

**Algorithmic Improvements for Modern Sequencing Technologies: Long-Read
Polishing, Alignment, and Single-Cell Clustering**

by

Weerakoon Mudiyanseelage Minindu Bandara Weerakoon

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
August 8, 2026

Keywords: Quality scores, Machine learning, Sequence analysis, HiFi reads, Single Cell
RNA Sequencing, Spatial ScRNA, Demultiplexing, Clustering

Copyright 2026 by Weerakoon Mudiyanseelage Minindu Bandara Weerakoon

Approved by

Haynes Heaton, Assistant Professor of Computer Science and Software Engineering
Sathyanarayanan Aakur, Assistant Professor of Computer Science and Software
Engineering

Tao Shu, Associate Professor of Computer Science and Software Engineering
Neha Potnis, Associate Professor of College of Agriculture

Abstract

Sequence analysis involves the examination and interpretation of genetic and genomic data, particularly the order of nucleotides, adenine, cytosine, guanine, and thymine, in DNA and RNA molecules. Accurate and precise sequence analysis is critical, as it enables the detection of genetic variations and mutations linked to diseases such as cancer and informs the development of effective treatments. This dissertation presents a multi-faceted approach to enhancing modern sequencing pipelines, addressing four complementary problems across long-read and single-cell technologies.

First, we investigate the per-base quality scores produced by Pacific Biosciences high fidelity (HiFi) circular consensus sequencing. By comparing these scores against a validation dataset, we identify systematic deviations from empirical error rates and describe a deep learning method, *TopoQual*, that corrects these discrepancies through topological analysis of partial order alignment graphs combined with learned recalibration. The improved scores are validated on an umbilical cord blood sample and shown to track empirical error rates more closely than both the original PacBio scores and DeepConsensus, while simultaneously polishing approximately 32% of the errors present in the input consensus reads.

Second, we address the computational cost of partial order alignment (POA), the multiple sequence alignment method that underlies both consensus generation and the proposed quality score recalibration. While POA preserves more information than sequential pairwise alignment, its quadratic time and memory complexity make it impractical on long reads such as PacBio HiFi. We extend the longest common subsequence with k-mer matches (LCSk++) algorithm to graph structures and present *LCSkPOA*, a banded POA method that achieves up to a 60 times speedup over unbanded SPOA on long synthetic sequences while maintaining alignment accuracy above 99% on real HiFi data.

Third, we focus on the demultiplexing of pooled single-cell ribonucleic acid sequenc-

ing (scRNA-seq) data, specifically enhancing the clustering performance of the Souporecell method. Existing genotype-based demultiplexing tools were primarily designed and benchmarked on pools of 16 or fewer donors, beyond which clustering quality degrades sharply. We present *Souporcell3*, which combines a 10x-merge cluster initialization strategy, deterministic annealing-enhanced K-Harmonic Means clustering, and iterative refinement with bad-cluster reinitialization and good-cluster locking. *Souporcell3* achieves near-perfect adjusted Rand index scores and zero incorrectly merged clusters at donor counts of up to 64, substantially outperforming Vireo, overclustered Vireo, and the original Souporecell.

Fourth, we extend the clustering paradigm to spatial single-cell transcriptomics, where the goal shifts from demultiplexing donors to assigning cells to anatomically distinct tissue domains. By analysing Slide-seqV2 mouse cerebellum and Visium human dorsolateral prefrontal cortex (DLPFC) data under a suite of probabilistic models, we show that the negative binomial likelihood and generalised linear model principal component analysis (GLM-PCA) embedding exhibit complementary failure modes: the negative binomial loss is precise but has narrow range, while GLM-PCA provides wide global separation but is insensitive to small gene-level differences between transcriptionally similar layers. Motivated by this observation, we present *SpaceCell*, a three-stage method that first clusters cells with a Gaussian mixture model on the GLM-PCA embedding, then refines the partition by iteratively selecting the most representative cells under a negative-binomial discriminability score, and finally assigns the remaining cells using Leiden community detection on a graph whose edge weights combine negative binomial dissimilarity with spatial distance. *SpaceCell* improves cluster assignment accuracy over baseline k-means++ and Gaussian mixture model clustering on both datasets and offers a flexible framework for emerging spatial platforms such as Slide-seq and Visium HD.

Together, these contributions improve the accuracy, efficiency, and applicability of sequence analysis pipelines across both long-read and single-cell technologies, with direct

implications for variant calling, genome assembly, and population-scale single-cell studies.

Artificial Intelligence (AI) Use Disclosure Statement

In the preparation of this dissertation, the following Artificial Intelligence (AI) tools were used: Claude. These tools were used primarily to assist with grammar, spelling correction, and formatting assistance with $\LaTeX$. The author acknowledges full responsibility for the intellectual content of this work and has ensured that all AI-assisted sections have been reviewed and revised for accuracy and appropriate academic style. All AI-generated content was reviewed and validated for relevance, appropriateness, and accuracy before incorporation into the final document to maintain scholarly integrity of this research.

Digital Accessibility Disclosure Statement

In the preparation of this dissertation, the following digital accessibility tools were used to ensure this document complies with federal requirements: PAC (PDF Accessibility Checker). The author acknowledges full responsibility for the intellectual content of this work and has made a good faith effort to comply with digital accessibility requirements in publishing, wherein the nature of the content does not significantly change in order to do so. Furthermore, all content has been reviewed and revised to meet these requirements prior to final publication.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my advisor, Dr. Haynes Heaton, without whom I may never have embarked on the journey toward a Ph.D. His extensive knowledge of computer science and biology, along with his unwavering guidance and mentorship, has been invaluable throughout my time at Auburn University.

I would also like to thank the members of my dissertation committee: Dr. Sathyanarayanan Aakur, Dr. Tao Shu, and Dr. Neha Potnis. Their thoughtful feedback during my proposal defense, as well as their insightful questions and valuable suggestions, helped shape the direction of this dissertation and strengthened it in meaningful ways. I am deeply grateful for the time, expertise, and support they generously contributed throughout this process.

Finally, I would like to express my heartfelt gratitude to my family and friends. My parents, Sanath Weerakoon and Ghana Dissanayake, have been unwavering sources of love, encouragement, and support throughout my journey, both as a graduate student and as a person. I am especially grateful to my girlfriend, Piyumali Weerasuriya, whose patience, understanding, and constant encouragement sustained me through the most challenging moments of this Ph.D. journey.

Table of Contents

Abstract	2
Artificial Intelligence (AI) Use Disclosure Statement	5
Digital Accessibility Disclosure Statement	6
Acknowledgements	7
List of Abbreviations	17
1 Introduction	18
1.1 Motivation for studying HiFi long-read consensus and alignment	19
1.2 Motivation for improving demultiplexing in single-cell RNA sequencing	25
1.3 Dissertation Organization	28
2 Quality Score Correction and Sequence Polishing	29
2.1 Motivation	29
2.2 Evidence for Overestimation of Quality Scores	30
2.3 Method	36
2.4 Results	40
2.5 Discussion	45
3 Speeding Up Partial Order Alignment with Banding	47
3.1 Motivation	47
3.2 Implementation	48
3.3 Results	56
3.4 Discussion	61
4 Demultiplexing High-Donor Single-Cell RNA-seq	63
4.1 Motivation	63
4.2 Method	65
4.3 Datasets	75
4.4 Evaluation Metrics	79

4.5	Cluster Initialization Evaluation	80
4.6	K-Harmonic Means Evaluation	83
4.7	Main Results	85
4.8	Discussion	95
5	Clustering Spatial scrnaseq	98
5.1	Introduction	98
5.2	Analysis of Data	100
5.3	Method	113
5.4	Results	116
5.5	Discussion	117
6	Conclusion	119
6.1	Summary of Contributions	119
6.2	Closing Remarks	121
	References	122

List of Tables

2.1	Percentage of errors corrected by TopoQual using each technique. Total bases analyzed: 11Gb; errors present: 1.8Mb.	42
2.2	Results of different methods on the validation dataset, average subread depth = 10.	43
3.1	Comparison of alignment accuracy (alignment score similarity with respect to SPOA unbanded alignment) for SPOA, LCSkPOA (sequential), and LCSkPOA (parallel) at different values of k. Values are averages of (10) runs on synthetic data and PacBio HiFi data.	60
4.1	Composition of the balanced datasets used in evaluation.	78

List of Figures

- 1.1 PacBio HiFi sequencing overview. Hairpin adapters are ligated to both ends of a DNA molecule to form a circular construct. A strand-displacing DNA polymerase replicates this circular template, generating multiple subreads that are consolidated into a high-accuracy consensus sequence. Adapted from PacBio. 21
- 1.2 Overview of the droplet-based single-cell RNA sequencing scRNA-seq procedure, from sample preparation to sequencing library. Individual cells are encapsulated in droplets with uniquely barcoded beads, lysed within each droplet, and their messenger RNA is captured, reverse-transcribed, and tagged with cell-specific barcodes. The barcoded cDNA from all cells is then pooled and prepared as a single sequencing library, with the cell-of-origin recoverable from the barcode after sequencing. Adapted from 10x Genomics. 25
- 2.1 Comparison of base quality scores at a mismatch reference position chr3:5,084,593 between DeepConsensus (bottom) and the original PacBio (top), as viewed in the Integrative Genomics Viewer (IGV). 32
- 2.2 Comparison of base quality scores at a mismatch reference position chr3:5,170,774 between TopoQual and the original PacBio, as viewed in the Integrative Genomics Viewer (IGV). 34
- 2.3 Comparison of base quality scores at a mismatch reference position chr21:13,220,919 between TopoQual and the original PacBio, as viewed in the Integrative Genomics Viewer (IGV). 35
- 2.4 The quality score estimation strategy of TopoQual. Subreads are aligned together with the current consensus sequence using partial order alignment. Potential alternative bases in the multiple alignment are detected by identifying parallel bases in the POA graph. These signals, along with several others, are passed to a deep learning model that outputs a recalibrated quality score. . . 36

2.5	Topoqual Algorithm	39
2.6	Errors present before (top) and after polishing (bottom) by TopoQual in the validation dataset for T to X mutations, stratified by the three correction types (parallel bases prefer a different base, POA consensus deletes a base, and POA consensus substitutes a base) on chromosome 2.	41
2.7	Quality score validation comparison of different methods on chromosome 2. Well-calibrated quality scores are expected to fall on the diagonal. Marginal plots show the distribution of base counts.	44
3.1	Basic dynamic programming example with $k = 2$. Each cell shows $dp(i, j)$, the LCS _k ++ alignment score for the i th column and j th row.	50
3.2	Matching k -mers between the query and the POA graph with $k = 2$. Five match pairs are present, denoted a through e. Starts are shown as circles and ends as squares. In this example, “b continues a” and “c precedes e” both hold. . .	52
3.3	Results of running LCS _k POA on synthetic data. (A) Speedup of ABPOA and LCS _k POA compared to unbanded SPOA. (B) Memory usage of banded methods ABPOA and LCS _k POA compared to unbanded SPOA.	57
3.4	Comparison of speedup of ABPOA and LCS _k POA when run on the synthetic dataset and the PacBio HiFi dataset, both compared against unbanded SPOA.	58
3.5	Speedup of LCS _k POA (parallel) across different datasets and thread counts. .	59

4.1	Method overview. (A) Randomly initialize (10k) cluster centers to ensure broad coverage of the genotype space. (B) Merge nearby cluster centers using a distance metric defined as the sum of allele fraction differences weighted by allele counts until (k) clusters remain. (C) Cluster using K-Harmonic Means (KHM) to ensure enhanced convergence and reduced sensitivity to initialization. (D) Identify low- and high-quality clusters based on the number of assigned cells and the loss value per cluster. (E) Reinitialize low-quality clusters. (F) Rerun KHM on the refined set while locking the high-quality clusters.	66
4.2	K-means++ Initialization: (A) Randomly select a cell as the first cluster center. (B-D) Iteratively select cells as cluster centers based on their squared distance to the closest existing cluster center.	67
4.3	Overclustering Initialization: (A) Select (k + k) random cells as cluster centers, where (k) is the desired number of clusters. (B) Cluster using the selected cluster centers. (C) Select (k) cluster centers with the maximum number of associated cells.	68
4.4	10x-Merge Initialization: (A) Randomly initialize (10k) cluster centers. (B) Iteratively merge clusters in close proximity until the required cluster size is reached.	69
4.5	Number of cells assigned to each cluster center on a no-doublet balanced dataset with (64) donors. Empty or near-empty clusters indicate the presence of incorrectly merged clusters elsewhere in the assignment.	73
4.6	Total log loss for each cell's assigned cluster, normalized by the mean total log loss, on a no-doublet balanced dataset with (64) donors.	74
4.7	(A) After the first clustering run, bad clusters are detected and reinitialized while good clusters are locked. (B) With the good clusters locked against updates, the KHM clustering converges to a refined solution.	75

4.8	Illustration of the IMC metric. (A) Before clustering, cluster centers are initialized. (B) After the clustering method converges, an empty cluster center is present (not assigned to any cells) and one cluster center is assigned to cells from two donors, yielding (IMC = 1).	80
4.9	Log loss versus iterations for EM with five different initialization methods (random, k-means++, overclustering, optimal, and 10x-merge) on a no-doublet balanced dataset with (64) donors.	81
4.10	The right panel shows the ARI values for EM runs with different cluster initialization methods. The left panel illustrates the incorrectly merged clusters IMC present for the same methods.	82
4.11	Log loss versus iterations for clustering methods DA-EM, DA-KHM, V-EM, and V-KHM on a no-doublet balanced dataset with (64) donors.	83
4.12	The right panel shows the ARI values for clustering methods DA-EM, DA-KHM, V-EM, and V-KHM. The left panel illustrates the incorrectly merged clusters present for the same methods.	84
4.13	Number of incorrectly merged donor assignments for Vireo, overclustered Vireo, Souporcell, and Souporcell3 across different donor counts and doublet percentages on balanced datasets.	86
4.14	Adjusted Rand Index for Vireo, overclustered Vireo, Souporcell, and Souporcell3 across different donor counts and doublet percentages on balanced datasets.	87
4.15	Cluster maps using UMAP for the x and y axes, showing the result of the 10 percent doublet, 64-donor balanced dataset after running Souporcell, Souporcell3, Vireo, and overclustered Vireo prior to doublet detection.	89
4.16	Heat maps showing cell counts for assigned cluster versus ground truth for (64) clusters of the (10 percent)-doublet balanced dataset after running Souporcell, Souporcell3, Vireo, and overclustered Vireo, prior to doublet detection.	90

4.17	Number of incorrectly merged donor assignments for Vireo, overclustered Vireo, Souporcell, and Souporcell3 on the three unbalanced datasets.	91
4.18	Adjusted Rand Index for Vireo, overclustered Vireo, Souporcell, and Souporcell3 on the three unbalanced datasets.	92
4.19	Heat maps showing cell counts for assigned cluster versus ground truth for unbalanced dataset 1, after running Souporcell, Souporcell3, Vireo, and overclustered Vireo.	93
4.20	Heat maps showing cell counts for assigned cluster versus ground truth for unbalanced dataset 2, after running Souporcell, Souporcell3, Vireo, and overclustered Vireo.	94
4.21	Heat maps showing cell counts for assigned cluster versus ground truth for unbalanced dataset 3, after running Souporcell, Souporcell3, Vireo, and overclustered Vireo.	95
5.1	Ground-truth annotation of the Slide-seqV2 mouse cerebellum sample, with cells coloured by their assigned cell type.	101
5.2	Negative binomial likelihood-based cell separation for each cell type in the mouse cerebellum sample. Left: spatial view with cells coloured by ground-truth label. Right: scatter of own-cluster loss against other-cluster loss; cells below the diagonal are closer to their assigned cluster's average.	103
5.3	GLM-PCA based cell separation for each cell type in the mouse cerebellum sample. Cells more widely dispersed in the loss space than under the NB model, but a meaningful fraction sit on the wrong side of the diagonal.	104
5.4	K-means++ clustering on the GLM-PCA embedding of the cerebellum data. Ground truth (left) compared to k-means++ cluster assignments (right). ARI = 0.448, F1 = 0.515.	105

5.5	Gaussian mixture model clustering on the GLM-PCA embedding of the cerebellum data. Ground truth (left) compared to GMM cluster assignments (right). ARI = 0.545, F1 = 0.520.	105
5.6	Ground-truth annotation of Visium DLPFC section 151673, with spots coloured by manually assigned cortical layer (L1–L6) and white matter (WM).	107
5.7	Negative binomial likelihood-based cell separation for each layer in the DLPFC sample. Spots fall very close to the diagonal but consistently on the correct side.	109
5.8	GLM-PCA based cell separation for each layer in the DLPFC sample.	110
5.9	K-means++ clustering on the GLM-PCA embedding of the DLPFC data. ARI = 0.395, F1 = 0.537.	111
5.10	Gaussian mixture model clustering on the GLM-PCA embedding of the DLPFC data. ARI = 0.433, F1 = 0.557.	111
5.11	Method overview. (A–C) Cluster with a Gaussian mixture model on the GLM-PCA embedding of the count matrix and spatial coordinates. (D–F) Refine with negative binomial likelihood-based cell selection. (G–I) Assign the unassigned cells using Leiden community detection, connecting neighbouring cells with a combined negative binomial and spatial loss.	113
5.12	Clustering of the Slide-seqV2 mouse cerebellum sample. SpaceCell recovers spatially coherent layers that closely follow the annotated cerebellar architecture, whereas the CellCharter assignments are more fragmented across layer boundaries.	117

List of Abbreviations

RNA	ribonucleic acid	25
SNP	single-nucleotide polymorphism	65
UMI	unique molecular identifier	66
HiFi	high fidelity	2
CCS	circular consensus sequencing	38
POA	partial order alignment	2
LCS	longest common subsequence	24
LCSk++	longest common subsequence with k-mer matches	2
DFS	depth-first search	54
ZMW	zero-mode waveguide	22
scRNA-seq	single-cell ribonucleic acid sequencing	2
ENA	European Nucleotide Archive	78
EM	expectation-maximization	70
KHM	K-Harmonic Means	63
DA	deterministic annealing	72
GMM	Gaussian mixture model	98
ARI	Adjusted Rand Index	79
IMC	Incorrectly Merged Cluster	79
UMAP	Uniform Manifold Approximation and Projection	89
DLPFC	dorsolateral prefrontal cortex	3
GLM-PCA	generalised linear model principal component analysis	3

Chapter 1

Introduction

This dissertation studies two related sets of problems in sequence analysis, with a focus on improving the accuracy and applicability of modern sequencing technologies. The first part addresses long-read sequencing, where we investigate base quality score correction, read polishing, and multiple sequence alignment in Pacific Biosciences HiFi reads [1]. We present *TopoQual* [2], a method for correcting base-level quality scores and polishing consensus reads, and *LCSkPOA* [3], a banding strategy that enables fast partial order alignment on long reads. Together, these methods improve the accuracy of base-level information available to downstream bioinformatics pipelines and reduce the computational cost of the alignment substrate they depend on. This is particularly important for somatic variant calling in cancer genomics [4–6], where the true mutations occur at frequencies below the per-base error rate of any single sequencing read, and where miscalibrated base quality scores yield false-positive variant calls that must be filtered downstream [7, 8].

The second part addresses single-cell sequencing, where improvements to demultiplexing and clustering enable cost-effective multiplexed experiments [9–11] and unlock the analysis of emerging spatial transcriptomics modalities. A common limitation in droplet-based scRNA-seq demultiplexing is the breakdown of clustering quality as the number of pooled donors increases [9], while methods adapted to spatial single-cell technologies such as Slide-seq and Visium HD remain scarce. We present *Souporcell3* [12], an enhanced genotype-based demultiplexing method that scales to high donor counts, and

SpaceCell, a clustering method designed for spatial single-cell datasets. Beyond single-cell applications, these methods also help with the broader problem of clustering high-dimensional, sparse, non-Gaussian data [13–15], where classical clustering algorithms tend to fail.

Section 1.1 elaborates the motivation for the first part of the dissertation, while Section 1.2 proposes the reasoning behind the second part. Section 1.3 describes the organization of this dissertation.

1.1 Motivation for studying HiFi long-read consensus and alignment

Sequence analysis is the process of examining and interpreting sequences of biological data, particularly in the context of genetics and genomics [16]. In genetics, sequence analysis involves studying the order of nucleotides (adenine, thymine/uracil, cytosine, and guanine) in DNA or RNA molecules. This analysis can include tasks such as identifying genes, predicting protein structures and functions, studying evolutionary relationships, and detecting variations or mutations associated with diseases. There are various techniques and tools used in sequence analysis, including alignment algorithms to compare sequences [17–19], statistical models to infer evolutionary relationships [20], and bioinformatics software to analyze large datasets [21]. Sequence analysis plays a crucial role in fields like molecular biology, genetics, evolutionary biology, and personalized medicine, contributing to our understanding of biological systems and informing clinical decision-making.

