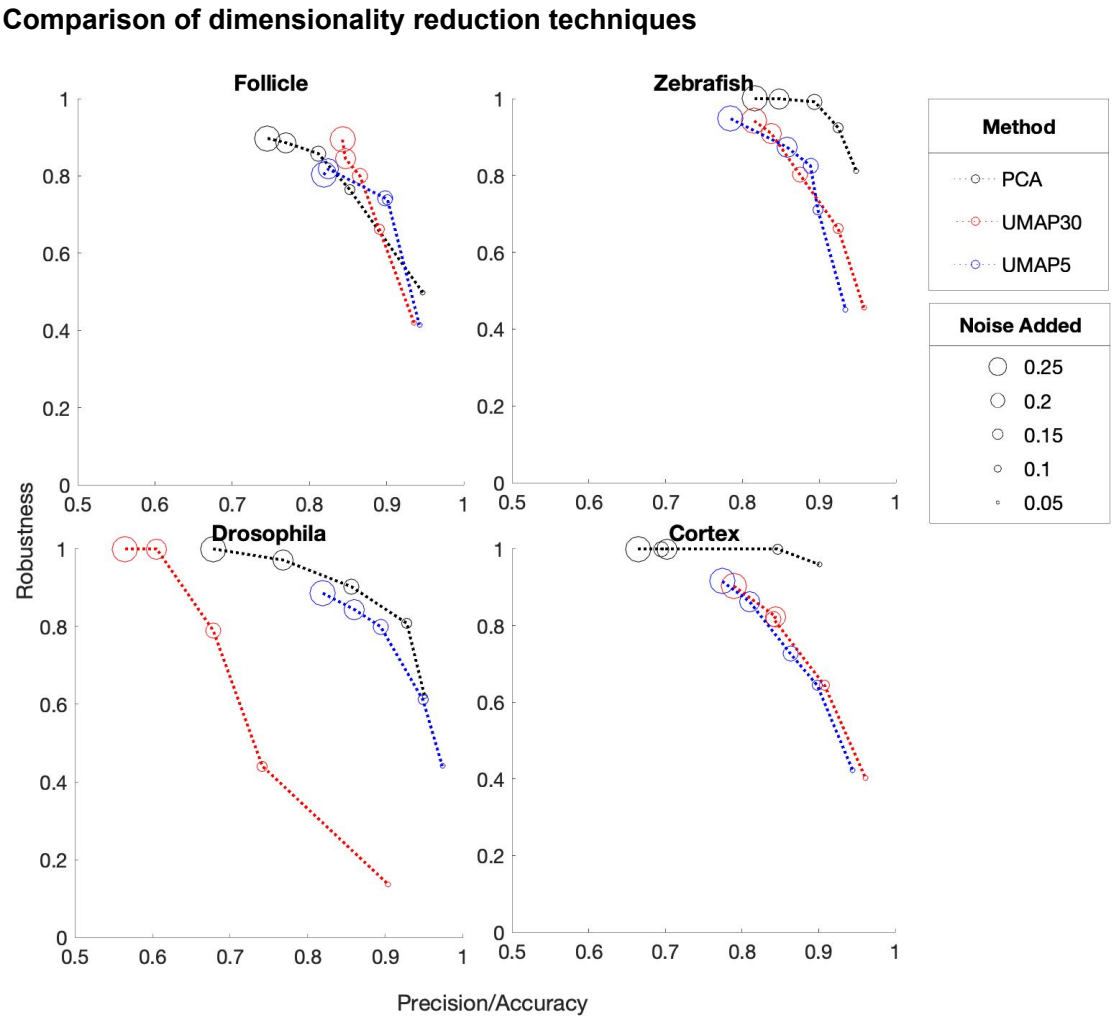

Figure 2.8: A comparison of the performance of DEEPsc networks using different dimensionality reduction methods on each of the biological systems for various levels of added noise during training. We compare principal component analysis (PCA) to Uniform Manifold Approximation and Projection (UMAP) with n_neighbors = 30 (UMAP30) and n_neighbors = 5 (UMAP5). Each of these methods reduce the dimensionality of the initial dataset to n_dimensions = 8. These scores are each defined to be one minus the corresponding penalty term in the performance score, so that a higher score is better. Since most methods have near perfect accuracy scores, the x-axis shows a mean of the precision and accuracy scores. The y-axis shows the robustness scores for each method.

the generality of its neural network architecture and the multiple checks for robustness employed during training on the reference atlas. The various parameters for training a DEEPsc network, though chosen empirically, appear to translate to multiple systems effectively, so we expect DEEPsc to continue to perform well across more biological systems in future study.

One notable weakness of DEEPsc is the significant amount of training time required to produce a final mapping. While most existing reference atlas methods simply involve a deterministic calculation to produce a mapping, DEEPsc requires an initial training, and the training time depends on the number of locations in the spatial atlas. The training process of DEEPsc can be effectively accelerated by iterating over a subset of possible location pairs. Due to the dimension reduction step, DEEPsc can still be trained efficiently on datasets with large amount of genes, for example, the spatial transcriptomics data on the murine frontal cortex. Though the predefined metrics including the 2-norm and inf-norm perform well in terms of accuracy and precision, they are less robust to noise. This is further the case for LMNN as it tries to amplify any small variations. This drawback in robustness is mitigated by DEEPsc by controlling the balance between precision and robustness.

Learning a metric from high-dimensional datasets can be generally useful for analysis and integration of omics datasets. A future research interest is to decrease training time in such framework by developing a better method for reducing the size of the training set to a small, targeted fraction of relevant examples, particularly for very large atlases such as those derived from spatial transcriptomics assays. Since the size of the training set can increase quadratically with the number of positions in the atlas, it is beneficial to develop a more efficient training pipeline. We have developed a method of sparsifying the training set (section 2.5), so that its size only increases linearly with the number of positions in the atlas, though further improvement may be warranted. The largest atlas studied here was that of *Drosophila* ($P = 3039$), the training of which took several hours even with the sparsified training set. Typical numbers of distinct spatial locations in a spatial transcriptomics dataset

can be orders of magnitude larger.

DEEPsc aside, the performance score we have created can serve as a comprehensive measure of mapping performance for future work. The performance score is able to be calculated for any mapping method that assigns a likelihood of origin from each spatial location, particularly within the reference atlas framework. It is not dependent on any specific system or mapping method, and the individual terms which constitute it allow for a detailed analysis and comparison of various methods. Potential improvements might include incorporating some amount of spatial awareness into the calculation. Currently each spatial position is treated as completely independent from every other spatial position, so the precision term, for example, can yield unintuitive results if a method maps a cell, for example, with high probability to two positions on opposite sides of a system and low probability everywhere else, compared to a different method mapping the same cell with high probability to five positions in a tightly clustered, spatially compact region of the system. If, for example, the various correspondence scores for each position with high probability were weighted by their physical distance from other cells with high probability, this term might more accurately reflect the intuitive idea of precision. Other improvements might include simplifying the calculation of the robustness term to require fewer intensive calculations.

2.4 Conclusion

DEEPsc achieves an accuracy comparable to several existing methods while attaining improved precision and robustness. It also has a more consistent performance across the four different biological systems tested thanks to the system-adaptive design. As spatially resolved gene expression data becomes more readily available, our method will serve as a useful tool to infer spatial origins from non-spatial scRNA-seq data.

Additionally, our comprehensive performance score and the collection of reproductions of previously developed methods in a single software framework will serve as useful tools for future comparisons of spatial mapping methods. This systematic approach to imputing spatial information to scRNA-seq data is crucial to studying the spatial impact on cell fate dynamics.

2.5 Materials and methods

2.5.1 Data preparation for DEEPsc

Given a matrix of scRNA-seq read counts where each row is a different gene and each column is a different cell, and a matrix representing a spatial reference atlas where each row is a different gene and each column is a different spatial position, we first select common genes by eliminating rows in each corresponding to genes not in the other matrix. Once we have eliminated genes not in common, we are left with a number of cells (C) $\times$ number of genes (G) matrix for the scRNA-seq data and a number of positions (P) $\times$ number of genes (G) matrix for the spatial reference atlas.

We then apply dimensionality reduction to the atlas in the form of a PCA projection, selecting a user-configurable number of principal components to serve as feature vectors. We find in our analysis that keeping the top eight principal components yields satisfactory results. The same PCA coefficients are used to project the scRNA-seq matrix into these principal components. After projection, both matrices are normalized by dividing by the largest element in each, so that the elements are all in $[0, 1]$.

For the comparisons in section 2.2.5 we use the UMAP implementation by Meehan et al.[58] found on the MATLAB Central File Exchange at `https://www.mathworks.com/`

matlabcentral/fileexchange/71902. Specifically, we ran the run_umap() function on the spatial reference atlas with n_dimensions = 8 and n_neighbors = 30 or n_neighbors = 5 for UMAP30 and UMAP5, respectively.

2.5.2 Training a DEEPsc network

To train the DEEPsc network, we use the spatial position feature vectors themselves as simulated scRNA-seq data. The training data is a set of P^2 vectors of length $2N$, where N is the reduced dimensionality of the reference atlas. The first N components correspond to a feature vector of one position in the reference atlas (functioning as a simulated cell) and the last N components correspond to some other position in the reference atlas. Each simulated cell is compared pairwise with every position in the spatial reference atlas; if the simulated cell is an exact match to a given position, the target output is chosen to be 1 (a high likelihood of origin), and if the simulated cell and chosen position are not an exact match, the target output is chosen to be 0 (a low likelihood of origin).

The DEEPsc architecture is an artificial neural network with $2N$ inputs, two fully connected hidden layers with N nodes each and a single node in the output layer. Sigmoid activation functions are attached to each node, including the output node, so that the resulting output is in $[0, 1]$ and can be interpreted as a likelihood that the input cell originated from the input spatial position. To preserve robustness and avoid overfitting the training data, a layer of Gaussian noise is added to the simulated cells so that the network is pushed to learn complex nonlinear relationships among the spatial positions in the reference atlas rather than simply activate when an exact match is encountered. This Gaussian noise layer allows the user to configure the standard deviation of the added noise, as well as to configure the probability that any noise will be added in a given training epoch. We find empirically that a noise level of about 0.10 and a probability of 0.5 yield reasonable robustness to noise, though this may

vary from system to system.

Since the training data will naturally consist of many more non-matches than matches, and thus the target data will contain many more zeros than ones, we use a novel custom objective function,

$$L(Y,T) = \sum_{i=1}^{P} (y_i - t_i)^2 \frac{1}{1.001 - t_i} \tag{2.2}$$

where y_i is the network's predicted output and ti is the target output ($t_i = 1$ if exact match and $t_i = 0$ if not), to more heavily penalize the network when it gives a false negative (low likelihood when it should be high) than when it gives a false positive (high likelihood when it should be low). This acts to counteract the tendency of the network to "learn" to simply return 0 for every single input and "ignore" any comparably rare training data with $t_i = 1$.

To further account for the sparsity of exact matches in the training set, we split it into a test and validation set, the former consisting of a configurable fraction of the inputs corresponding to exact matches as well as a configurable multiple of the inputs corresponding to non-matches. If trainFrac $= 0.9$ and trainingMultiple $= 99$, for example, 90% of the exact matches will be added to the training set and 99x more non-matches will be added, so that the exact matches make up 1% of the training set. The rest of the inputs are reserved for the (generally much larger) validation set. This is beneficial in reducing training time because it allows us to train with a much smaller fraction of the P^2 input vectors, giving preference to the exact matches. Indeed, this reduces the size of the actual training set to scale linearly with the size of the atlas rather than quadratically.

Training is performed in MATLAB using the trainNetwork() function in the Deep Learning Toolbox,[59], for which we implemented the above-described custom network layers. Since the input data is already normalized in preprocessing, we disable the default normalization of trainNetwork(). We use the default Glorot[60] initialization of weights and biases in the fully connected layers. We then train each network for a maximum of 50,000 epochs of

standard gradient descent with a learning rate of $\eta = 0.01$, shuffle the order of the data each epoch, and use the ADAM optimization method (Kingma and Ba, 2014) with the default parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 10^{-8}$. In addition to the custom objective function layer we describe above, trainNetwork() by default adds an L^2-regularization term to the loss with a regularization factor of $\lambda = 0.0001$. We monitor the RMSE of the validation set throughout training and manually stop training if it is no longer improving before the maximum number of epochs has been reached. The trainNetwork() function also allows for parallel computation via the Parallel Computing Toolbox,[61] which is highly recommended but not strictly required for training.

2.5.3 Creating a reference atlas for the murine follicle

To create a spatial reference atlas for the murine follicle system, we patterned the spatial coordinates of each position in the atlas off of a standard diagram of a mouse follicle found in Figure 1 of Joost et al.[12] We constructed a Voronoi diagram around each of the cell centers and made manual adjustments to the vertices as we saw fit aesthetically. We then selected the eight genes in the atlas from the systematic staining catalog made available by Joost. We chose the genes based on a combination of high image quality and spatial diversity. Gene expression levels in $[0, 1]$ were chosen manually to best represent the images, though to eliminate any implicit bias we also added a small level of Gaussian noise to the atlas. For all methods requiring a binary atlas, we chose a cutoff of 0.2 to represent "on" expression in this atlas.

2.5.4 Large margin nearest neighbor baseline

To implement a LMNN baseline for benchmarking comparison, we used code from the MATLAB Toolbox for Dimensionality Reduction found at `https://lvdmaaten.github.`

`io/drtoolbox/` and modified it for our uses. Specifically, we used the lmnn() function in the "techniques" subfolder, and modified the code to set mu = 1, i.e., to remove the "pull" term, as well as setting the number of targets to 1 (the point itself) and treating all other points as imposters. Further, we modified the slack variables to enforce a minimum separation of $\sqrt{D}$, where D is the dimensionality of the space ($D = G$ for our applications). For the numerical experiments of the LMNN method with the cortex dataset, a PCA dimension reduction (50 PCs) was performed before applying LMNN to accommodate the large number of genes.

Chapter 3

AtlasGeneratorOT: Automating the creation of a reference atlas

In order to train DEEPsc networks for the biological systems studied in Chapter 2, we made use of already existing reference atlases of the biological systems where available,[1, 2, 56] and we created by hand a novel reference atlas of the murine hair follicle using fluorescence imaging data of eight different genes known to have spatially coherent expression patterns.[12] Steps involved in the manual creation of this reference atlas included creating a standardized diagram of the follicle onto which we manually ascribed 233 different "spots" of approximate cellular resolution. We then examined each of the eight fluorescence imaging slides and manually decided an expression level for each spot.

