Probabilistic and Optimization Methods for Biomedical and Single-Cell Studies

Yining Liu

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
under the Executive Committee
of the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2025

© 2025

Yining Liu

All Rights Reserved

Abstract

Probabilistic and Optimization Methods for Biomedical and Single-Cell Studies

Yining Liu

In this thesis, we explore probabilistic and optimization methods to address critical challenges in biomedical and single-cell studies. These methods encompass diverse applications, from group testing strategies to advanced computational approaches for single-cell data analysis, highlighting their utility across scales and disciplines.

First, we develop a novel probabilistic framework for one-stage noisy group testing, a technique that efficiently identifies infected individuals within large populations. Motivated by practical needs during the COVID-19 pandemic, we propose a pooling design guided by maximizing pool entropy and a maximum-likelihood recovery algorithm. Our findings reveal the interplay between pooling parameters and randomness in infection vectors, offering a robust and adaptable group testing strategy.

Second, we introduce CellStitch, a 3D cell segmentation method leveraging optimal transport to overcome the challenges of segmenting anisotropic microscopy images. Unlike most existing segmentation approaches, CellStitch circumvents the need for large 3D training datasets by aligning cellular correspondences across imaging layers. Benchmarking on diverse plant microscopy datasets demonstrates that CellStitch outperforms state-of-the-art segmentation methods, particularly on images with high anisotropy, enabling accurate analysis of 3D cellular structures.

Finally, we present RefCM, an algorithm for automating cell-type annotation in single-cell RNA sequencing data based on computing Wasserstein distance between cell populations. RefCM

measures gene expression distribution similarities and solves an integer program to align query clusters with reference annotations. Our method achieves superior accuracy in cross-technology, cross-tissue, and cross-species mappings, addressing a critical need for robust annotation in single-cell studies and broadening the applicability of scRNA-seq technologies.

Together, these contributions demonstrate the power of probabilistic modeling and optimization methods in advancing the efficiency and accuracy of biomedical and single-cell analyses, providing tools to tackle challenges in diverse biological contexts.

Table of Contents

Acknowledgments vii

Dedication ix

Chapter 1: Introduction 1

1.1 Contributions and Structure of the Thesis 2

Chapter 2: Background and Related Works 4

2.1 Background 4

2.2 Related Works 6

2.2.1 Metric Learning 7

2.2.2 Spatial Position Reconstruction 7

2.2.3 Data Integration 8

Chapter 3: Probabilistic Modeling for One-Stage Group Testing 10

3.1 Introduction 10

3.2 Methods 11

3.2.1 Noisy Group Testing under the Probabilistic Framework 11

3.2.2 A Group Testing Protocol 13

3.2.3 Recovery Algorithm 15

3.3 Results 17
3.3.1 Performance of the Recovery Algorithm 18
3.3.2 Pooling Matrices 19
3.4 Discussion 23

Chapter 4: Optimal Transport for 3D Cell Segmentation 24
4.1 Introduction 24
4.2 Methods 25
4.2.1 Optimal Transport for Stitching Adjacent Layers 26
4.2.2 CellStitch Stitching Algorithm 28
4.3 Results 31
4.3.1 Datasets 31
4.3.2 Evaluation Metrics and Benchmark Methods 32
4.3.3 Experimental Results 33
4.4 Discussion 36

Chapter 5: Optimal Transport for Cell Type Annotation in scRNA-Seq Data 38
5.1 Introduction 38
5.2 Methods 40
5.2.1 Algorithm Overview 40
5.2.2 Integer programming for relaxed graph matching 42
5.2.3 Datasets pre-processing 43
5.2.4 Performance benchmarking 45
5.3 Results 46

5.4 Discussion 50

Conclusion or Epilogue 52

References 56

List of Figures

2.1 Optimal transport motivation 5

3.1 Four classes of pooling designs 12

3.2 Noisy group testing under the probabilistic framework 13

3.3 Theoretically optimal pool size as a function of infection rate 14

3.4 Robustness against measurements errors 18

3.5 Entropy of each pool against the performance of constant-pool designs 19

3.6 Average recovery accuracy of designs with varying pool sizes and number of pools 20

3.7 Performance of different designs under simulated data 21

3.8 Distributions of the inner products of column vector 22

4.1 Overview of CellStitch framework 26

4.2 Poor correspondence of cell centroids 28

4.3 Stitching rejection voting mechanism 30

4.4 Qualitative pipeline benchmark results on consecutive Ovules slices 36

5.1 Overview of RefCM 41

5.2 Integer programming relaxation in RefCM algorithm 43

5.3 Benchmark performance as cell type annotation tools 48

5.4 Benchmark performance as reference cluster mapping methods 49

5.5 Augmenting RefCM with Leiden clustering for reference mapping 50

List of Tables

4.1 Summary of benchmarking datasets 32

4.2 Performance benchmarks on the ovules dataset 34

4.3 Performance on the ATAS dataset 34

4.4 Performance on Arabidopsis 3D Digital Tissue Atlas 35

5.1 Summary of the pancreas dataset. 44

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my advisors, Professors Andrew J. Blumberg and Itsik Pe'er. Their mentorship has shaped not only my research but also my growth as an individual. I am immensely grateful for their patience, insightful feedback and encouragement in times of doubt. I would also like to extend my thanks to my committee members, Professors Elham Azizi, Simon Tavaré, and David Knowles, for their generous investment in time and invaluable guidance.

I am extremely grateful for the talented individuals with whom I had the pleasure of discussing and collaborating on research throughout the past ve years; many of the ideas in this thesis contribute to them. To members of the Pe'er lab, Andrew's Geometric Data Analysis group, and the Herbert and Florence Irving Institute for Cancer Dynamics, thank you for all your insights and support.

I am incredibly fortunate to have become friends with many of my fellow graduate students. To Raiyan Khan, Melanie Subbiah, Yinuo Jin, Vivian Liu, Junyoung Kim, Philippe Chlenski, Sitara Persad, Justin Hong, and Saeyoung Rho, your companionship has made the challenging times more bearable and the good times even better.

My decision to pursue a Ph.D. is largely driven by a passion for university-level teaching; I would also like to thank my mentors and teachers, many of whom I also had the pleasure of becoming friends with, who constantly inspire me with their passion and knowledge. To Adam Cannon, Kevin Lin, Nakul Verma, Dan Rubenstein, and Tony Dear, thank you for your support and guidance in my career. To Dan Daniel Erdmann-Pham, Owen Jow, Satish Rao, Koji Shimizu,

David Nadler, Dan Garcia, Sinho Chewi, Jeffrey Regier, Romain Lopez, Peter Friis, Ms. Jiang and Mr. Liu, your early guidance and encouragement were crucial in setting me on this path. I deeply appreciate the impact you all have had on my journey.

I would not have enjoyed my graduate studies nearly as much without the support of my friends outside of graduate school. To Kun Wang, Grace He, Xueqi Jiang, Litao Deng, Qian Sun, and Qianzi Cong, thank you for always being there for me despite the physical distance; your unwavering belief in me has been a constant reminder of the importance of lifelong friendships. To Sasha Chen, Muye Wang, Wenxuan Guo, Yiming Shi, Deborah Nixon, Peter Fancovic, Meghan Crocker, and the skaters on Gotham City Synchro, I am grateful for the much-needed breaks from academia that helped me stay grounded.

Finally, I could not have made it through the ups and downs of this journey without the constant support and encouragement of my family. To my parents and grandparents, your love and support have been my biggest source of strength; thank you for believing in me. To Steven Yin, thank you for being my rock throughout this process; your love, patience, and understanding have made all the difference.

Thank you all for being part of this journey with me. This thesis would not have been possible without any of you.

Dedication

To my Lao Lao, Rui Hao.

Chapter 1: Introduction

Modern biological research increasingly relies on computational tools to address challenges at the intersection of data analysis, resource ef ciency, and biological insight. In this thesis, we explore the use of probabilistic and optimization methods to solve diverse problems in biomedical and single-cell studies, spanning group testing strategies, 3D cellular segmentation, and transferring cell type annotations between single-cell RNA sequencing (scRNA-seq) datasets. Despite the distinct nature of these problems, they share a common need for innovative computational frameworks to process complex data with accuracy and ef ciency.