Single-molecule sequencing accurately determines the nucleotide composition of individual DNA molecules without the need for amplification before library preparation [22, 23]. This is a critical distinction from short-read platforms such as Illumina [24], which require PCR or bridge amplification prior to sequencing. During amplification, errors are introduced that propagate into every downstream read derived from that template [25], and no matter how accurate the subsequent base calling is, these amplification-introduced errors

place a hard ceiling on the achievable per-base accuracy of the platform. This ceiling is especially problematic for somatic variant calling, where the true variants of interest occur at frequencies of roughly 25 mutations per cell per year of life [26, 27], below the error rates introduced during PCR amplification [25]. Single-molecule platforms avoid this amplification cap entirely, making them uniquely suited for high-sensitivity variant detection and for applications such as cancer genomics [4, 28] where the signal of interest lies at very low allele frequencies.

Pacific Biosciences and Oxford Nanopore [29] are leading providers of single-molecule sequencing platforms. The focus of this work lies on Pacific Biosciences' circular consensus sequencing, also known as high fidelity (HiFi) technology [30].

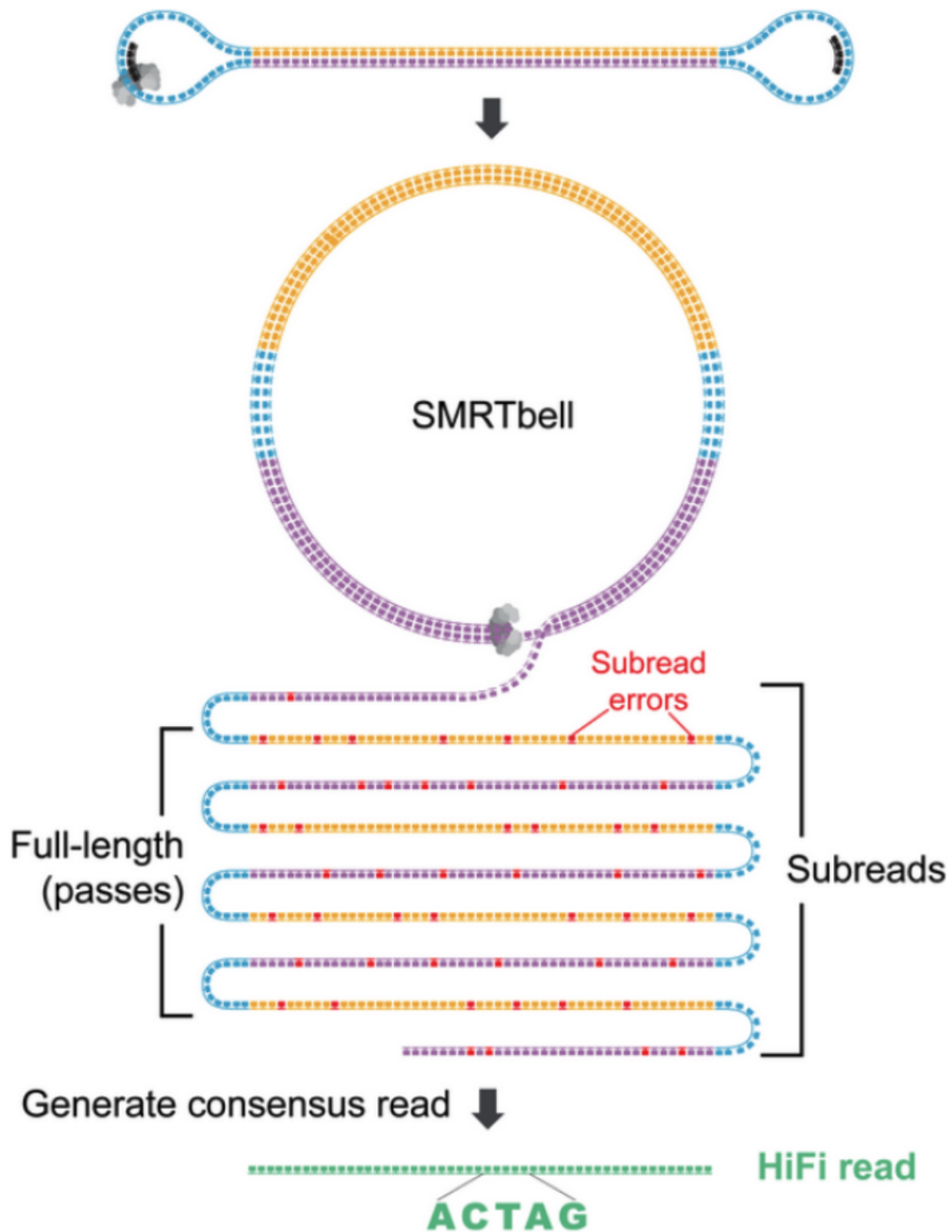


Figure 1.1: PacBio HiFi sequencing overview. Hairpin adapters are ligated to both ends of a DNA molecule to form a circular construct. A strand-displacing DNA polymerase replicates this circular template, generating multiple subreads that are consolidated into a high-accuracy consensus sequence. Adapted from PacBio.

In PacBio HiFi sequencing, hairpin adapters are ligated to both ends of a DNA molecule to form a circular construct [31]. A strand-displacing DNA polymerase then continuously replicates this circular template, enabling multiple passes around the molecule. The resulting subreads, comprising sequences from both the forward and reverse strands, are extracted and used to generate a high-accuracy consensus sequence [32]. Combined with enhanced polymerase processivity, this approach allows for numerous complete passes over the same molecule, as illustrated in Figure 1.1.

The polymerase itself is held inside a zero-mode waveguide (zero-mode waveguide (ZMW)) [23], a nanoscale chamber smaller than the wavelength of visible light. This sub-wavelength design enables the optical isolation of a single polymerase molecule, but the resulting signal-to-noise constraints limit the accuracy of any single subread to roughly 85%. HiFi sequencing overcomes this per-pass limit by consolidating the many subreads of a single circular template into a consensus sequence [1]. Under the assumption that errors across passes are largely uncorrelated [1, 31], the theoretical accuracy of this consensus is bounded only by the number of independent passes, in principle allowing for per-base accuracies on the order of $1 - 10^{-9}$ or better.

The adoption of HiFi sequencing benefits a variety of downstream applications, namely, (1) enabling more complete *de novo* genome assemblies [33–35], (2) improving the detection of structural variants [36], (3) resolving repetitive and previously inaccessible regions of the genome [37, 38], and (4) supporting accurate small-variant calling in both germline and somatic contexts [39–41]. Despite these advantages, the HiFi pipeline contains several components that remain either proprietary or under-explored in the literature. In particular, the method used to generate consensus sequences from subreads is not publicly available in full detail, the per-base quality scores reported by the platform are systematically overestimated relative to empirically observed error rates [2], and the multiple sequence alignment step required for consensus calling is computationally expensive at long read lengths [19, 42].

The per-base quality scores associated with HiFi reads are critical inputs to virtually every downstream bioinformatics tool, including variant callers [7, 8, 40, 41] and polishing pipelines [39, 43]. If these quality scores are systematically miscalibrated, as we observe to be the case for HiFi reads, downstream callers weight unreliable bases as if they were reliable, leading to inflated false-positive rates for low-frequency variants such as somatic mutations [26, 27]. Existing consensus tools also treat all base positions uniformly without explicitly accounting for the systematic error patterns of single-molecule sequencing [39, 43], leaving consensus errors that are not well corrected by the standard procedure. Accurate base quality estimation and post-consensus polishing are therefore direct prerequisites for reliable variant detection in clinical and research applications.

The consensus sequence is built by aligning all subreads of a single circular template to each other, which is itself a multiple sequence alignment problem. Optimal multiple sequence alignment of k sequences of length n has a worst-case runtime of $\mathcal{O}(n^k)$ under standard dynamic programming [17, 18], which is intractable for HiFi reads where n is greater than 10,000 and k can exceed 10 subreads. Partial order alignment (POA) [19] approximates this problem by iteratively aligning each subread to a growing sequence graph and finding the consensus as the highest-supported path through that graph, reducing the effective runtime to roughly $\mathcal{O}(n^2)$ per subread. While an approximation, POA works well in practice [19, 43] and forms the basis of most modern long-read consensus pipelines. However, the quadratic memory and time complexity still makes plain POA impractical for HiFi reads without aggressive banding [42], and standard banding strategies that work well for short reads [44, 45] do not necessarily transfer to long reads where insertion and deletion errors shift the alignment band across long stretches.

Improving the HiFi pipeline is a complicated task that can be approached from a variety of directions. One method is to introduce model components that explicitly handle the dominant error modes of the platform through post-consensus polishing [2, 43]. A second direction is to recalibrate the quality scores produced by the platform against empirically

observed error rates, so that downstream tools receive better-calibrated inputs [46, 47]. A third method involves redesigning the alignment substrate of consensus calling to maintain accuracy while reducing the computational footprint on long reads [42, 48, 49]. Each of these directions addresses a distinct failure mode of the existing pipeline and together they target the consensus, quality estimation, and alignment stages of HiFi processing.

The work in Chapters 2 and 3 follows the aforementioned directions to design and evaluate improvements to the HiFi processing pipeline. First, we develop a more accurate method for predicting base-level quality scores by combining a topological analysis of the partial order alignment graph with a deep learning model trained against empirical error rates observed in alignments to high-quality reference assemblies, and we use the same framework to polish residual errors in the consensus read. Second, we investigate alternative banding strategies for subread POA based on longest common subsequence (LCS) variants [48–50] to enable efficient alignment without sacrificing the accuracy required by the consensus step. The proposed methods are evaluated against a cord blood sample which we sequenced and is now publicly available (SRA biosample accession PRJNA1128051) and simulated reads [51] to quantify their effect on quality score calibration, polishing accuracy, and alignment runtime and memory.

The results of the studies in Chapters 2 and 3 confirm a variety of hypotheses regarding the HiFi pipeline. It is validated that the proposed polishing strategy improves consensus accuracy without compromising accuracy in unaffected regions of the read. This improved consensus, in turn, has a positive impact on the quality of downstream variant calls [7, 8]. The recalibrated quality scores are shown to track empirical error rates more closely than the scores reported by the platform, addressing a known limitation of existing HiFi quality estimation [2]. The alternative banding strategies are also shown to substantially reduce the memory and runtime cost of subread POA while maintaining alignment accuracy comparable to standard approaches [19, 42]. As such, the results provide a strong argument for integrating these refinements into the HiFi processing pipeline to improve the reliability

of long-read sequence analysis.

1.2 Motivation for improving demultiplexing in single-cell RNA sequencing

Although improvements to long-read sequencing can provide more accurate base-level information for downstream pipelines (see Chapters 2 and 3), they do not address a complementary class of problems that arise specifically in single-cell experiments. scRNA-seq is a technology developed by 10x Genomics that isolates individual cells and sequences the messenger ribonucleic acid (RNA) (mRNA) within each cell [52, 53]. This allows researchers to analyze gene expression at the single-cell level, enabling the study of cellular heterogeneity, identification of distinct cell types, and insights into complex biological processes [54, 55].

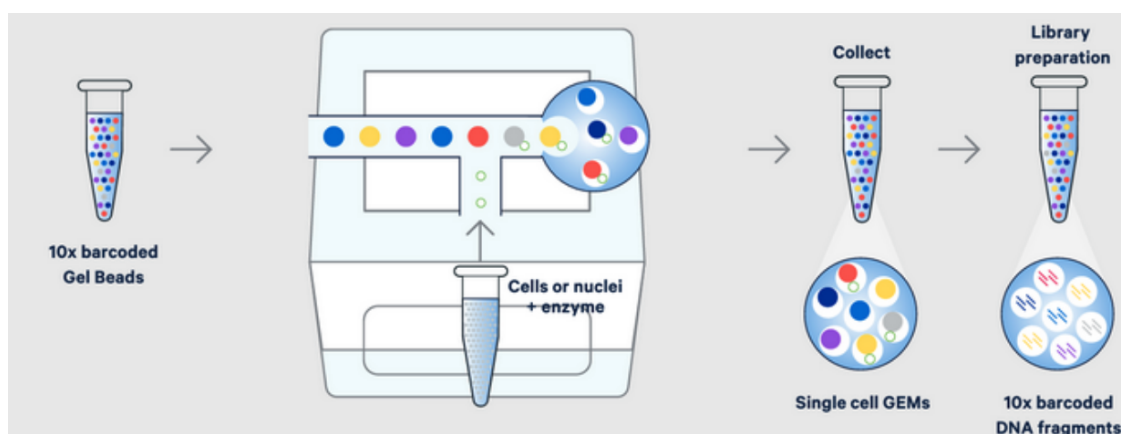


Figure 1.2: Overview of the droplet-based single-cell RNA sequencing scRNA-seq procedure, from sample preparation to sequencing library. Individual cells are encapsulated in droplets with uniquely barcoded beads, lysed within each droplet, and their messenger RNA is captured, reverse-transcribed, and tagged with cell-specific barcodes. The bar-coded cDNA from all cells is then pooled and prepared as a single sequencing library, with the cell-of-origin recoverable from the barcode after sequencing. Adapted from 10x Genomics.

While the primary purpose of scRNA-seq is to measure per-cell expression profiles and characterize cellular heterogeneity [54, 55], the same reads also incidentally capture genetic variants distributed across the transcribed regions of the genome. When multiple

individuals are pooled into a single experiment to reduce per-donor cost, minimize technical batch effects, and improve doublet [56, 57] and ambient RNA [58] detection, the natural genetic differences between individuals, visible as variant calls in the scRNA-seq reads themselves, can be used to assign each cell back to its donor of origin without requiring a separate genotyping step. Souporcell is a state-of-the-art method that achieves this by using expectation-maximization clustering on the variants observed in each cell [9]. Souporcell and similar tools [10, 11] have established the genotype-based demultiplexing paradigm, but they were primarily designed and benchmarked on pools of relatively few donors [59].

Clustering under any non-trivial loss function is an NP-hard problem in the number of data points [13, 60], and the difficulty scales further with the number of clusters: no algorithm with better runtime than $\mathcal{O}(k^n)$ is known for k clusters and n data points [61]. As the donor count increases, the variant signal available for distinguishing any single pair of donors also becomes sparser per cell. This sparsity is intrinsic to scRNA-seq: any given cell expresses only a fraction of the genome at detectable levels, and combined with limited experimental yield, only about 5% of variant loci are sampled per cell [52]. Empirically, while Souporcell can accurately demultiplex pools of up to 16 donors [9], beyond that scale existing methods tend to produce incorrectly merged clusters and low adjusted Rand index scores [62, 63] relative to ground truth. This is a problem because the economic and statistical benefits of multiplexing scale with the number of donors per pool [64]. A robust demultiplexing method that scales to higher donor counts is therefore needed to fully exploit the multiplexing capabilities of modern droplet-based platforms [65].

A related but distinct clustering problem arises in emerging spatial single-cell technologies such as the Slide-seq platform and Visium HD. Unlike the demultiplexing task discussed above, where the goal is to assign cells to their donor of origin using genotype information, the spatial clustering problem is one of identifying distinct *cell types* based on the expression patterns of imaging-resolved targeted gene panels [66]. These

platforms differ from droplet-based scRNA-seq in both the structure of their measurements (imaging-based targeted panels with spatial coordinates rather than droplet-isolated whole-transcriptome libraries) and in the analyses they enable, but they share with droplet-based scRNA-seq the underlying challenge of clustering high-dimensional, sparse expression data into biologically meaningful groups [66]. Existing clustering methods developed for whole-transcriptome scRNA-seq do not necessarily translate to the targeted, imaging-based modality of these platforms, and a separate methodological effort is needed to enable robust cell-type assignment in this setting.

The studies in Chapters 4 and 5 address these limitations and are guided by the following research questions:

1. How can genotype-based clustering be extended to robustly demultiplex pools of more than 16 donors in droplet-based scRNA-seq?
2. Which algorithmic choices in the clustering pipeline (e.g., variant selection, initialization, regularization) are responsible for the breakdown of existing methods at high donor counts?
3. How can the clustering paradigm be adapted to identify distinct cell types in spatial single-cell data, where measurements come from imaging-based targeted gene panels rather than droplet-isolated whole-transcriptome reads?

The results of Chapters 4 and 5 provide empirical evidence that, despite the popularity of expectation-maximization clustering on variant matrices for demultiplexing droplet-based scRNA-seq [9, 10], modifications to initialization [67] and clustering objectives [14, 15, 68, 69] can substantially improve the adjusted Rand index in high-donor pools relative to the existing state of the art. Furthermore, we show that a clustering approach adapted to imaging-based spatial measurements enables robust cell-type identification on platforms such as Slide-seq and Visium HD, where existing methods developed for whole-transcriptome scRNA-seq do not directly apply. We show examples in which the

proposed methods recover donor identities and cell-type assignments more accurately than the baselines when tested against real-world multiplexed and spatial datasets [70–72]. To the best of our knowledge, this is among the first works to comprehensively assess genotype-based demultiplexing at high donor counts and to extend the clustering paradigm to spatial single-cell modalities.

1.3 Dissertation Organization

This dissertation is organized as follows. Chapter 2 presents methods for quality score correction and sequence polishing of PacBio HiFi reads, including a post-consensus polishing strategy and a recalibrated method for predicting base-level quality scores from empirical error rates. The proposed methods are evaluated against publicly available HiFi datasets and reference genomes. Chapter 3 describes alternative banding strategies for partial order alignment of HiFi subreads, leveraging LCS-based seeding to substantially reduce the runtime and memory footprint of subread alignment without sacrificing accuracy. Chapter 4 presents a new method for genotype-based demultiplexing of droplet-based scRNA-seq at high donor counts, including modifications to initialization, and clustering objectives that extend the practical donor-count limit of existing tools. Chapter 5 describes a clustering approach for assigning spatial single-cell data into distinct cell types on platforms such as Slide-seq and Visium HD, where existing methods developed for whole-transcriptome scRNA-seq do not directly apply. Chapter 6 concludes the dissertation with a summary of the contributions made and notes for future research.

Chapter 2

Quality Score Correction and Sequence Polishing

This chapter presents *TopoQual*, a method for correcting per-base quality scores and polishing the consensus sequences of Pacific Biosciences HiFi reads. This method is published in BMC bioinformatics journal [2]. Section 2.1 motivates the study and describes the evidence for systematic overestimation of HiFi base quality scores. Section 2.2 provides evidence for overestimation of base quality scores of HiFi reads. Section 2.3 describes the proposed method, including the partial order alignment-based extraction of parallel bases and the deep learning model used to predict recalibrated quality scores. Section 2.4 presents the results of the proposed method on a validation dataset and compares its performance against the original PacBio quality scores and DeepConsensus. Section 2.5 discusses the implications of the results and limitations of the proposed approach.

2.1 Motivation

Pacific Biosciences recently released their HiFi sequencing technology, in which a hairpin adapter is attached to each end of a DNA molecule to form a circular construct [1, 31]. Using a strand-displacing DNA polymerase, the platform sequences the full circular construct multiple times [1]. The resulting subreads, corresponding to the forward and reverse strands, are then separated and used to construct a consensus sequence. Combined with improvements in DNA polymerase processivity, this approach allows for many independent passes over the same circular molecule [30].

Because the errors on each sequencing pass are largely independent of one another,

the accuracy of the resulting consensus sequence is theoretically limited only by the number of passes and the quality of the multiple alignment used to reconcile the subreads. This produces long (15 kb+) and highly accurate (99.9%+) reads, with some bases reaching a theoretical accuracy on the order of $1 - 10^{-9}$ or higher, assuming no upstream error sources, optimal multiple alignment, and truly independent measurements [1]. This technology has revolutionized genome assembly [33, 37, 38], structural variant analysis [36], and other aspects of genomics.

However, when PacBio estimates the quality of each base, it assigns many bases a Phred-scaled quality score [47] of 93, corresponding to a theoretical accuracy of $1 - 10^{-9.5}$. When compared against a sample with nearly perfect ground-truth knowledge, these bases only validate at approximately Q45. Therefore, the reported quality score estimates cannot be trusted at face value. The goal of this chapter is to develop a system that not only estimates these quality scores accurately, but also corrects some of the errors introduced by the consensus base calling algorithms currently in use.

2.2 Evidence for Overestimation of Quality Scores

This section presents five lines of evidence supporting the claim that HiFi base quality scores are systematically overestimated.

2.2.1 Experiment with Umbilical Cord Blood

The observation that PacBio HiFi sequences overestimate their base qualities is substantiated by our analysis of a 9-month-old umbilical cord blood sample, which should theoretically harbor minimal somatic variants due to its young age. Contrary to expectations, our results revealed a substantial number of apparent somatic variants (mismatches), suggesting the presence of errors in the sequencing data rather than true somatic mutations.

The primary dataset is a 9-month-old cord blood granulocyte sample (PD47269d) with

an average read length between 16 kb and 20 kb and approximately $30\times$ coverage. This sample is expected to contain only a small number of somatic mutations (approximately 675 variants across the genome) given the average somatic mutation rate of roughly 25 per cell per year [26, 27]. Because we treat every mismatch that is not a known germline variant as a potential error, the theoretical maximum validatable quality value, under the assumption that all observed mismatches were true somatic variants, would be Q76, computed as follows.

$$\text{Expected somatic mutations (whole genome)} = 675 \quad (2.1)$$

$$\text{Expected somatic mutations (35\% genome used)} = 236 \quad (2.2)$$

$$\text{Total bases used in testing} = 10.67 \text{ Gb} \quad (2.3)$$

$$\text{Theoretical maximum quality} = -10 \log_{10} \left(\frac{236}{10.67 \times 10^9} \right) \approx 76 \quad (2.4)$$

However, when we analyzed the actual errors and computed the empirical quality score, an effective base quality of only Q45 was obtained. This discrepancy indicates that the true confidence in the accuracy of these bases is much lower than the reported quality score, supporting the claim that HiFi sequences systematically overestimate their base qualities.