This manual process was slow and tedious, and would have been even more so if the number of genes in the reference atlas were larger. Further, using a similar manual creation process to develop future novel reference atlases may potentially introduce unintentional biases into the atlas, which may influence the training of a DEEPsc network and therefore the spatial mapping of future scRNA-seq datasets. In this chapter, we describe our efforts to more fully

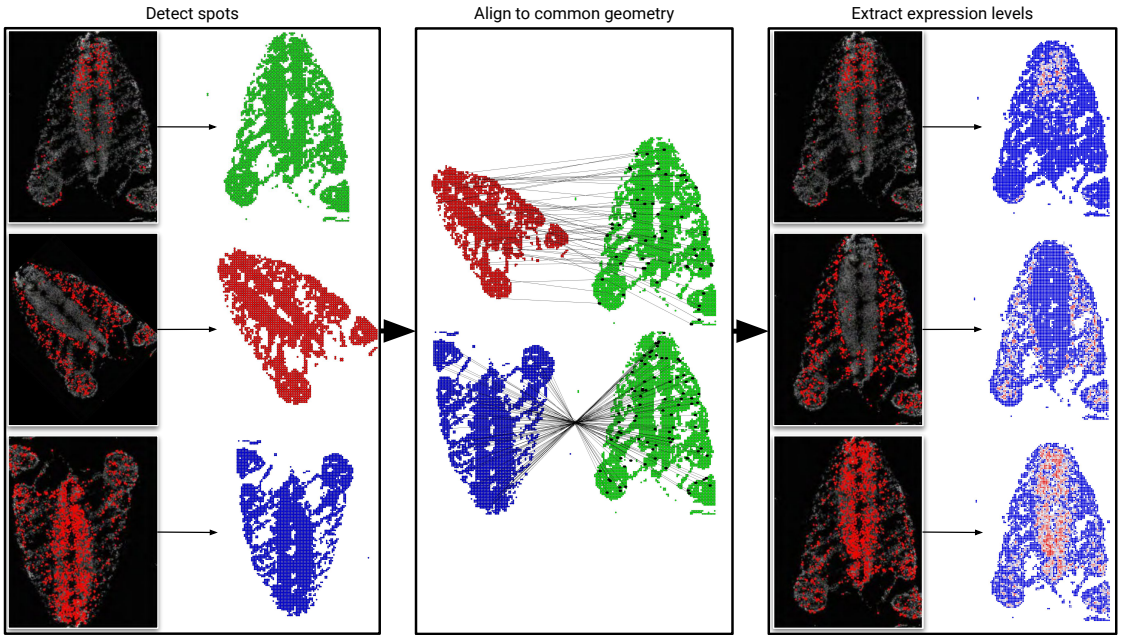

Figure 3.1: Flowchart describing the workflow for AtlasGeneratorOT. Beginning with a collection of images, AtlasGeneratorOT detects spots in each image (left), uses an optimal transport-based algorithm to align the spots to a common geometry (middle), and extracts expression information from each of the aligned images (right). Images depict Slice 15 of the murine neural crest in [4].

automate the process of creating a novel reference atlas from a collection of fluorescence or other imaging data, speeding up the workflow and reducing bias in the atlas creation process.

3.1 Background

A typical reference atlas for a biological system consists of a $P \times G$ matrix A representing the expression level of G genes for each of P positions. The spatial arrangement $\mathcal{X}$ of these P positions is also given, e.g. as a set of coordinates $\mathcal{X} = \{(x_i, y_i)\}_{i=1}^{P} \in \mathbb{R}^2$ or $\{(x_i, y_i, z_i)\}_{i=1}^{P} \in \mathbb{R}^3$, or as a collection of patches or regions $\mathcal{X} = \{U_i\}_{i=1}^{P} \subset \mathbb{R}^2$ or $\mathbb{R}^3$ in a 2- or 3-dimensional standardized diagram of the biological system. Many such reference atlases exist of varying spatial resolutions and with wildly varying numbers of genes (cf. Table 1 in [18]), and more are being created constantly as fast, high-resolution imaging technology

becomes more ubiquitous. Particularly with the recent advent of spatial transcriptomic methods,[19, 20, 21, 22, 23, 24] the ease with which a collection of images of a large number of genes can be quickly produced is increasing rapidly.

However, the process of extracting the information necessary to create an atlas matrix A or even a spatial description $\mathcal{X}$ from the collection of images is sorely lacking. A typical starting point for the creation of a reference atlas is to gather a collection of images depicting the spatial gene expression pattern for some number of genes. Often times these images may be sourced from multiple samples of the system, produced by different labs, under vastly different experimental setups. Because of this, the process by which expression levels are extracted from the images is a nontrivial problem, perhaps requiring complicated mapping of one geometry onto another, distinguishing the output of disparate imaging technologies, and dealing with images of vastly different quality and resolution.

There do exist several platforms for spot or cell detection in 2-D images which can be used to produce a spatial description $\mathcal{X}$ of the spots or cells in an image, among them CellProfiler[62] and StarDist,[63, 64] with the latter also able to handle 3-D images. However, the learning curve for many of these platforms is quite steep, and there appear to be no existing platforms dedicated to the creation of a spatial reference atlas directly. We therefore introduce here a novel automation platform, AtlasGeneratorOT, which, with minimal user input, produces an atlas matrix A and set of 2-D spatial coordinates $\mathcal{X}$ corresponding to a common set of spots compiled from all images in a collection.

AtlasGeneratorOT makes a few basic assumptions about the dataset from which the reference atlas will be created. We first assume that at least one image of the expression of one gene depicts the entire biological system in a convenient geometry with a clearly distinguishable solid color background and will thus serve as the "anchor" onto which all other images will be mapped. We also assume that all images of the system are stored in a single folder with no other files. Finally, we assume that each image depicts only one gene with one color. It is,

however, possible using AtlasGeneratorOT to extract the expression of multiple genes from a single image, provided the image is duplicated multiple times with different filenames in the parent folder.

Throughout this chapter, we use as a motivating biological system the murine neural crest, with imaging data provided by Soldatov et al. (cf. Data S11 in [4]). We begin with images of the expression level of 32 different genes known to be spatially regulated, obtained by *in situ* sequencing of 15 serial sections of the neural crest of an E9.5 murine embryo. Within each serial slice, each of the 32 images depicts gene expression over the same underlying geometry, so no alignment is necessary for this dataset; all images share coordinates exactly and the extraction process of section 3.2 can be applied directly. However, we describe in section 3.3 a method of aligning images with differing geometries based on optimal transport theory, which may be required for other datasets. A flowchart of the AtlasGeneratorOT workflow is shown in Figure 3.1.

3.2 Extracting expression levels from images

3.2.1 Detecting spots in the anchor image

Given a 2-D anchor image in RGB format (i.e. $I \in [0,1]^{w \times h \times 3}$), we allow the user to define an integer spot size $\Delta x \in (0, w]$, default $\Delta x = \lceil w/50 \rceil$, and construct a regular $n_x \times n_y$ grid atop the image consisting of square cells of $\Delta x \times \Delta x$ pixels, allowing for non-square cells around the borders if necessary. The pixels in each cell are compared to a background color $\mathbf{c} = [r, g, b] \in [0,1]^3$, by default defined as the most common color in the image, $\mathbf{c} = \text{mode}(\mathbf{I}_{x,y})$, $x = 1, 2, \ldots, w$, $y = 1, 2, \ldots, h$, but also user-configurable if necessary. We

use for the distance between colors the Euclidean norm on the sRGB color space,

$$d(\mathbf{c}_1, \mathbf{c}_2)^2 = d([r_1, g_1, b_1], [r_2, g_2, b_2])^2 = (r_1 - r_2)^2 + (g_1 - g_2)^2 + (b_1 - b_2)^2 \tag{3.1}$$

and calculate for each grid cell $\mathcal{G}_{i,j},\ i = 1, 2, \ldots, n_x,\ j = 1, 2, \ldots, n_y$, the average of the distance from each pixel to the background color $\mathbf{c}$,

$$\bar{d}_{i,j} = \frac{1}{|\mathcal{G}_{i,j}|} \sum_{k,l \in \mathcal{G}_{i,j}} d(\mathbf{I}_{k,l}, \mathbf{c}) \tag{3.2}$$

since each $\bar{d}_{i,j} \in [0, 1]$ by definition, we then use a threshold $\eta \in (0, 1)$, default $\eta = 0.1$, to define whether or not a cell is a background cell ($\bar{d}_{i,j} < \eta$) or a detected spot ($\bar{d}_{i,j} \geq \eta$). The coordinates of the centers of each detected spot are then stored as an array $X \in \mathbb{R}^{P \times 2}$, where P is the number of spots detected.

Other more complex methods of defining a measure of color difference exist, including those in several perceptually uniform color spaces such as CIEXYZ, CIELUV, or CIELAB. In particular CIE76, defined over the $L^*a^*b^*$ color space (itself a nonlinear transformation of RGB space), and CIE94, defined over the closely related $L^*C^*h^*$ color space, can be shown to more accurately reflect human perception of color difference. However, due to the relatively higher computational complexity of these measures, we choose to use the more computationally efficient sRGB measure described above. To account for less ideal results, we allow the user to manually select false positive spots to remove from the array X if desired.

Experimental results of the spot detection algorithm on an image extracted from Data S11 in [4] of the expression of Car11 in Slice 1 of the murine neural crest are shown for various spot sizes Δx and thresholds η in Figure 3.2. We note that the optimal value of $\eta \approx 0.1$ for each of the chosen spot sizes.

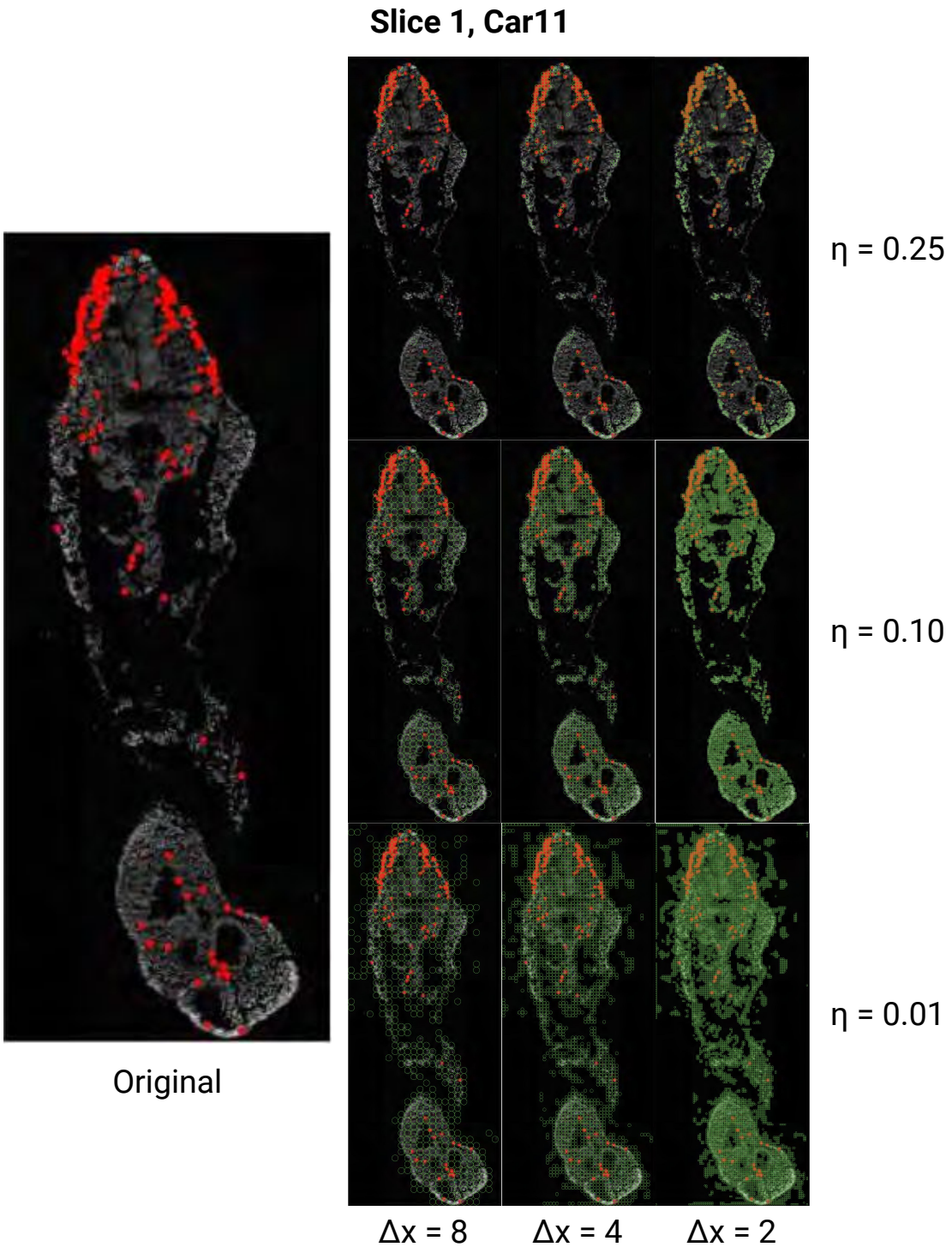

Figure 3.2: Results of the spot detection algorithm, applied to a 201×494-pixel RGB image depicting expression of the gene Car11 in Slice 1 of the murine neural crest from Soldatov.[4] Results are shown for spot sizes $\Delta x = \{8, 4, 2\}$, background color $\mathbf{c} = [0, 0, 0]$ (black) with threshold values $\eta = \{0.01, 0.10, 0.25\}$.

3.2.2 Extracting gene expression from each image

Given a collection of spot coordinates $\{(x_i, y_i)\}_{i=1}^{P}$ stored in an array $X \in \mathbb{R}^{P\times 2}$ extracted from the anchor slide, we now seek to extract gene expression levels at each of the P spots in each of G images, $I_1, I_2, \ldots, I_G$, depicting the fluorescence expression of each gene to be included in the $n \times G$ reference atlas. To do so, we require for each image a color $\mathbf{c}_g \in [0,1]^3$ indicating expression of the gth gene, and we use a similar process as in (3.2) to determine the average color difference $\bar{d}_i$ between pixels in a $\Delta x \times \Delta x$ square centered at (x_i, y_i) in image I_g, where Δx is the same spot size used to extract spots in section 3.2.1.

We find that the unnormalized distance between colors gives in general some nonzero expression at every spot, which is biologically unlikely, and may or may not give a large expression value to spots where a human would likely do so. To remedy this, after calculating $\bar{d}_i$ for all P spots, we rescale the group to the interval $[0,1]$ by subtracting the minimum expression, then dividing by the resulting maximum. Finally, we subtract the result from 1 so that the resulting values represent a similarity measurement rather than a dissimilarity measurement. A value close to 1 therefore signifies high expression while a value close to 0 signifies low expression. To further sparsify the expression pattern, we again specify a threshold $\delta \in (0,1)$ below which a gene is assumed to have zero expression.

We repeat this process for each of the G images, storing the resulting values into columns of a $P \times G$ matrix A, which, along with the coordinate matrix X, we call the reference atlas. We include the expression levels extracted from images of genes Ets1 and Sox2 in Slices 1 and 15 of the murine neural crest in Figure 3.3. We note that the extraction process robustly captures gene expression for different values of Δx.