A unifying theme across the algorithms proposed in this thesis is the use of mathematical frameworks of probabilistic modeling and optimization principles to design effective algorithms for analyzing biological data. Probabilistic methods are particularly powerful for handling uncertainty and randomness, which are ubiquitous in biological systems. Optimization, on the other hand, provides a structured way to identify solutions to complex computational problems, such as matching patterns across datasets or maximizing the utility of experimental designs.

The rst project in this thesis focuses on the design of one-stage noisy group testing protocols, which became crucial during the COVID-19 pandemic. Group testing is an ef cient method that reduces the number of tests required by pooling samples, but its practical implementation is often hindered by uncertainty in infection rates and noise in test results. Using a probabilistic framework, we developed a recovery algorithm and pooling design strategy that optimizes test performance under real-world conditions. This work highlights how probabilistic modeling and entropy maximization can inform resource-ef cient strategies in biomedical applications.

The second and third projects shift focus to single-cell biology, an area that has seen tremendous growth with the advent of technologies like single-cell RNA sequencing and 3D microscopy. These datasets, while rich in information, pose unique computational challenges due to their high

dimensionality and inherent noise. Both projects leverage the mathematical framework of optimal transport (OT), which provides a natural way to compare and align distributions of biological data.

In the second project, we address the problem of 3D cellular segmentation in anisotropic microscopy images, where resolution differences across imaging dimensions complicate accurate cell morphology reconstruction. We developed CellStitch, an OT-based pipeline that aligns cellular correspondences across imaging layers and interpolates missing slices to recover isotropic cell structures. By using OT to infer optimal matches between image layers, CellStitch enables robust and high- delity segmentation without requiring extensive 3D training datasets, bridging a critical gap in biological image analysis.

The third project focuses on the annotation of single-cell RNA sequencing datasets, where the goal is to assign meaningful cell type labels to transcriptional pro les. We developed RefCM, a computational method that uses OT to align query clusters with reference datasets. By formulating the problem as an optimization task, we enhance the accuracy of cell type annotation across technologies, tissues, and species, demonstrating the broad applicability of OT in transferring cell type annotations across single-cell studies.

Through these projects, this thesis showcases the versatility of probabilistic and optimization methods in solving diverse challenges in biomedical research. While the group testing project demonstrates the utility of probabilistic modeling in designing cost-effective protocols for disease control, the single-cell studies leverage OT to address fundamental questions about cellular identity and structure. Together, these contributions highlight the power of mathematical frameworks in advancing biological discovery and resource-ef cient experimental designs.

1.1 Contributions and Structure of the Thesis

This thesis contributes to the elds of computational biology, optimization, and probabilistic modeling through the development of novel methods and their application to real-world biomedical problems. Speci cally, the contributions can be summarized as follows:

1. A probabilistic modeling framework for one-stage group testing, advancing diagnostic ef -

ciency in resource-limited settings.

2. An optimization-based approach for 3D cell segmentation, leveraging optimal transport to improve accuracy and generalizability.

3. A robust framework for cell type annotation in scRNA-seq data, addressing batch effects and enabling the discovery of novel biological insights.

The remainder of this thesis is organized as follows:

- Chapter 2: Background and Related Work. This chapter introduces the theoretical foundations of optimal transport and reviews its applications in single-cell studies. By focusing on these topics, the chapter provides the necessary context for the subsequent chapters.

- Chapter 3: Probabilistic Modeling for One-Stage Group Testing. Detailed methodology, experiments, and results for the group testing framework are presented.

- Chapter 4: Optimal Transport for 3D Cell Segmentation. This chapter describes the segmentation framework, benchmarking experiments, and comparative analyses.

- Chapter 5: Optimal Transport for Cell Type Annotation in scRNA-Seq Data. The methodology and applications of the annotation framework are discussed in this chapter.

- Chapter 6: Conclusion and Future Directions. This chapter summarizes the contributions of the thesis and outlines potential avenues for future research.

Chapter 2: Background and Related Works

In this chapter, we begin with an overview of the mathematical foundations of OT, including the Kantorovich formulation, Wasserstein distances, and entropy-regularization. We then review how OT has been applied speci cally in single-cell biology, highlighting its role in trajectory inference, dataset integration, and other key challenges. By establishing this background, we aim to provide a clear context for the methodological developments and applications presented in subsequent chapters.

Optimal transport (OT) is a central focus of this chapter because it lies at the intersection of probabilistic modeling and optimization, two core principles that underpin many biological problems. In biology, systems often involve uncertainty and variability, which require probabilistic frameworks, while optimization provides tools to address structured, high-dimensional data. OT embodies both, offering a principled way to compare and map distributions, making it particularly well-suited for the challenges posed by single-cell biology. In addition, OT plays an essential role in most of this thesis, serving as a key framework for multiple projects.

2.1 Background

Technological advances in the past decades have enabled researchers to study the biology of a system at single-cell resolution. With methods such as single cell RNA sequencing (scRNA-seq) [1] and spatial transcriptomics (ST) [2], biologists are able to measure gene expression levels as well as the spatial location of single cells; the rich single cell information continues to help researchers gain more understanding on cell states, cell types, and tissue architecture [3].

Optimal transport is an active area of research in mathematics that is motivated by nding the optimal allocation of resources [4]; it provides a exible framework that has enabled researchers to

Figure 2.1: Optimal transport is motivated by nding the optimal allocation of resources.

develop novel tools in various elds, including computer vision [5], machine learning [6], and signal processing [7]. Optimal transport has begun to help researchers solve various tasks in biology, but its full potential in single-cell biology is far from being fully realized.

Originally, the eld of optimal transport is motivated by nding the optimal allocation of resources. A classical example is that there are factories producing products and stores demanding products. Since the sizes of the factories (and correspondingly the sizes of the stores) are different, the supply and demand are nonuniform across factories and stores, respectively. Considering the location of the factories and the stores, the cost of transporting products between each factory and store is different. In this example, the goal of optimal transport is to nd an allocation plan to transport products from factories to stores so that the transportation cost is minimized.

Formally, for discrete measures, the Kantorovich formulation of the optimal transport problem is as follows: given two discrete measures $\mu : \mathcal{X} \to \mathbb{R}$ and $\mu' : \mathcal{X} \to \mathbb{R}$, a transport plan $\pi \in \mathbb{R}^{\mathcal{X} \times \mathcal{X}}$ is a product distribution on $\mathcal{X} \times \mathcal{X}$ that has μ and μ' as marginals; in other words, π_{G-H} describes the amount of mass in bin G that ows to bin H. Additionally, given a cost matrix $C \in \mathbb{R}^{\mathcal{X} \times \mathcal{X}}$ such that C_{G-H} describes the price associated with moving a unit of mass from bin G to bin H, the Kantorovich optimal transport solves for

$$\pi^* := \arg\min_{\pi \in \Pi(\mu, \mu')} \langle C, \pi \rangle = \sum_{G-H} C_{G-H} \pi_{G-H} \tag{2.1}$$

where

$$\Pi(\mu, \mu') = \{ \pi \in \mathbb{R}^{\cdot \times \cdot} : \pi \mathbb{1}_{\cdot} = \mu, \pi^T \mathbb{1}_{\cdot} = \mu' \} \tag{2.2}$$

The choice of the cost matrix determines the properties of the optimal transport problem such as the uniqueness of solutions and the computational cost of solving the optimization problem. A popular choice to compute the cost matrix is the distance, where $c(G, H) := \|G - H\|^2$. In this case, optimal transport induces a distance on the space of distributions, referred to as the Wasserstein distance. In particular, the 2-Wasserstein distance is obtained by using the Euclidean distance as the cost matrix; it has gained popularity in the computer vision community for performing interpolation, color transfer, and geometry processing [8, 9].