2.2.2 DeepConsensus Recalibrates Quality Scores

There is independent evidence to suggest that PacBio HiFi base quality scores are overestimated. DeepConsensus [39], a deep learning-based method designed to improve the accuracy of PacBio HiFi sequences, implicitly performs a form of quality score recalibration. In our analysis, DeepConsensus produces base quality scores that are generally lower than the original values provided by the sequencing platform, as illustrated in Figure 2.1. This indicates that the initial base quality scores in PacBio HiFi reads are often inflated and do not accurately reflect the true error rates. By providing more realistic base

quality assessments and improving the underlying sequence quality, DeepConsensus enhances the precision of downstream processes such as germline and somatic variant detection, supporting the claim that using uncorrected HiFi quality scores can lead to errors, particularly in somatic variant calling. The method proposed in this chapter, TopoQual, improves on this approach by combining algorithmic analysis of partial order alignments with deep learning over those and other signals.



Figure 2.1: Comparison of base quality scores at a mismatch reference position chr3:5,084,593 between DeepConsensus (bottom) and the original PacBio (top), as viewed in the Integrative Genomics Viewer (IGV).

2.2.3 Somatic Mutation Likelihood and Examples

Both Strelka2 [7] and Mutect2 [8] utilize base quality scores in their statistical models for somatic mutation detection. To illustrate the practical consequences of overestimated quality scores, we present a simplified somatic mutation likelihood calculation analogous to those used by these callers.

The probability of observing the data given an error is derived directly from the Phred scaling of the base quality score:

$$\text{Qual} = -10 \log_{10} P(D | E) \implies P(D | E) = 10^{-\text{Qual}/10}. \quad (2.5)$$

For multiple independent reads supporting the same base, the per-base error probabilities are multiplied. For example, two Q93 bases yield $P(D | E) = 5 \times 10^{-10} \times 5 \times 10^{-10} = 2.5 \times 10^{-19}$.

The posterior likelihood that a base is a true somatic mutation can then be expressed as

$$P(S | D) = \frac{P(D | S) P(S)}{P(D | S) P(S) + P(D | E) P(E)}, \quad (2.6)$$

where $P(S) = 10^{-6}$ is the average somatic mutation prior used by both Strelka2 [7] and Mutect2 [8], derived from the expected rate of somatic variants in an average human sample [28]; $P(E) = 1 - P(S)$ is the probability of an error, with germline mutation rates and reference bias ignored for simplicity; and $P(D | S) \approx 1$.

We present two examples in which the recalibrated quality score from TopoQual does not call the base as a somatic mutation, whereas the PacBio quality score would. A posterior cutoff threshold of 0.9 (Q10) is used.

Example 1 (chr3:5,170,774). See Figure 2.2.

- **PacBio:** Qual = 93; error rate = 5.0×10^{-10} ; $P(S | D) = 0.9995$ (Q33, PASS for somatic).

- **TopoQual:** Qual = 23; error rate = 5.0×10^{-3} ; $P(S | D) = 2.0 \times 10^{-4}$ (Q0, below threshold).

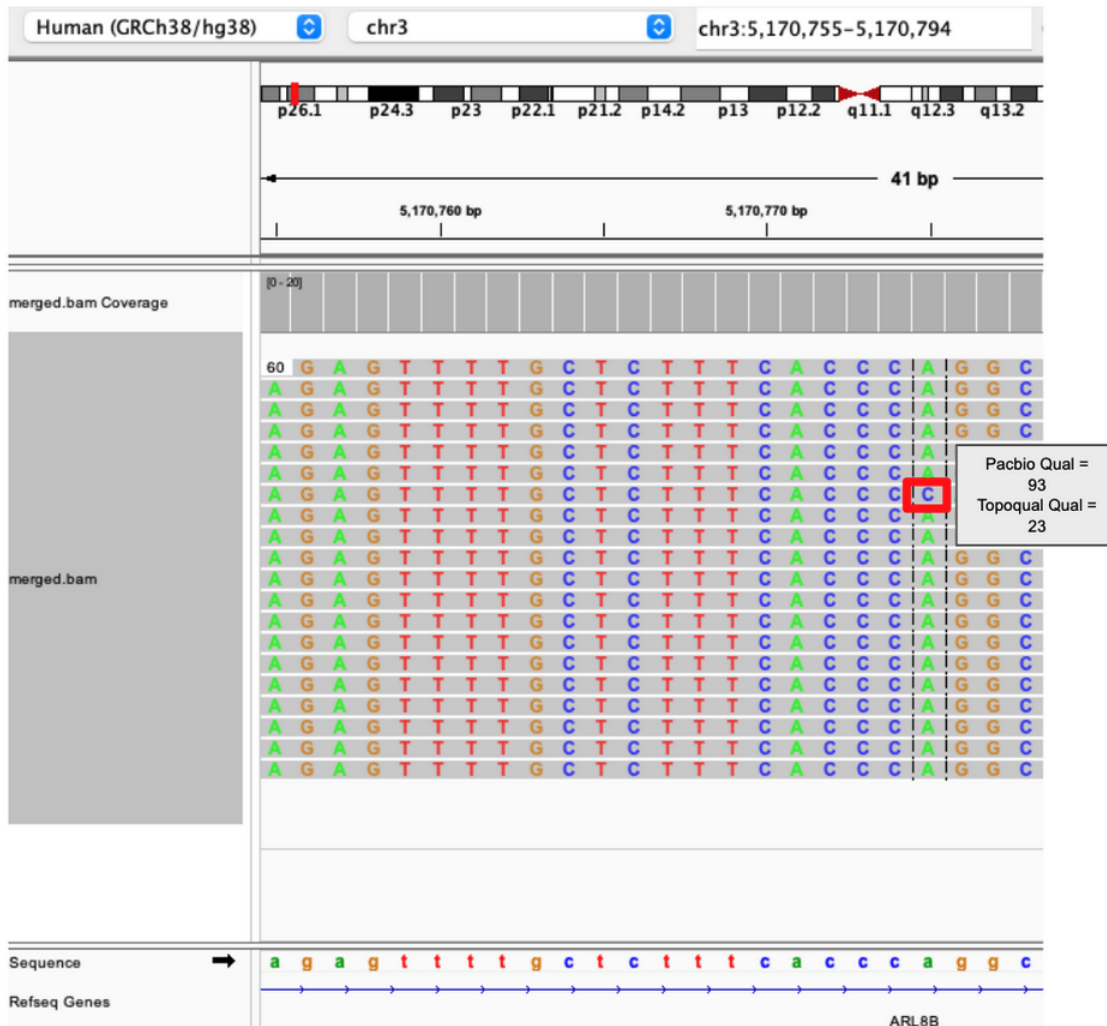


Figure 2.2: Comparison of base quality scores at a mismatch reference position chr3:5,170,774 between TopoQual and the original PacBio, as viewed in the Integrative Genomics Viewer (IGV).

Example 2 (chr21:13,220,919). See Figure 2.3.

- **PacBio:** Qual = 93; error rate = 5.0×10^{-10} ; $P(S | D) = 0.9995$ (Q33, PASS for somatic).
- **TopoQual:** Qual = 28; error rate = 1.6×10^{-3} ; $P(S | D) = 6.3 \times 10^{-4}$ (Q0, below

threshold).

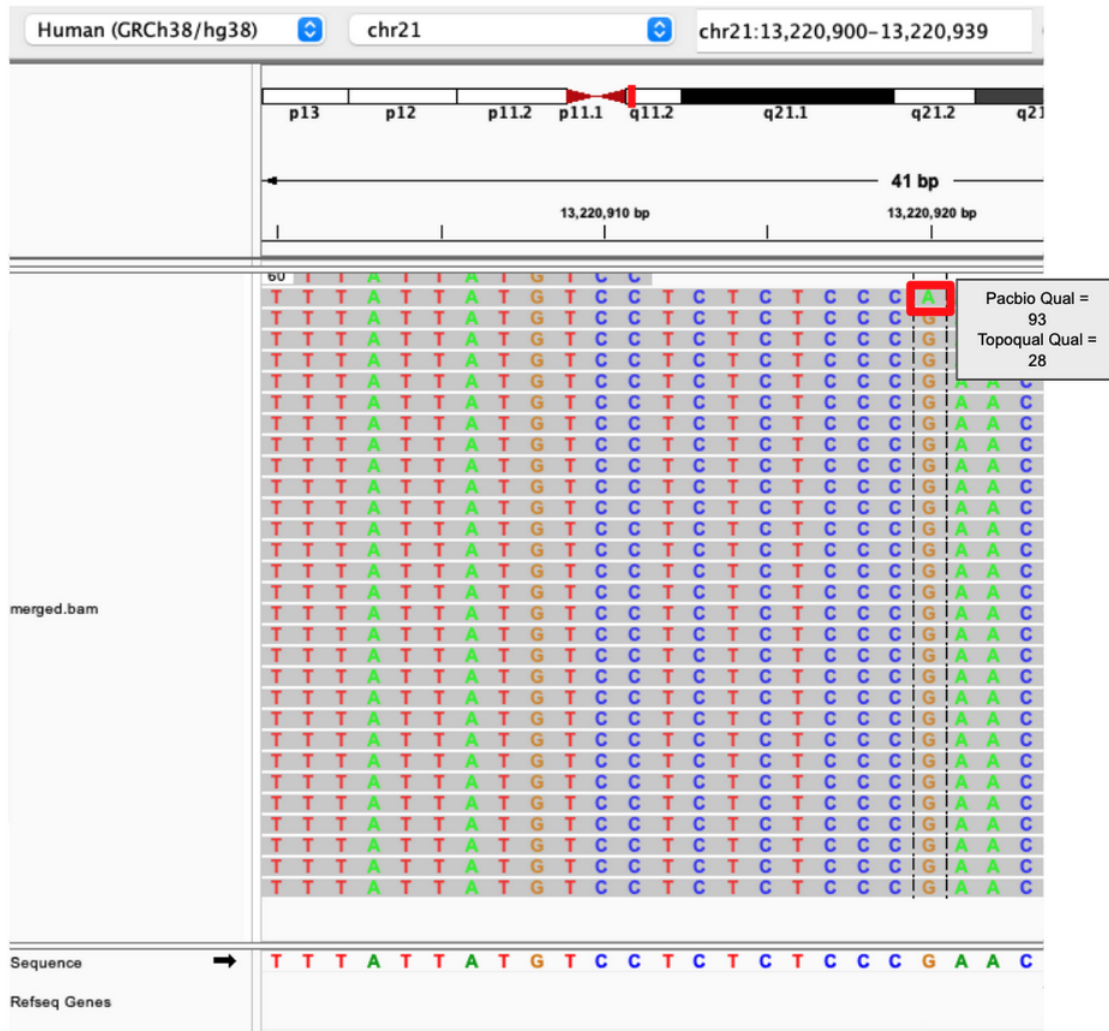


Figure 2.3: Comparison of base quality scores at a mismatch reference position chr21:13,220,919 between TopoQual and the original PacBio, as viewed in the Integrative Genomics Viewer (IGV).

In both examples, the recalibrated quality score from TopoQual indicates that the mismatch is most likely an error, while the original PacBio quality score would lead a downstream variant caller to report a false-positive somatic mutation.

2.2.4 Communication with PacBio

Through private communication with PacBio, including discussions with a collaborator employed directly at PacBio, we have confirmed that the issue of overestimated quality scores in certain regions of the genome is known and recognized internally. PacBio has acknowledged that while HiFi reads offer high accuracy in most contexts, in some specific regions, particularly repetitive or structurally complex areas of the genome [73, 74], the reported quality scores can be inflated. PacBio is aware of these limitations and has indicated that they are actively working on refining their algorithms to address the issue. This confirmation strengthens the claim that the overestimation of quality scores, while not yet widely reported in the literature, is a real and acknowledged problem in specific genomic contexts.

2.3 Method

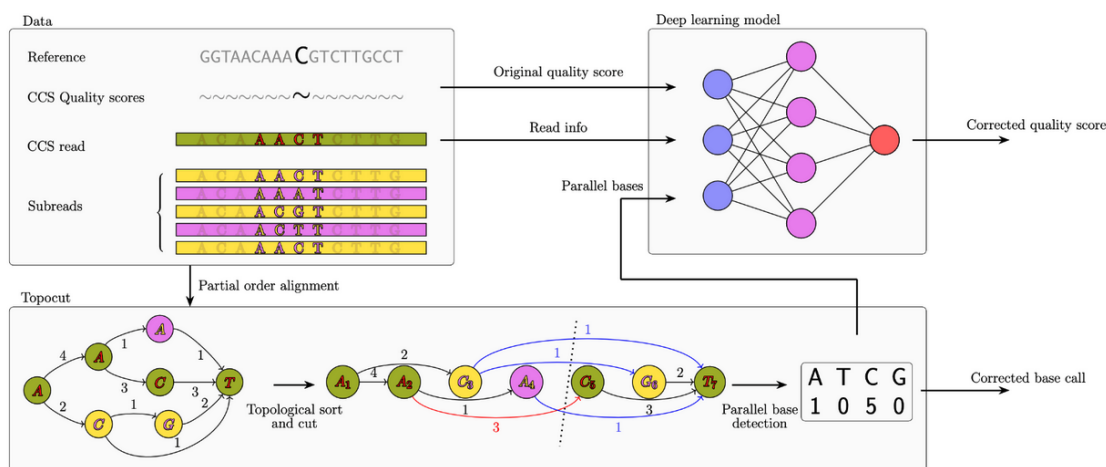


Figure 2.4: The quality score estimation strategy of TopoQual. Subreads are aligned together with the current consensus sequence using partial order alignment. Potential alternative bases in the multiple alignment are detected by identifying parallel bases in the POA graph. These signals, along with several others, are passed to a deep learning model that outputs a recalibrated quality score.

We present TopoQual, a tool for polishing HiFi consensus sequences and producing precise base quality scores through the use of *parallel* (potential alternative) bases identified in the partial order alignment of subreads. The overall workflow of TopoQual is illustrated in Figure 2.4.

To begin, TopoQual performs POA multiple alignment of the subreads together with the current consensus sequence [19]. We then use our algorithm, *TopoCut* (described below), to identify the parallel bases of the calling base in the POA graph. These parallel bases from TopoCut are used to correct the original base call if an alternate base has a higher count than the original base. Additionally, the parallel bases, in conjunction with the trinucleotide sequence context of the read and the target base's reported quality score, are passed as input to a deep learning model which outputs a corrected quality score.

During training, the deep learning model treats mismatched bases that are not known germline variants as errors, because the number of true somatic mutations in the umbilical cord blood data is expected to be much smaller than the number of observed mismatches. While a high-quality reference genome is necessary for the training of this model, it is not required at inference time: new datasets can be polished and recalibrated using only the subreads.

2.3.1 Partial Order Alignment and Parallel Bases

The partial order alignment data structure [19] is a graph that captures rich details about the structure of aligned sequences, allowing analysis of alternate pathways from the target base's path. We define these alternate pathways as *parallel bases*. We use POA because it guarantees the optimal alignment of a new sequence with respect to the set of sequences already in the graph. POA works by extending standard dynamic programming sequence alignment [17, 18] to operate on partial order graphs, adding a new sequence to the graph at each step.

2.3.2 TopoCut: Identifying Parallel Bases

TopoCut is the algorithm used by TopoQual to extract parallel bases from the partial order alignment graph. To accurately find the parallel bases, TopoCut first performs POA multiple sequence alignment using the circular consensus sequencing (CCS) read and its associated subreads. This yields a partially ordered graph in which sequence letters are represented by nodes and the number of agreeing sequences at each step is represented by edge weights. The nodes are then sorted in topological order and ranked accordingly. In this sorted graph, TopoCut makes a cut immediately in front of the calling base and identifies the edges that intersect this cut; these edges define the parallel bases.

For example, consider the calling base *C* with topological rank 5 in Figure 2.4. First, the parent edge weight of the calling base *C* is added to the parallel base count. Then, parent–child rank pairs that sandwich the calling base *C* are identified ($3 \rightarrow 6$, $3 \rightarrow 7$, $4 \rightarrow 7$), and their corresponding edge weights are added to the parallel base count, yielding the final parallel base distribution [$A = 1$, $C = 5$, $G = 0$, $T = 0$]. The total parallel base count is 6, which agrees with the total number of sequences, so no further action is required.

Algorithm 1: Identify parallel bases.

Require:

Partial order graph G
The calling base C_b
Num of sequences Num

- 1: $F \rightarrow [0,0,0,0]$ // Forward cut
- 2: ADD C_b 's parent edge weight to F
- 3: **for** $k \rightarrow$ all parent-child pairs in G not containing C_b **do**
- 4: **if** $k.child\ rank \geq C_b\ rank \ \&\& \ k.parent\ rank \leq C_b\ rank$ **then**
- 5: ADD $k.parent$ edge weight to F
- 6: **end if**
- 7: **end for**
- 8: **if** $F\ count == Num$ **then**
- 9: **return** F
- 10: **end if**
- 11: $B \rightarrow [0,0,0,0]$
- 12: **for** $k \rightarrow$ all parent-child pairs in G not containing C_b **do**
- 13: **if** $k.child\ rank \geq C_b\ rank \ \&\& \ k.parent\ rank \leq C_b\ rank$ **then**
- 14: ADD $k.child$ edge weight to B
- 15: **end if**
- 16: **end for**
- 17: **if** $B\ count > F\ count$ **then**
- 18: **return** B
- 19: **end if**
- 20: **return** F

Figure 2.5: Topoqual Algorithm

Algorithm. The TopoCut_Identify_Parallel_Bases routine accomplishes the above by adding the calling base's weight in the graph to the parallel bases array, then adding the corresponding parent edge weights for any parent-child rank pairs that sandwich the calling base's rank. If the count of parallel bases does not sum to the total number of sequences, the process is repeated in reverse to capture any remaining contributions.

2.3.3 Deep Learning Model

The deep learning model is the core of TopoQual: it takes in various features computed from the read and outputs a recalibrated quality score. The inputs to the model include the trinucleotide sequence context of the read, the PacBio CCS quality score, the parallel base distribution from TopoCut, the average inter-pulse duration, the average pulse width,

and the signal-to-noise ratio of the bases [31]. During the training phase, the model is supervised with labels of 0 for reference-mismatched bases and 1 otherwise. To prevent overfitting, dropout [75] is applied between hidden layers, and the model is optimized using Adam [76].

2.4 Results

2.4.1 Sequence Polishing

TopoQual performs partial order alignment using PacBio CCS reads and their sub-reads to obtain parallel bases. Within this process, various mismatches with the reference genome are corrected using three distinct techniques: (i) the parallel base distribution prefers a different base, (ii) the POA consensus deletes a base, and (iii) the POA consensus substitutes a base. Figure 2.6 illustrates the polishing of T→X mutations stratified by the three-base sequence context.

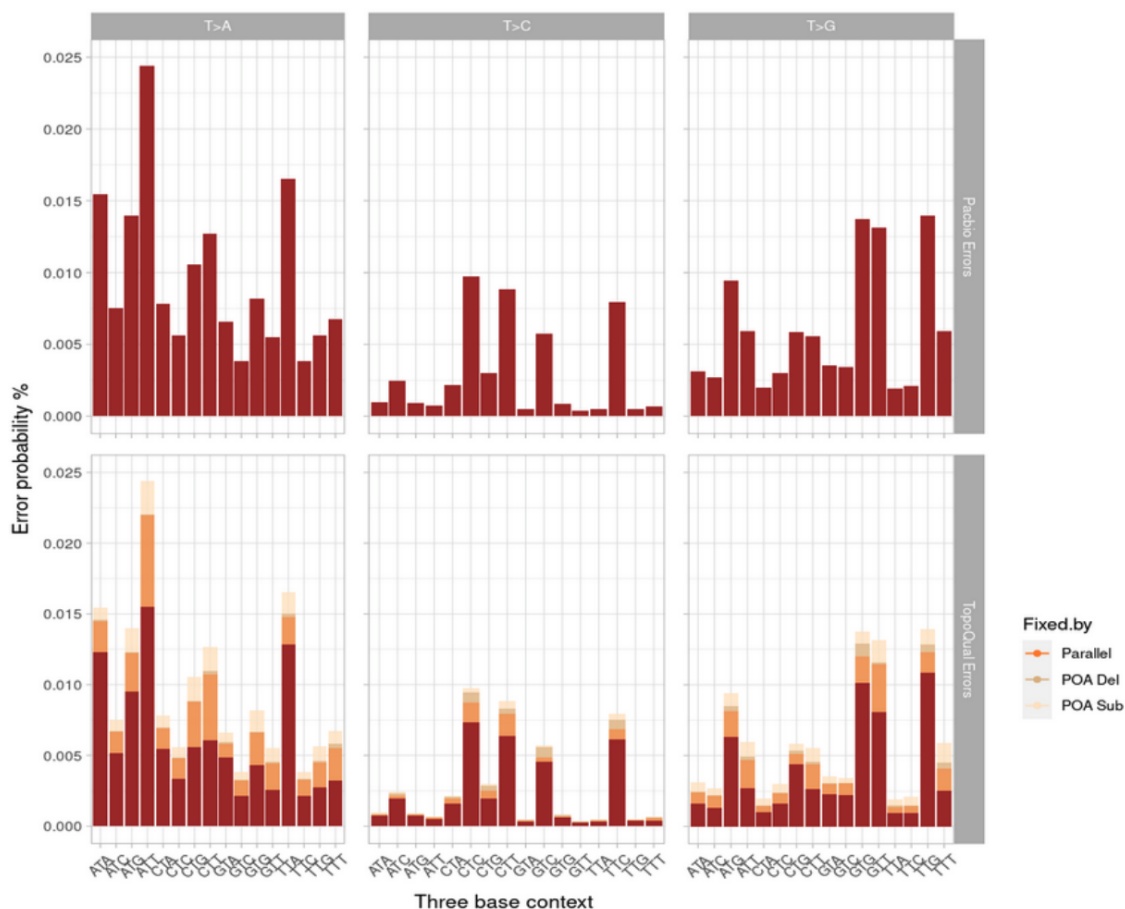


Figure 2.6: Errors present before (top) and after polishing (bottom) by TopoQual in the validation dataset for T to X mutations, stratified by the three correction types (parallel bases prefer a different base, POA consensus deletes a base, and POA consensus substitutes a base) on chromosome 2.