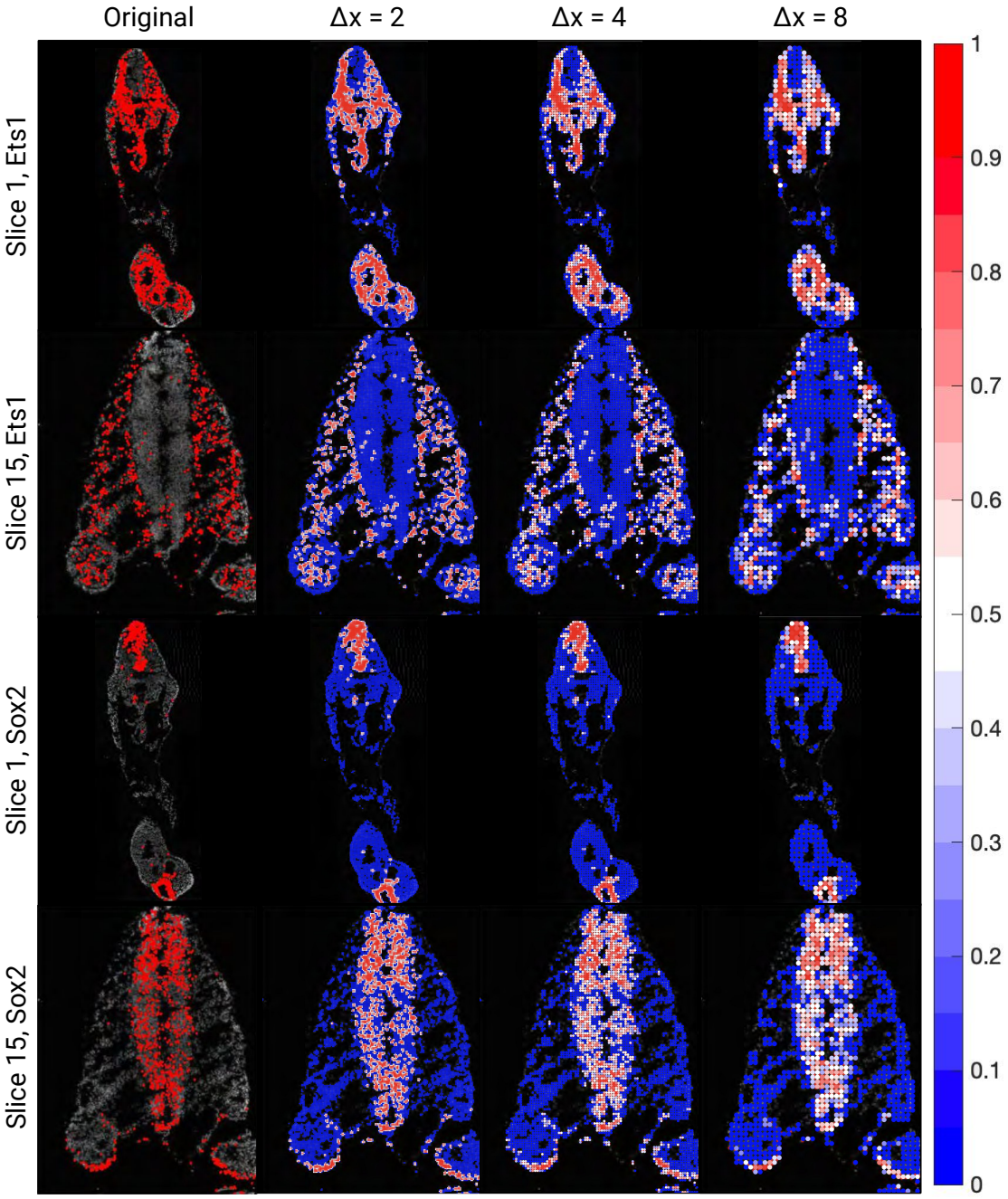

Figure 3.3: Extracted gene expression for genes Ets1 and Sox2 from images of Slice 1 and Slice 15 of the murine neural crest from Soldatov[4] for spot sizes $\Delta x = \{2, 4, 8\}$. Gene expression color $\mathbf{c}_g$ is taken to be $[1, 0, 0]$ (red), expression threshold $\delta = 0.1$, in all images. Expression levels are detected using the sRGB distance (3.2) and scaled from 0 (low/no expression, blue) to 1 (highest expression, red).

3.3 Aligning images of different genes with optimal transport

During the creation of a reference atlas from a collection of images as in section 3.2, it is often the case that, unlike in the murine neural crest dataset, images of the expression patterns of different genes do not conform to the same geometry, i.e. the coordinates of the spots in the anchor image do not translate directly to the other images, and thus some type of alignment is necessary. Manual alignment using photo editing software is a slow and tedious process, leading to a bottleneck in the workflow and is a possible source of further unintended bias in the atlas creation process. In this section, we describe a process based on optimal transport (OT) theory which we use to semi-automate the alignment process.

3.3.1 Optimal transport background

The output of the spot detection algorithm outlined in section 3.2 is a set of points $\{(x_i, y_i)\}_{i=1}^n \in \mathbb{R}^2$, where n is the number of spots detected in the image. If we view these spots as a discrete distribution on $\mathbb{R}^2$, $\mu = \frac{1}{n}\sum_{i=1}^n \delta(x_i, y_i)$, where $\delta(x)$ is the Dirac distribution at point x, then given the spots in two images, we can formulate an optimal transport problem to couple the spots $\{\mathbf{x}_i\}_{i=1}^n$ in image 1 to the spots $\{\mathbf{y}_j\}_{j=1}^m$ in image 2. Mathematically, a (discrete) optimal transport problem is described as follows:[65]

Given two distributions

$$\alpha = \sum_{i=1}^{n} \mathbf{a}_i \delta(\mathbf{x}_i), \qquad \beta = \sum_{j=1}^{m} \mathbf{b}_j \delta(\mathbf{y}_j) \tag{3.3}$$

such that $\mathbf{a}_i \geq 0$, $\sum_i \mathbf{a}_i = 1$, $\mathbf{b}_i \geq 0$, $\sum_j \mathbf{b}_j = 1$, along with a cost matrix $C \in \mathbb{R}_+^{n \times m}$, find a

coupling matrix T that solves

$$\underset{T\in\mathbb{R}_+^{n\times m}}{\operatorname{argmin}} \sum_{i,j} C_{i,j}T_{i,j} \quad \text{subject to } T\mathbf{1}_m = \mathbf{a}, T^\mathsf{T}\mathbf{1}_n = \mathbf{b} \tag{3.4}$$

where $\mathbf{1}_n \in \mathbb{R}^n$ is a length n vector of all ones and $T_{i,j}$ can be interpreted as the fraction of "mass" transported from point $\mathbf{x}_i$ in the source distribution to the point $\mathbf{y}_j$ in the target distribution. The constraints on T are often referred to as "mass constraints", as they can be interpreted physically as requiring transport of all mass from one distribution to the other.

In typical applications, the cost matrix C consists of pairwise Euclidean distances between points in each distribution, i.e. $C_{i,j} = \|\mathbf{x}_i - \mathbf{y}_j\|_2$; however, in principle, this matrix can be computed with any power of any distance metric, $C_{i,j} = d(\mathbf{x}_i, \mathbf{y}_j)^p$. In this case, the minimum value of the product computed using the optimal T in (3.4) is referred to as the pth Wasserstein distance between the two distributions (which depends on d), and is often denoted

$$W_p = \left(\min_{T\in\mathbb{R}_+^{n\times m}} \sum_{i,j} d(\mathbf{x}_i, \mathbf{y}_j)^p T_{i,j}\right)^{1/p} \implies W_p^p = \min_{T\in\mathbb{R}_+^{n\times m}} \sum_{i,j} C_{i,j}T_{i,j} \tag{3.5}$$

One drawback to using the Euclidean distance to form a cost matrix is the implicit assumption that the two distributions inhabit the same space. Indeed even if a collection of spots ostensibly inhabits the same space, the classical OT algorithm is not invariant under simple transformations like rescalings, translations, and rotations. A relevant example of using a non-standard cost matrix is that of Motta, et al.,[45] wherein they make use of an underlying graph structure to form a cost matrix consisting of node-to-node and neighborhood-to-neighborhood dissimilarities in order to align two images of a retinal fundus.

In the context of spots extracted from images, since the images may be of different sizes and/or may have nontrivial differences in geometry, that the spots extracted from two images inhabit the same space is not a valid assumption. Furthermore, early attempts at defining

novel cost matrices for this task did not produce sufficiently satisfactory alignment results. Instead we rely on an extension of classical optimal transport, Gromov-Wasserstein, which does not require the two distributions to inhabit the same space.[66, 67] Indeed, we need only calculate two separate distance matrices for each space. If α is defined on the metric space (M_1, d_1) and β is defined on the metric space (M_2, d_2), we can compute the two matrices $D_1 \in \mathbb{R}_+^{n\times n}$ and $D_2 \in \mathbb{R}_+^{m\times m}$ where $(D_1)_{i,i'} = d_1(\mathbf{x}_i, \mathbf{x}_{i'})$ and $(D_2)_{j,j'} = d_2(\mathbf{y}_j, \mathbf{y}_{j'})$. Then the pth Gromov-Wasserstein distance is defined by

$$GW_p^p = \min_{T\in\mathbb{R}_+^{n\times m}} \sum_{i,i',j,j'} |(D_1)_{i,i'} - (D_2)_{j,j'}|^p \, T_{i,j} T_{i',j'} \tag{3.6}$$

where the matrix T is subject to the same mass constraints, $T\mathbf{1}_m = \mathbf{a}$, $T^\mathsf{T}\mathbf{1}_n = \mathbf{b}$, as in classical optimal transport.

The classical OT optimization problem can be viewed as a linear program and thus enjoys the existence of many different efficient solvers based on, e.g., Dantzig's Simplex method or interior point methods.[65] However, finding the coupling matrix which solves the Gromov-Wasserstein OT problem is a quadratic assignment problem and in general NP-hard to solve.

To allow faster approximation of a solution (indeed even of the classical OT problem), an entropic regularization term is often added to the loss function,[68] typically of the form $\varepsilon H(T)$, where

$$H(T) = -\sum_{i,j} T_{i,j} \left(\log T_{i,j} - 1\right) \tag{3.7}$$

As detailed by Benamou in [69], this regularization term allows the solution to (3.4) to admit a factorization

$$T_{i,j} = \mathbf{u}_i e^{-C_{i,j}/\varepsilon} \mathbf{v}_j \equiv \mathbf{u}_i K_{i,j} \mathbf{v}_j \tag{3.8}$$

for some scaling vectors $\mathbf{u} \in \mathbb{R}_+^n$, $\mathbf{v} \in \mathbb{R}_+^m$. The mass constraints on the optimal T then

require

$$\mathbf{u} \odot (K\mathbf{v}) = \mathbf{a}, \qquad \mathbf{v} \odot (K^{\mathsf{T}}\mathbf{u}) = \mathbf{b} \tag{3.9}$$

where $\odot$ is the Hadamard componentwise product. This factorization thus allows for the use of iterative procedures such as the Sinkhorn algorithm,[70] which defines for $\ell = 0, 1, 2, \ldots$

$$\mathbf{u}^{(\ell+1)} = \frac{\mathbf{a}}{K\mathbf{v}^{(\ell)}}, \qquad \mathbf{v}^{(\ell+1)} = \frac{\mathbf{b}}{K^{\mathsf{T}}\mathbf{u}^{(\ell+1)}} \tag{3.10}$$

where $\mathbf{v}^{(0)} \equiv \mathbf{1}_m$ and the division is understood to be componentwise.

Large ε values typically lead to more dense coupling matrices, and small values lead to more sparse coupling, and the solution to the regularized problem approaches that of the unregularized problem in the limit $\varepsilon \to 0$. In general, the goal is to determine an optimal coupling matrix with as small a value of ε as possible; however, due to the factor of $1/\varepsilon$ in the definition of K in (3.8), small values of ε can lead to numerical overflow errors, especially for distributions with a large number of spots.

For the Gromov-Wasserstein formulation (3.6), the entropic regularization term allows the problem to be successively linearized,[71] amounting to an update for $\ell = 0, 1, 2, \ldots$

$$T^{(\ell+1)} = \underset{T \in \mathbb{R}_+^{n \times m}}{\operatorname{argmin}} \sum_{i,j} C_{i,j}^{(\ell)} T_{i,j} - \varepsilon H(T) \tag{3.11}$$

where $C^{(\ell)} = -D_1 T^{(\ell)} D_2$, with $T^{(0)} = \mathbf{a}\mathbf{b}^{\mathsf{T}}$. Then each update can be viewed as an entropically regularized classical OT problem, which can be approximated iteratively with the Sinkhorn algorithm above.

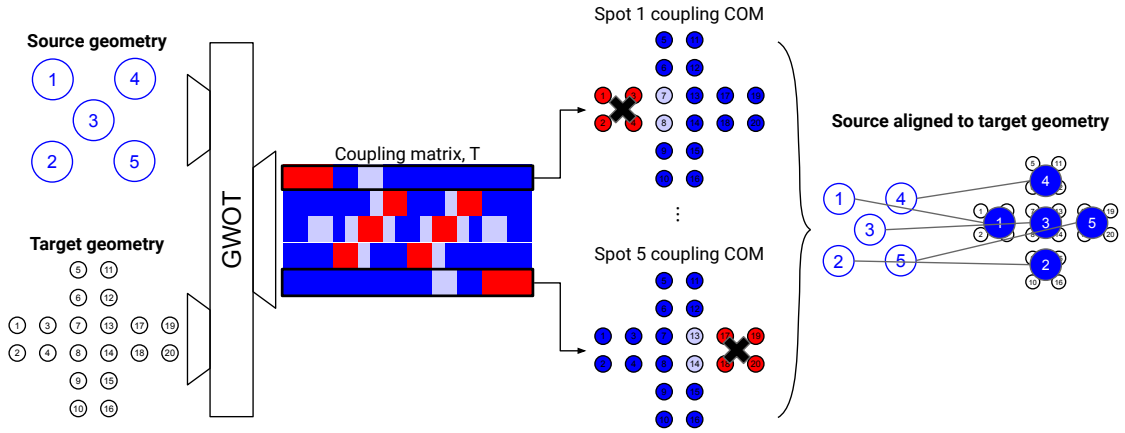

Figure 3.4: Overview of control point determination using the coupling matrix between two point clouds provided by Gromov-Wasserstein optimal transport (GWOT). GWOT accepts the two distance matrices D_1 and D_2 and outputs coupling matrix T satisfying (3.4) in classical OT or (3.6) in GWOT. This coupling matrix is then used to determine a set of control points representing the source image in the target image geometry by computing the weighted sum (3.12).

3.3.2 Identification of control points

Once an optimal coupling matrix T has been determined (with either classical OT or Gromov-Wasserstein OT), we use it to identify pairs of control points by transforming the spots in one image (hereafter, the source image) into the coordinate system of the other (hereafter, the target image). Given the coordinates of the m points in the source image, we use the coupling matrix T to form a weighted average of the coordinates of the spots in the target image, i.e.

$$\tilde{\mathbf{x}}_j = \frac{1}{S_j} \sum_{i=1}^{n} T_{i,j}^8 \mathbf{y}_i \tag{3.12}$$

where $S_j = \sum_{i=1}^{n} T_{i,j}^8$. The exponent of eight was chosen empirically to further sparsify the coupling since each element satisfies $0 \leq T_{i,j} \leq 1$ due to the mass constraints. These mapped coordinates then serve as control points for the source image spots in the target coordinate system. An overview of this process is depicted in Figure 3.4.