In order to solve the optimization problem 2.1, one can reduce it to a max- ow problem on a bipartite graph, where the vertices are cell slices in the two layers, and there is an edge between two vertices if and only if they are in different layers. The vertex and edge weights are given correspondingly by the proportions of label masses and the cost matrix. As a result, an optimal solution can be found using the max- ow algorithms. The network-simplex algorithm can solve Equation 2.1 in $O(n^3 \log(n))$. Entropy regularization [10] can be used to reduce the runtime. The entropy-regularized optimal transport problem is as follows:

$$\hat{\pi}_n := \arg\min_{\pi \in \Pi(\mu, \mu')} \langle -\pi, C_n \rangle - \varepsilon H(\pi) \tag{2.3}$$

where $H(\pi) := \sum_{i,j} \pi_{ij} \log \pi_{ij}$ is the entropy of the optimal transport plan.

The solution to Equation 2.3 is unique, and has the form of $\pi_n = D_1 K D_2$ for some diagonal matrices D_1, D_2. The Sinkhorn algorithm [10] finds $\pi_n = D_1 K D_2$ by alternatively rescaling rows and columns to match μ and μ'; it reduces the runtime from $O(n^3 \log(n))$ to $O(n^2)$.

2.2 Related Works

Optimal transport has been found helpful for single-cell tasks, including metric learning, spatial position reconstruction, and data integration. We now introduce each task and present a summary

of the existing OT-based method for each task.

2.2.1 Metric Learning

A common goal in single-cell data analysis is identifying the cell types and states in a sample. Cell-type and cell-state identi cation is typically achieved data-driven through unsupervised clustering. After pre-processing steps such as normalization and dimensionality reduction, a cell-cell distance matrix is computed, after which unsupervised clustering algorithms will be applied to the cell-cell distances to identify different cell types. Learning a cell-cell distance matrix that captures the global relationship of sampled cells is the central goal of metric learning.

scOmics-OT [11] de nes a similarity of pairs of cells using optimal transport distance. The authors used scRNA-seq, scATAC-seq, and scDNA-methylation datasets. They viewed normalized gene expression data as distributions and computed optimal transport distance as the cell-cell distance matrix. The gene costs were computed using Euclidean, Manhattan, cosine, and Pearson correlation. Empirically, the authors found that the best gene cost is data-type dependent, where Pearson correlation is the best for scRNA-seq simulated and Methylation data, and cosine is the best for scRNA-seq and scATAC-seq data. In their follow-up work [12], the authors developed an algorithm to learn an optimal cost matrix in an unsupervised setting.

2.2.2 Spatial Position Reconstruction

In a single-cell RNA protocol [13], cells are encapsulated in droplets where cell barcodes are attached for reverse transcription reaction, droplets are broken up, and reads are ampli ed. At the end, each read is assigned to a cell according to the barcode identity. During a single-cell RNA seq experiment, cells are removed from their innate tissue positions before being sequenced. As a result, spatial location is lost during scRNA-seq experiments. One goal of spatial position reconstruction is to map gene expression vectors of the cell population to some prede ned tissue coordinates.

Authors of novoSpaRc [14] model spatial position reconstruction under the optimal transport

framework to nd a matching between expression pro les and pre-de ned spatial location. Since expression pro les and spatial location are of different spaces, Gromov-Wasserstein optimal transport [9], a variant of the OT framework used to match distributions sampled from spaces without an underlying common metric, was used. Gromov-Wasserstein optimal transport nds the optimal matching between two distributions 0 2 < − 1 2 < 0 de ned on different spaces by solving the following optimization problem:

$$, \quad - \; {}^{0}{}^{1}0 - 1^{2} := \min_{\%2^{*}\,{}^{1}0-1} \; \tilde{O} \; j \; {}^{8-0}{}^{8} \quad {}^{0}{}_{9-0} j^{2} \% \; {}^{8-9}{}^{8-9} \% \qquad (2.4)$$

where 2 R< < and 0 2 R< 0 < 0 are distance matrices within each space. novoSpaRc [14] operates under the assumption that cells that are physically close share similar expression pro les; it solves the entropy-regularized Gromov-Wasserstein optimal transport problem with quadratic cost to nd a mapping from cells to tissue space such that the structural correspondence is preserved.

Alternatively, spatial location can be directly measured using single-cell RNA imaging technology [15], which enables measuring gene expression vector in the tissue coordinates by attaching uorescent probes; however, such technology is expensive and hence not widely used on the same scale as scRNA-seq in terms of the number of genes sequenced. Another goal of spatial position reconstruction is to infer the spatial location of scRNA-seq data using a spatial dataset with overlapping genes. In SpaOTsc [16], the authors construct a cost between cells and spatial locations using the common genes of the spatial and the scRNA-seq datasets and then use structured optimal transport [17] to recover spatial positions of scRNA-seq data using spatial measurements of a small number of genes.

2.2.3 Data Integration

Cells are complex environments where scRNA-seq only pro les partial information. Integrating different single-cell measurements can discover relationships across different cellular modalities and learn a more holistic representation of the cell state. The goal of data integration is to nd

a representation for cells that combines multiple single-cell measurement modalities [18].

Optimal transport supports a natural way to integrate different single-cell measurements sequenced on different cell populations. In particular, SCOT [19] performs OT-based data integration by rst computing the graph distance from the NN graph for each measurement domain to get a distance matrix within each dataset. Assuming uniform distribution over cells on each dataset, they used Gromov-Wasserstein optimal transport to compute an optimal matching. They used optimal matching to project one dataset onto another through barycentric projection. In the end, the integrated dataset is obtained by projecting one measurement domain onto the other.

Pamona [20] is another OT-based method to perform data integration but allows the two datasets to only partially overlap for the same underlying cellular structure. The authors relax the SCOT framework by only matching cells with the same underlying cellular structure. The partial optimal transport [21] framework was used to achieve the relaxation. In particular, partial optimal transport allows only a fraction of the total mass needs to be transported:

$$, \; {}^{B_1}0-1 := \min_{\%2 * {}^{B_1}0-1} h-\%i \tag{2.5}$$

where

$$* \; {}^{1}0-1 := f\%2 \; R^{<}_{,} \; {}^{<0} j \; \%1_{<} \quad 0-\%1_{<0} \quad 1-1^{)}_{<} \; \%1_{<0} = Bg \tag{2.6}$$

and $B 2 \; {}^{1}0-1^{o}$ is an additional parameter of the partial mass. Equation 2.5 could be reduced to the classical optimal transport by adding virtual points and hence can be ef ciently solved by the Sinkhorn algorithm if an entropy regularization is added. After solving the partial optimal transport problem, Pamona uses Laplacian eigenmaps [22] to embed both the shared and the dataset-speci c cells in a common low-dimensional space.

Chapter 3: Probabilistic Modeling for One-Stage Group Testing

This chapter is joint work with Sachin Kadyan and Itsik Pe'er and was published as [23] in International Conference on Algorithms for Computational Biology (AlCoB)

3.1 Introduction

Group testing is a procedure to nd positives in a cohort by applying tests for the presence of any positives to cohort subsets (groups), instead of testing each individual separately. When the fraction of positives among the samples is low, implementing group testing saves time and resources. Group testing has applications in genetics [24], drug screening [25], communications [26] and epidemiology [27].

One-stage group testing [28] addresses the scenario of groups being set in advance, independently of test results - a common practical requirement, e.g. due to testing speed constraints. Existing algorithms on infection vector recovery for noisy group testing under the combinatorial prior (the number of positives among samples is xed) include LP relaxation [29], belief propagation [30], Markov Chain Monte Carlo (MCMC) [31], Noisy Combinatorial Orthogonal Matching Pursuit (NCOMP) [32], separate decoding [33] and De nite positives (DD) [34]. However, group testing under a fully probabilistic framework has not been extensively studied. In particular, to the best of our knowledge, there has been a lack of work on non-asymptotic results on noisy group testing in the realistic scenario where number of positives in not known in advance, as in [35, 36], but rather is a random variable.

As part of the global efforts to control the COVID-19 pandemic, there has been an emerging body of work on implementing one-stage group testing for COVID-19 [37, 38, 39, 27, 40]. However, testing scenarios vary signi cantly due to uctuating infection rates across time and geogra-

phy often caused by emerging variants, as well as different testing objectives, such as screening of health care workers versus large scale monitoring of the community. Group testing protocols should thus be adjusted according to the infection rate among the tested samples, measurement error rates, and the recovery error tolerance level [41].