The sensitivity and specificity of sequence polishing are 31.9% and 99.6%, respectively. The percentages of errors corrected at each step are summarized in Table 2.1.

Table 2.1: Percentage of errors corrected by TopoQual using each technique. Total bases analyzed: 11Gb; errors present: 1.8Mb.

Polished by method	Number of bases polished (% of errors)
POA substitution	166 Kb (9.0%)
POA deletion	66 Kb (3.5%)
Parallel bases	361 Kb (19.5%)
TopoQual (total)	594 Kb (31.9%)

Table 2.2 compares the error rates and maximum validated quality scores (Max Q) of PacBio, DeepConsensus [39], and TopoQual across seven chromosomes (chr2, chr3, chr4, chr18, chr19, chr20, and chr21). For each chromosome, TopoQual consistently exhibits lower error rates and, in most cases, higher Max Q scores than the other methods. Specifically, TopoQual achieves a higher Max Q for all chromosomes except chr4 and produces approximately 0.06% lower error rates across all chromosomes. These improvements demonstrate TopoQual’s superior performance in reducing errors and producing better-calibrated base quality scores compared to PacBio and DeepConsensus.

Table 2.2: Results of different methods on the validation dataset, average subread depth = 10.

Chr	PacBio		DeepConsensus		TopoQual	
	Err/Total	Max Q	Err/Total	Max Q	Err/Total	Max Q
chr2	562K/3.3G	46	585K/3.8G	49	263K/2.7G	54
chr3	431K/2.6G	47	530K/3.1G	45	283K/2.6G	52
chr4	379K/2.4G	48	431K/2.7G	48	246K/2.4G	47
chr18	149K/0.96G	49	178K/1.1G	47	96K/0.96G	54
chr19	144K/0.74G	44	166K/0.9G	44	100K/0.74G	48
chr20	126K/0.78G	49	145K/0.92G	50	82K/0.78G	59
chr21	63K/0.37G	46	85K/0.52G	47	42K/0.37G	51

2.4.2 Comparison of Quality Scores Against PacBio and DeepConsensus

PacBio and DeepConsensus both report Phred-scaled quality score outputs [1, 39] ranging from 1 to 93, corresponding to base call accuracies of 20% to 99.99999995%. Consensus and polishing algorithms aim both to identify the correct base and to assign an accurate assessment of the likelihood of that base being erroneous. To evaluate calibration, we count mismatches to the reference genome that are not known germline variants as errors. This procedure overlooks true somatic mutations and counts them as errors; however, because the validation dataset is derived from umbilical cord blood, the number of true somatic mutations is much smaller than the number of observed mismatches (675 versus 524,575). This yields a theoretical maximum validatable quality of approximately Q80 if the only observed mismatches were somatic mutations (see the supplementary derivation in Section 2.2.1).

Figure 2.7 illustrates the algorithm-reported base quality scores (x-axis) compared to the corresponding empirical base qualities computed from mismatches in chromosome 2 of the validation dataset (y-axis). The two marginal plots show the density distribution of

base counts at each quality value.

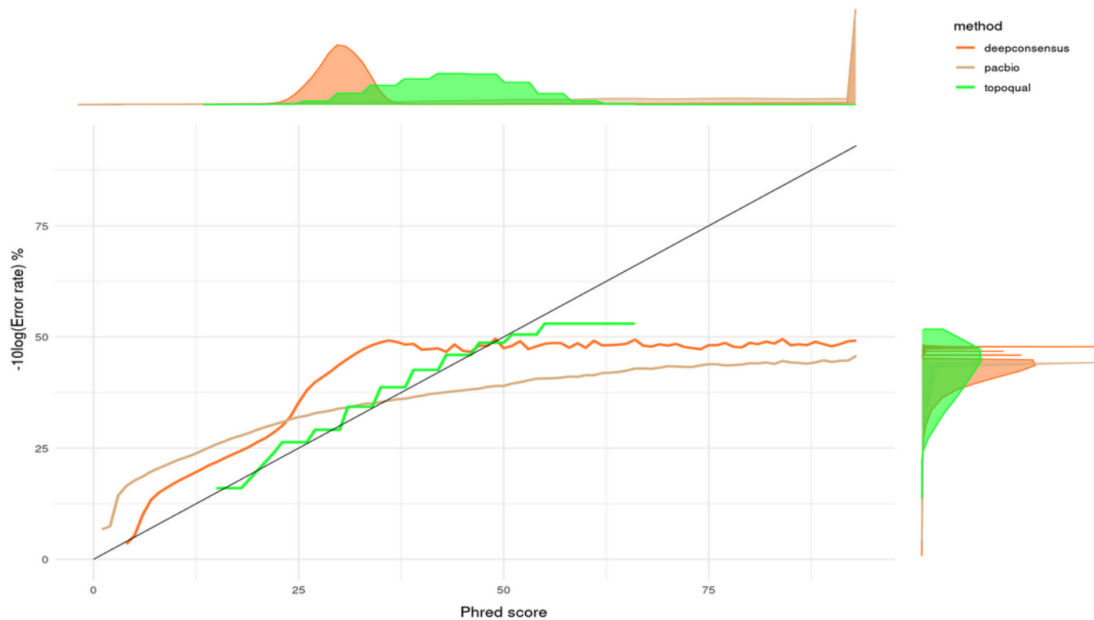


Figure 2.7: Quality score validation comparison of different methods on chromosome 2. Well-calibrated quality scores are expected to fall on the diagonal. Marginal plots show the distribution of base counts.

At lower quality levels, both PacBio and DeepConsensus exhibit fewer errors than anticipated, but at higher quality levels both methods surpass the expected error rates. PacBio reaches a maximum validated quality of 46, while DeepConsensus achieves 49. TopoQual produces quality scores that align with the actual observed error rates at both lower and higher quality levels, reaching a maximum validated quality of 54.

The distribution of quality scores in PacBio and DeepConsensus is heavily concentrated near the maximum value of 93, even though the actual measured quality is well below this value. TopoQual more accurately measures the validated quality scores, which are roughly normally distributed as expected. Despite the broader range of quality scores produced by TopoQual, the count of high-quality ($> Q45$) bases is equivalent to that of PacBio to within $\pm 1\%$.

2.5 Discussion

The results presented in this chapter demonstrate that the per-base quality scores produced by PacBio HiFi sequencing are systematically overestimated, particularly at the high end of the Phred scale, and that this overestimation has practical consequences for downstream variant calling. The proposed method, TopoQual, addresses this problem by combining a topological analysis of the partial order alignment graph (TopoCut) with a deep learning model trained against empirical error rates. The result is a recalibrated quality score that tracks validated error rates more closely than either the original PacBio scores or those produced by DeepConsensus, while simultaneously polishing approximately 32% of the errors present in the input CCS reads.

These improvements have direct implications for somatic variant calling, where the inflation of base quality scores can lead to false-positive variant calls (Section 2.2.3). By providing better-calibrated inputs to variant callers such as Strelka2 [7], Mutect2 [8], and the more recent long-read-specific callers ClairS [40] and DeepSomatic [41], TopoQual stands to reduce the false-positive rate of somatic variant detection in clinical and research applications that rely on HiFi data [4, 27].

Several limitations of the current method merit discussion. First, TopoQual requires a high-quality reference genome for training, which constrains the populations and species to which the trained model can be confidently applied. Second, the polishing component currently corrects only a fraction (31.9%) of the errors present in the input data; further improvements may be possible by combining TopoQual with complementary polishing approaches such as Racon [43] or by integrating signal-level features more deeply into the model. Third, the validation strategy treats all non-germline mismatches as errors, which slightly underestimates the true achievable quality given the presence of true somatic mutations in the cord blood sample. Future work should explore evaluation strategies that explicitly distinguish errors from low-frequency somatic variants, for example by using duplex sequencing [28, 77] as an orthogonal ground truth.

The next chapter addresses a complementary bottleneck in the HiFi pipeline: the computational cost of the partial order alignment step that underlies both consensus generation and the TopoQual method described here. By introducing alternative banding strategies for POA based on LCS variants, Chapter 3 aims to make accurate subread alignment tractable on long reads at scale.

Chapter 3

Speeding Up Partial Order Alignment with Banding

This chapter presents *LCSkPOA*, a banding strategy for partial order alignment that extends the Longest Common Subsequence with k -mer matches (LCSk++) algorithm to graph-based alignment. This method is published in BMC bioinformatics journal [3]. Section 3.1 motivates the need for efficient banded POA on long reads and reviews the limitations of existing tools. Section 3.2 describes the proposed algorithm, including its dynamic programming formulation, an efficient event-based implementation, and the data structures used to achieve its time complexity. Section 3.3 presents results on both synthetic and real PacBio HiFi data, comparing LCSkPOA against SPOA [43] and ABPOA [42] in terms of speedup, memory usage, and alignment accuracy. Section 3.4 concludes the chapter with a discussion of the implications and limitations of the proposed approach.

3.1 Motivation

Graph-based aligners, such as GraphAligner [78], have become increasingly important in genomics and bioinformatics due to their ability to provide a more flexible and comprehensive representation of genetic variation compared to linear sequence-based reference mapping techniques such as minimap2 [45], HISAT-3N [79], and BWA [44]. These graph-based reference alignment techniques decrease reference bias and enhance the accuracy of variant detection and genome assembly [33, 35]. However, despite these advantages, graph-based reference alignment is substantially slower than linear sequence-based methods. Because the creation of graph-based references depends on multiple

sequence alignment, improving the speed and quality of multiple sequence alignment will help reduce the performance gap between graph-based and linear-reference approaches. While multiple sequence alignment is a well-established problem with a long history of algorithmic work [17–19], challenges remain when scaling to the long reads produced by modern sequencing platforms [1].

This chapter introduces a novel adaptation of the Longest Common Subsequence with k -mer matches (LCSk++) algorithm [48] specifically designed for graph structures, with a focus on enabling banded local and semi-global POA [19]. POA graphs are directed acyclic graphs that represent multiple sequence alignments, capturing the relationships between aligned sequences and accommodating variations such as insertions, deletions, and substitutions more flexibly and comprehensively than traditional pairwise alignment methods [19]. Existing methods such as ABPOA [42] and SPOA [43] improve POA performance by using greedy adaptive banding and by computing the full alignment matrix with SIMD optimizations, respectively. However, ABPOA’s banding method does not support local and semi-global alignment, which are crucial for many use cases, while SPOA, which calculates the full alignment matrix, suffers from poor scalability on long reads. The approach proposed here extends the LCSk++ algorithm [48, 49] from the sequence-to-sequence setting [50, 80] to the graph-based setting, enabling efficient banded semi-global POA for long reads.

3.2 Implementation

This section develops the proposed algorithm in three stages: a basic dynamic programming formulation that establishes correctness but is too slow to be practical, an efficient event-based reformulation that achieves a significant asymptotic speedup, and a description of the data structures and parallelization strategies used in the implementation.

3.2.1 Basic Dynamic Programming

Finding the optimal graph-to-sequence LCSk++ path can be expressed using a basic dynamic programming recurrence similar to that of Pavetić et al. [49], with modifications to handle the graph structure. We adopt the definitions presented by Benson et al. [81], augmented with additional notation to accommodate the partial order graph:

$$dp(i, j) = \max \begin{cases} 0 \\ dp(i, j-1) \\ dp(p_{i,1}, j) \\ dp(p_{i,q-1}, j-q+1) + q \quad \text{for all } q \geq k \text{ s.t. } G_{p_{i,q-1} \dots p_{i,0}} = Y_{j-q+1 \dots j} \end{cases} \quad (3.1)$$

where $dp(i, j)$ is the dynamic programming table entry at column i and row j , with i indexing the i th topologically ordered node in graph G and j indexing the j th position in query Y . The notation $p_{i,m}$ denotes the predecessor node m edges back from the i th topologically ordered node in G .

Figure 3.1 shows an example where this equation is applied to each cell of the matrix, starting from the top-left cell $dp(0, 0)$. The alignment scores are computed row by row until the bottom-right cell is reached. At $dp(3, 2)$, the first 2-mer (“AC”) is encountered, and the score is updated as $dp(3, 2) = dp(1, 1) + 2 = 0 + 2$. The final cell evaluates to $dp(7, 4) = dp(6, 3) + 2 = dp(3, 2) + 4 = dp(1, 1) + 6 = 6$.

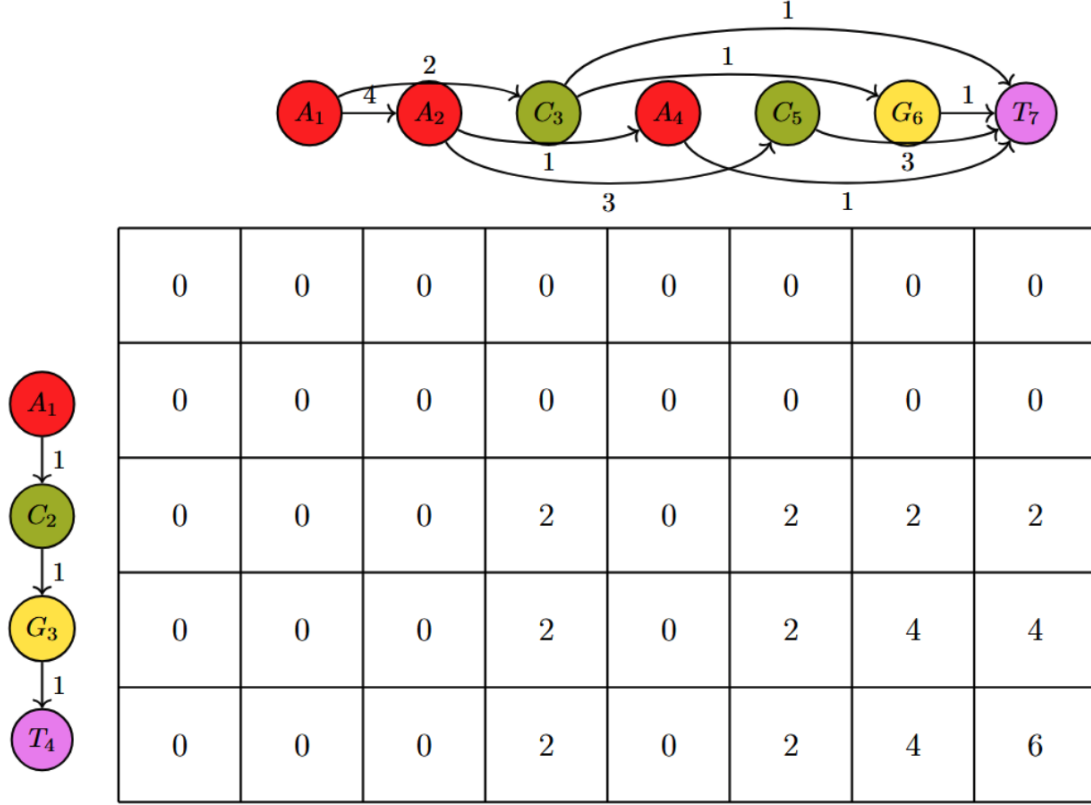


Figure 3.1: Basic dynamic programming example with $k = 2$. Each cell shows $dp(i, j)$, the LCSk++ alignment score for the i th column and j th row.

Because the algorithm must compute each cell in the alignment matrix of size $L \times M$, and for each cell may need to look back up to $\min(L, M)$ positions to find $k+$ length matches, the resulting time complexity is $\mathcal{O}(LM \min(L, M))$, where L is the query length and M is the number of nodes in the graph. Performing basic dynamic programming to obtain the LCSk++ path for banded POA is therefore not a viable solution: the time complexity of basic dynamic programming for LCSk++ alone is equivalent to that of full POA, defeating the purpose of using LCSk++ as a banding seed.

3.2.2 Efficient Algorithm

An efficient algorithm presented by Pavetić et al. [48] achieves a significant speedup over the basic dynamic programming formulation and is a crucial component in banded

sequence-to-sequence dynamic programming alignment [82]. We propose a query-to-graph version that achieves similar improvements for partial order alignment. The definition of a match pair is adapted from the Longest Common Subsequence in k -length substrings [81], with modifications to accommodate the graph-based structure.

Definitions. Let $s_{i,m}$ denote the successor node m edges forward of the i th topologically ordered node in graph G . We define:

$$\text{kMatch}(i, j) = \begin{cases} 1 & \text{if } G_{s_{i,f}} = Y_{j+f} \text{ for every } 0 \leq f \leq k-1 \\ 0 & \text{otherwise} \end{cases} \quad (3.2)$$

If $\text{kMatch}(i, j) = 1$, we call (i, j) a *match pair*, where (i, j) is the *start* of the match pair and $(s_{i,k-1}, j + k - 1)$ is the *end* of the match pair. This definition differs from that in Benson et al. [81], where using $(s_{i,k}, j + k)$ could imply multiple possible ends for a match pair. For example, in Figure 3.2, applying the original definition to the match pair for “a” could result in two different end values: $(5, 5)$ and $(6, 5)$. To resolve this ambiguity, we adjust the algorithm to accommodate the modified definition.

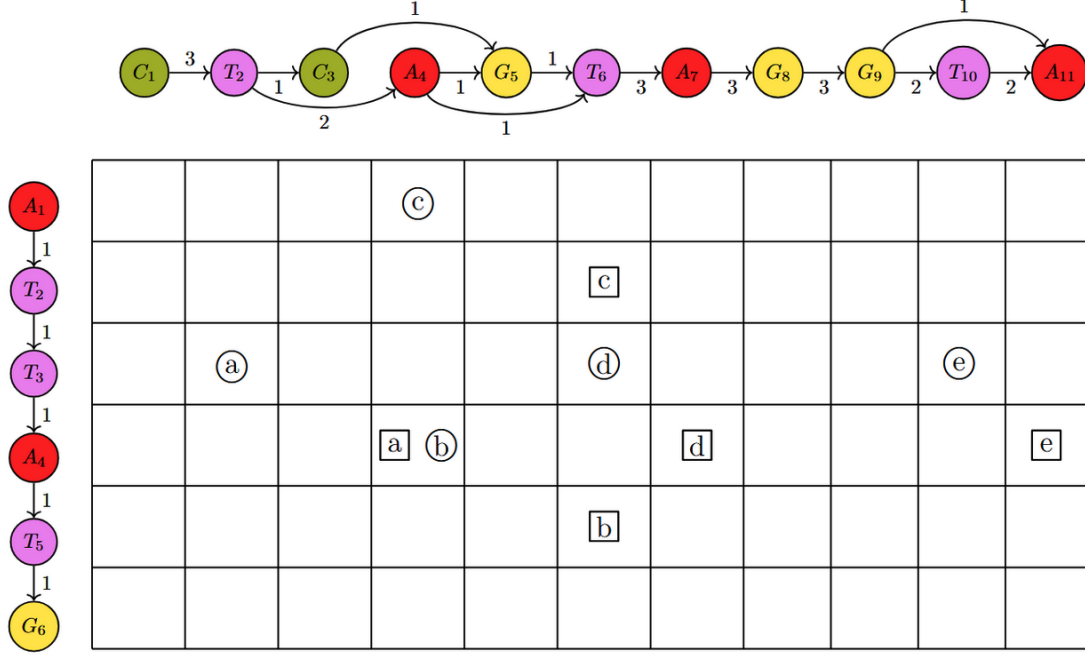


Figure 3.2: Matching k -mers between the query and the POA graph with $k = 2$. Five match pairs are present, denoted a through e. Starts are shown as circles and ends as squares. In this example, “b continues a” and “c precedes e” both hold.