We show in Figure 3.5 a sampling of the set of control points determined through classi-

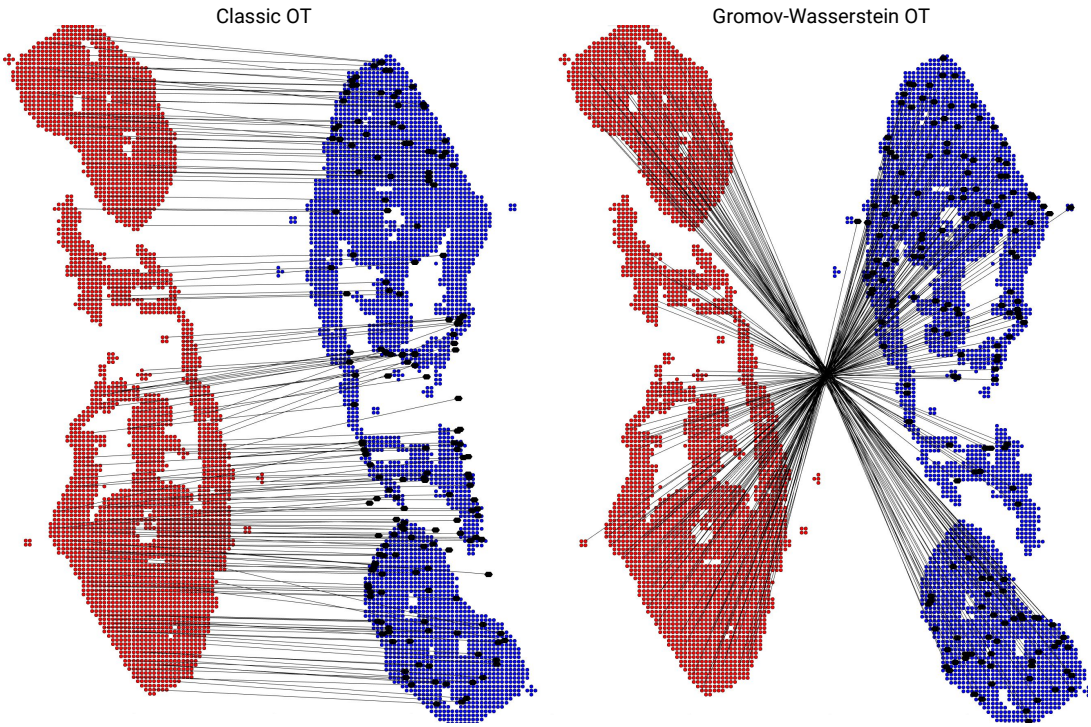

Figure 3.5: Control points determined using classical optimal transport (3.4, left) and Gromov-Wasserstein optimal transport (3.6, right) between two sets of identical points, extracted from the same image of Slice 1 of the murine neural crest from Soldatov.[4] In both panels, the source points (red) have been rotated by an angle of 180°, and lines connect these points to the corresponding mapped coordinates given by (3.12) using the relevant coupling matrix T. To reduce clutter, we have suppressed all but a randomly chosen 5% of control point pairs to indicate coupling. We see that GWOT is invariant to the source rotation while classic OT produces control points that are not desirable.

cal OT (3.4) and Gromov-Wasserstein OT (3.6) between two identical sets of coordinates representing Slice 1 of the murine neural crest, the second having been rotated through an angle of 180°. We see that classical OT is unable to produce realistic control points with this rotation while Gromov-Wasserstein is invariant to the rotation, still producing meaningful control points.

3.3.3 Determining a global transformation for image registration

After a set of control points has been identified between two images, we infer a global transformation using these points which can be applied pointwise to the source image to warp it into the coordinates of the target image so that its expression data can be extracted as in section 3.2. We do so with the help of the fitgeotrans () function in MATLAB's Image Processing Toolbox,[72] which allows the inference of several different types of transformations.

The simplest type of transformation allowed is a rigid transformation consisting of a rotation, a horizontal and vertical translation, and scale factor. Given a set of points $\{\mathbf{x}_j\}_{j=1}^m$ in the coordinate system of the source image and a corresponding set of points $\{\tilde{\mathbf{x}}_j\}_{j=1}^m$ in the coordinate system of the target image, a regression algorithm is used to fit a 3×3 matrix

$$A = \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ e & f & 1 \end{bmatrix} \quad \text{such that} \quad \begin{bmatrix} \tilde{\mathbf{x}}_j^\mathsf{T} & 1 \end{bmatrix} \approx \begin{bmatrix} \mathbf{x}_j^\mathsf{T} & 1 \end{bmatrix} A \quad \forall j = 1, 2, \ldots, m \tag{3.13}$$

This is the general matrix form of any composition of the following three elementary trans-

formations:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix} \text{(horizontal translation by } t_x\text{, vertical translation by } t_y\text{)}$$

$$\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{(horizontal scaling by } s_x\text{, vertical scaling by } s_y\text{)}$$

$$\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{(CCW rotation by } \theta\text{)}$$

Once the matrix A is determined, this informs a global transformation that can be applied to every point in the source image (e.g. using MATLAB's imwarp() function),

$$f(\mathbf{x}) = \begin{bmatrix} \mathbf{x} & 1 \end{bmatrix} A \tag{3.14}$$

A more general nonlinear transformation allowed by the fitgeotrans () function is a polynomial transformation of degree 2, 3, or 4. This transformation uses a similar regression algorithm to determine a set of coefficients to polynomials of the specified degree. If the coordinates of a point in the source image are given by (x, y), the output coordinates (u, v) in the target image are given by

$$u = \sum_{i+j=0}^{k} a_{i,j} x^i y^j, \qquad v = \sum_{i+j=0}^{k} b_{i,j} x^i y^j \tag{3.15}$$

where $k = 2$, 3 or 4, for quadratic, cubic, or quartic polynomial transformations, respectively, and the sum is over all valid (i, j) combinations. A quadratic polynomial requires at least

six control points to allow approximation of the six coefficients, a cubic polynomial requires ten, and a quartic polynomial requires fifteen.

A third type of transformation allowed by the fitgeotrans () function is a piecewise linear transformation. This transformation computes a Delaunay triangulation for the points in the source image and fits a linear transformation to each triangle.

The final transformation allowed by fitgeotrans () is a local weighted mean function, which for each point in the source image fits a quadratic polynomial using the k nearest neighbors in the source image ($k \geq 6$, as above), then applies a weighted average of each polynomial to each point in the source image.

Due to the relevant coefficients or parameters for each type of transformation being derived from a regression algorithm (the core of which requires inverting some matrix), there are certain regularity conditions the control points must meet to fit a transformation successfully. If too large of a regularization parameter ε is used in the entropic GWOT algorithm, the resulting dense coupling may lead to local discontinuities or overlap in the control point definitions. For example, point A may be above point B in the source space but in the target space point $\tilde{B}$ is above point $\tilde{A}$ while the surrounding points remain in the original orientation, or perhaps point $\tilde{A}$ and $\tilde{B}$ end up at exactly the same coordinates. In this case, the matrix to be inverted as part of the regression will not be full rank or be very ill-conditioned and thus can lead to numerical errors.

To ameliorate this issue, we make use of a sparsity parameter s to eliminate points mapped "too close" to each other in the target geometry, ensuring a more even spread of control points to better facilitate inference of a transformation. To sparsify the control points, we iteratively loop through the $i = 1, 2, \ldots, m$ mapped source spots in the target geometry, and any point mapped within $s\Delta x$ of point $\tilde{\mathbf{x}}_i$ is eliminated from the set of control points. Here, Δx is the grid spacing used in the spot detection algorithm in section 3.2. A check is performed after

iterating through all points to ensure that the minimum number of control points for each transformation type remains. This sparsification process is shown to empirically reduce the number of errors encountered when inferring a global transformation.

Once the desired transformation is fitted to the set of (sparsified) control points, it is applied pointwise to the source image to create a warped image in the target space, with any intermediate points interpolated linearly by default, though options for nearest neighbor or cubic spline interpolation are also supported. The resulting image is then able to have gene expression data extracted as in section 3.2.2, using the coordinates of the anchor spots. If desired, a new set of spots can be detected in the warped image, and the gene expression for the new spots in the anchor image can be redetermined. Once all G images have been aligned to the anchor, one final pass is made over all G images to ensure that every spot is associated with the correct expression information.

3.3.4 Accelerating the Gromov-Wasserstein computation

The Gromov-Wasserstein optimal transport (GWOT) algorithm used to register and align two different images is computationally expensive and thus any speed improvement would significantly speed up the total workflow of creating a reference atlas from a collection of images. Indeed, the amount of time taken to run the GWOT algorithm is experimentally shown (Figure 3.6D) to scale linearly with the product of the number of spots in the source and target images. On an iMac with a 3.6 GHz 8-Core Intel Core i9 CPU and 64 GB of 2667 MHz DDR4 RAM, typical run times for two images with ~ 3000 spots is upwards of 20-30 minutes. For this reason, we have developed an algorithm to reduce the number of spots considered for GWOT, as detailed in this section.

Given a set of m points in a source image and n points in a target image, as well as grid spacings Δx_s and Δx_t, respectively, we define a scaling parameter μ to reduce the resolution

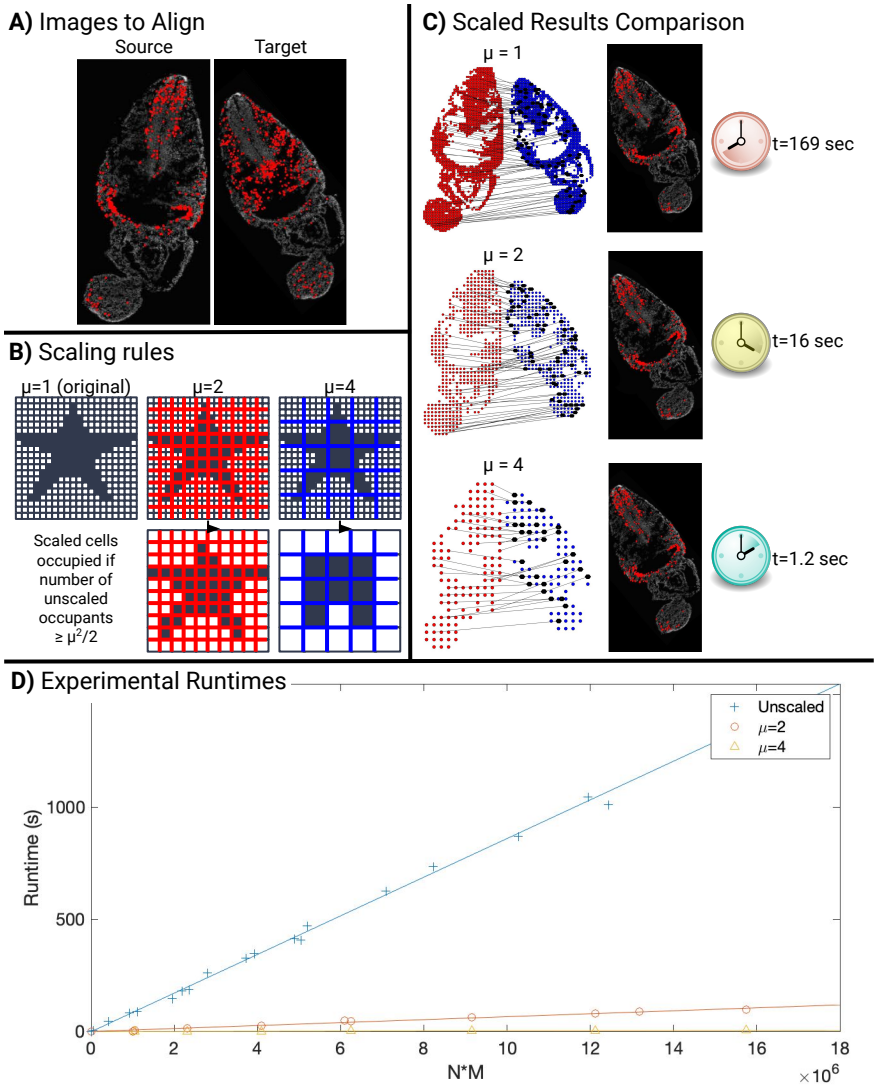

Figure 3.6: **(A)** Example images of cell expression to be aligned. The source image depicts expression of gene Gbx2 in Slice 13 of the murine neural crest from Soldatov[4] and the target image depicts expression of gene Foxd3 in the same. **(B)** Example scalings with $\mu = \{1, 2, 4\}$ for an initial point cloud. Cells in the low-resolution grid are marked occupied if more than $\mu^2/2$ corresponding cells in the high-resolution grid are occupied. **(C)** Resulting alignment of the source image into the target geometry based on the control points determined by GWOT with a scaling parameter of $\mu = 1$ (original, top), $\mu = 2$ (center), and $\mu = 4$ (bottom). In each, we learn a local weighted mean transformation, choosing $k = 50$ nearest neighbors to inform the local quadratic polynomial transformation, as well as a sparsity parameter $s = 5$. Also included are the runtimes required to obtain the aligned source images. **(D)** Experimentally observed runtime to determine the coupling matrix T using GWOT for various numbers of input points $N \cdot M$. The runtime scales linearly with this product, and the scale factor μ decreases computation time by a factor of $\mathcal{O}(\mu^4)$.

of the atlas in order to compute an optimal transport coupling more efficiently. To do so, we overlay a new, lower resolution grid atop the existing grids in the source and target images, with grid spacing $\mu\Delta x_s$ and $\mu\Delta x_t$, respectively. We then iterate over the cells in this low resolution grid, marking a cell as occupied if the number of spots inside the cell is greater than or equal to $\mu^2/2$. When $\mu = 2$, this corresponds to a doubling of the grid spacing so that at most four spots would occupy each cell, and cells are marked occupied if $\mu^2/2 = 2$ or more spots are inside the cell. For a value of $\mu = 3$, a tripling of the grid spacing for a maximum of nine spots in each cell, cells are marked occupied if $\mu^2/2 = 4.5$ or more spots occupy each cell. Figure 3.6B explains the scaling rules with a toy example.

The centers of the occupied cells in the low resolution source and target grids then serve as the support of the distribution supplied to the GWOT algorithm. Figure 3.6C shows scaled spots extracted from the source and target images in Figure 3.6A, a sample of the control points determined by GWOT as in section 3.3.2, and the resulting warped source image determined by a local weighted mean transformation as in section 3.3.3. In each, we choose $k = 50$ nearest neighbors to inform the local quadratic polynomial transformation, as well as a sparsity parameter $s = 5$. We also include the runtimes required to obtain the aligned source images for each scaling parameter.

Because the same scaling is applied to the grid spacing in the source and target atlas, and the runtime of GWOT is directly proportional to the product of the number of spots in the source and target, the runtime with a scaling factor of μ decreases by a factor of $\mathcal{O}(\mu^4)$, a significant improvement. We show in Figure 3.6D experimental results for computation time required to determine a coupling matrix various numbers of input points using both unaccelerated ($\mu = 1$) and accelerated ($\mu = \{2, 4\}$) GWOT. Furthermore, the quality of the mapping empirically does not appear to suffer, even in combination with the sparsification process described in section 3.3.3.

3.4 Conclusion

We have developed the software suite AtlasGeneratorOT, accessible as a standalone app via the free MATLAB Runtime Environment, to serve as a useful tool to create future reference atlases in a largely automated fashion with minimal user input. The graphical user interface, a screenshot of which is shown in Figure 3.7, allows the user to select an anchor image, detect spots in the anchor image using the various thresholds and options described in section 3.2.1, then extract expression levels from the anchor and all other images using the settings described in section 3.2.2. If any image in the collection is not aligned to the same geometry as the anchor image, AtlasGeneratorOT gives the user the option to automatically register and align the two images using the Gromov-Wasserstein optimal transport-based algorithm described in section 3.3. A manual override of the alignment and of all parameters involved is also provided for extra fine tuning.