As a result, systematically studying one-stage noisy group testing with a random number of positives is of both theoretical interest and practical importance. In this paper, we study noisy group testing under the probabilistic framework in order to address the practical challenges posted by group testing implementation for COVID-19.

The rest of the paper is organized as follows. We begin with an introduction of the noisy group testing under the probabilistic framework in Section 3.2, including a group testing protocol and a novel recovery algorithm under the probabilistic framework. The performance of the recovery algorithm and the pooling designs is in Section 3.3. Finally, we conclude with a discussion on future work in Section 3.4.

3.2 Methods

3.2.1 Noisy Group Testing under the Probabilistic Framework

We assume a tested cohort of n individuals, with some infection rate δ among the population the cohort is sampled form. Note that the actual number of true positives is not known in advance, as combinatorial priors unrealistically assume [35, 36]. Also, this is the ground truth rate of infected individuals, as opposed to the observed positivity rate. Speci cally:

De nition 1. An infection vector $\mathbf{r}$ is a random vector with n i.i.d. Bernoulli δ entries.

We aim to design a protocol to test n samples with $C \ll n$ tests. We arrange the pool assignments into a pooling matrix $\mathbf{A} \in \{0,1\}^{C \times n}$ such that $\mathbf{A}_{ij} = 1$ if and only if individual j is included in pool i. Notice that each row i sum of $\mathbf{A}$ corresponds to a pool size R_i, and each column j sum corresponds to the number of pools in which a sample participates, which we de ne as redundancy A_j.

Figure 3.1: Four classes of pooling designs.

Following [42] and [43], we focus on the following four classes of pooling designs that are of practical interest (see Fig. 3.1):

1. (constant-pool design) each test combines a fixed number $B = B$ of samples, where B samples are chosen uniformly at random among the n samples;

2. (constant-redundancy design) each individual participates in a fixed number $A = A$ of tests, where A tests are chosen uniformly at random among the C tests;

3. (doubly-regular design) draw a design uniformly at random from the designs that are both constant-pool and constant-redundancy;

4. (Reed-Solomon design) the pool assignment for each individual is obtained as a concatenation of Reed-Solomon error correcting code; see [43] for more details on the explicit construction.

Under the noiseless setting, a test result is negative if and only if all samples in the pool are negative. In practice, test results might suffer from measurement errors. Suppose the test we use for each pool has a false negative rate of ε_0 and a false positive rate of ε_1, i.e., consider the asymmetric noisy channel (where ε_0 and ε_1 can be distinct) that combines the additive model and the dilution model in [35]. See Fig. 3.2 for an example of a doubly-regular pooling matrix under the probabilistic framework.

Figure 3.2: Noisy group testing under the probabilistic framework. The pooling matrix is a doubly-regular design with $B = 3$ and $A = 2$. A pool is negative if and only if no positive sample participates in the pool. Each pool is then passed to an asymmetric noisy channel to model measurement errors.

3.2.2 A Group Testing Protocol

Let Y_p be the indicator that the test result for the pool p is negative; then

$$Y_p \sim \text{Bernoulli}\left(\left(1-p_0\right)\left(1-p\right)^{B_p} + p_1\left(1-\left(1-p\right)^{B_p}\right)\right).$$

Extending the idea in [39] to the noisy setting, a pool design should maximize the entropy of Y_p, hence the optimal pool size is

$$\arg\max_{B_p} H(Y_p; B_p) = \frac{\ln\left(\frac{0.5 - p_0}{1 - p_1 - p_0}\right)}{\ln\left(1-p\right)}. \tag{3.1}$$

Infection rates therefore dramatically affect the theoretical optimal pool size, while measurement errors slightly offset such optima (see Fig. 3.3).

We will show in Section 3.3 that given a fixed set of parameters A, B, C, the four classes of designs (and that of each member of a given class) perform comparably under this probabilistic

Figure 3.3: Theoretically optimal pool size as a function of infection rate; false positives (resp. negatives) slightly reduce (increase) optimal pool size.

framework. Hence, practitioners can safely choose a design class with practical advantages and then optimize the parameters A, B, C.

We thus propose the following protocol for implementing one-stage group testing under a given infection rate:

1. From the potential pool sizes (subject to practical constraints), choose B to be closest to the theoretical optimal pool size, per Equation 3.1.
2. Simulate the performance curve using the recovery algorithm in Section 3.2.3.
3. Choose the number of pools C based on the error tolerance level.
4. Perform one-stage group testing with the same recovery algorithm in step 2.

3.2.3 Recovery Algorithm

We now introduce an algorithm to recover the infection vector for noisy group testing under the probabilistic framework.

Assume we are given a pooling design with C pools; the pool results is

$$\ldots := \text{sign}\left(\ldots - \ldots \ldots\right)$$

where $\ldots$ are false-negatives vector and false-positives vector respectively, i.e. $\ldots$ for $1, \ldots, C$

$$\ldots \begin{cases} \text{Bernoulli}(\ldots) & \text{if } \ldots = 0 \\ \text{Bernoulli}(\ldots) & \text{if } \ldots = 0 \\ \text{Bernoulli}(\ldots) & \text{if } \ldots = 0 \\ \text{Bernoulli}(\ldots) & \text{if } \ldots = 0 \end{cases}$$

A recovery algorithm is given a pool result $\ldots \in \{0,1\}^C$ and outputs $\ldots$, an estimate for $\ldots$. Our recovery algorithm $\ldots$ (Algorithm 1) is given by solving the following integer

linear program:

$$
\begin{aligned}
&\text{minimize} \quad \|\mathbf{c}\|\ln\left(\frac{5}{15}\right) \quad \|\mathbf{a}^{(0)}\|\ln\left(\frac{5}{15}\right) \quad \|\mathbf{a}^{(1)}\|\ln\left(\frac{5}{15}\right) \\
&\text{subject to} \quad 0.5\,\mathbf{a}_i^{(0)} \le (1-\mathbf{c})_i \le \mathbf{a}_i^{(0)}, \quad 0.5 \qquad \{1,\ldots,n\} \text{ s.t. } \mathbf{b}_i = 0 \\
&\qquad\qquad 0.5\,\mathbf{a}_i^{(1)} \le \mathbf{a}_i^{(1)}, \quad 0.5 \le (1-\mathbf{c})_i \le \mathbf{a}_i^{(1)}, \quad 0.5 \qquad \{1,\ldots,n\} \text{ s.t. } \mathbf{b}_i = 1 \\
&\qquad\qquad \mathbf{c} \in \{0,1\}^n \\
&\qquad\qquad \mathbf{a}^{(0)}, \mathbf{a}^{(1)} \in \{0,1\}^n
\end{aligned}
$$

Here, for a given binary vector $\mathbf{c} \in \{0,1\}^n$, we use $\|\mathbf{c}\|$ to denote the Hamming weight of $\mathbf{c}$, i.e. $\|\mathbf{c}\| = \sum_{i=1}^{n} c_i$.

We now analyze the correctness and optimality of Algorithm 1.

Theorem 1. Algorithm 1 returns a realizable output.