The recurrence for the efficient algorithm is:

$$dp(P) = \max \begin{cases} k \\ \max_Q dp(Q) + k & \text{over all } Q \text{ preceding } P \text{ with a path available} \\ dp(Q) + 1 & \text{if } P \text{ continues } Q \text{ and shares the same path} \end{cases} \quad (3.3)$$

That is, a k -mer match pair P can start its own k -common subspace, extend a k -common subspace if a path is available from the start of P to the end of Q , or extend a k -common subsequence by one position if the continuing nodes lie on the same path.

In the example shown in Figure 3.2, once the start and end positions of the k -mers are identified, the equation $dp(P)$ is applied to each end position. The computation proceeds row by row, starting from the top-left corner. The end of c is reached first, where $dp(c\text{-end}) = k = 2$ since no preceding k -mer exists. Next, the end of a is processed, yielding $dp(a\text{-end}) = k = 2$. The end of d follows, with $dp(d\text{-end}) = k = 2$ as no prior k -mer is

found. Subsequently, the end of e is reached, where $dp(e\text{-end}) = dp(c\text{-end}) + k = 2 + 2 = 4$ since c precedes e . Finally, the end of b is processed, with $dp(b\text{-end}) = dp(a\text{-end}) + 1 = 3$ since b continues a . The resulting LCSk++ path is $[e\text{-end}, c\text{-end}]$, containing two k -mers with a total score of 4.

3.2.3 Algorithm Implementation

Algorithm 1 Sequence to graph LCSK++ computation

Require:

```

    Query  $X$ 
    Partial order graph
    Sequences  $Seq_{i \in 0..N}$ 
1:  $Seq\_paths \leftarrow$  find the  $Seq_{i \in 0..N}$  paths in graph
2:  $MaxColDp \leftarrow$  2D array filled with L zeros * N, where L is the sequence
   length
3:  $MatchPairs \leftarrow$  find all  $k$ -match pairs between  $X$  and  $Seq_{i \in 0..N}$ , save the
   corresponding paths  $paths[]$  and graph nodes  $j$ 
4:  $events \leftarrow$  all starts, ends, and paths of Match Pairs sorted in row-major
   order, if some start  $S = (i_S, j_S, paths[]_S)$  and some end  $E = (i_E, j_E,$ 
    $paths[]_E)$  share the same indices, E should come first
5: for all event  $\epsilon$  events do
6:   if event is a start  $P = (i_p, j_p, paths[]_p)$  then
7:     for for all path  $\epsilon$   $paths[]_p$  do do
8:        $dp(P) \leftarrow k + \max_{x \in 0..j_P} MaxColDp[path][x]$ 
9:     end for
10:  else if event is an end  $P = (i_p + k, j_p + k, paths[]_p)$  then
11:    if  $\exists Q$  s.t P continues Q and share same path then
12:       $dp(P) \leftarrow \max dp(P), dp(Q) + 1$ 
13:    end if
14:    for for all path  $\epsilon$   $paths[]_p$  do do
15:       $MaxColDp[path][j_p + k] \leftarrow \max MaxColDp[path][j_p + k], dp(P)$ 
16:    end for
17:  end if
18: end for
19: return  $\max_p dp(p)$ 

```

The following variables and notations are used throughout the algorithm description and complexity analysis:

- V : the number of vertices in the graph;

- E : the number of edges in the graph;
- N : the number of sequences aligned to the graph;
- L : the sequence (query) length;
- r : the number of k -mer matches between the query and the aligned sequences.

The algorithm obtains the maximum k -mer path of a query sequence X against an existing POA graph. The high-level process is illustrated in Figure 3.2. The inputs are the query sequence X , the current POA graph (comprising the previously aligned sequences), and the sequences aligned to the graph. The output is the LCSk++ path with the maximum number of k -mers. The algorithm proceeds as follows.

Line 1. Using a depth-first search (depth-first search (DFS)), the sequence paths in the POA graph are identified. The DFS leverages the previously aligned sequences (N) to ensure that all relevant paths are accounted for.

Line 2. A two-dimensional array of dimensions $N \times L$ is initialized. This array stores the maximum column values for each node in the graph, corresponding to each sequence path.

Line 3. All k -mer match pairs between the query sequence X and the graph paths are identified. This step hashes the query sequence and identifies corresponding matches in each graph path.

Line 4. The k -mer matches are sorted in row-major order. For each match, the algorithm records relevant details such as start and end positions, paths, and the corresponding graph nodes. These details are encapsulated in an event structure that serves as the basis for subsequent processing.

Line 5. The algorithm processes events one by one to update alignment scores and column values.

Lines 6–7. If the event represents the start of a match pair, the $d(P)$ value for the current path is updated. The value is computed as the maximum column value for the current path plus k .

Lines 11–13. If the event continues from a previous event, the $d(P)$ value is updated using $d(P) = d(Q) + 1$, where Q is the predecessor match pair.

Lines 14–16. After processing an event, the maximum column values for all paths are updated in the two-dimensional array.

Line 19. The `MaxColDP` array is returned; this includes the max k -mer path.

Complexity analysis. The complexity of the constituent operations can be analyzed as follows. Finding sequence paths in the graph has a time complexity of $\mathcal{O}(N(V + E))$. Finding match pairs involves hashing the query in $\mathcal{O}(L)$ and matching in $\mathcal{O}(NL)$, for a total of $\mathcal{O}(NL)$. Sorting events has a complexity of $\mathcal{O}(r \log r)$. Lines 7 through 15 are handled by N Fenwick trees [82], which operate in $\mathcal{O}(N \log L)$ per update; processing all events therefore takes $\mathcal{O}(rN \log L)$. The total time complexity is

$$\mathcal{O}(N(V + E) + NL + r \log r + rN \log L), \quad (3.4)$$

and the total space complexity is $\mathcal{O}(r)$. By integrating sequence-finding in the graph using DFS while recording nodes during the POA, the time complexity can be further reduced to

$$\mathcal{O}(NL + r \log r + rN \log L), \quad (3.5)$$

significantly enhancing performance when LCSkPOA is used to compute a backbone for downstream POA.

3.3 Results

3.3.1 System Configuration

All experiments were conducted on the following hardware and software configuration:

- **CPU:** AMD EPYC 7313 16-core processor (64 threads);
- **Memory:** 420 GB DDR4;
- **Language:** Rust 1.65.0;
- **Libraries:** Rust-bio [82] for the POA backbone and `std::thread` for the parallel processing implementation.

3.3.2 Datasets

Two datasets were used in this evaluation. The synthetic dataset was generated by PBSIM3 [51] using a reference sequence and default parameters. PBSIM3 introduces variations such as insertions, deletions, and substitutions, simulating realistic genetic changes for testing alignment methods. The PacBio dataset was obtained from real sequenced reads, the same dataset used in the evaluation of TopoQual in Chapter 2 [2]. All tests were performed using global alignment.

3.3.3 Results on Synthetic Data

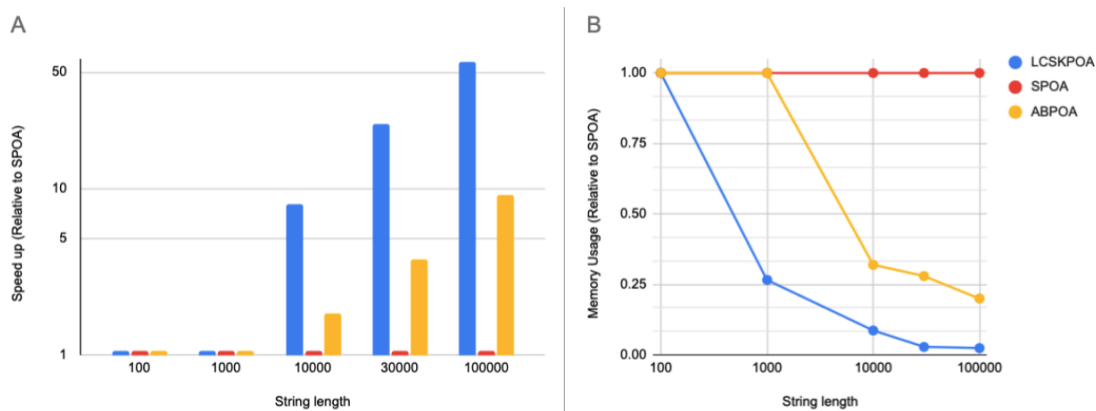


Figure 3.3: Results of running LCSkPOA on synthetic data. (A) Speedup of ABPOA and LCSkPOA compared to unbanded SPOA. (B) Memory usage of banded methods ABPOA and LCSkPOA compared to unbanded SPOA.

Figure 3.3A shows the speedup factors of ABPOA [42] and LCSkPOA compared to unbanded SPOA [43] across different string lengths. LCSkPOA delivers substantial speed improvements, reaching up to $60\times$ for the longest sequences tested. Although ABPOA performs well for short sequences, its speedup diminishes significantly for longer sequences. Figure 3.3B illustrates the memory usage of the same methods relative to SPOA. LCSkPOA consumes only a fraction of the memory used by SPOA for long strings. In contrast, ABPOA initially uses high memory but reduces consumption substantially for longer strings due to its banding. However, LCSkPOA still maintains memory usage roughly ten-fold lower than ABPOA at string lengths of 30,000 and 100,000. These results highlight the efficiency of the LCSkPOA algorithm, particularly with parallel processing on long sequences, in providing substantial speedup while maintaining low memory usage, making it a robust solution for alignment in POA graphs.

3.3.4 PacBio HiFi Data versus Synthetic Data

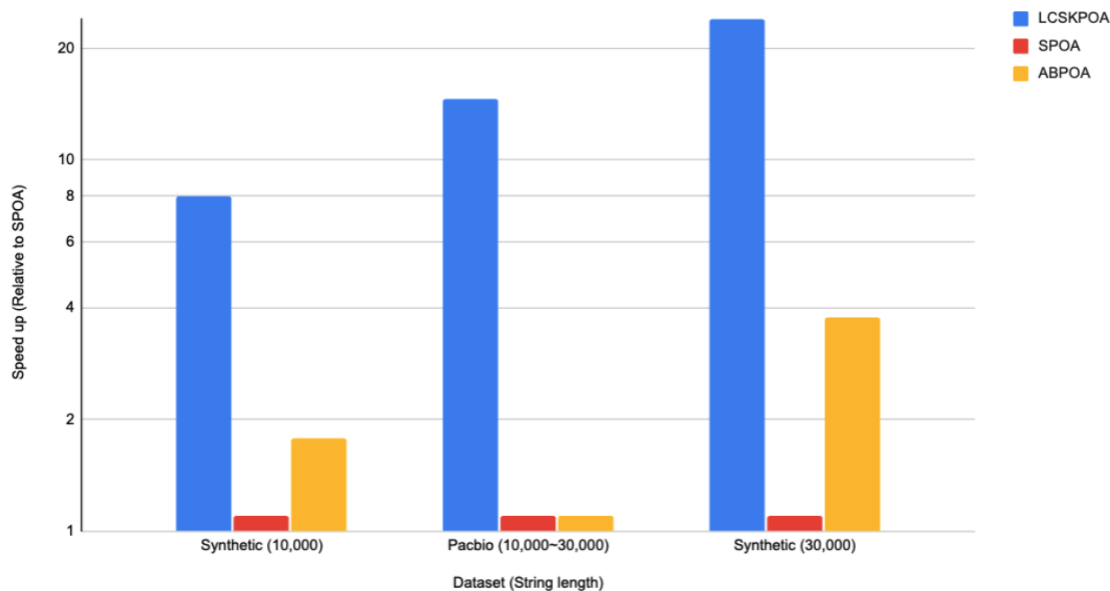


Figure 3.4: Comparison of speedup of ABPOA and LCSkPOA when run on the synthetic dataset and the PacBio HiFi dataset, both compared against unbanded SPOA.

Figure 3.4 shows that single-threaded LCSkPOA exhibits a linear speedup pattern from synthetic datasets of length 10,000 through PacBio HiFi datasets of length 10,000 to 30,000 and on to synthetic datasets of length 30,000. ABPOA does not achieve this linear improvement on the PacBio HiFi dataset. This discrepancy is attributable to ABPOA reverting to its fully banded SIMD implementation, as evidenced by its high memory usage, possibly because alignments go out of the adaptive band. As a result, ABPOA is less effective on PacBio HiFi datasets compared to synthetic datasets of comparable length. By contrast, LCSkPOA's seed-based banding strategy generalizes consistently from synthetic to real HiFi data, demonstrating its robustness to the indel patterns that characterize single-molecule long reads [1, 2].

3.3.5 Parallel Implementation Results

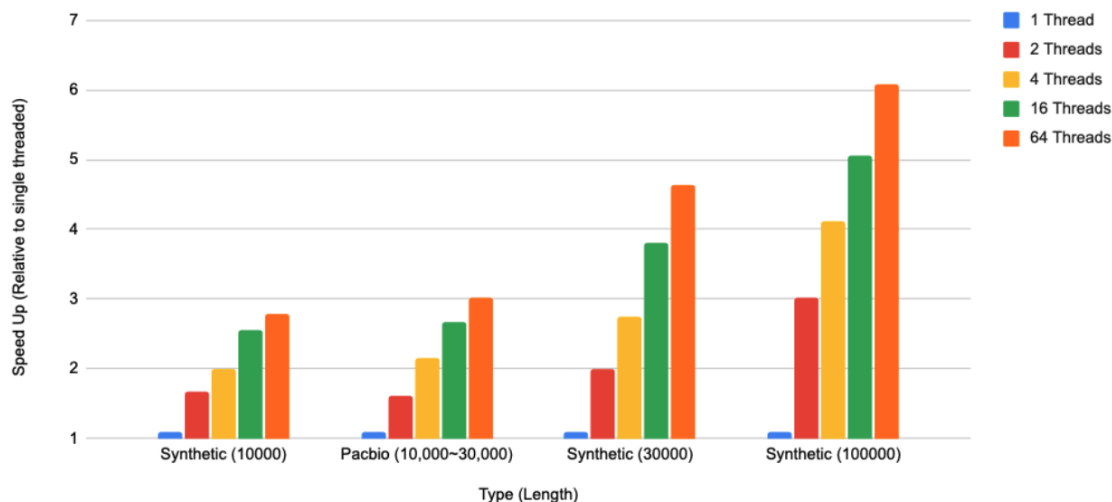


Figure 3.5: Speedup of LCSkPOA (parallel) across different datasets and thread counts.

Figure 3.5 illustrates the speedup achieved by the LCSkPOA (parallel) implementation across different numbers of threads and datasets of varying length. As expected, the speedup increases with the number of threads and with the length of the input sequences, with the highest performance gain observed at 64 threads on synthetic datasets of length 100,000, reaching over $6\times$ speedup over the single-threaded version. The speedup typically does not match the number of threads used because several components of the algorithm, such as partition search, graph construction for each thread, and concatenation of the resulting subgraphs, cannot be parallelized. However, for the synthetic 100,000-base dataset, the speedup observed with 2 and 4 threads exceeds the thread count. This is due to the use of dynamic banding, which effectively reduces the band size compared to the single-threaded run, leading to additional performance gains.

A noticeable discrepancy appears when comparing the synthetic dataset of length 10,000 to the PacBio HiFi dataset (10,000–30,000 bases). While single-threaded performance remains consistent across both datasets, multi-threaded performance is lower on the PacBio HiFi data. This reduction is attributable to the presence of long gaps, represented as long edges in the POA graph, within HiFi reads, which complicate the iden-

tification of suitable anchors (partition points) required for effective parallelization. As a result, fewer segments can be processed in parallel, limiting the achievable speedup on real-world data compared to synthetic sequences.

3.3.6 Alignment Quality

Table 3.1: Comparison of alignment accuracy (alignment score similarity with respect to SPOA unbanded alignment) for SPOA, LCSkPOA (sequential), and LCSkPOA (parallel) at different values of k . Values are averages of (10) runs on synthetic data and PacBio HiFi data.

String length	SPOA score	k	LCSkPOA (seq.)	LCSkPOA (par.)
10	16	2	16 (100%)	16 (100%)
		4	16 (100%)	16 (100%)
100	142	4	140 (99.01%)	133 (94.21%)
		8	126 (89.42%)	136 (96.05%)
1,000	1,411	4	1,320 (93.57%)	1,022 (72.46%)
		8	1,297 (91.96%)	1,322 (93.69%)
10,000	14,891	8	14,891 (100%)	14,065 (94.45%)
		12	14,696 (98.69%)	14,399 (96.70%)
30,000	46,182	8	46,182 (100%)	42,994 (93.09%)
		12	43,926 (95.11%)	44,026 (95.33%)
10,000–30,000 (PacBio HiFi)	27,742	8	24,217 (87.29%)	27,305 (98.42%)
		12	27,644 (99.66%)	27,393 (98.75%)

Table 3.1 presents a comparative analysis of SPOA, LCSkPOA (sequential), and LCSkPOA (parallel) across various string lengths, focusing on alignment accuracy. Accuracy is defined as the similarity of the alignment score, computed using a scoring scheme of +2 for matches and −2 for mismatches and gaps, relative to SPOA’s unbanded alignment.

For short strings of length 10 and 100, all three methods achieve similarly high POA alignment scores, indicating that even with limited input size, the LCSkPOA-based approaches maintain high alignment quality.

As the string length increases to 1,000 bases, the alignment scores of both LCSkPOA variants decline (below 94% accuracy). This decline is primarily due to some reads containing long indels that span beyond the banded region, causing portions of the alignment to fall outside the computed area. For longer sequences of 10,000 and 30,000 bases, the accuracy of LCSkPOA (parallel) remains consistently high across increasing values of k . At 30,000 bases and $k = 12$, LCSkPOA (parallel) achieves a score nearly identical to the sequential version, confirming that the introduction of parallelism does not degrade accuracy.

On real sequencing data, specifically PacBio HiFi reads (10,000–30,000 bases), LCSkPOA (parallel) with $k = 12$ maintains a high alignment score of 99.66%, virtually equivalent to both LCSkPOA (sequential) and SPOA. These results confirm that LCSkPOA (sequential) and LCSkPOA (parallel) achieve alignment accuracy comparable to SPOA, making them robust choices for high-fidelity sequence alignment even in large-scale genomic applications.

3.4 Discussion

The results presented in this chapter demonstrate that the proposed LCSkPOA algorithm provides substantial improvements in both runtime and memory consumption for partial order alignment, particularly on the long reads produced by PacBio HiFi sequencing. By extending the LCSk++ framework [48, 49] to the graph-based setting and integrating it with an efficient event-driven processing scheme backed by Fenwick trees [82], LCSkPOA achieves up to a $60\times$ speedup over unbanded SPOA [43] on long synthetic sequences while maintaining alignment accuracy above 99% on real HiFi data at $k = 12$.

These improvements directly benefit the consensus and polishing pipeline described

in Chapter 2: because TopoQual relies on POA of subreads as its underlying alignment substrate, the speed and memory gains from LCSkPOA translate into a tractable polishing pipeline at the genome scale. More broadly, the results suggest that graph-based aligners such as GraphAligner [78] and downstream graph-based assemblers [33–35] stand to benefit from improvements in the multiple sequence alignment substrate they rely on.

Several limitations of the current method merit discussion. First, the accuracy drop observed at intermediate string lengths (around 1,000 bases) suggests that the chosen banding strategy can miss alignments that contain long indels spanning beyond the band; this could be addressed by adaptively widening the band in regions where the seed density is low. Second, the parallel implementation observes diminishing returns on real HiFi data due to long gaps in the POA graph; future work could explore graph-aware partitioning schemes that explicitly account for the indel structure of HiFi subreads. Third, the current evaluation focuses on global alignment; extending the empirical evaluation to local and semi-global alignment, both of which the method theoretically supports, is an important direction for future work.

Together, Chapters 2 and 3 address two complementary bottlenecks in the HiFi processing pipeline: the calibration of base quality scores and the runtime of the alignment substrate that underlies both consensus and polishing. The next chapter shifts focus to a different class of problem, namely the demultiplexing of high-donor pools in droplet-based scRNA-seq [9], where the algorithmic challenge lies not in aligning long reads but in clustering short-read variant signal across many samples.

Chapter 4

Demultiplexing High-Donor Single-Cell RNA-seq

This chapter presents *Souporcell3*, a method for robust demultiplexing of pooled droplet-based single-cell RNA-seq (scRNA-seq) data at high donor counts. This method is published in Bioinformatics journal [12]. Section 4.1 motivates the work by describing the limitations of existing genotype-based demultiplexing tools at scale. Section 4.2 describes the three principal algorithmic improvements introduced in *Souporcell3*: a $10\times$ -merge cluster initialization strategy, a K-Harmonic Means (K-Harmonic Means (KHM)) clustering objective with deterministic annealing, and an iterative refinement procedure that reinitializes low-quality clusters while locking high-quality ones. Section 4.3 describes the benchmark datasets used for evaluation, including both synthetically merged balanced datasets at up to 64 donors and experimentally merged unbalanced datasets at 18–24 donors. Section 4.4 introduces the evaluation metrics used throughout the chapter. Sections 4.5 and 4.6 present ablation studies of the initialization and clustering components, respectively. Section 4.7 presents the main empirical results comparing *Souporcell3* against Vireo [10], overclustered Vireo, and the original *Souporcell* [9]. Section 4.8 discusses the implications and limitations of the proposed method.