Using the novel GWOT acceleration algorithm in section 3.3.4, AtlasGeneratorOT can greatly reduce the time and overhead required to generate novel (two-dimensional) reference atlases, such as those for each of the 15 serial slices through the murine neural crest provided by Soldatov.[4] As more and more imaging data becomes available due to high throughput spatial transcriptomic techniques, AtlasGeneratorOT can serve as a useful tool to speed up the workflow required to integrate scRNA-seq data with spatial imaging data.

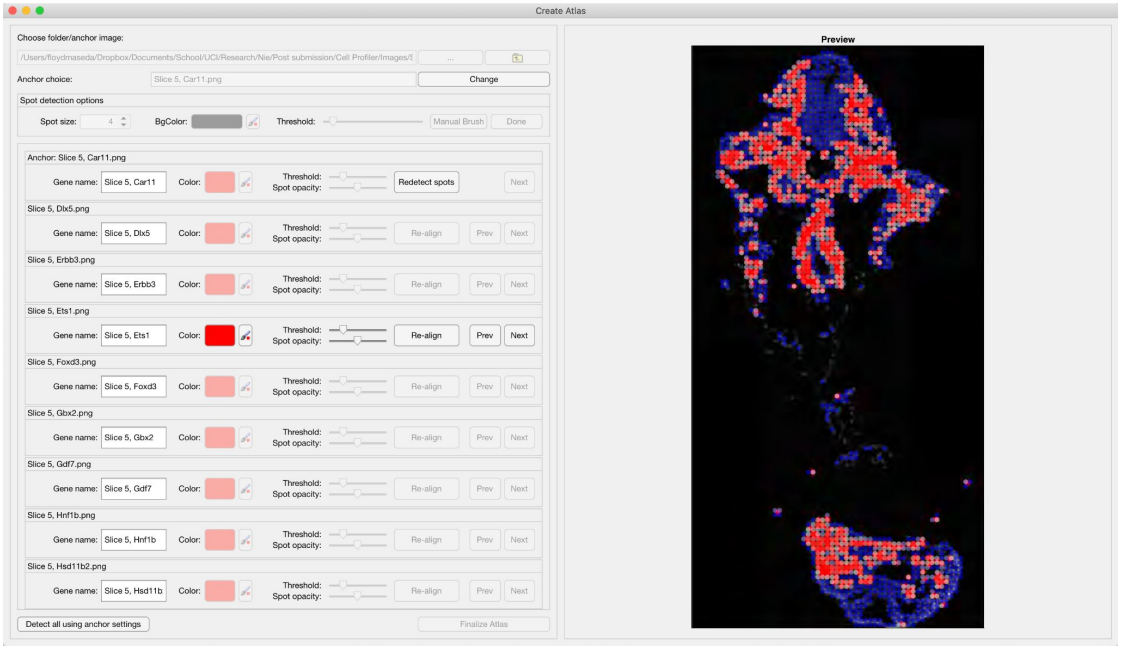

Figure 3.7: Screenshot of the AtlasGeneratorOT GUI for extracting gene expression information from a collection of images to create a reference atlas. Shown is the current detection of gene Ets1 in Slice 5 of [4]. Options are available to change the threshold δ for each image as well as the gene color $\mathbf{c}_g$ (cf. Section 3.2.2), and to re-align the current image to the anchor geometry if necessary.

Chapter 4

Combining multiple 2-D reference atlases into a cohesive 3-D reference atlas

As with the neural crest dataset of Soldatov[4] examined in Chapter 3, it is often the case that a three-dimensional biological system is imaged in multiple two-dimensional slices. These two-dimensional slices may then need to be recombined to form a coherent three-dimensional structure. The extraction and alignment process in sections 3.2 and 3.3 can be used to construct separate two-dimensional reference atlases of each slice, which consists of a set of spot coordinates, $\{\mathbf{x}_i\}_{i=1}^{n} \in \mathbb{R}^2$, and a corresponding $n \times G$ matrix, where G is the number of genes in the reference atlas, the columns of which describe the expression level of each gene at each spot on a scale from 0 (not at all expressed) to 1 (highly expressed).

Similar to how images of the expression of genes *within* a slice may not conform to the same geometry and must therefore be aligned to a common geometry as in section 3.3, the collection of two-dimensional reference atlases may also need to be aligned into a common

geometry. We therefore include a separate GUI as part of AtlasGeneratorOT which allows the combination of two-dimensional slices into a coherent three-dimensional atlas by again making use of optimal transport to couple spots between two atlases, identify pairs of control points, and determine a global mapping of one slide onto another. After all slices are aligned to the same 2-D coordinate system, each can be assigned a z value according to the inter-slice distance(s), thus forming a coherent three-dimensional reference atlas.

We describe in this chapter the various methods by which AtlasGeneratorOT can align reference atlases of two or more slices to the same geometry. We further outline in section 4.3 a separate function which allows optimal transport-based interpolation *between* existing two-dimensional slices, further improving the quality of the resulting three-dimensional atlas.

4.1 Fused Gromov-Wasserstein

If the spacing between 2-D slices of a 3-D system is small enough that there is not much geometric variation between them, coupling spots in two slices with classical optimal transport (3.4) may lead to reasonable results. However, if the slices are spaced far apart from each other or there is a lot of geometric variation between them, more advanced OT variations may be required. One possibility is to again make use of the Gromov-Wasserstein formulation (3.6) used to align images within the same slice, which considers the structure of each reference atlas (i.e. the intra-spot distances) to determine a coupling matrix. As in the alignment of images of the same slice, the Gromov-Wasserstein OT (GWOT) formulation typically leads to more reasonable results in this setting than classical OT, as in Figure 4.1 where we show alignment of spots extracted from Slide 2 of the murine neural crest from [4] to those extracted from Slice 1 of the same system for various scale parameters $\mu = \{1, 2, 4\}$ (cf. section 3.3.4). Because the two slices have very similar structures, GWOT is able to provide reasonable couplings here, even for large μ.

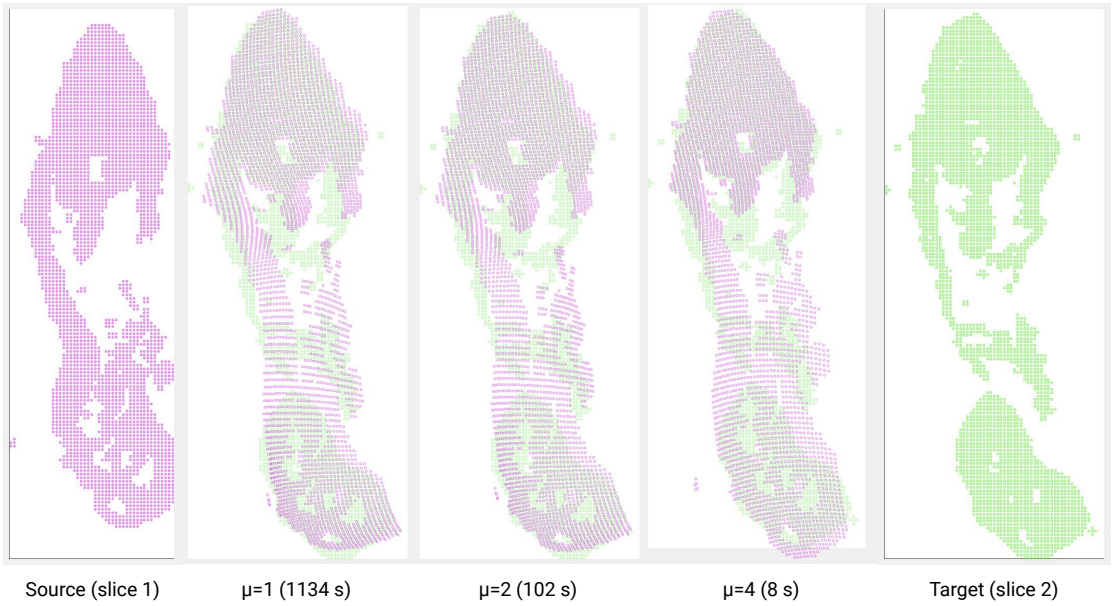

Figure 4.1: An example mapping of two point clouds extracted by the procedure in section 3.2.1 from an image of Slice 1 (far right, green) and Slice 2 (far left, magenta) of the murine neural crest from Soldatov.[4] We superimpose in magenta on top of the target spots in green the result of the pointwise application to the source spots of the global map inferred from the set of control points obtained by GWOT with $\mu = 1$ (unscaled, left), $\mu = 2$ (middle), and $\mu = 4$ (right). Each is labelled with the runtime required to produce such a mapping.

However, in general the output from GWOT may still exhibit non-ideal behavior such as unwarranted reflections or rotations between neighboring slides. In these cases, not all is lost, however. Indeed, in contrast to the task of aligning two images of gene expression within the same slice, the task of aligning two atlases might ideally make use of the existence of not only coordinate information, $\{\mathbf{x}_i\}_{i=1}^n \in \mathbb{R}^2$, but also feature information in the form of gene expression data, $\{\mathbf{g}_i\}_{i=1}^n \in [0,1]^G$, for each spot. GWOT, however, ignores the features and only incorporates structure data.

Another variation of optimal transport called Fused Gromov-Wasserstein optimal transport, introduced by Vayer et al,[73, 74] incorporates *both* structure *and* feature information to provide a coupling. The base assumption in the Fused Gromov-Wasserstein formulation is that the two distributions α and β are viewed as tuples $\{(\mathbf{x}_i, \mathbf{g}_i)\}_{i=1}^n \in M_1 \times [0,1]^G$ and $\{(\mathbf{y}_j, \mathbf{g}_j)\}_{j=1}^m \in M_2 \times [0,1]^G$, where (M_1, d_1) is the metric space for the spots in the source

image, (M_2, d_2) is the metric space for the spots in the target image, and $([0,1]^G, d)$ is the *shared* feature space for both images. Defining $\delta(\mathbf{x}, \mathbf{g})$ as the Dirac distribution over the product space, we can thus write

$$\alpha = \sum_{i=1}^{n} \mathbf{a}_i \delta(\mathbf{x}_i, \mathbf{g}_i), \qquad \beta = \sum_{j=1}^{m} \mathbf{b}_j \delta(\mathbf{y}_j, \mathbf{g}_j)$$

Given this assumption, we compute two matrices $D_1 \in \mathbb{R}_+^{n \times n}$ and $D_2 \in \mathbb{R}_+^{m \times m}$ where $(D_1)_{i,i'} = d_1(\mathbf{x}_i, \mathbf{x}_{i'})$ and $(D_2)_{j,j'} = d_2(\mathbf{y}_j, \mathbf{y}_{j'})$ as in Gromov-Wasserstein, as well as the feature matrix $C \in \mathbb{R}_+^{G \times G}$ where $C_{i,j} = d(\mathbf{g}_i, \mathbf{g}_j)$ as in classical OT. The Fused Gromov-Wasserstein distance is then defined using a tradeoff parameter $t \in [0,1]$ and $p, q \geq 1$ as

$$FGW_{p,q}^p = \min_{T \in \mathbb{R}_+^{n \times m}} \sum_{i,i',j,j'} \left((1-t) C_{i,j}^p + t \left| (D_1)_{i,i'} - (D_2)_{j,j'} \right|^p T_{i',j'} \right)^q T_{i,j} \tag{4.1}$$

where T is subject to the same mass constraints as in classical and Gromov-Wasserstein OT, $T\mathbf{1}_m = \mathbf{a}$, $T^\mathsf{T}\mathbf{1}_n = \mathbf{b}$. Vayer shows that this quantity satisfies all metric axioms iff $q = 1$, which we henceforth assume for simplicity. We note that if $t = 0$, this reduces to the pth Wasserstein distance in the classical OT formulation, and if $t = 1$ this reduces to the pth Gromov-Wasserstein distance; thus, the Fused Gromov-Wasserstein distance can be interpreted as the most general distance of the three.

In the case $q = 1$, the FGW optimization problem can be viewed as a classical OT problem (3.4) with cost matrix

$$\bar{C}_{i,j} = (1-t) C_{i,j}^p$$

and a nonentropic regularization term,

$$\bar{H}(T) = \sum_{i,i',j,j'} \left| (D_1)_{i,i'} - (D_2)_{j,j'} \right|^p T_{i',j'} T_{i,j}.$$

so that (4.1) reads

$$FGW_p^p = \min_{T \in \mathbb{R}_+^{n \times m}} \sum_{i,j} \bar{C}_{i,j} T_{i,j} + t\bar{H}(T) \tag{4.2}$$

Because of this nonentropic regularization term, the standard Sinkhorn algorithm (3.10) cannot be applied. Instead, Vayer[74] utilizes a Franke-Wolfe style conditional gradient method wherein during each iteration, the gradient of (4.2) with respect to T,

$$\frac{\partial FGW_p^p}{\partial T}(T) \equiv G(T) = (1-t)C_{i,j}^p + 2t \sum_{i',j'} |(D_1)_{i,i'} - (D_2)_{j,j'}|^p \, T_{i',j'} \tag{4.3}$$

is evaluated at $T = T^{(\ell-1)}$, with $T^{(0)} = \mathbf{a}\mathbf{b}^\mathsf{T}$, and an unregularized classic OT problem is solved with cost function

$$C^{(\ell)} = G(T^{(\ell-1)}) \tag{4.4}$$

producing a transport matrix $\tilde{T}^{(\ell)}$. A line search algorithm (Algorithm 2 of [73]) is then used to find the optimal step size $\tau^{(\ell)} \in [0, 1]$ along a second degree polynomial fitted to the non-convex loss, the solution of which defines the next iterate,

$$T^{(\ell)} = (1 - \tau^{(\ell)})T^{(\ell-1)} + \tau^{(\ell)}\tilde{T}^{(\ell)} \tag{4.5}$$

This iteration is repeated until convergence, which is only guaranteed to be local due to the nonconvexity of the problem.

When applying Fused Gromov-Wasserstein optimal transport (Fused GWOT) to align two reference atlases, we use a similar acceleration technique as that used to accelerate GWOT in section 3.3.4. In addition to obtaining a low-resolution representation of the atlas structure in the form of bins on a regular grid, the gene expression information is averaged over the spots contributing to each bin in the low-resolution atlases. This averaged gene expression is then passed to the Fused GWOT algorithm as feature information for each bin. For a given scaling parameter μ as in accelerated GWOT, we also observe a similar $\mathcal{O}(\mu^4)$ decrease in

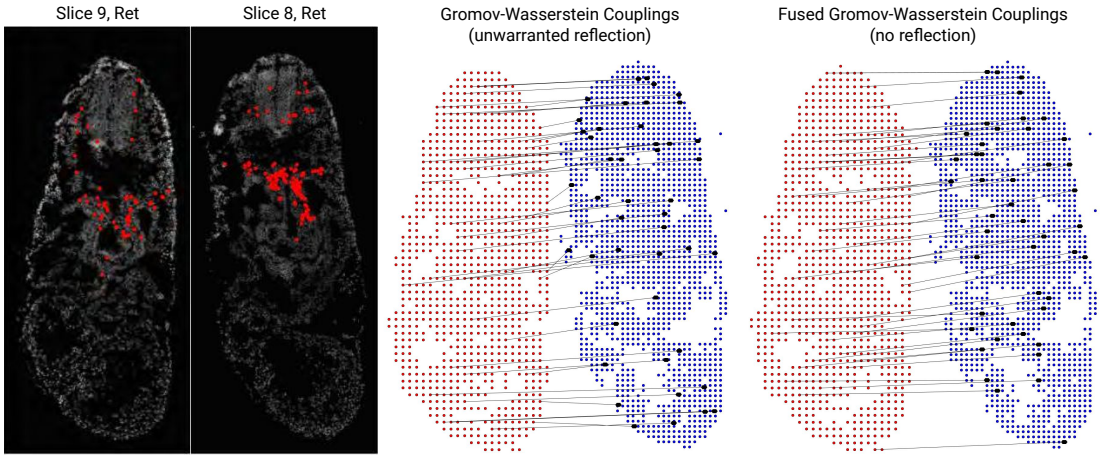

Figure 4.2: Example couplings of atlases obtained from Slice 8 and Slice 9 of the murine neural crest from [4] (example images, left) provided by GWOT (center) and Fused GWOT (right). Since Fused GWOT not only considers structure information but also feature information in the form of gene expression for each spot, it is more able to correctly couple spots in each atlas without introducing an unwarranted horizontal reflection than is GWOT, which only incorporates structure information.

runtime for accelerated Fused GWOT. Similarly, we make use of the same sparsity parameter s described in section 3.3.3 for Fused GWOT to ensure sufficiently well-conditioned matrices for the regression used by fitgeotrans().