Proof. In order for an output $\mathbf{c} = \mathbf{a}^{(0)} - \mathbf{a}^{(1)}$ to be realizable, each row of the equation $\mathbf{b} = \mathbf{c}$ with sign $(1-\mathbf{c})$, $\mathbf{a}^{(0)}$, $\mathbf{a}^{(1)}$ must fall into one of the following cases:

$$
\begin{cases}
b_i = 0 \text{ and } (1-c_i) = 0 \text{ and } a_i^{(0)} = 0; \\
b_i = 0 \text{ and } (1-c_i) = 1 \text{ and } a_i^{(0)} = 1; \\
b_i = 1 \text{ and } (1-c_i) = 1 \text{ and } a_i^{(1)} = 0; \\
b_i = 1 \text{ and } (1-c_i) = 0 \text{ and } a_i^{(1)} = 1.
\end{cases}
$$

Notice that for $b_i = 0$, we have

$$
\begin{aligned}
a_i^{(0)} = 1 &\iff 0.5 \le (1-c_i) \le 1 \iff 0.5 \le (1-c_i) \le 1 \\
a_i^{(0)} = 0 &\iff 0 \le (1-c_i) \le 0.5 \iff (1-c_i) = 0.
\end{aligned}
$$

Similarly, for $H_8 = 1$, we have

$$4_8^{11o} = 0 \quad () \qquad 0•5 \quad {}^{1} \text{"} \quad G^{o}_{8} \quad C, \; 0•5 \; () \qquad 1 \quad {}^{1} \text{"} \quad G^{o}_{8} \quad C-$$

$$4_8^{11o} = 1 \quad () \qquad 0 \quad {}^{1} \text{"} \quad G^{o}_{8} \quad 0•5 \; () \; {}^{1} \quad \text{"} \quad G^{o}_{8} = 0•$$

Therefore, the constraints in the integer linear program guarantees the output is realizable. □

Theorem 2. Algorithm 1 returns a maximum likelihood estimate (MLE), i.e. an output $(G, 4^{10o}, 4^{11o})$ maximizes $Pr^{1}- = G- {}^{10o} = 4^{10o}- {}^{11o} = 4^{11o}o•$

Proof. Let $=_0$ be the number of pools with no positive sample and $=_1$ be the number of pools containing at least one positive sample. The log-likelihood is

$$\ln Pr^{1}- = G- {}^{10o} = 4^{10o}- {}^{11o} = 4^{11o}o$$

$$= \ln Pr^{1}- = G^{o} \; , \; \ln Pr^{1} \quad {}^{10o} = 4^{10o}- {}^{11o} = 4^{11o}j- = G^{o}$$

$$= jjGjj \ln 5 \; , \; {}^{1} = \; jj \; Gjj^{o} \ln^{1}1 \quad 5^{o} \; , \; jj \; 4^{10o}jj \ln 5_0 \; , \; {}^{1} =_1 \; jj \; 4^{10o}jj^{o} \ln^{1}1 \quad 5_0^{o}$$

$$, \; jj \; 4^{11o}jj \ln 5_1 \; , \; {}^{1} =_0 \; jj \; 4^{11o}jj^{o} \ln^{1}1 \quad 5_1^{o}$$

$$= jjGjj \ln^{1} \frac{5}{1 \quad 5}^{o} \; , \; jj \; 4^{10o}jj \ln^{1} \frac{5_0}{1 \quad 5_0}^{o} \; , \; jj \; 4^{11o}jj \ln^{1} \frac{5_1}{1 \quad 5_1}^{o} \; , \; \text{const.}$$

Therefore, the objective function in the integer linear program maximizes the likelihood. □

Open source code implementing all our methods is available at [44].

3.3 Results

For biological experiments, sample size $= 2 \; f \; 96–384–1536 \; g$ are of particular interests because the experiments tend to be conducted on 96-well plates, 384-well plates, or 1536-well plates [43]. The savings in resources offered by pooled design compared to individual testing stops being worth the increased complexity of the experimental paradigm when infection rates are high. For example, under an infection rate $5 = 0•1$, using a constant-pool design with optimal pool size $B = 7$, even a

Figure 3.4: Robustness against measurements errors.

2-fold saving, with $) = =\bullet 2 = 192$ pools to recover $= = 384$ samples, the average accuracy over 1000 trials is unacceptably low $96\bullet 5\%$. As a result, we believe our group testing protocol should not be applied when the infection rate is over $0\bullet 1$, and limit our analysis to infection rate up to $0\bullet 025$ in this section.

3.3.1 Performance of the Recovery Algorithm

In order to test the performance of the recovery algorithm, we used the pooling design $f\,0\text{–}1g^{48\ \ 384}$ (testing 384 individuals in 48 pools) in [27].

We simulated 1000 infection vectors $G2\ f\,0\text{–}1g^{384}$ under the population infection rate $= 0\bullet 02$ and evaluated recovery under different rates of false positives and false negatives. We observe error rate to only slightly detract from the accuracy of noiseless measurements (see Fig. 3.4).

Figure 3.5: Entropy of each pool against the performance of constant-pool designs with pool size $B \in \{8, 16, \ldots, 80\}$ and $C = 48$ pools; infection vectors are simulated with 384 Bernoulli entries. The direction of the triangular markers indicates increasing S

3.3.2 Pooling Matrices

Optimal Pool Size

In order to test the theoretically optimal pool size given in Equation 3.1, we simulated 1000 infection vectors under various infection rates and evaluated the performance of constant-pool matrix with $C = 48$ and $B \in \{8, 16, 24, \ldots, 80\}$. The performance of the pooling matrices is strongly correlated with the entropy of each pool, which supports the guidance of maximizing pool entropy in one-stage group testing designs (see Fig. 3.5).

In order to test the robustness of the optimal pool size S with respect to varying the number of pools C, we tested the performance of constant-pool designs with $B \in \{8, 16, 24, \ldots, 80\}$ based on

Figure 3.6: Average recovery accuracy of designs with the pool size $B = 2, 8, 16, 24, \ldots, 80$ and varying the number of pools under the infection rate $\delta = 0.02$; the pool size $B = 32$ achieves the maximum pool entropy.

1000 simulated infection vectors under $\delta = 0.02$. The optimal pool size ($B = 32$) achieves nearly maximal recovery accuracy for all A (see Fig. 3.6).

Explicit Constructions

Combinatorial structure of pooling matrices has been studied under the combinatorial framework [45]. Recent development of explicit pooling matrices construction for noisy group testing is based on error correcting codes [42]. Pooling design based on Reed-Solomon code was introduced in [46] and shown to be optimal under a more restricted probabilistic setting (assuming a random set of positives of a xed size) [47].

Authors in [27] use a design based on Reed-Solomon error correcting code; the matrix assigns each individual in 6 pools such that each pool has 48 individuals. We compare the performance of the Reed-Solomon design, a randomly drawn doubly-regular design (with $A = 6$ and $B = 48$), a

Figure 3.7: Performance of the Reed-Solomon (red), the doubly-regular ($A = 6$, $B = 48$) (blue), the constant redundancy ($A = 6$) (green), and the constant pool ($B = 48$) (orange) designs under simulated data with $N = 384$ individuals, $C = 48$ pools. The box shows the four quartiles, the whiskers show the rest of the distribution, and the points show the outliers.

randomly drawn constant-redundancy design (with $A = 6$ and varying pool sizes), and a randomly drawn constant-pool design (with $B = 48$ and varying redundancies).

We found that under the probabilistic framework, the performance of the three random designs is comparable to that of the Reed-Solomon design in terms of both the average and the standard deviation of the recovery accuracy (see Fig. 3.7).

For each of the three random designs, we computed the distribution of the inner products of column vectors (which are derived from code words) of 1000 randomly-drawn matrices. The performance guarantees of Reed-Solomon design under the combinatorial setting follows from the minimum pairwise distance of the code words [42]. The authors in [48] studied explicit pooling matrices and showed the dependency of the error probability on both the minimum and the aver-

Figure 3.8: Distributions of the inner products of column vector. Error bars are computed based on 1000 matrices.

age distance under the assumption of a xed number of positives. In the same spirit, we found that maximizing the number of pairs of column vectors with zero inner products could lead to performance guarantees under the probabilistic framework (see Fig. 3.8).

As a rst step towards understanding the role of combinatorial structures of the pooling matrices under the probabilistic framework, we randomly drew 1000 constant-pool designs with $B = 48$ to test $= = 384$ individuals with $C = 48$ pools, and computed the average accuracy on two sets of 1000 simulated infection vectors under the infection rate $= 0.02$. We observe low correlation (0.0884, ? 0.00516) between the performance of designs on the two sets of vectors, which suggests that under the probabilistic framework, the combinatorial structure of the pooling matrices plays a less signi cant role than the parameters B, A, C.

3.4 Discussion

We presented a framework that addresses several practical issues, including randomness of the infection vector and noisy measurements. Our ndings suggest that in practice, the combinatorial structure of pooling design matrices plays a lesser role compared to parameters such as redundancy, pool sizes, and the number of pools. Furthermore, we provide a protocol and an implementation for practitioners to choose the parameters in order to design pooling matrices.