4.1 Motivation

Multiplexing cells from multiple individuals into a single scRNA-seq sample has become a popular experimental design because it reduces costs, avoids batch effects, increases statistical power, improves doublet detection, and enables the measurement and

removal of ambient RNA [9, 58]. Demultiplexing such samples was originally enabled by *demuxlet* [11], which required *a priori* knowledge of the genotypes of every individual in the mixture. When this information was unavailable, additional costly DNA sequencing of each sample was required. Subsequently, Souporcell [9] and Vireo [10] were developed to demultiplex cells by genotype without prior knowledge of donor genotypes, using various sparse clustering methods.

However, clustering is known to be an NP-hard problem under any non-trivial loss function [13], and this difficulty increases with the number of clusters [60, 61]. The historical solution to overcoming local optima has been to use multiple random restarts; however, the number of restarts required to probabilistically reach the global optimum grows exponentially in the number of clusters. More recently, the k-means++ cluster initialization strategy [67] has provided a simple but effective heuristic for selecting initial cluster centers that are more likely to converge to the global optimum. Because k-means++ uses individual data points as initial cluster centers, with new cluster centers chosen randomly weighted by squared distance to the closest existing cluster center, three problems arise when dealing with sparse data:

1. It is unclear what value to assign a cluster center along dimensions for which the data point has no observation.
2. The distance of a data point to other cluster centers can only be judged along dimensions for which it has observed data.
3. Because not every cell expresses every gene, and experimental yield is imperfect, scRNA-seq data is roughly 95% sparse: only about 5% of the dimensions have observed data for any given cell.

For these reasons, k-means++ is not directly suitable for scRNA-seq data and other very sparse data modalities.

An alternative is to apply a dimensionality reduction step prior to clustering. The sparse and binomial nature of scRNA-seq variant data complicates this approach, but recent work by Chung and Huang [83] introduced SNPManifold, which uses a binomial variational autoencoder to learn a lower-dimensional latent space of single-nucleotide polymorphism (SNP) data. This is promising but has not yet been validated on larger numbers of donors [83].

This chapter presents multiple algorithmic improvements for clustering scRNA-seq data by genotype. Previously, Souporcell was shown to cluster up to approximately 21 individuals with very high-quality data, but the original publication did not recommend designing experiments with more than 16 individuals [9]. With the improvements described here, we demonstrate robust genotype-based clustering for up to 64 individuals.

4.2 Method

Souporcell3 introduces several key improvements over its predecessor [9], enabling robust clustering of up to 64 donor samples. Figure 4.1 provides an overview of the method.

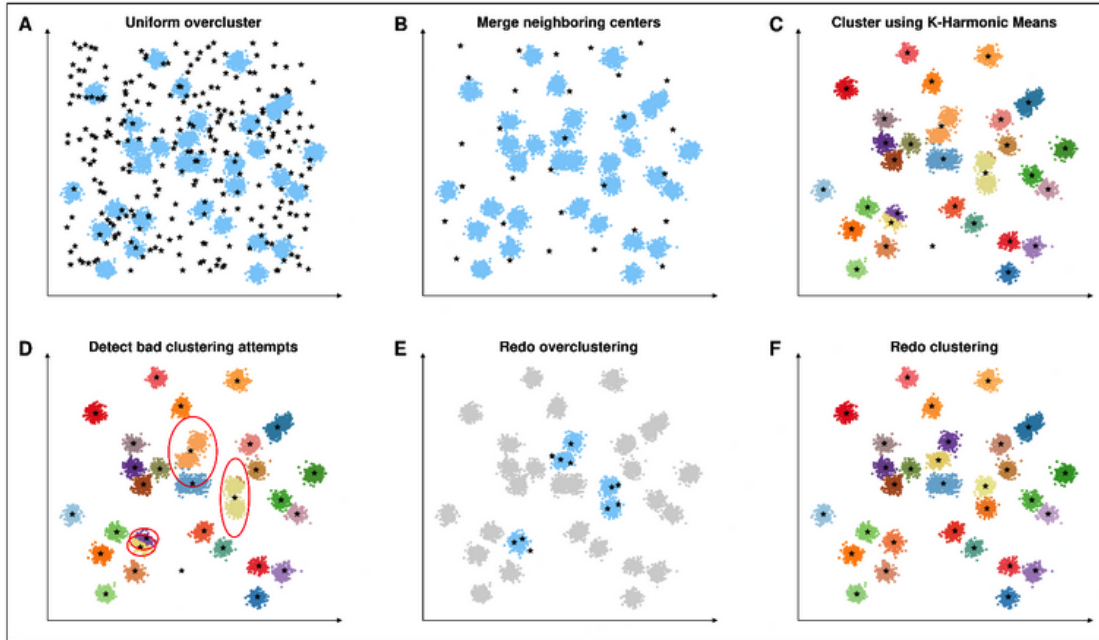


Figure 4.1: Method overview. (A) Randomly initialize (10k) cluster centers to ensure broad coverage of the genotype space. (B) Merge nearby cluster centers using a distance metric defined as the sum of allele fraction differences weighted by allele counts until (k) clusters remain. (C) Cluster using K-Harmonic Means (KHM) to ensure enhanced convergence and reduced sensitivity to initialization. (D) Identify low- and high-quality clusters based on the number of assigned cells and the loss value per cluster. (E) Reinitialize low-quality clusters. (F) Rerun KHM on the refined set while locking the high-quality clusters.

4.2.1 Input and Data Preprocessing

Data preprocessing in Souporcell3 follows the approach used by Souporcell [9]. To prepare the data for genotype-based cell clustering, allele information is first measured for each cell. Instead of using STAR-aligned BAM files, which can introduce false-positive variants and reference bias, reads are remapped using minimap2 [45] to obtain more accurate alignments. Putative single-nucleotide polymorphisms (SNPs) are then called with FreeBayes, and allele counts per cell are obtained using Vartrix, which reduces reference bias and corrects duplicate reads through unique molecular identifiers (unique molecular identifiers (UMIs)). When available, a set of reliable common variants can be used in place of FreeBayes-derived candidate variants. The alternate/reference read depth at

each variant locus is used to determine whether the locus is included in the clustering task. A default threshold of 4 is set in Souporcell3, which can be adjusted by the user. The allele frequency at each passing locus is used for the binomial loss calculation and, in turn, for the clustering step.

4.2.2 Cluster Initialization Strategies

Almost all commonly used clustering algorithms implicitly assume the data is dense and continuous, which ensures a relatively smooth cost-function landscape. In contrast, genetic variant data is both sparse and discrete, with each dimension typically limited to three possible values (homozygous reference, heterozygous, homozygous alternate). This sparsity poses significant challenges for greedy clustering algorithms such as k-means, which are prone to becoming trapped in local minima. A standard remedy is to perform multiple random restarts; however, the number of restarts needed to reliably approach the global optimum increases exponentially with the number of clusters k [13, 61]. To address this, three initialization strategies were considered.

K-means++ Initialization

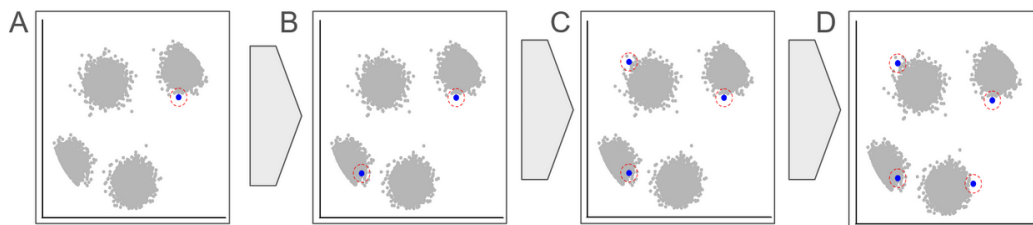


Figure 4.2: K-means++ Initialization: (A) Randomly select a cell as the first cluster center. (B-D) Iteratively select cells as cluster centers based on their squared distance to the closest existing cluster center.

The k-means++ initialization method [67] was proposed to provide a simple, parameter-free, and generally robust seeding strategy. Despite its widespread adoption, k-means++

still struggles in sparse settings, where limited shared information between data points across dimensions undermines the effectiveness of its distance-based seeding. In *k*-means++, the initial cluster center is selected randomly from the dataset. A distance metric is then used iteratively to determine the probability of selecting a new cluster center, weighted by the squared distance of each data point from its closest existing cluster center. For this metric, we use the beta-binomial distance, as it accounts for genotype positions where no data is present.

Overclustering Initialization

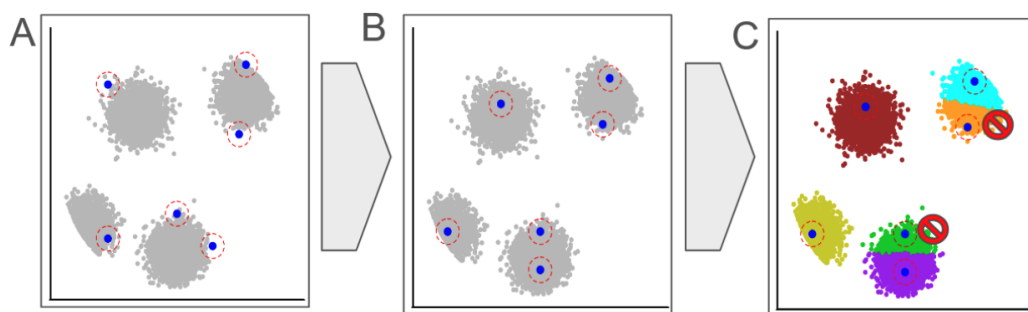


Figure 4.3: Overclustering Initialization: (A) Select $(k + k)$ random cells as cluster centers, where (k) is the desired number of clusters. (B) Cluster using the selected cluster centers. (C) Select (k) cluster centers with the maximum number of associated cells.

Another strategy aimed at improving cluster initialization is *overclustering*, which is suited for challenging genetic datasets that are sparse and discrete. By temporarily increasing the number of clusters beyond the desired value, overclustering aims to better capture the underlying structure of the data, especially since the effective number of clusters can be higher than expected due to the presence of doublets. This strategy is employed by Vireo and SC3 [10, 66] to enhance cluster initialization.

We begin clustering with $k + k$ clusters instead of the target k . After clustering, we evaluate the resulting clusters based on the number of cells assigned to each. The k clusters with the highest cell counts are then selected to form the final set of cluster centers, similar

to Vireo. This selection criterion ensures that the most representative and well-supported clusters are retained while smaller, potentially spurious clusters are discarded. However, this method can be prone to errors in high-donor-count scenarios, where the genotype space is constrained and substantial overlap between donors exists. In such cases, selecting clusters solely based on cell count may result in incorrectly merged donors, where one cluster gets assigned cells from multiple donors.

10x-Merge Initialization

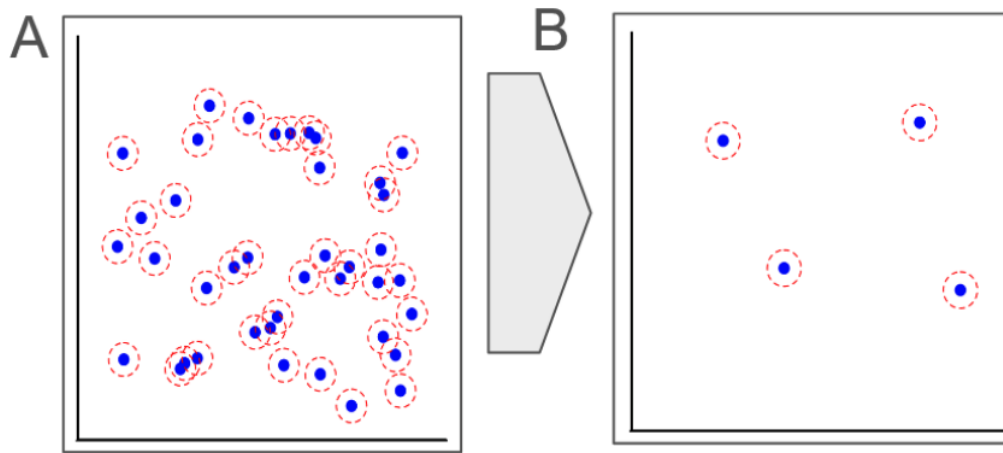


Figure 4.4: 10x-Merge Initialization: (A) Randomly initialize (10k) cluster centers. (B) Iteratively merge clusters in close proximity until the required cluster size is reached.

10x-merge is the primary initialization strategy employed by Souporcell3. The algorithm generates a large number of preliminary cluster centers, ten times the expected number of donors, and then performs an iterative merging step. Highly similar cluster centers in genotype space, typically differing by only a few informative alleles, are combined into a more concise and representative set. The distance metric used is the sum of squared differences between randomly assigned cluster center values, weighted by the number of alleles at each variant position. The two closest cluster centers are identified and merged by computing the mean values at each variant position and using that as the new cluster center. This process is repeated iteratively until k cluster centers remain.

This strategy reduces redundancy while preserving essential genotype distinctions, ensuring that the subsequent clustering step begins with informative and well-separated cluster centers. This is particularly helpful in scenarios where multiple overlapping donor cells occupy a dense genotype space, as widely separated cluster centers are unlikely to claim cells from multiple donors at the outset. The intuition is similar to the spacing achieved by k-means++ [67], but adapted to the sparsity and discreteness of scRNA-seq variant data.

4.2.3 K-Harmonic Means Clustering

Soft clustering methods such as Expectation-Maximization (expectation-maximization (EM)) applied to Gaussian Mixture Models have been widely used due to their ability to recover more optimal clusterings than hard clustering methods such as Lloyd’s algorithm for k-means [84]. However, these methods remain prone to converging to local optima if the cluster initialization is poor. To address this, Souporcell3 adopts K-Harmonic Means (KHM) [14, 68, 69], a soft clustering objective that computes the contribution of each data point to all cluster centers based on inverse distances.

Definitions. Similar to Souporcell, cluster centers in Souporcell3 are represented by allele fraction values for each variant locus. The negative binomial likelihood probability mass function of allele counts at each cluster center’s allele fraction is used as the distance metric for the KHM loss function. The notation used throughout is as follows:

- K : the number of genotype clusters, fixed at the outset. Lower-case k is used for indexing a specific cluster.
- C : the number of cells. Lower-case c indexes a specific cell barcode. A barcode may contain 0, 1, or more cells; the model assumes that the majority of barcodes contain a single cell.

- L : the number of variant loci. Lower-case l indexes a specific locus; only biallelic variants are considered. L_c is the list of loci with observed data in cell c .
- A : allele counts. $A_{l,c}$ is a tuple of size 2 with the first entry representing the number of reference alleles and the second entry representing the number of alternate alleles observed at locus l in cell c .
- $\phi_{k,l}$: the mixture parameter for the allele fraction of cluster k at locus l . This is a real number representing the fraction of reference alleles in this cluster at this locus. Values are expected to be near 1.0 (homozygous reference), 0.5 (heterozygous), or 0.0 (homozygous alternate), but they can be skewed by noise, doublets, or ambient RNA.

Loss function. The KHM loss for cluster centers $\Phi = \{\phi_k\}_{k=1}^K$ given the observed allele counts is given by

$$\mathcal{L}_{\text{KHM}}(\Phi) = \sum_{c=1}^C \frac{K}{\sum_{k=1}^K \frac{1}{d(A_c, \phi_k)^p}}, \quad (4.1)$$

where $d(A_c, \phi_k)$ is the negative binomial likelihood distance between cell c and cluster center k , and p is a tunable parameter that controls the influence of distance magnitudes. The harmonic-mean form of the loss allows every cell to influence all cluster centers during each iteration, leading to more robust convergence and improved clustering outcomes. KHM has been shown to outperform traditional k-means in both optimality and resilience to poor or adversarial initializations [14, 68, 69].

To further adapt KHM to high-dimensional genotype space, Souporecell3 employs a generalized variant, KHM_p [15], which uses L_p -norm normalization to address symmetry issues known to affect clustering performance in such contexts. We found that setting $p = 25$ minimized the loss across a range of tested values, and this value is used in all reported results.

4.2.4 Deterministic Annealing

The original Souporcell used a deterministic annealing variant of EM (deterministic annealing (DA)-EM) to manage non-convexity via a temperature parameter that starts high and is gradually lowered, allowing the solution to more often fall into the global optimum [9]. Similarly, Souporcell3 employs a deterministic annealing-enhanced KHM (DA-KHM), combining the robustness of KHM with the annealing strategy. The temperature parameter is slightly adjusted using a temperature constant to account for the difference in convergence speed between the two clustering algorithms. When the temperature reaches 1.0, the loss function effectively becomes the standard KHM loss (Equation 4.1). An empirical comparison of the deterministic annealed methods and the vanilla methods is presented in Section 4.6.

4.2.5 Iterative Refinement with Reinitialization and Locking

After the initial clustering run, Souporcell3 evaluates the quality of each cluster using both the number of cells assigned to the cluster and the associated loss value. Clusters identified as outliers, clusters deemed suboptimal, are reinitialized using the 10x-merge cluster initialization strategy. Ten times the number of outlier clusters are initialized and merged to generate well-separated cluster centers equal in number to the original outliers. This procedure allows the algorithm to improve clustering in the next iteration.

At the same time, high-quality clusters are randomly selected from the remaining set, after excluding the outliers, and are *locked* to preserve their values during subsequent iterations. This strategy is inspired by node freezing and dropout in machine learning, which are used to reduce overfitting and accelerate fine-tuning [75, 85]. This iterative refinement ensures accurate donor assignment and enhances the overall robustness of the clustering process.

Bad Cluster Detection

Analysis of cell-to-cluster assignments in the 64-donor, no-doublet dataset after algorithm convergence revealed that the number of incorrectly merged clusters corresponds closely to the number of empty or near-empty clusters. As shown in Figure 4.5, the cell-assignment distribution after convergence reveals three incorrectly merged cluster centers that align with the presence of three empty clusters.

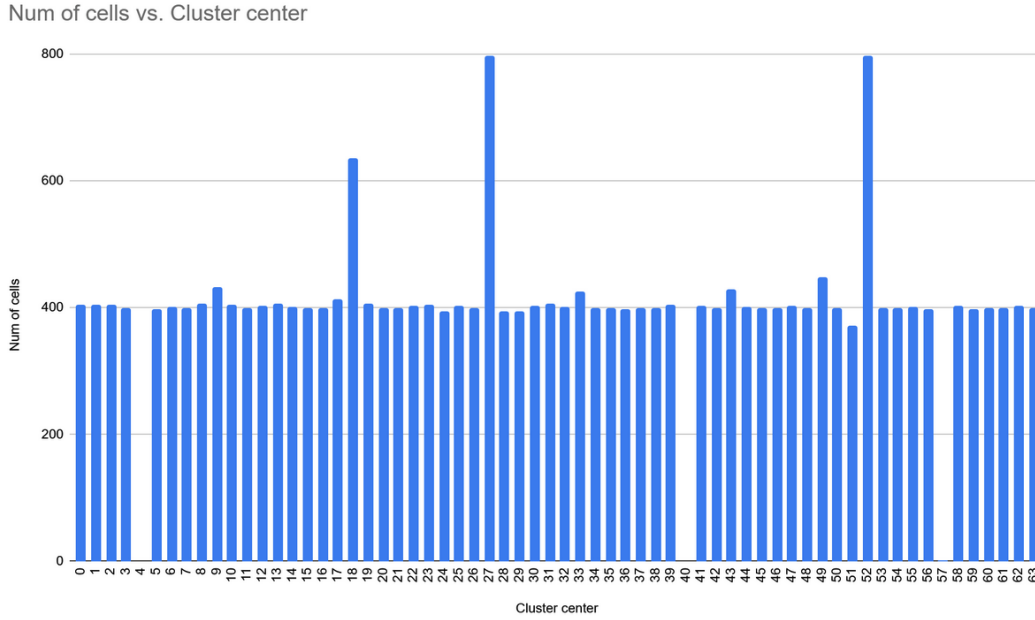


Figure 4.5: Number of cells assigned to each cluster center on a no-doublet balanced dataset with (64) donors. Empty or near-empty clusters indicate the presence of incorrectly merged clusters elsewhere in the assignment.