Because Fused GWOT also considers feature information, the coupling is often more accurate than in Gromov-Wasserstein OT (GWOT). For example, the atlases extracted from images of Slice 8 and Slice 9 of the murine neural crest from [4] have very similar structures, as shown in Figure 4.2. GWOT tends to introduce a horizontal reflection into the mapping (Figure 4.2, center), which at first glance may seem warranted even to a human. However, when incorporating feature data such as the expression of gene Ret (Figure 4.2, left), which appears to skew slightly right of center, it becomes evident that this reflection is likely erroneous. The Fused GWOT algorithm is able to incorporate this feature information into the coupling and results in more accurate control points (Figure 4.2, right).

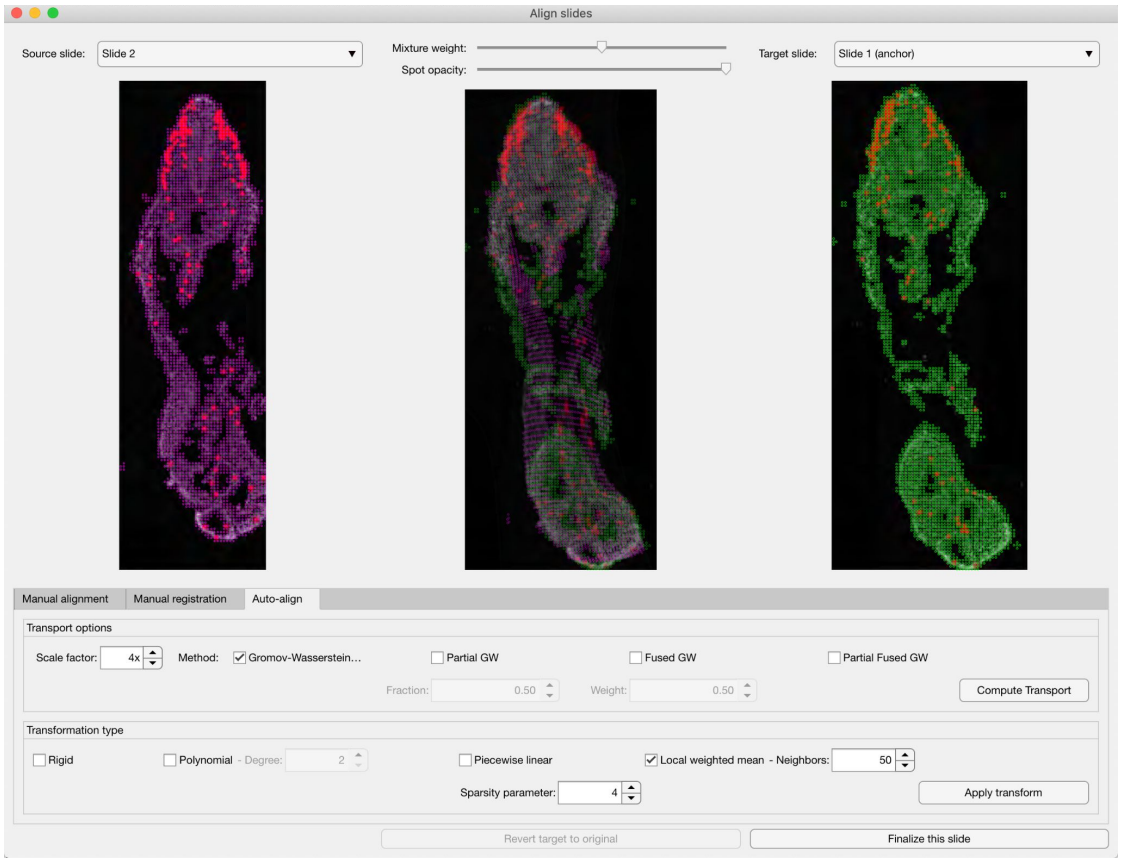

Figure 4.3: Screenshot of the AtlasGeneratorOT GUI for aligning a collection of two-dimensional reference atlases to a common anchor geometry. Shown are the reference atlases generated by AtlasGeneratorOT as in Chapter 3 for Slice 1 and 2 of the murine neural crest.[4]. Options for each of the formulations of optimal transport are included as described in the text.

4.2 Partial Optimal Transport

In classical optimal transport (3.4), the coupling matrix T is assumed to transport all of the mass from the source distribution α to the target distribution β. As such, both α and β are assumed to have unit mass ($\sum_i \mathbf{a}_i = \sum_j \mathbf{b}_j = 1$) to facilitate transfer. In some applications, it may be useful to specify that only part of the mass need be transferred. For example, given neighboring two-dimensional slices of an atlas, it may be that one atlas fails to depict part of the underlying biological system. In this case, we can reformulate a generalization of the classical OT problem to only transport part of the mass, called the partial OT problem.[75] Whereas in classical OT, the restrictions on the coupling matrix T were equalities, requiring $T\mathbf{1}_m = \mathbf{a}$, $T^\mathsf{T}\mathbf{1}_n = \mathbf{b}$, in partial OT, these are replaced with inequalities,

$$T\mathbf{1}_m \leq \mathbf{a}, \qquad T^\mathsf{T}\mathbf{1}_n \leq \mathbf{b}, \qquad \mathbf{1}_n^\mathsf{T} T \mathbf{1}_m = \gamma \tag{4.6}$$

where $\gamma \in (0, 1)$ is the fraction of mass to be transported.

In practice, this can be reformulated into a balanced classical OT problem with equality constraints (3.4) by adding virtual points $\mathbf{x}_{n+1}$ to the source and $\mathbf{y}_{m+1}$ to the target and extending the standard cost matrix as

$$\tilde{C} = \begin{bmatrix} C & \xi\mathbf{1}_m \\ \xi\mathbf{1}_n^\mathsf{T} & 2\xi + A \end{bmatrix} \tag{4.7}$$

for any $A > \max(C_{i,j})$ and $\xi > 0$. Setting the mass of these points to $\mathbf{a}_{n+1} = \mathbf{b}_{m+1} = 1 - \gamma$ results in a balanced OT problem with the augmented source distributions $\tilde{\mathbf{a}} = [\mathbf{a}, 1-\gamma]$, $\tilde{\mathbf{b}} = [\mathbf{b}, 1-\gamma]$. The optimal coupling matrix for the partial OT problem can be shown to be the optimal coupling matrix from the augmented problem, deleting the last row and column.

A similar extension exists for partial Gromov-Wasserstein optimal transport (PGWOT),

wherein similar to balanced GWOT, the extended problem (3.6) is successively linearized as in (3.11), and the final coupling matrix is similarly taken to be the solution to the extended problem, deleting the last row and column. To the current author's knowledge, however, no similar extension to allow for partial Fused Gromov-Wasserstein exists; however, a similar approach with extended matrices may be possible.

Figure 4.3 shows a screenshot of the portion of the AtlasGeneratorOT graphical user interface devoted to aligning two-dimensional atlases into the same underlying geometry. Shown are all options related to the optimal transport algorithms described in this chapter, including the scale parameter μ, the sparsity parameter s, and the fraction (γ) and weight/tradeoff (t) parameters from partial and fused GWOT, respectively. Also included are the relevant options for each of the global transformations able to be inferred from the control points decided by the chosen OT algorithm using the fitgeotrans() function described in section 3.3.3. Also included in the software is the ability to override any of the OT couplings and manually align two slides using a rigid transformation.

We further show in Figure 4.4A a sample of the fully aligned reference atlas for all 15 slices in the murine neural crest. All images depict the expression levels of gene Car11 and have been aligned using either Fused Gromov-Wasserstein or Gromov-Wasserstein optimal transport if results are reasonable, though some have been manually aligned using the override options provided by AtlasGeneratorOT. Although we describe a partial OT algorithm above, we did not make use of partial OT to align any slides. However, particularly for the alignment between Slides 2 and 3 and between Slides 4 and 5, where Slices 3 and 4 appear to only contain a fraction of the system compared to Slices 2 and 5, a partial OT mapping may be preferable. For these slides, we resorted to a manual alignment as we were unable to obtain a sufficiently accurate mapping using any of the optimal transport methods described herein. We also show in Figure 4.4B an example of Slices 1-4 endowed with a z coordinate to form a (portion of a) coherent three-dimensional atlas. We show the atlases before (left) and after

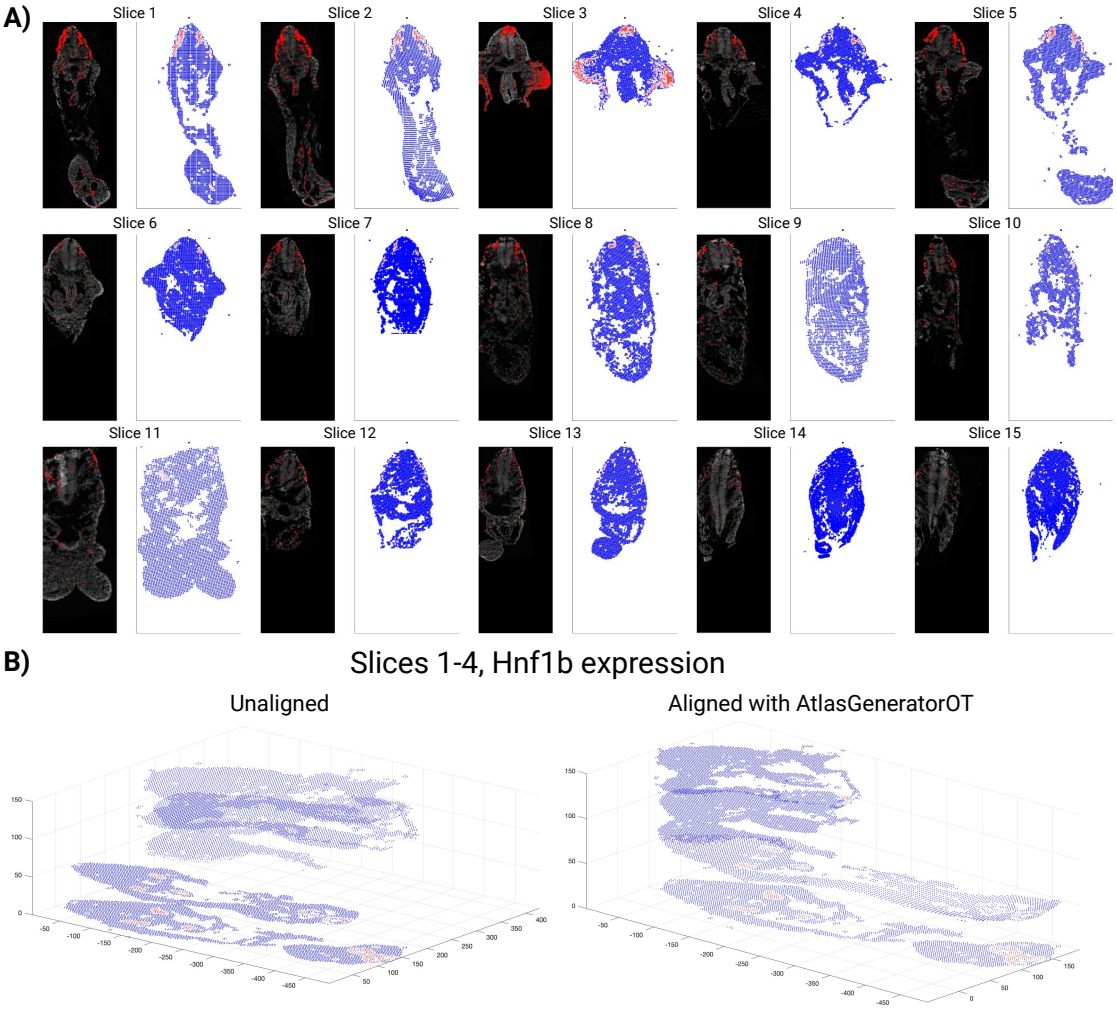

Figure 4.4: **(A)** Finished alignment of all 15 slices of the murine neural crest created with AtlasGeneratorOT to a common geometry. Each slice depicts the expression level of gene Car11; however the full atlas includes expression levels of all 32 genes provided by Soldatov.[4] All slices have been aligned using Fused Gromov-Wasserstein optimal transport where results were reasonable, or manual alignment where optimal transport failed to give expected results (e.g. between slices 2 and 3 and between slices 4 and 5). **(B)** Three-dimensional plot of expression of gene Hnf1b in Slices 1-4 pre-alignment (left) and post-alignment (right) with AtlasGeneratorOT.

(right) alignment with AtlasGeneratorOT. We note that the aligned 3-D atlas appears to show a more coherent structure than the collection of unaligned 2-D atlases.

4.3 Interpolating between slices in a 3-D reference atlas

After aligning the reference atlases of all two-dimensional slices in a biological system to a common two-dimensional coordinate system and assigning a z coordinate to each to form a coherent three-dimensional reference atlas as in Figure 4.4B, the resolution of the atlas within each two-dimensional slice is often much higher than the resolution in the z direction due to sparse slicing. We discuss in this section a method also based on optimal transport which allows interpolation between two-dimensional slices to further improve the quality of the final three-dimensional atlas.

Suppose that we have two 2-D reference atlases in the planes $z = a$ (the source atlas) and $z = b$ (the target atlas). Denote the spot centers in the source atlas as $\{(x_i^{(a)}, y_i^{(a)})\}_{i=1}^n \equiv \{\mathbf{x}_i^{(a)}\}_{i=1}^n$, and those in the target atlas as $\{(x_j^{(b)}, y_j^{(b)})\}_{j=1}^m \equiv \{\mathbf{x}_j^{(b)}\}_{j=1}^m$. For some $t \in (0, 1)$, we wish to form an interpolated reference atlas at $z = (1-t)a+tb$, somewhere in the interval (a, b). For example $t = 0.5$ would produce a new interpolated atlas at $z = (a+b)/2$, halfway between the two given atlases.