From a theoretical perspective, there are several promising future directions for this work. The ILP problem presented in Section 3.2.3 can be viewed as a generalization of the smallest satisfying set (SSS) problem [42], which is NP-hard. One direction of future work is to design polynomial-time approximation recovery algorithm under the probabilistic framework.

In addition, non-asymptotic theoretical understanding of one-stage group testing under the probabilistic framework is still limited. An interesting future research direction is to explore theoretical justi cations on the effect of the number of pairs of column vectors with zero inner product on the performance, as observed in Section 3.3.

From a practical perspective, while no signi cant difference in performance among the four classes of designs is observed in the simulations performed in this study, future research should be conducted to investigate the signi cance of the combinatorial structure of the pooling matrices under other scenarios such that parameter values with a higher range of recovery accuracy and sample sizes other than $n = 384$ that are germane to biological experiments.

Chapter 4: Optimal Transport for 3D Cell Segmentation

This chapter is joint work with Yinuo Jin, Elham Azizi, and Andrew J. Blumberg and was published as [49] in BMC Bioinformatics.

4.1 Introduction

Spatial pro ling of transcriptional states enables researchers to understand how the organization of cells in uences function by providing spatial context to single cells. Indeed, spatially resolved transcriptomics was crowned as the Method of the Year in 2020 due to the valuable biological insights it provides [50]. As a rst step in the pipeline, segmentation de nes cell boundaries from uorescent staining signals, assisting the assignment of RNA amplicons to reconstruct a gene-cell matrix; hence it plays a crucial role in characterizing cell types, their morphology, and location in the context of their microenvironment [2]. Segmentation is also a critical step in analyzing multiplexed imaging of proteins [51, 52].

Deep learning has been successful at 2D cell segmentation, leveraging the availability of diverse labeled 2D training data and specialized deep learning architectures such as U-Net [53]. Trained on labeled 2D images, 2D cell segmentation pipelines such as Mesmer [54], StarDist [55], and Cellpose [56] can segment cells from 2D images with minimal human supervision and achieve expert-level accuracy. On the other hand, due to the lack of large and diverse 3D training datasets, approaches that utilize 3D neural networks [57, 58, 59, 60] do not have comparable accuracy and generalizability, particularly in complex or dense tissues composed of heterogeneous and abnormally shaped cells such as cancer cells. Additionally, these methods often lead to the over-segmentation of cells or noisy masks, thus impacting downstream analyses. As 3D transcriptomics studies increase in scale and computational cost [61], the need for a robust, generalizable,

and user-friendly 3D cellular instance segmentation pipeline has become increasingly urgent [58]. To leverage the accuracy of 2D segmentation, [56] performs 2D segmentation layer by layer along a stitching direction and declares two cell slices as coming from the same cell if their overlap exceeds a prede ned threshold. However, when cells are stacked roughly on top of each other along the stitching direction, the resulting masks are inaccurate along the non-stitching directions. Alternative approaches have been proposed to segment cell slices on different projections of the images with subsequent 3D reconstruction [56], but the performance of such methods often suffers because of anisotropy in the microscopic images, i.e., inhomogeneity of image resolution among different dimensions due to experimental constraints. For instance, the thickness of tissue resections can be determined by imaging technology, tissue availability, and ensuring preserved tissue architecture, hence leading to higher resolution in the imaging (xy or z) dimensions than along the slicing (z or xy) dimension for softer tissues.

4.2 Methods

We developed CellStitch [49], a pipeline that applies optimal transport to segment and reconstruct cells from 3D images without requiring large 3D training datasets. CellStitch focuses on stitching 2D masks along one dimension obtained from any method. Throughout this paper, we leverage high-quality 2D segmentation masks obtained from Cellpose [56]. Hence, CellStitch does not require end-to-end training of any 3D network. Additionally, in contrast to a commonly-used stitching method [56] that fails to incorporate crucial information from the other directions, CellStitch uses optimal transport to trace cells across image layers and then uses segmentations from the other two directions to guide the declaration of new cells. In particular, to nd correspondence between cells in adjacent layers, we model the layers as certain associated discrete distributions and obtain the optimal correspondence of cells based on the optimal transport plan. The CellStitch framework is illustrated in Fig. 4.1.

Figure 4.1: Overview of CellStitch framework. (a) CellStitch consists of a 3-step pipeline to reconstruct 3D cells: backbone 2D segmentation along l-axis direction, stitching module, and an optional interpolation module. (b) Stitching pipeline: CellStitch first computes the source and target distributions based on the cell masses on given adjacent layers, and a cost matrix based on the pairwise overlap. The solved optimal transport plan is then used to deduce the optimal correspondence of cells across the adjacent layers. Finally, it reassigns instance labels of cell slices to enforce labeling consistency across all layers and creates new labels when new cells emerge based on the segmentations masks on the other two projections. (c) Interpolation pipeline: After finding optimal instance stitching, CellStitch leverages pixel-wise boundary matching followed by morphology interpolation to predict the internal layers between adjacent slices.

4.2.1 Optimal Transport for Stitching Adjacent Layers

Suppose there are $r - 1$ labels in layer l and $s - 1$ labels in layer $l + 1$ along l-axis, where the label 0 is reserved for background pixels and any label $i, j > 0$ represents a cell label; the goal is to relabel pixels in layer $l + 1$ so that the pixels corresponding to the same cell have the same label in the two layers.

In terms of optimal transport, we want to find the optimal transport plan between the two

discrete distributions $\%: \mathcal{X} \to [0,1] \in \mathbb{R}$ and $\%^{0}: \mathcal{X}' \to [0,1] \in \mathbb{R}$ where

$$\%(G) := \frac{\text{number of pixels of cell } G \text{ in layer } l}{\text{total number of pixels (including background) in layer } l}$$

$$\%^{0}(H) := \frac{\text{number of pixels of cell } H \text{ in layer } l-1}{\text{total number of pixels (including background) in layer } l-1}$$

i.e. the distributions are proportions of label masses.

The cost matrix $C \in \mathbb{R}^{|\mathcal{X}| \times |\mathcal{X}'|}$ is defined as

$$C_{G,H} = 1 - J(G,H) = 1 - \frac{|G \cap H|}{|G \cup H|}$$

where

- $J(G,H)$ is the Jaccard index of G and H;
- $|G \cap H|$ is the number of pixels in the intersection of cell G in layer l and cell H in layer $l-1$;
- $|G \cup H|$ is the number of pixels in the union of cell G in layer l and cell H in layer $l-1$.

The solved transport plan $\pi^{*} \in \mathbb{R}^{|\mathcal{X}| \times |\mathcal{X}'|}$ tells us the optimal way of moving labels in layer l to labels in layer $l-1$.

The Jaccard index is a similarity measure that has been commonly used as a statistic to measure the similarity of sample sets [62]; it is not, however, the most common measure used to define the cost matrix in optimal transport. The most common way of defining cost matrices is using the Euclidean distance, but in our case, using pairwise distances between cell centroids would not give us an optimal assignment. Consider the example shown in Fig. 4.2: the green and red cells would be incorrectly matched if we were to use the pairwise distance between the centroids as the cost matrix.

Figure 4.2: Poor correspondence of cell centroids. Using the pairwise distances between cell centroids would result in mismatched cells; as a result, this motivates the use of the union over intersection to compute the cost matrix. Here, the color code is based on the ground truth correspondence while the edges highlight the misassignments based on cell centroids. The image is taken from cell labels from the ovules dataset [63].

4.2.2 CellStitch Stitching Algorithm

We now describe how we use the optimal transport plan $\hat{\pi}$ computed in Sect. 4.2.1 to relabel cells in layer $l + 1$ consistently with the labels in layer l.