By computing the total log loss for each cell's assigned cluster, normalizing it by the mean total log loss, and identifying statistical outliers, we can pinpoint cluster centers that are poorly or incorrectly assigned. We define the cluster-center value as

$$\text{CC Value} = \frac{\sum_{\text{all cells}} \log \text{loss for assigned CC}}{\text{Mean log loss}}. \quad (4.2)$$

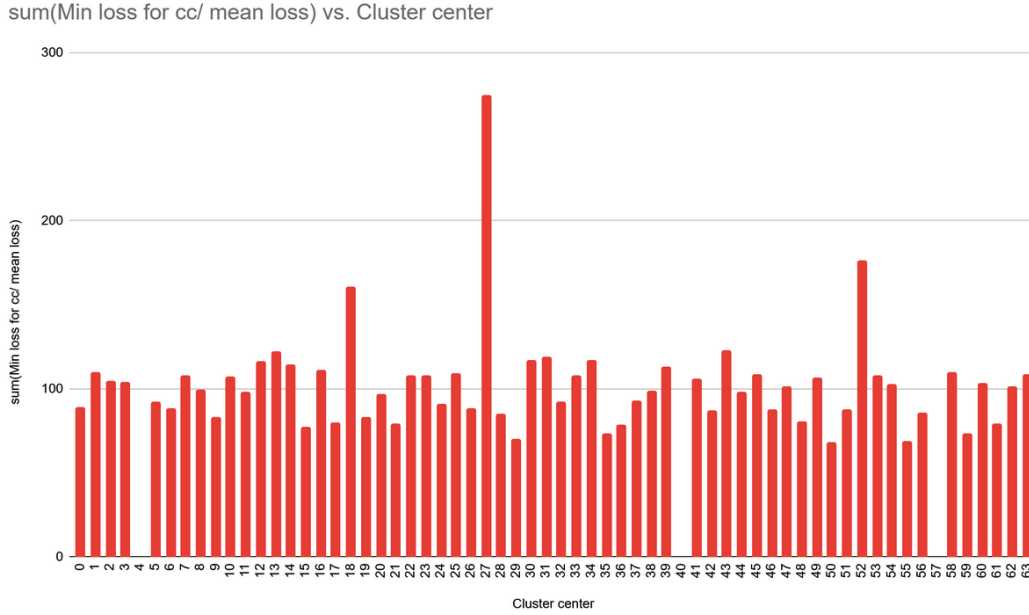


Figure 4.6: Total log loss for each cell's assigned cluster, normalized by the mean total log loss, on a no-doublet balanced dataset with (64) donors.

Outliers with a high number of cell assignments (greater than the median plus 1.5 times the interquartile range) can be considered the culprits that have claimed ownership of the low-assignment outliers (less than the median minus 1.5 times the interquartile range). By reinitializing these cluster centers, the clustering can be substantially improved.

In scenarios where the distribution of cells per donor is uneven, it may not be possible to detect outliers using this method alone. However, empty or near-empty clusters can still be identified by defining an appropriate threshold on the cell-assignment count. The corresponding incorrectly merged clusters typically have CC values at the high end of the distribution. In such cases, a number of high-loss clusters equal to the number of empty clusters is selected and treated as outliers for reinitialization. Even if a correctly clustered center is mistakenly selected as an outlier, the subsequent clustering should not significantly affect the final outcome, as the same cells should be reassigned to that cluster.

Reinitialization and Locking Procedure

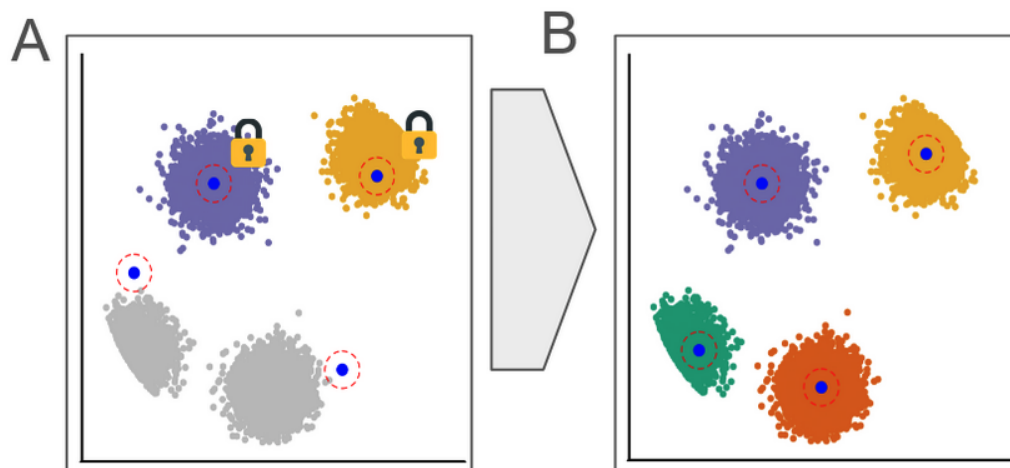


Figure 4.7: (A) After the first clustering run, bad clusters are detected and reinitialized while good clusters are locked. (B) With the good clusters locked against updates, the KHM clustering converges to a refined solution.

Once poorly formed clusters can be identified after convergence (Section 4.2.5), Souporecell3 reinitializes the corresponding cluster centers and reruns the algorithm to improve clustering accuracy. Since most donors are correctly clustered in the initial run, it is unnecessary to update all of the cluster centers. Instead, the cluster centers deemed high quality are locked, effectively freezing them, while only the reinitialized (bad) clusters are allowed to update. This strategy is analogous to node freezing and dropout in machine learning, where parts of a pretrained model remain fixed while selected components are fine-tuned on new data [75, 86]. The procedure can be iteratively repeated, detecting bad clusters and reapplying the locked KHM algorithm with reinitialized centers, until no bad clusters remain.

4.3 Datasets

To evaluate Souporecell3 under both controlled and realistic experimental conditions, we constructed two classes of benchmark datasets: synthetically merged balanced datasets

at high donor counts, and experimentally merged unbalanced datasets from previously published studies.

4.3.1 Synthetically Merged Balanced Datasets

To assess performance with high donor counts (~ 64 donors), we obtained individual datasets from established publications [54, 55] and synthetically merged them. Four datasets from the Human Cell Atlas [70, 71, 87] and the European Nucleotide Archive [72] were used, each comprising human-derived samples generated using the 10x Genomics 3' chemistry (v1 and v2). All samples were derived from *Homo sapiens*, and the data collectively span a wide range of immune and developmental contexts, providing a diverse foundation for evaluating demultiplexing performance across donor pools and biological conditions.

The four source datasets are summarized below.

- **Source 1:** Immune landscape of viral- and carcinogen-driven head and neck cancer [54, 70]. 37 donors, 126.0k cells.
- **Source 2:** Human Cell Atlas project [87]. 28 donors, 1.5M cells.
- **Source 3:** Mapping the developing human immune system across organs [55, 71]. 30 donors, 922.3k cells.
- **Source 4:** ENA cell lines [72]. 5 donors, 50k cells.

Collectively, these datasets span 100 unique donors and over 2.5 million single-cell transcriptomes. After applying quality control to remove samples with corrupted data, insufficient read depth, and redundant samples collected across different organs from the same individual, we curated a high-quality balanced dataset comprising 64 distinct donors for evaluation.

Dataset generation. To produce balanced datasets tailored for benchmarking, we extracted the required number of cells for each experimental scenario. Specifically, we constructed datasets with controlled doublet rates of approximately 0%, 5%, and 10% and varying donor counts. Each subset was limited to a total of approximately 10,000–30,000 cells, aligning with the practical upper limit for many current scRNA-seq experiments using 10x Genomics platforms [64]. This cell count threshold reflects a balance between throughput and data quality, as higher cell input often increases the likelihood of doublet formation and misclassification errors. Published reports suggest that, at this scale, the expected doublet rate for standard 10x Chromium 3' chemistry ranges between 5% and 10% when targeting approximately 20,000 cells [52, 56, 57, 64]. Table 4.1 summarizes the composition of the balanced datasets.

Table 4.1: Composition of the balanced datasets used in evaluation.

Type	Donors	Total cells	Cells/donor	Doublet count	Doublet %
No doublets	4	9,300	2,325	0	0.00
	8	10,000	1,250	0	0.00
	16	12,800	800	0	0.00
	32	12,800	400	0	0.00
	48	19,200	400	0	0.00
	64	25,600	400	0	0.00
5% doublets	4	7,123	1,781	373	5.24
	8	8,303	1,038	428	5.15
	16	11,200	700	578	5.16
	32	11,777	368	617	5.24
	48	18,239	380	959	5.26
	64	20,061	313	1,053	5.25
10% doublets	4	6,740	1,685	740	10.98
	8	8,120	1,015	888	10.94
	16	10,799	675	1,199	11.10
	32	11,159	349	1,239	11.10
	48	16,917	352	1,877	11.10
	64	22,679	354	2,519	11.11

4.3.2 Experimentally Merged Unbalanced Datasets

To evaluate Souporcell3 under unbalanced conditions in which some clusters contain far fewer cells than the average, we used three well-established imbalanced datasets [59] that have been used in multiple prior publications [83], comprising 18–24 donors per pool. All three datasets were obtained from the European Nucleotide Archive (European Nu-

cleotide Archive (ENA)) and consist of human-derived samples generated using the 10x Genomics 3' chemistry (v1 and v2):

- **Unbalanced 1 (Pool 4):** 18 donors, 8k cells.
- **Unbalanced 2 (Pool 6):** 24 donors, 6k cells.
- **Unbalanced 3 (Pool 8):** 19 donors, 8k cells.

The ground-truth donor assignments were determined using Demuxlet [11], as reported in the original publication [59]. The minority clusters in the three datasets comprise approximately 0.84%, 1.4%, and 1.13% of the cells, respectively, testing Souporecell3's ability to recover small donor populations.

4.4 Evaluation Metrics

Clustering performance is evaluated using two key metrics: the Adjusted Rand Index (Adjusted Rand Index (ARI)) and the number of Incorrectly Merged Clusters (Incorrectly Merged Cluster (IMC)).

Adjusted Rand Index. The Rand Index is a standard and widely used metric in clustering analysis; it measures the similarity between two data clusterings [62]. The Adjusted Rand Index corrects the Rand Index for the random chance that the clustering is correctly assigned [63, 88]. In this work, ARI is used to compare the clustering output of each method against the known ground truth, which is available because the datasets were generated from sources with known donor origins.

Incorrectly Merged Clusters. The IMC metric captures instances where two or more distinct ground-truth donors are incorrectly assigned to the same cluster by the method. For example, if both donor 3 and donor 4 cells are incorrectly grouped into the same cluster by the clustering algorithm, the IMC value is incremented by one.

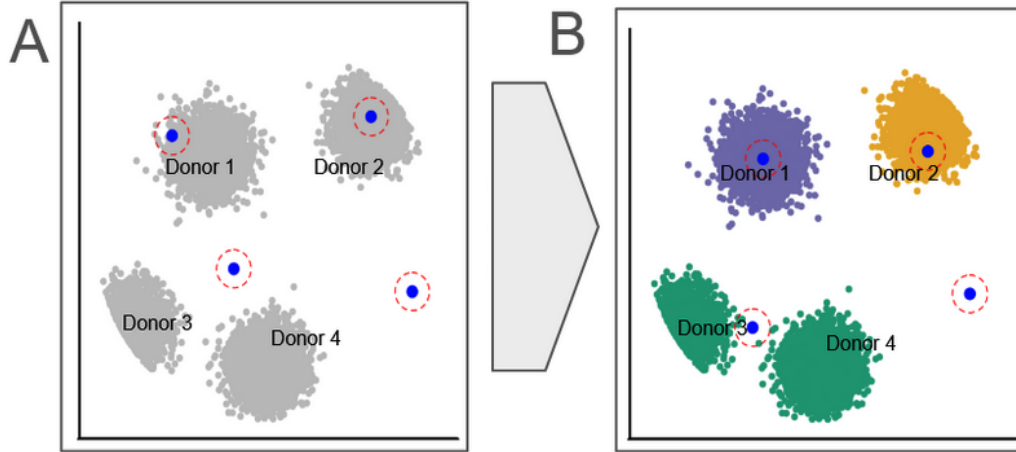


Figure 4.8: Illustration of the IMC metric. (A) Before clustering, cluster centers are initialized. (B) After the clustering method converges, an empty cluster center is present (not assigned to any cells) and one cluster center is assigned to cells from two donors, yielding (IMC = 1).

4.5 Cluster Initialization Evaluation

To evaluate the different cluster initialization strategies, we tested them on the 64-donor balanced dataset without doublets. We also included an *optimal* cluster initialization blueprint, in which clusters were initialized using ground-truth donor information, in order to assess whether perfect initialization alone could yield perfect clustering or whether additional refinement steps were still necessary in high-donor scenarios.

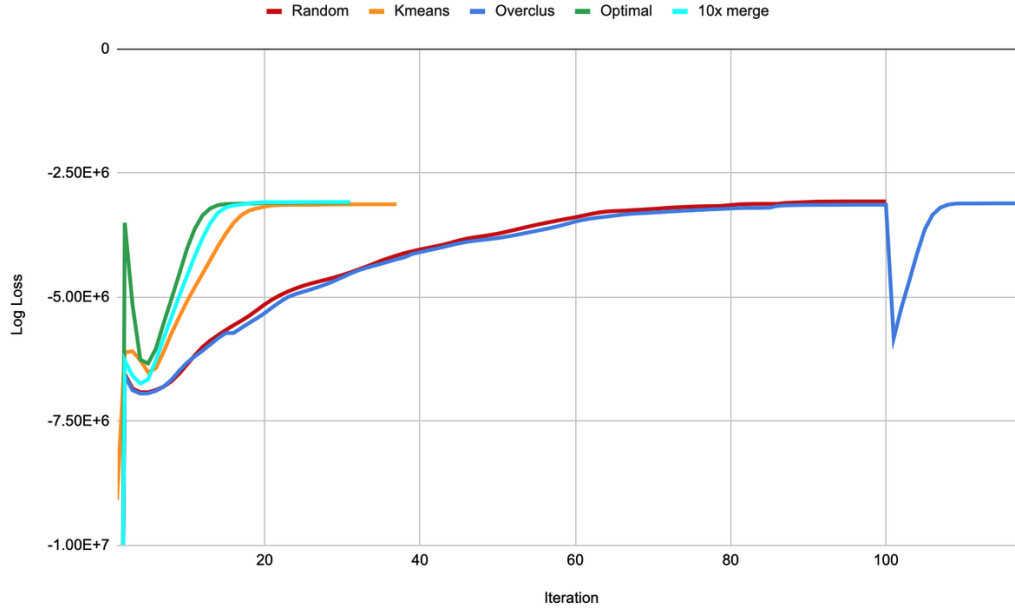


Figure 4.9: Log loss versus iterations for EM with five different initialization methods (random, k-means++, overclustering, optimal, and 10x-merge) on a no-doublet balanced dataset with (64) donors.

The log loss and number of iterations for the no-doublet 64-donor dataset are shown in Figure 4.9. EM with optimal cluster initialization starts with the lowest log loss but eventually reaches the same log loss as k-means++ and 10x-merge, all of which follow a similar convergence pattern. Both random and overclustering methods take more iterations to converge; overclustering runs for additional iterations after removing the extra k clusters (100+ iterations) and converges to the same log loss as random initialization.

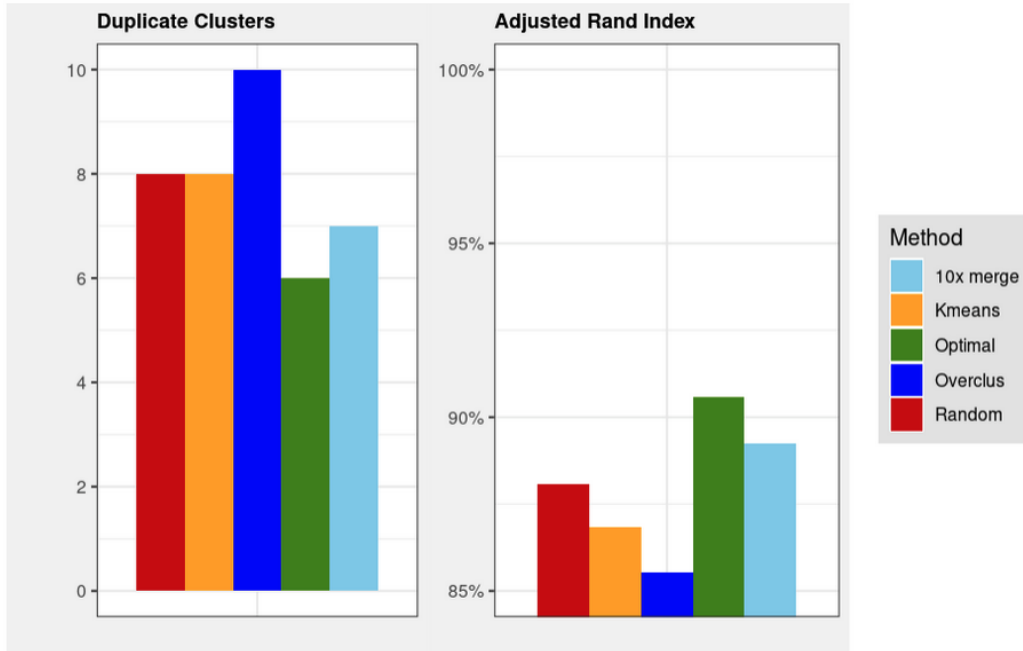


Figure 4.10: The right panel shows the ARI values for EM runs with different cluster initialization methods. The left panel illustrates the incorrectly merged clusters IMC present for the same methods.

Figure 4.10 displays the two key clustering metrics across five initialization methods: 10x-merge, k-means++, optimal, overclustering, and random. The number of incorrectly merged clusters is lowest for the optimal initialization method, which is expected since the cluster centers were initialized with ground-truth values. Overclustering and random initializations produce the highest number of incorrectly merged clusters, with overclustering producing 10. Among the practical initialization methods, 10x-merge performs the best, producing approximately 7 incorrectly merged clusters. The ARI values reinforce this trend: the optimal method achieves the highest accuracy ($> 90\%$), while overclustering and random methods show lower performance ($< 88\%$).

Interestingly, when using the optimal initialization, which uses ground-truth information to set the cluster centers, the ARI value improves only marginally by 1.3% over 10x-merge, and incorrectly merged cluster assignments are reduced by just one. This is unexpected, as perfect clustering is theoretically possible under such conditions, where no further up-

dates to the cluster centers should be needed. However, even after convergence, incorrectly merged cluster assignments still occurred. This may be due to the large number of donors involved, where cells are densely packed in the genotype space. As the optimization progresses, log-loss updates may inadvertently shift a cluster center toward a dense region, causing it to claim ownership of cells from neighboring donors with similar genotypes. This observation motivated the iterative refinement procedure described in Section 4.2.5.

4.6 K-Harmonic Means Evaluation

Given that cluster initialization alone cannot solve the high-donor clustering problem (Section 4.5), we evaluate KHM clustering to determine what additional improvements can be achieved. Four methods are compared: DA-EM (Expectation-Maximization with deterministic annealing), DA-KHM (KHM with deterministic annealing), V-EM (vanilla EM without deterministic annealing), and V-KHM (vanilla KHM without deterministic annealing).

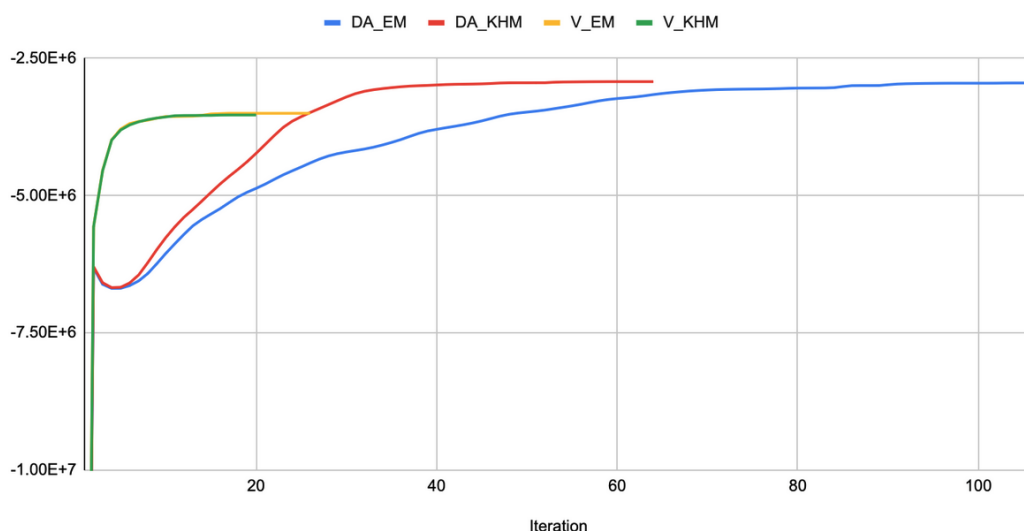


Figure 4.11: Log loss versus iterations for clustering methods DA-EM, DA-KHM, V-EM, and V-KHM on a no-doublet balanced dataset with (64) donors.

The log loss versus iteration is shown in Figure 4.11. EM and KHM without deterministic annealing (V-EM and V-KHM) follow a similar pattern and converge to approximately the same log loss. Both deterministic annealing methods require more iterations to converge, with DA-EM taking approximately twice as many iterations as DA-KHM. DA-EM converges to a slightly worse log loss of $-2,951,437.3$ compared to $-2,927,414.5$ for DA-KHM.