To produce such an atlas, we formulate the problem in terms of a Wasserstein barycenter computation,[69, 67, 65] wherein given a set of distributions $\{\alpha_k\}_{k=1}^K$ over some metric space (M, d) and a corresponding set of weights $\{\lambda_k\}_{k=1}^K$ such that $\sum_k \lambda_k = 1$, we seek a new distribution $\alpha^* \in M$ that solves

$$c \tag{4.8}$$

where W_p is the pth Wasserstein distance, as in 3.5. This can be viewed as a special case of the more general Fréchet mean computation over any metric space $(\mathcal{X}, d)$,

$$\operatorname*{argmin}_{x \in \mathcal{X}} \sum_{k=1}^{K} \lambda_k d(x, x_k)^p$$

For example, when $\mathcal{X} = \mathbb{R}^n$, $d(x, y) = \|x - y\|_2$, $p = 2$, the resulting minimizer is the arithmetic mean of input points, $\sum_k \lambda_k x_k$.

Even for discrete distributions, the barycenter problem (**??**) can be computationally expensive to solve. Most formulations involve discretizing the metric space M into a regular grid with $N \gg \max(n, m)$ cells and representing the distributions α_k as histograms over the N cells.[65] Computing the common cost matrix $C_k = C = D^p$, where $D \in \mathbb{R}_+^{N \times N}$ is a distance matrix containing pairwise distances between grid cells, the problem can be viewed as a large linear program wherein we seek a set of coupling matrices $\{T_k\}_{k=1}^K \in \mathbb{R}_+^{N \times N}$ between the histogram of each input distribution α_k and the histogram of the barycenter distribution α^*, which are each subjected to the same mass constraint, $T_k^\mathsf{T} \mathbf{1}_N = \mathbf{a}$, where $\mathbf{a}$ is the histogram representing α^*. That is, we seek a solution to

$$\operatorname*{argmin}_{\mathbf{a} \in \mathbb{R}_+^N,\, T_1, T_2, \ldots T_K \in \mathbb{R}_+^{N \times N}} \sum_{i,j,k} \lambda_k (T_k)_{i,j} (C_k)_{i,j} \quad \text{such that} \quad \forall k, T_k \mathbf{1}_N = \mathbf{a},\ T_k^\mathsf{T} \mathbf{1}_N = \mathbf{a}_k \tag{4.9}$$

The scale of this linear program makes computation of a solution unwieldy in general, though several approaches exist for the general case, including using entropic regularization (3.7) to make the problem smooth and convex so that a modified Sinkhorn algorithm can be used as in classical OT (3.10).

In certain special cases, however, the barycenter problem (4.9) has known explicit solutions. For our case where $K = 2$, i.e. we seek the barycenter of only two discrete distributions with weights $\lambda_1 = t$, $\lambda_2 = 1 - t$, $t \in (0, 1)$, it is shown in [65] that the optimal solution is

equivalent to McCann's interpolation,[76]

$$\alpha_t^* = \sum_{i,j} T_{i,j}\delta((1-t)\mathbf{x}_i^{(a)} + t\mathbf{x}_j^{(b)}) \equiv \sum_{i,j} T_{i,j}\delta(\mathbf{x}_{(i,j,t)}^*) \tag{4.10}$$

where $T \in \mathbb{R}^{n\times m}$ is the solution to the classical OT problem (3.4) with cost matrix $C_{i,j} = \|\mathbf{x}_i^{(a)} - \mathbf{x}_j^{(b)}\|_2$. This explicit solution speeds up computation tremendously compared to the general case, with the added benefit that no discretization of the metric space is required, resulting in smaller matrices. Furthermore, multiple interpolated reference atlases can be generated from the same coupling matrix.

An exact solution T to (3.4) can be shown to have at most $n+m+1$ nonzero elements,[65] each of which is interpreted as a spot in the interpolated atlas; however, if entropic regularization is used as we do, there may be many more nonzero elements in T. To prevent the interpolated atlas from containing too many spots, we set a threshold value ξ on T, only keeping the spot $\mathbf{x}_{(i,j,t)}^*$ in the interpolated atlas if $T_{i,j} \geq \xi$. To further prune the interpolated atlas, we limit each spot $\mathbf{x}_i^{(a)}$ in the source to be associated with at most three interpolated spots $\mathbf{x}_{(i,j,t)}^*$, choosing the three corresponding spots $\mathbf{x}_{j_1}^{(b)}$, $\mathbf{x}_{j_2}^{(b)}$, and $\mathbf{x}_{j_3}^{(b)}$ to be those with the largest coupling coefficients in the ith row of T. We also prune in the other direction, selecting for each target spot $\mathbf{x}_j^{(b)}$ only the top three coupled spots in the source atlas. This heuristic appears to lead empirically to reasonably sized interpolated atlases.

Once the spots in the interpolated atlas are determined, it is straightforward to interpolate the expression information from the source and target slices. For the spot $\mathbf{x}_{(i,j,t)}^* = (1-t)\mathbf{x}_i^{(a)} + t\mathbf{x}_j^{(b)}$, we simply assign the linearly interpolated gene expression $\mathbf{g}_{(i,j,t)}^* = (1-t)\mathbf{g}_i^{(a)} + t\mathbf{g}_j^{(b)}$, where $\mathbf{g}_i^{(a)} \in [0,1]^G$ is the gene expression associated with spot $\mathbf{x}_i^{(a)}$ in the source, and $\mathbf{g}_j^{(b)} \in [0,1]^G$ is the gene expression associated with spot $\mathbf{x}_j^{(b)}$ in the target.

We show in Figure 4.5A computed interpolations between Slices 1 and 2 and in Figure 4.5B those between Slices 4 and 5, for interpolation parameter values $t = \{1/6, 1/3, 1/2, 2/3, 5/6\}$.

To demonstrate the interpolation not only of geometry but also of gene expression, we label each spot with the interpolated expression level of gene Msx1. We note that with the similarity of Slices 1 and 2, the interpolation appears to work almost flawlessly; however, the interpolation between Slices 4 and 5 is not as ideal. This may be due to the fact that the spots in Slice 4 can be recognized as depicting only a portion of the spots in Slice 5, excluding the bottom part of the system. Because of this, optimal transport appears to have trouble figuring out which spots in the source are coupled to that region in the target, if any. Obtaining a coupling matrix using partial optimal transport as in section 4.2 may lead to better results.

We further show in Figure 4.5C an interpolated version of the expression of gene Hnf1b in slices 1-4 originally shown figure 4.4B. Compared to the uninterpolated version, the interpolated version fills in the gaps with relevant detail, providing for a more high quality three-dimensional reference atlas.

4.4 Conclusion

We have extended the capabilities of AtlasGeneratorOT to be able to not only extract and compile a two-dimensional atlas from a collection of images as in Chapter 3, but in the case of a three-dimensional biological system imaged in multiple serial two-dimensional slices, to be able to align those slices to a common geometry and supplement the existing slices with any number of intermediate interpolations to provide a more complete structure of the three-dimensional system. With an intuitive GUI as in Figure 4.3, AtlasGeneratorOT can largely automate a process which in the absence of software may take many hours. Using the powerful framework of the various formulations of optimal transport, AtlasGeneratorOT can serve as a useful tool to speed up the workflow required to integrate scRNA-seq data with spatial imaging data.

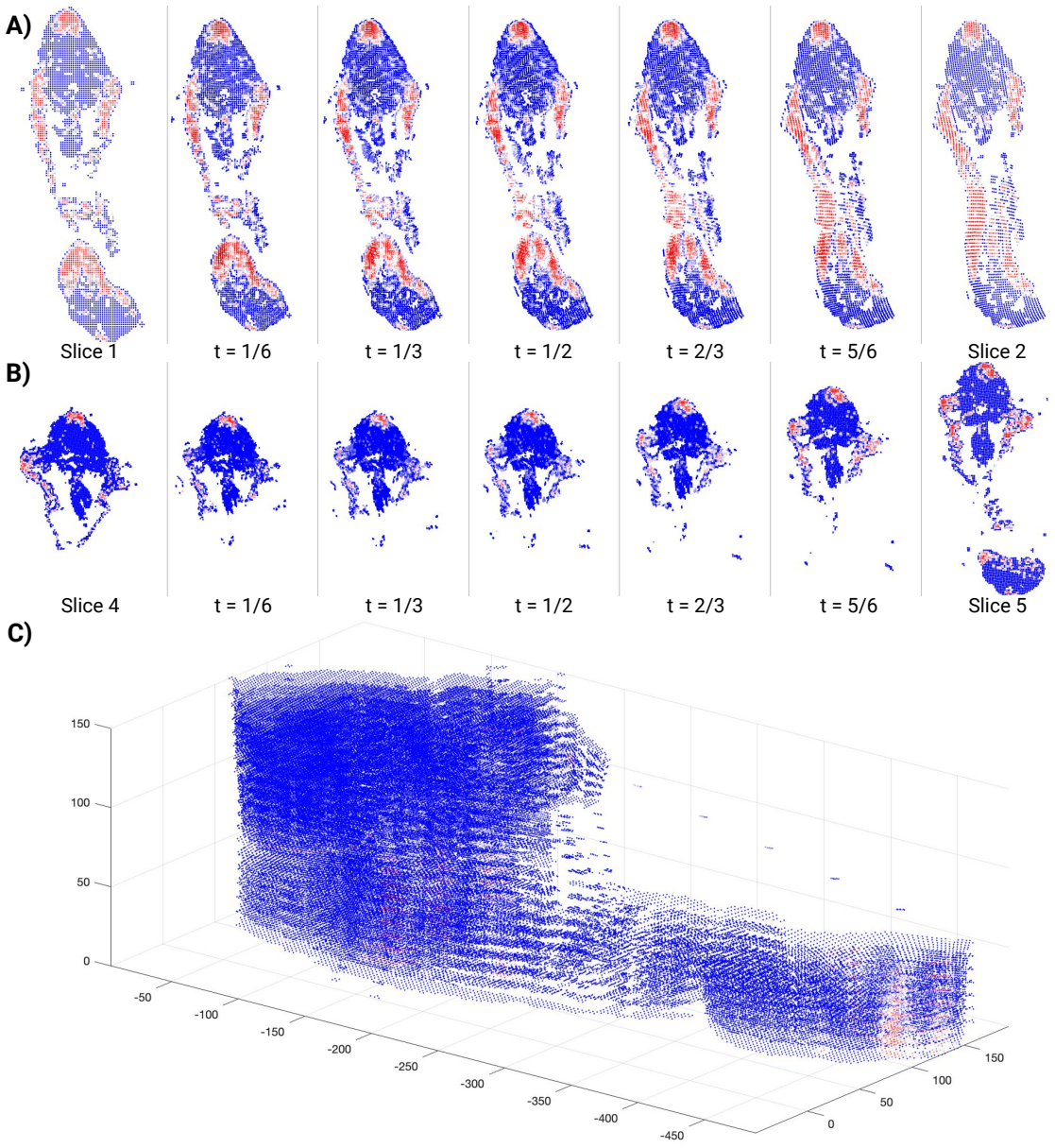

Figure 4.5: Example interpolations between **(A)** Slice 1 and Slice 2 **(B)** Slice 4 and Slice 5, for values of $t = \{1/6, 1/3, 1/2, 2/3, 5/6\}$. Displayed are the interpolated positions and expression levels for all spots $\mathbf{x}^*_{i,j,t}$ which have coupling constant $T_{i,j} \geq \xi = 0.01$ and satisfy the pruning constraints. With $n_1 = 3630$ spots in the atlas for Slice 1 and $m_2 = 3309$ spots in the atlas for Slice 2, we find there are $n_{1,2} = 8191$ spots in each interpolated atlas. With $n_4 = 3696$ spots in the atlas for Slice 4 and $m_5 = 3745$ spots in the atlas for Slice 5, we find there are $n_{4,5} = 6619$ spots in each interpolated atlas. For visualization, we have represented the known expression levels of gene Msx1 in each of the fixed slices, as well as the interpolated expression levels in each of the interpolated slices. **(C)** Hnf1b expression in slices 1-4 of the resulting interpolated 3-D atlas. Compared with Figure 4.4B, much more detail is evident.

Bibliography

[1] 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.

[2] Nikos Karaiskos, Philipp Wahle, Jonathan Alles, Anastasiya Boltengagen, Salah Ayoub, Claudia Kipar, Christine Kocks, Nikolaus Rajewsky, and Robert P. Zinzen. The drosophila; embryo at single-cell transcriptome resolution. *Science*, 358(6360):194, 2017.

[3] Bosiljka Tasic, Vilas Menon, Thuc Nghi Nguyen, Tae Kyung Kim, Tim Jarsky, Zizhen Yao, Boaz Levi, Lucas T. Gray, Staci A. Sorensen, Tim Dolbeare, Darren Bertagnolli, Jeff Goldy, Nadiya Shapovalova, Sheana Parry, Changkyu Lee, Kimberly Smith, Amy Bernard, Linda Madisen, Susan M. Sunkin, Michael Hawrylycz, Christof Koch, and Hongkui Zeng. Adult mouse cortical cell taxonomy revealed by single cell transcriptomics. *Nature Neuroscience*, 19(2):335–346, 2016.

[4] Ruslan Soldatov, Marketa Kaucka, Maria Eleni Kastriti, Julian Petersen, Tatiana Chontorotzea, Lukas Englmaier, Natalia Akkuratova, Yunshi Yang, Martin Häring, Viacheslav Dyachuk, Christoph Bock, Matthias Farlik, Michael L. Piacentino, Franck Boismoreau, Markus M. Hilscher, Chika Yokota, Xiaoyan Qian, Mats Nilsson, Marianne E. Bronner, Laura Croci, Wen-Yu Hsiao, David A. Guertin, Jean-Francois Brunet, Gian Giacomo Consalez, Patrik Ernfors, Kaj Fried, Peter V. Kharchenko, and Igor Adameyko. Spatiotemporal structure of cell fate decisions in murine neural crest. *Science*, 364(6444), 2019.

[5] Paul A McGettigan. Transcriptomics in the rna-seq era. *Current Opinion in Chemical Biology*, 17(1):4–11, 2013. Omics.

[6] Fuchou Tang, Catalin Barbacioru, Yangzhou Wang, Ellen Nordman, Clarence Lee, Nanlan Xu, Xiaohui Wang, John Bodeau, Brian B Tuch, Asim Siddiqui, Kaiqin Lao, and M Azim Surani. mrna-seq whole-transcriptome analysis of a single cell. *Nature Methods*, 6(5):377–382, 2009.

[7] Alexander B. Rosenberg, Charles M. Roco, Richard A. Muscat, Anna Kuchina, Paul Sample, Zizhen Yao, Lucas T. Graybuck, David J. Peeler, Sumit Mukherjee, Wei Chen, Suzie H. Pun, Drew L. Sellers, Bosiljka Tasic, and Georg Seelig. Single-cell profiling of the developing mouse brain and spinal cord with split-pool barcoding. *Science*, 360(6385):176, 2018.

[8] Valentine Svensson, Roser Vento-Tormo, and Sarah A. Teichmann. Exponential scaling of single-cell rna-seq in the past decade. *Nature Protocols*, 13(4):599–604, 2018.

[9] Guo-Cheng Yuan, Long Cai, Michael Elowitz, Tariq Enver, Guoping Fan, Guoji Guo, Rafael Irizarry, Peter Kharchenko, Junhyong Kim, Stuart Orkin, John Quackenbush, Assieh Saadatpour, Timm Schroeder, Ramesh Shivdasani, and Itay Tirosh. Challenges and emerging directions in single-cell analysis. *Genome Biology*, 18(1):84, 2017.

[10] Malte D. Luecken and Fabian J. Theis. Current best practices in single-cell rna-seq analysis: a tutorial. *Molecular Systems Biology*, 15(6):e8746, 2019.