For each cell G in layer l, notice that $H^* := \arg\max_H \pi_{G,H}$ is the cell in layer $l + 1$ such that cell G most likely moves to. Define the cell tracing function $\phi_\pi : \{1, \ldots, L_l\} \to \{1, \ldots, L_{l+1}\}$ induced by the transport plan π as

$$\phi_\pi(G) = \arg\max_H \pi_{G,H}$$

If $\phi_\pi^{-1}(H) = \emptyset$ (i.e. no cells in layer l to trace the cell H), we declare H as a new cell and assign it a new label; $\forall\, \{1, \ldots, L_{l+1}\}$. Note that ϕ_π might not be an injection; for a given cell in layer $l + 1$, there can be more than one cell in layer l that traces to H. As a result, for each cell $H \in \{1, \ldots, L_{l+1}\}$, we

match it to the cell tracing to it in layer l with the least cost:

$$H^{*} = \arg\min_{G \in \mathcal{T}(H)} \; G \cdot H$$

As opposed to deterministic methods that match cell slices based on local IoU overlap (e.g. Cellpose2D), our formulation enables probabilistic matching learned by solving an optimal transport problem which is a global formulation that captures overlapping information of all pairs of cell slices all at once. In addition to the principled mathematical formulation, our formulation also provides a natural way of handling over-segmentation errors in layer l. Since we fix the labeling of layer l and use the labels in layer l to label cell slices in layer $l+1$, when there is over-segmentation in layer l, there will be multiple cell slices $G_0, G_1, \ldots, G_n$ in layer $l+1$ that trace to the same cell slice H in layer l by the tracing function $\mathcal{T}(\cdot)$. Since we only relabel cell slice H in layer $l+1$ with the one of the G_i's (in particular, the one with the maximum IoU with H), the rest of the cell slices G_i's will not be used to relabel any cell slices in layer $l+1$. The unused cell slices G_i's will then only appear in isolated layers, which can be removed after a final pass through the labeled masks. On the other hand, since our procedure focuses on labeling 2D masks in a consistent manner instead of modifying the shapes of the 2D masks, the procedure will not be able to correct under-segmentation errors. We therefore advise the users to perform pre-processing steps such as increasing the contrast of the raw images as attempts to prevent under-segmentation errors in the 2D masks.

Notice that the matched cell slices should not always be assigned to the same labels, because they might come from distinct cells if the two cells are stacked on top of each other in z-axis direction. In order to resolve the ambiguity imposed by the placement of cells, we use a voting scheme to decide whether two OT-matched cell slices will be stitched or not (see Fig. 4.3). Each cell pixel in the matched cell in layer $l+1$ gets two votes to reject stitching. For a given pixel, we use the 2D segmentation masks (x-masks and y-masks) in the projections to x- and y-axis to determine whether they vote for rejecting stitching or not. In particular, a cell pixel in layer $l+1$ will

use one rejection vote if it gets assigned different cell label than the corresponding cell pixel in layer I using the $/$ -masks (similarly for the $/$ -masks). Intuitively, each pixel in layer $I_{¸}$ 1 will use a $-/$ -mask (or $/$ -mask) vote if the pixel does not get a same cell label as the corresponding pixel in layer z using the $/$ -masks (i.e. projections to the z-axis). In other words, we use the proportion of rejection votes as a proxy to the likelihood of two cell slices coming from two cells lying on top of each other (as opposed to coming from the same cell). At the end, the two cell slices will only be stitched (i.e. assigned as the same cell labels) if the proportion of rejection votes is smaller than a user-de ned threshold. Algorithm 1 summarizes the stitching algorithm. In practice, we nd that stitching from the top to the bottom layer produces nearly identical results to stitching from the bottom to the top layer; open source code implementing the stitching algorithm from the top to the bottom layer is available at https://github.com/imyiningliu/cellstitch .

Figure 4.3: Stitching rejection voting mechanism. The diagram shows the case of how the voting mechanism prevents incorrectly stitching two cells stacked on top of each other in the z-direction. Left: An example of "splitting" scenario that would have been matched by the OT formulation. The blue pixel (middle) will vote to reject stitching due to different cell labels in the corresponding index I & $I_{¸}$ 1 in the $/$ -masks (right).

Algorithm 1 CellStitch: Stitching Algorithm

Input:

Cells $\mathcal{Z}_0, \mathcal{Z}_1, \ldots, \mathcal{Z}_{\mathcal{L}-1}$ in layer $\mathcal{L}$ in $\mathcal{D}$ masks;
Cells $\mathcal{Z}'_0, \mathcal{Z}'_1, \ldots, \mathcal{Z}'_{\mathcal{L}'-1}$ in layer $\mathcal{L}+1$ in $\mathcal{D}'$ masks;
$B_0, B_1, \ldots, B_{\mathcal{L}-1}$ indicating number of cell pixels assigned to the different labels in $\mathcal{D}$ masks;
$B'_0, B'_1, \ldots, B'_{\mathcal{L}'-1}$ indicating number of cell pixels assigned to the different labels in $\mathcal{D}'$ masks;
$\tau \in [0, 1]$ indicating the maximum proportion of disagreeing pixels for the matched cell slices

```
/* Create source and target distribution*/
for G = 0, 1, ..., L-1 do
    p_G = size(Z_G) / Σ_G size(Z_G)
end for
for H = 0, 1, ..., L'-1 do
    p'_H = size(Z'_H) / Σ_H size(Z'_H)
end for
/* Create cost matrix. */
for G = 0, 1, ..., L-1 do
    for H = 0, 1, ..., L'-1 do
        C_G-H = 1 - intersection(Z_G, Z'_H) / union(Z_G, Z'_H)
    end for
end for
π = argmin_π Σ_G,H π_G-H C_G-H
for H = 1, ..., L'-1 do
    G = argmin_G π_G-H
    if π_G-H < τ and τ ≤ B_G / B'_G ≤ 2 size(Z_G) then
        Relabel Z_H as label(G)
    else
        Create new label for Z_H
    end if
end for
```

4.3 Results

4.3.1 Datasets

We evaluated CellStitch on eight publicly available *Arabidopsis thaliana* datasets with ground-truth segmentation labels (Table 4.1). The first dataset (ovules) [63] contains 31 anisotropic images of ovule cells at all developmental stages using confocal laser scanning microscopy (anisotropy = 4); the ovules dataset was used to benchmark CellStitch's stitching and pipeline performance. The second dataset (ATAS) [64] contains 125 isotropic images of apical stem cells; the ATAS

dataset was used to evaluate CellStitch's performance at increasing anisotropy levels. Finally, in order to evaluate CellStitch's performance under a realistic setting where the anisotropy of the dataset is unknown, we further generated six additional datasets by subsampling 600 images from six different plant organs from the Arabidopsis 3D Digital Tissue Atlas (https://osf.io/fzr56).

Dataset	Number of images	Average image size (/ . - pixels)			Resolution (/ . - `<)		
Ovules	31	317	910	949	0•240	0•063	0•063
Apical Stem	125	197	509	509	0•26	0•22	0•22
Anther	100	20	224	224	Unknown		
Filament	100	20	224	224	Unknown		
Leaf	100	20	224	224	Unknown		
Pedicel	100	20	224	224	Unknown		
Sepal	100	20	224	224	Unknown		
Valve	100	20	224	224	Unknown		

Table 4.1: Summary of benchmarking datasets

4.3.2 Evaluation Metrics and Benchmark Methods

We benchmarked our results with the state-of-the-art pipeline from each of the three classes of current deep learning-based 3D segmentation pipelines:

- Cellpose2D that stitches cell slices in adjacent layers if their IoU exceeds a user-de ned threshold (2D-based) [56],
- Cellpose3D (2.5D-based) [56],
- PlantSeg's pretrained confocal_unet_bce_dice_ds1x model (3D-based) [60].

In order to evaluate the segmentation accuracy, we rst matched the cells in the segmentation mask and the ground truth label if the two cells have an intersection over union greater than τ. Then, we computed the precision, recall, and average precision as

$$\text{precision}_{\tau} := \frac{)\%}{)\% + \%} \quad \text{recall}_{\tau} := \frac{)\%}{)\% + \#} \quad \%_{\tau} := \frac{)\%}{)\% + \# + \%}$$

where TP is the number of matched cells under the given threshold, FP is the number of unmatched cells in the segmentation mask, and FN is the number of unmatched cells in the ground truth labels. In the following section, we report precision, recall and AP computed at $IoU = 0.5$, as well as the mean average precision (mAP) averaged over $IoU \in \{0.25, 0.5, 0.75\}$.

4.3.3 Experimental Results

We first evaluate CellStitch's segmentation performance on the ovules dataset [63] under two settings:

- low anisotropy (anisotropy = 4): original data;
- high anisotropy (anisotropy = 8): sparsifying the z dimension by removing every other layer.

To compare the performance between the 2D-based, 2.5D-based, and 3D-based methods, we used the same training data (22 training images, 2 validation images, 7 test images) that was used to train PlantSeg's `confocal_unet_bce_dice_ds1x' model to train a Cellpose 2D segmentation model for 100 epochs with learning rate 0.2 and batch size 8; the trained model was then used as a backbone to generate 2D masks later used for 3D volume reconstruction from CellStitch, Cellpose2D and Cellpose3D. We see that Cellpose3D suffers the most from increased anisotropy in the raw data, that Cellpose2D is subject to under-segmentation issues from its hard-thresholded stitching, and that CellStitch consistently produces the best segmentation masks for both low and high anisotropy settings (Table 4.2).

Next, we used the *Arabidopsis thaliana* apical stem cells (ATAS) dataset [64] containing 125 isotropic images under $0.22\mu m \times 0.22\mu m \times 0.26\mu m$ resolution to further compare the performance of CellStitch and Cellpose; since the images in the ATAS dataset are isotropic, we are able to explore how the performance of CellStitch changes with increasing anisotropy. We followed a 7-3 train-test split to train a Cellpose model which is the backbone to generate 3D segmentations for CellStitch, Cellpose2D, and Cellpose3D; due to the lack of instructions on how to train PlantSeg models on additional datasets, we used PlantSeg's pretrained model for the rest of the

Setting	Method	Precision		Recall		AP		mAP	
Low Anisotropy	CellStitch	0.64	0.08	0.64	0.14	0.48	0.11	0.51	0.09
	Cellpose2D	0.42	0.07	0.57	0.09	0.31	0.05	0.36	0.05
	Cellpose3D	0.45	0.20	0.84	0.11	0.42	0.20	0.42	0.20
	PlantSeg	0.45	0.06	0.80	0.07	0.40	0.05	0.41	0.05
High Anisotropy	CellStitch	0.66	0.07	0.52	0.10	0.41	0.08	0.48	0.07
	Cellpose2D	0.48	0.05	0.54	0.09	0.34	0.05	0.40	0.04
	Cellpose3D	0.35	0.14	0.73	0.13	0.31	0.13	0.31	0.12
	PlantSeg	0.32	0.23	0.72	0.05	0.27	0.19	0.29	0.19

Table 4.2: Performance benchmarks on the ovules dataset; the best performance (within 0.03) is in bold. The method that achieves the best performance under the majority of the metrics is marked with .

experiments. We introduced anisotropy in the ATAS dataset by subsampling across layers in order to further test the robustness of CellStitch and Cellpose3D against anisotropy in the dataset. We found that Cellpose3D achieves higher average precision when there is no anisotropy in the dataset, whereas Cellpose2D does not achieve comparable performance on isotropic dataset due to its ignorance of the other two directions. Additionally, similar to the results on the ovules dataset, the average precision of Cellpose3D signi cantly dropped under the high anisotropy setting. The quantitative results are presented in Table 4.3.

Setting	Method	Precision		Recall		AP		mAP	
Anisotropy = 0	CellStitch	0.79	0.08	0.77	0.04	0.64	0.07	0.70	0.07
	Cellpose2D	0.61	0.10	0.69	0.04	0.48	0.07	0.54	0.07
	Cellpose3D	0.87	0.17	0.98	0.01	0.85	0.17	0.85	0.16
	PlantSeg	0.40	0.07	0.92	0.16	0.39	0.07	0.39	0.07
Anisotropy = 5	CellStitch	0.81	0.04	0.58	0.05	0.51	0.05	0.62	0.04
	Cellpose2D	0.67	0.04	0.61	0.05	0.47	0.04	0.56	0.03
	Cellpose3D	0.40	0.09	0.59	0.11	0.32	0.08	0.32	0.06
	PlantSeg	0.37	0.08	0.31	0.13	0.20	0.08	0.21	0.06
Anisotropy = 10	CellStitch	0.75	0.04	0.46	0.06	0.40	0.05	0.53	0.05
	Cellpose2D	0.64	0.04	0.55	0.05	0.42	0.05	0.52	0.03
	Cellpose3D	0.30	0.09	0.38	0.11	0.20	0.08	0.26	0.06
	PlantSeg	0.12	0.06	0.04	0.03	0.03	0.02	0.00	0.00

Table 4.3: Performance on the ATAS dataset; the best performance (within 0.03) is in bold. The method that achieves the best performance under the majority of the metrics is marked with .

Finally, to test CellStitch's performance under the practical setting, where the anisotropy of the datasets might be unknown, we generated six datasets by sampling manually annotated images

from the Arabidopsis 3D Digital Tissue Atlas (https://osf.io/fzr56). For each of the six datasets, we trained Cellpose's 2D segmentation model for 100 epoches with learning rate 0.2 and batch size 8 under a 7-3 train-test split as the backbone to further generate 3D segmentation results for CellStitch and Cellpose3D. We see in Table 4 that CellStitch is consistently the best performing method(s) on all datasets. Furthermore, CellStitch is also able to discover more morphology diversity, whereas Cellpose3D tends to over-segment non-spherical cells.

Dataset	Method	Precision		Recall		AP		mAP	
Anther	CellStitch	0.66	0.10	0.53	0.11	0.42	0.10	0.41	0.08
	Cellpose2D	0.48	0.11	0.53	0.10	0.33	0.08	0.34	0.07
	Cellpose3D	0.41	0.17	0.38	0.13	0.24	0.10	0.23	0.08
	PlantSeg	0.31	0.05	0.60	0.10	0.26	0.04	0.26	0.04
Filament	CellStitch	0.74	0.14	0.53	0.12	0.46	0.13	0.46	0.11
	Cellpose2D	0.51	0.15	0.57	0.13	0.38	0.13	0.38	0.12
	Cellpose3D	0.03	0.02	0.22	0.11	0.03	0.02	0.03	0.02
	PlantSeg	0.37	0.08	0.31	0.13	0.20	0.08	0.21	0.06
Leaf	CellStitch	0.83	0.17	0.75	0.19	0.68	0.21	0.66	0.18
	Cellpose2D	0.65	0.19	0.77	0.19	0.56	0.20	0.55	0.18
	Cellpose3D	0.07	0.05	0.53	0.20	0.06	0.04	0.06	0.04
	PlantSeg	0.30	0.08	0.71	0.20	0.27	0.09	0.27	0.08
Pedicel	CellStitch	0.57	0.23	0.44	0.16	0.35	0.16	0.36	0.13
	Cellpose2D	0.36	0.19	0.44	0.15	0.25	0.13	0.26	0.11
	Cellpose3D	0.61	0.25	0.30	0.13	0.27	0.13	0.28	0.09
	PlantSeg	0.29	0.07	0.39	0.13	0.20	0.06	0.21	0.05
Sepal	CellStitch	0.51	0.12	0.43	0.16	0.31	0.12	0.33	0.09
	Cellpose2D	0.33	0.11	0.44	0.17	0.23	0.09	0.25	0.08
	Cellpose3D	0.41	0.23	0.34	0.12	0.20	0.10	0.21	0.09
	PlantSeg	0.34	0.07	0.43	0.14	0.23	0.06	0.25	0.05
Valve	CellStitch	0.71	0.07	0.47	0.04	0.40	0.05	0.41	0.11
	Cellpose2D	0.58	0.09	0.47	0.04	0.35	0.05	0.37	0.05
	Cellpose3D	0.67	0.10	0.51	0.05	0.41	0.06	0.40	0.05
	PlantSeg	0.53	0.08	0.38	0.07	0.29	0.06	0.32	0.05

Table 4.4: Performance on Arabidopsis 3D Digital Tissue Atlas; best performance (within 0.03) is in bold. The method that achieves the best performance under the majority of the metrics is marked with .