[11] Vladimir Yu Kiselev, Tallulah S. Andrews, and Martin Hemberg. Challenges in unsupervised clustering of single-cell rna-seq data. *Nature Reviews Genetics*, 20(5):273–282, 2019.

[12] Simon Joost, Amit Zeisel, Tina Jacob, Xiaoyan Sun, Gioele La Manno, Peter Lönnerberg, Sten Linnarsson, and Maria Kasper. Single-cell transcriptomics reveals that differentiation and spatial signatures shape epidermal and hair follicle heterogeneity. *Cell Systems*, 3(3):221–237.e9, 2016.

[13] Sidharth V. Puram, Itay Tirosh, Anuraag S. Parikh, Anoop P. Patel, Keren Yizhak, Shawn Gillespie, Christopher Rodman, Christina L. Luo, Edmund A. Mroz, Kevin S. Emerick, Daniel G. Deschler, Mark A. Varvares, Ravi Mylvaganam, Orit Rozenblatt-Rosen, James W. Rocco, William C. Faquin, Derrick T. Lin, Aviv Regev, and Bradley E. Bernstein. Single-cell transcriptomic analysis of primary and metastatic tumor ecosystems in head and neck cancer. *Cell*, 171(7):1611–1624.e24, 2017.

[14] Shristi Pandey, Karthik Shekhar, Aviv Regev, and Alexander F. Schier. Comprehensive identification and spatial mapping of habenular neuronal types using single-cell rna-seq. *Current Biology*, 28(7):1052–1065.e7, 2018.

[15] Shuxiong Wang, Matthew Karikomi, Adam L. MacLean, and Qing Nie. Cell lineage and communication network inference via optimization for single-cell transcriptomics. *Nucleic Acids Research*, 47(11):e66–e66, 2019.

[16] Judy Sprague, Leyla Bayraktaroglu, Dave Clements, Tom Conlin, David Fashena, Ken Frazer, Melissa Haendel, Douglas G. Howe, Prita Mani, Sridhar Ramachandran, Kevin Schaper, Erik Segerdell, Peiran Song, Brock Sprunger, Sierra Taylor, Ceri E. Van Slyke, and Monte Westerfield. The zebrafish information network: the zebrafish model organism database. *Nucleic acids research*, 34(Database issue):D581–D585, 2006.

[17] Charless C. Fowlkes, Cris L. Luengo Hendriks, Soile V. E. Keränen, Gunther H. Weber, Oliver Rübel, Min-Yu Huang, Sohail Chatoor, Angela H. DePace, Lisa Simirenko, Clara Henriquez, Amy Beaton, Richard Weiszmann, Susan Celniker, Bernd Hamann, David W. Knowles, Mark D. Biggin, Michael B. Eisen, and Jitendra Malik. A quantitative spatiotemporal atlas of gene expression in the ¡em¿drosophila¡/em¿ blastoderm. *Cell*, 133(2):364–374, 2008.

[18] Kaia Achim, Jean-Baptiste Pettit, Luis R. Saraiva, Daria Gavriouchkina, Tomas Larsson, Detlev Arendt, and John C. Marioni. High-throughput spatial mapping of single-cell rna-seq data to tissue of origin. *Nature Biotechnology*, 33(5):503–509, 2015.

[19] Simone Codeluppi, Lars E. Borm, Amit Zeisel, Gioele La Manno, Josina A. van Lunteren, Camilla I. Svensson, and Sten Linnarsson. Spatial organization of the somatosensory cortex revealed by osmfish. *Nature Methods*, 15(11):932–935, 2018.

[20] Jeffrey R. Moffitt, Dhananjay Bambah-Mukku, Stephen W. Eichhorn, Eric Vaughn, Karthik Shekhar, Julio D. Perez, Nimrod D. Rubinstein, Junjie Hao, Aviv Regev, Catherine Dulac, and Xiaowei Zhuang. Molecular, spatial, and functional single-cell profiling of the hypothalamic preoptic region. *Science*, 362(6416):eaau5324, 2018.

[21] Sheel Shah, Eric Lubeck, Wen Zhou, and Long Cai. In situ transcription profiling of single cells reveals spatial organization of cells in the mouse hippocampus. *Neuron*, 92(2):342–357, 2016.

[22] Chee-Huat Linus Eng, Michael Lawson, Qian Zhu, Ruben Dries, Noushin Koulena, Yodai Takei, Jina Yun, Christopher Cronin, Christoph Karp, Guo-Cheng Yuan, and Long Cai. Transcriptome-scale super-resolved imaging in tissues by rna seqfish+. *Nature*, 568(7751):235–239, 2019.

[23] Xiao Wang, William E. Allen, Matthew A. Wright, Emily L. Sylwestrak, Nikolay Samusik, Sam Vesuna, Kathryn Evans, Cindy Liu, Charu Ramakrishnan, Jia Liu, Garry P. Nolan, Felice-Alessio Bava, and Karl Deisseroth. Three-dimensional intact-tissue sequencing of single-cell transcriptional states. *Science*, 361(6400):eaat5691, 2018.

[24] Samuel G. Rodriques, Robert R. Stickels, Aleksandrina Goeva, Carly A. Martin, Evan Murray, Charles R. Vanderburg, Joshua Welch, Linlin M. Chen, Fei Chen, and Evan Z. Macosko. Slide-seq: A scalable technology for measuring genome-wide expression at high spatial resolution. *Science*, 363(6434):1463, 2019.

[25] Guangdun Peng, Shengbao Suo, Jun Chen, Weiyang Chen, Chang Liu, Fang Yu, Ran Wang, Shirui Chen, Na Sun, Guizhong Cui, Lu Song, Patrick P L. Tam, Jing-Dong J Han, and Naihe Jing. Spatial transcriptome for the molecular annotation of lineage fates and cell identity in mid-gastrula mouse embryo. *Developmental Cell*, 36(6):681–697, 2016.

[26] Keren Bahar Halpern, Rom Shenhav, Orit Matcovitch-Natan, Beáta Tóth, Doron Lemze, Matan Golan, Efi E. Massasa, Shaked Baydatch, Shanie Landen, Andreas E. Moor, Alexander Brandis, Amir Giladi, Avigail Stokar-Avihail, Eyal David, Ido Amit, and Shalev Itzkovitz. Single-cell spatial reconstruction reveals global division of labour in the mammalian liver. *Nature*, 542(7641):352–356, 2017.

[27] 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.

[28] Tim Stuart, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M. Mauck, Yuhan Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. Comprehensive integration of single-cell data. *Cell*, 177(7):1888–1902.e21, 2019.

[29] 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.e17, 2019.

[30] 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.

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

[32] 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.

[33] Yuval Lieberman, Lior Rokach, and Tal Shay. Castle – classification of single cells by transfer learning: Harnessing the power of publicly available single cell rna sequencing experiments to annotate new experiments. *PLOS ONE*, 13(10):e0205499, 2018.

[34] 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.

[35] Florian Wagner and Itai Yanai. Moana: A robust and scalable cell type classification framework for single-cell rna-seq data. *bioRxiv*, page 456129, 2018.

[36] Yuqi Tan and Patrick Cahan. Singlecellnet: A computational tool to classify single cell rna-seq data across platforms and across species. *Cell Systems*, 9(2):207–213.e2, 2019.

[37] Katerina Boufea, Sohan Seth, and Nizar N. Batada. scid uses discriminant analysis to identify transcriptionally equivalent cell types across single-cell rna-seq data with batch effect. *iScience*, 23(3):100914, 2020.

[38] Jian Hu, Xiangjie Li, Gang Hu, Yafei Lyu, Katalin Susztak, and Mingyao Li. Iterative transfer learning with neural network for clustering and cell type classification in single-cell rna-seq analysis. *bioRxiv*, page 2020.02.02.931139, 2020.

[39] Feiyang Ma and Matteo Pellegrini. Actinn: automated identification of cell types in single cell rna sequencing. *Bioinformatics*, 36(2):533–538, 2020.

[40] Kilian Q. Weinberger and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res.*, 10:207–244, 2009.

[41] Mahmut Kaya and H. Bilge. Deep metric learning: A survey. *Symmetry*, 11:1066, 2019.

[42] Davide Chicco. *Siamese Neural Networks: An Overview*, pages 73–94. 2020.

[43] Brian Kulis. Metric learning: A survey. *Foundations and Trends® in Machine Learning*, 5(4):287–364, 2013.

[44] Liqun Chen, Zhe Gan, Yu Cheng, Linjie Li, Lawrence Carin, and Jingjing Liu. Graph optimal transport for cross-domain alignment. *CoRR*, abs/2006.14744, 2020.

[45] Danilo Motta, Wallace Casaca, and Afonso Paiva. Vessel optimal transport for automated alignment of retinal fundus images. *IEEE Transactions on Image Processing*, 28(12):6154–6168, 2019.

[46] Pinar Demetci, Rebecca Santorella, Björn Sandstede, William Stafford Noble, and Ritambhara Singh. Gromov-wasserstein optimal transport to align single-cell multi-omics data. *bioRxiv*, 2020.

[47] Zixuan Cang and Qing Nie. Inferring spatial and signaling relationships between cells from single cell transcriptomic data. *Nature Communications*, 11(1):2084, 2020.

[48] Alexander Tong, Jessie Huang, Guy Wolf, David van Dijk, and Smita Krishnaswamy. Trajectorynet: A dynamic optimal transport network for modeling cellular dynamics, 2020.

[49] Floyd Maseda, Zixuan Cang, and Qing Nie. Deepsc: A deep learning-based map connecting single-cell transcriptomics and spatial imaging data. *Frontiers in Genetics*, 12:348, 2021.

[50] Guoji Guo, Mikael Huss, Guo Qing Tong, Chaoyang Wang, Li Li Sun, Neil D. Clarke, and Paul Robson. Resolution of cell fate decisions revealed by single-cell gene expression analysis from zygote to blastocyst. *Developmental Cell*, 18(4):675–685, 2010.

[51] Romain Lopez, Achille Nazaret, Maxime Langevin, Jules Samaran, Jeffrey Regier, Michael I. Jordan, and Nir Yosef. A joint model of unpaired data from scrna-seq and spatial transcriptomics for imputing missing gene expression measurements. *CoRR*, abs/1905.02269, 2019.

[52] Qian Zhu, Sheel Shah, Ruben Dries, Long Cai, and Guo-Cheng Yuan. Identification of spatially associated subpopulations by combining scrnaseq and sequential fluorescence in situ hybridization data. *Nature Biotechnology*, 36(12):1183–1190, 2018.

[53] Ruben Dries, Qian Zhu, Rui Dong, Chee-Huat Linus Eng, Huipeng Li, Kan Liu, Yuntian Fu, Tianxiao Zhao, Arpan Sarkar, Feng Bao, Rani E George, Nico Pierson, Long Cai, and Guo-Cheng Yuan. Giotto, a toolbox for integrative analysis and visualization of spatial expression data. *bioRxiv*, 2020.

[54] Alma Andersson, Joseph Bergenstråhle, Michaela Asp, Ludvig Bergenstråhle, Aleksandra Jurek, José Fernández Navarro, and Joakim Lundeberg. Single-cell and spatial transcriptomics enables probabilistic inference of cell type topography. *Communications Biology*, 3(1):565, 2020.

[55] Mor Nitzan, Nikos Karaiskos, Nir Friedman, and Nikolaus Rajewsky. Gene expression cartography. *Nature*, 576(7785):132–137, 2019.

[56] Patrik L. Ståhl, Fredrik Salmén, Sanja Vickovic, Anna Lundmark, José Fernández Navarro, Jens Magnusson, Stefania Giacomello, Michaela Asp, Jakub O. Westholm, Mikael Huss, Annelie Mollbrink, Sten Linnarsson, Simone Codeluppi, Åke Borg, Fredrik Pontén, Paul Igor Costea, Pelin Sahlén, Jan Mulder, Olaf Bergmann, Joakim Lundeberg, and Jonas Frisén. Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. *Science*, 353(6294):78, 2016.

[57] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, February 01, 2018 2018.

[58] Connor Meehan, Jonathan Ebrahimian, Wayne Moore, and Stephen Meehan. Uniform manifold approximation and projection (umap). 2021.

[59] The Mathworks Inc. Matlab deep learning toolbox release 2019b, 2019.

[60] PMLR. *Understanding the difficulty of training deep feedforward neural networks*, volume 9. Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, March 31 2010.

[61] The Mathworks Inc. Matlab parallel toolbox release 2019b, 2019.

[62] Claire McQuin, Allen Goodman, Vasiliy Chernyshev, Lee Kamentsky, Beth A. Cimini, Kyle W. Karhohs, Minh Doan, Liya Ding, Susanne M. Rafelski, Derek Thirstrup, Winfried Wiegraebe, Shantanu Singh, Tim Becker, Juan C. Caicedo, and Anne E. Carpenter. Cellprofiler 3.0: Next-generation image processing for biology. *PLOS Biology*, 16(7):1–17, 07 2018.

[63] Uwe Schmidt, Martin Weigert, Coleman Broaddus, and Gene Myers. Cell detection with star-convex polygons. *CoRR*, abs/1806.03535, 2018.

[64] Martin Weigert, Uwe Schmidt, Robert Haase, Ko Sugawara, and Gene Myers. Star-convex polyhedra for 3d object detection and segmentation in microscopy. In *The IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 3655–3662, March 2020.

[65] Gabriel Peyré and Marco Cuturi. Computational optimal transport, 2020.

[66] Facundo Memoli. On the use of Gromov-Hausdorff Distances for Shape Comparison. In M. Botsch, R. Pajarola, B. Chen, and M. Zwicker, editors, *Eurographics Symposium on Point-Based Graphics*. The Eurographics Association, 2007.

[67] Gabriel Peyré, Marco Cuturi, and Justin Solomon. Gromov-Wasserstein Averaging of Kernel and Distance Matrices. In *ICML 2016*, Proc. 33rd International Conference on Machine Learning, New-York, United States, June 2016.

[68] Sira Ferradans, Nicolas Papadakis, Gabriel Peyré, and Jean-François Aujol. Regularized discrete optimal transport. *CoRR*, abs/1307.5551, 2013.

[69] Jean-David Benamou, Guillaume Carlier, Marco Cuturi, Luca Nenna, and Gabriel Peyré. Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015.

[70] Paul Knopp and Richard Sinkhorn. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343 – 348, 1967.

[71] Justin Solomon, Gabriel Peyré, Vladimir G. Kim, and Suvrit Sra. Entropic metric alignment for correspondence problems. *ACM Trans. Graph.*, 35(4), July 2016.

[72] The Mathworks Inc. Matlab imageprocessing toolbox release 2019b, 2019.

[73] Titouan Vayer, Laetita Chapel, Rémi Flamary, Romain Tavenard, and Nicolas Courty. Fused gromov-wasserstein distance for structured objects: theoretical foundations and mathematical properties, 2018.

[74] Titouan Vayer, Laetitia Chapel, Rémi Flamary, Romain Tavenard, and Nicolas Courty. Optimal transport for structured data with application on graphs, 2019.

[75] Laetitia Chapel, Mokhtar Z. Alaya, and Gilles Gasso. Partial optimal transport with applications on positive-unlabeled learning, 2020.

[76] Wilfrid Gangbo and Robert J. McCann. The geometry of optimal transportation. *Acta Mathematica*, 177(2):113 – 161, 1996.