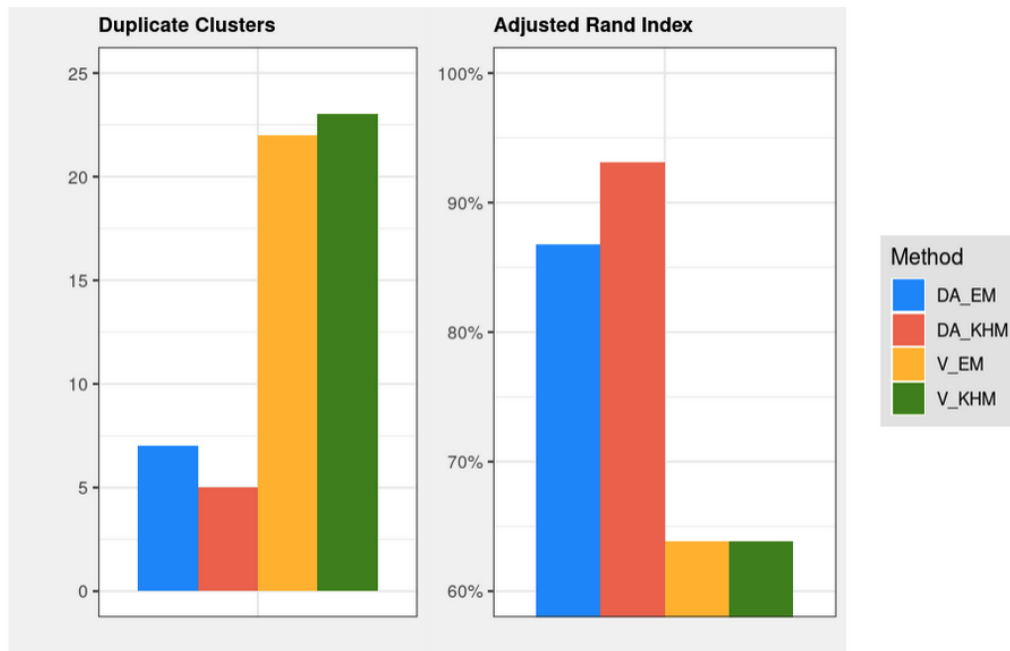


Figure 4.12: The right panel shows the ARI values for clustering methods DA-EM, DA-KHM, V-EM, and V-KHM. The left panel illustrates the incorrectly merged clusters present for the same methods.

Figure 4.12 presents the performance of the four clustering methods on a no-doublet dataset of 64 donors. Among the methods, DA-KHM achieves the highest ARI, exceeding 93%, and also has the fewest incorrectly merged clusters, at just 5. DA-EM also performs well, with an ARI close to 90% and only 7 incorrectly merged clusters. In contrast, both non-annealing variants (V-EM and V-KHM) perform significantly worse, with ARI values around 63% and more than 22 incorrectly merged clusters each. These results demonstrate the substantial benefit of deterministic annealing in clustering complex genotype data.

The more robust DA-KHM method outperforms DA-EM by 5% in ARI and produces 2 fewer incorrectly merged clusters. While this represents a significant improvement, perfect clustering is still not achieved, as some incorrectly merged cluster assignments remain. This residual error motivates the bad-cluster detection and reinitialization-with-locking procedure described in Section 4.2.5.

4.7 Main Results

This section presents the results of the four methods compared throughout this study: Vireo [10] (default configuration), overclustered Vireo (with pre-clustered donors specified as $10 \times k$ to match the cluster number in the 10x-merge setting), the original Souporcell [9] (default settings), and Souporcell3 (using 10x-merge initialization, KHM clustering with deterministic annealing, and iterative refinement with bad-cluster reinitialization and good-cluster locking).

4.7.1 Balanced Dataset Results

Incorrectly Merged Clusters



Figure 4.13: Number of incorrectly merged donor assignments for Vireo, overclustered Vireo, Soupocell, and Soupocell3 across different donor counts and doublet percentages on balanced datasets.

Figure 4.13 compares the performance of the four methods on datasets with varying numbers of donors (4, 8, 16, 32, 48, 64) under three doublet conditions (0%, 5%, and 10%). Under the no-doublet condition, all methods show an increasing number of incorrectly merged clusters as the donor count grows. Vireo performs the worst, peaking at 13 incorrectly merged clusters at 64 donors, followed by overclustered Vireo and Soupocell,

while Souporcell3 consistently yields no incorrectly merged clusters. With 5% doublets, Vireo performs slightly better than in the no-doublet scenario but still produces 12 incorrectly merged clusters at 64 donors, while Souporcell maintains moderate performance with only 7 incorrectly merged clusters. In the most challenging 10% doublet scenario, the pattern persists: Vireo and overclustered Vireo exhibit a high number of incorrectly merged clusters, Souporcell performs moderately, and Souporcell3 remains the most robust, maintaining zero incorrectly merged clusters across all donor counts.

Adjusted Rand Index

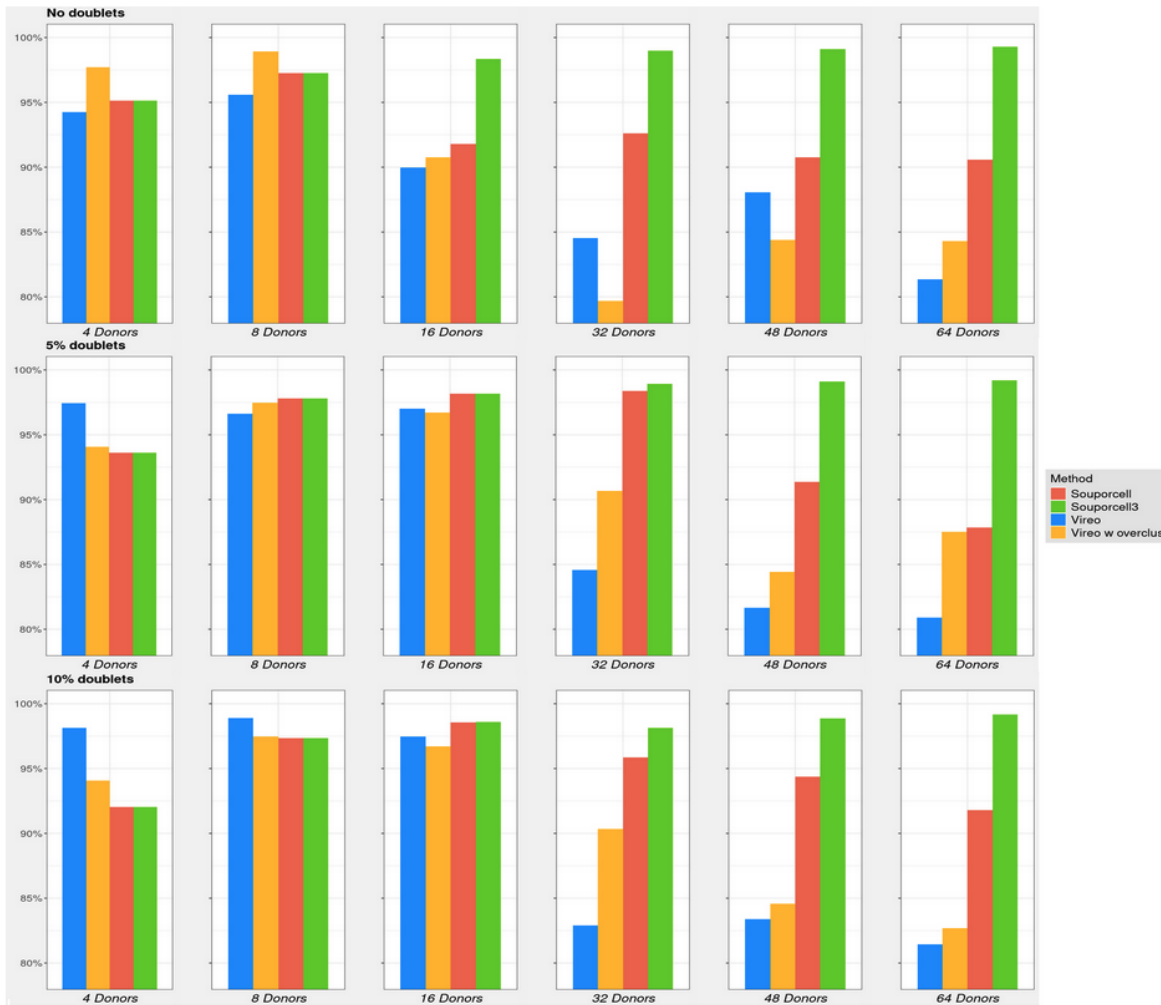


Figure 4.14: Adjusted Rand Index for Vireo, overclustered Vireo, Souporcell, and Souporcell3 across different donor counts and doublet percentages on balanced datasets.

Figure 4.14 compares the ARI across the four demultiplexing methods. In the no-doublet condition, all methods perform well at low donor counts (4–16 donors), with ARI values near or above 0.95. As the donor count increases to 32, 48, and 64, Vireo’s accuracy declines sharply, dropping below 0.86 at 48 donors and to 0.81 at 64 donors. Overclustered Vireo also deteriorates significantly beyond 32 donors, while Souporcell maintains better performance, and Souporcell3 consistently achieves near-perfect clustering ($\text{ARI} > 0.99$) across all donor levels.

In the 5% doublet condition, similar trends emerge: Souporcell3 remains highly accurate across all donor counts, while Vireo’s performance declines at 64 donors (ARI of 0.82) and overclustered Vireo exhibits a similar drop. The 10% doublet condition presents the greatest challenge, but Souporcell3 continues to maintain a nearly perfect clustering score, while Vireo experiences a pronounced decline ($\text{ARI} < 0.79$ at 64 donors). Souporcell performs similarly to its results in the 5% doublet condition.

The overall trend shows that for datasets with more than 32 donors, overclustered Vireo outperforms Vireo due to its better cluster center selection strategy. However, both Souporcell and Souporcell3 still perform better in comparison. Among all methods, Souporcell3 consistently outperforms the alternatives, demonstrating the highest clustering accuracy across all dataset complexities. It maintains near-perfect ARI values regardless of donor count or doublet rate, making it the most robust and reliable demultiplexing method evaluated in this study.

Cluster Maps and Heat Maps

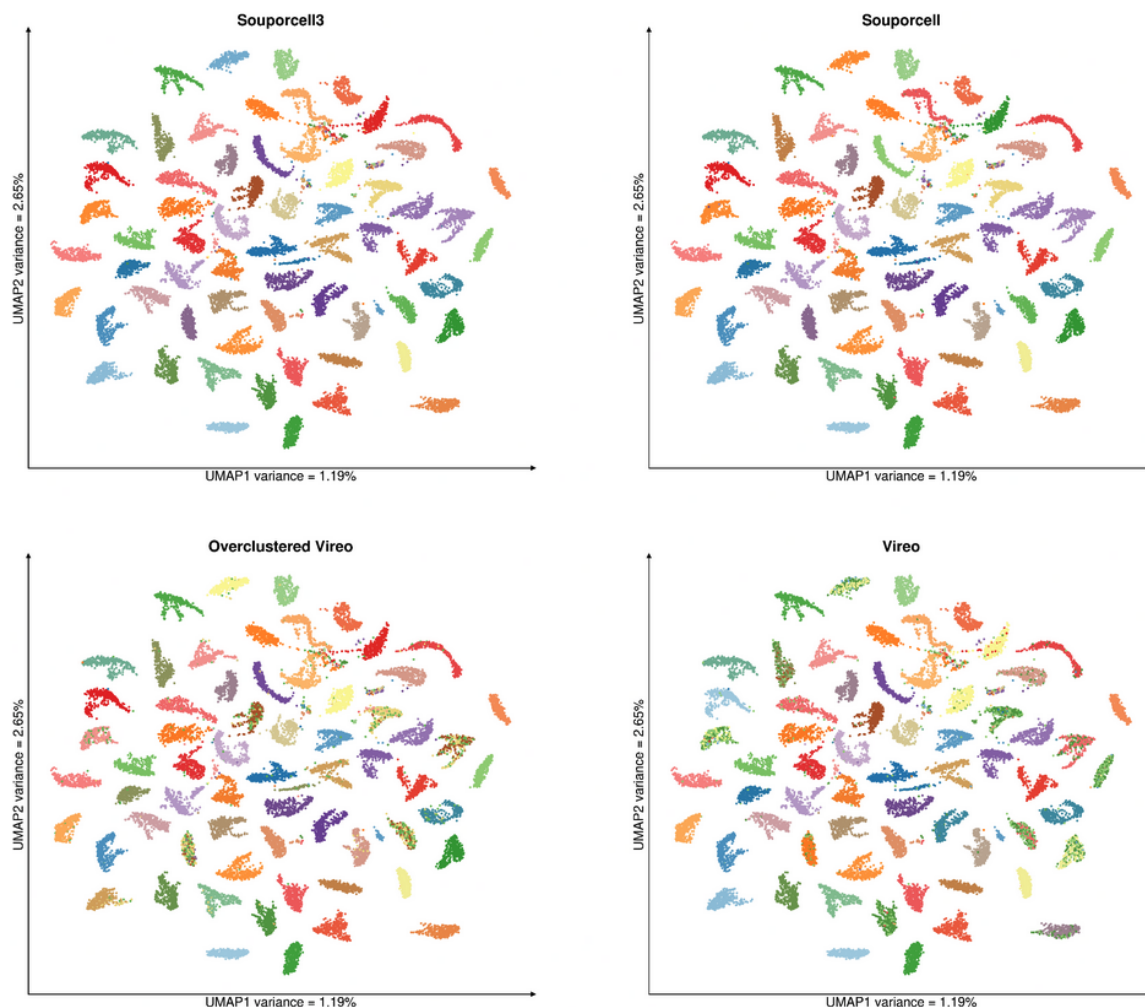


Figure 4.15: Cluster maps using UMAP for the x and y axes, showing the result of the 10 percent doublet, 64-donor balanced dataset after running Souporecell, Souporecell3, Vireo, and overclustered Vireo prior to doublet detection.

Figure 4.15 displays the clustering of the 64-donor, 10%-doublet dataset, with doublets excluded from visualization as the figures are generated before doublet detection for all methods. Each point represents a cell colored by its ground-truth donor identity, and the Uniform Manifold Approximation and Projection (UMAP) axes capture the primary structure of the high-dimensional cluster data in two dimensions. In the Souporecell3 panel, clusters are well-separated throughout the UMAP space, indicating effective donor de-

multiplexing. Even in the central region, where many clusters appear close together, they remain distinct and clearly partitioned with only a few outliers, demonstrating Souporcell3's ability to resolve densely packed but separate donor profiles. In contrast, Vireo and over-clustered Vireo exhibit many ground-truth clusters where two or more clusters are merged.

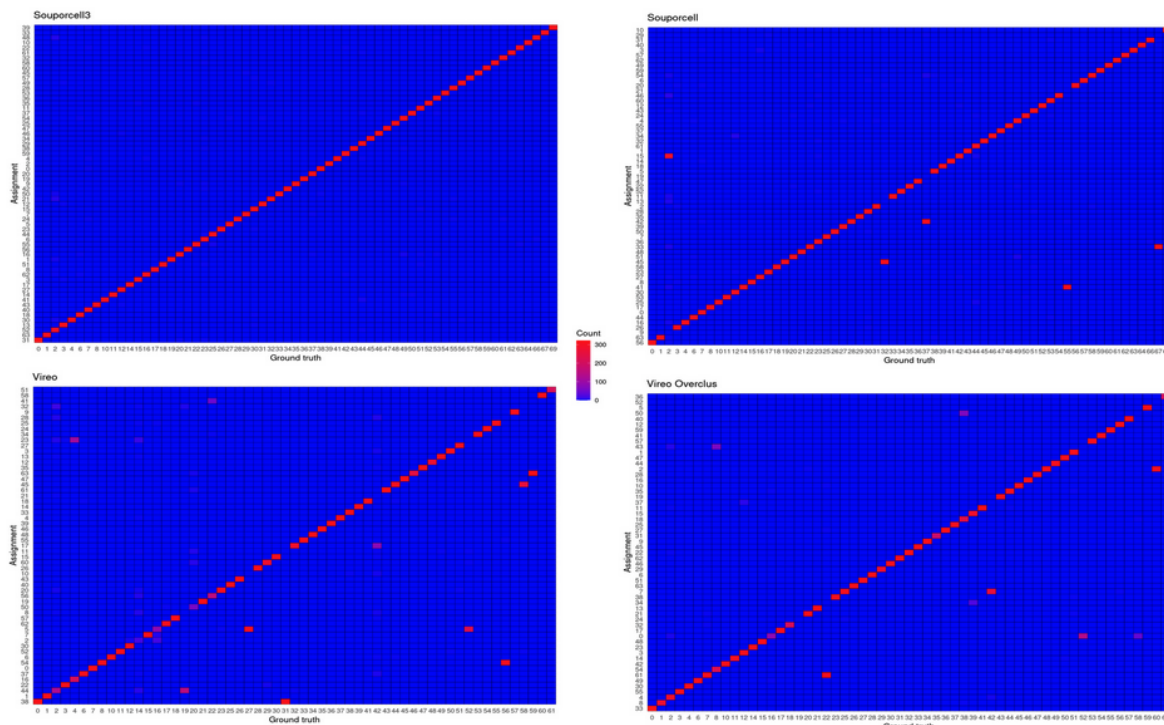


Figure 4.16: Heat maps showing cell counts for assigned cluster versus ground truth for (64) clusters of the (10 percent)-doublet balanced dataset after running Souporcell, Souporcell3, Vireo, and overclustered Vireo, prior to doublet detection.

Figure 4.16 shows heat maps of cell counts across 64 clusters for the 64-donor dataset with 10% doublets, with each row corresponding to the cluster assignment by the method and each column representing the ground-truth cluster. Color intensity reflects the number of cells assigned, with a median cell count per donor of 315. The Souporcell3 heat map shows a strong diagonal pattern, indicating that the majority of cells are correctly assigned to their respective clusters with minimal cross-contamination. The other methods show off-diagonal cell counts, indicating ground-truth clusters where two or more clusters are merged. The absence of significant off-diagonal cell counts in the Souporcell3 heat map

further supports its effectiveness in distinguishing between donors even in the presence of 10% doublets.

4.7.2 Unbalanced Dataset Results

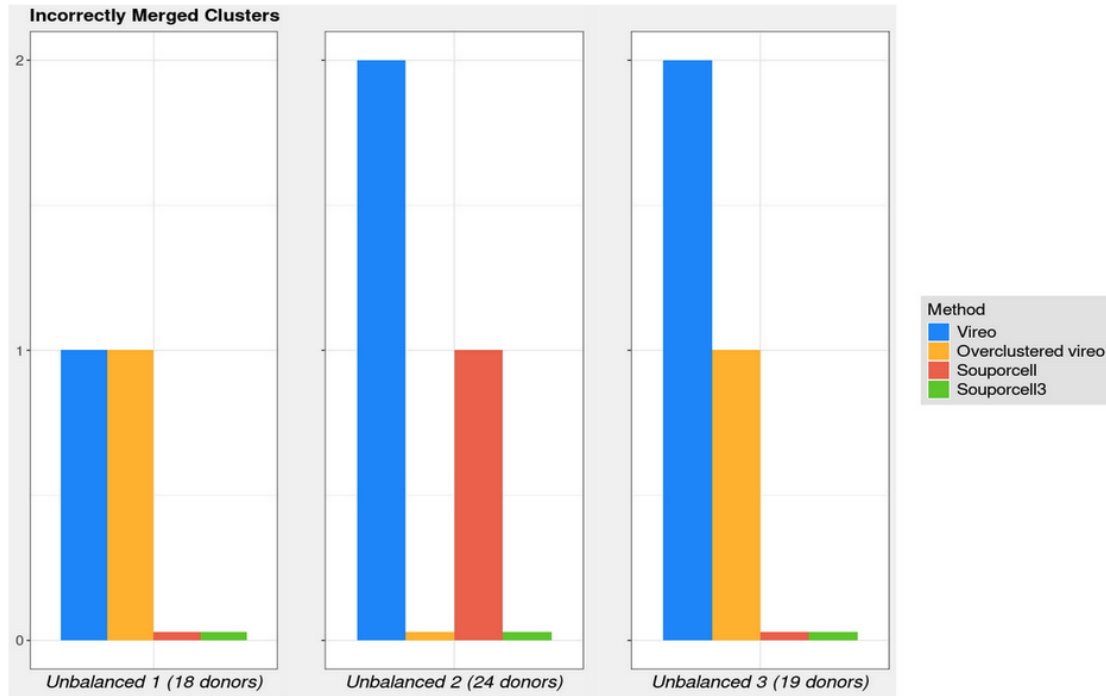


Figure 4.17: Number of incorrectly merged donor assignments for Vireo, overclustered Vireo, Souporcell, and Souporcell3 on the three unbalanced datasets.

Figure 4.17 compares the four methods on the three unbalanced datasets in terms of incorrectly merged clusters. Souporcell3 performs perfectly, producing zero incorrectly merged clusters across all three datasets. In Unbalanced 1, Vireo and overclustered Vireo each produce a single incorrectly merged cluster. In Unbalanced 2, Vireo and Souporcell each produce a single incorrectly merged cluster. In Unbalanced 3, Vireo produces two incorrectly merged clusters and overclustered Vireo produces one, while the other methods remain error-free.

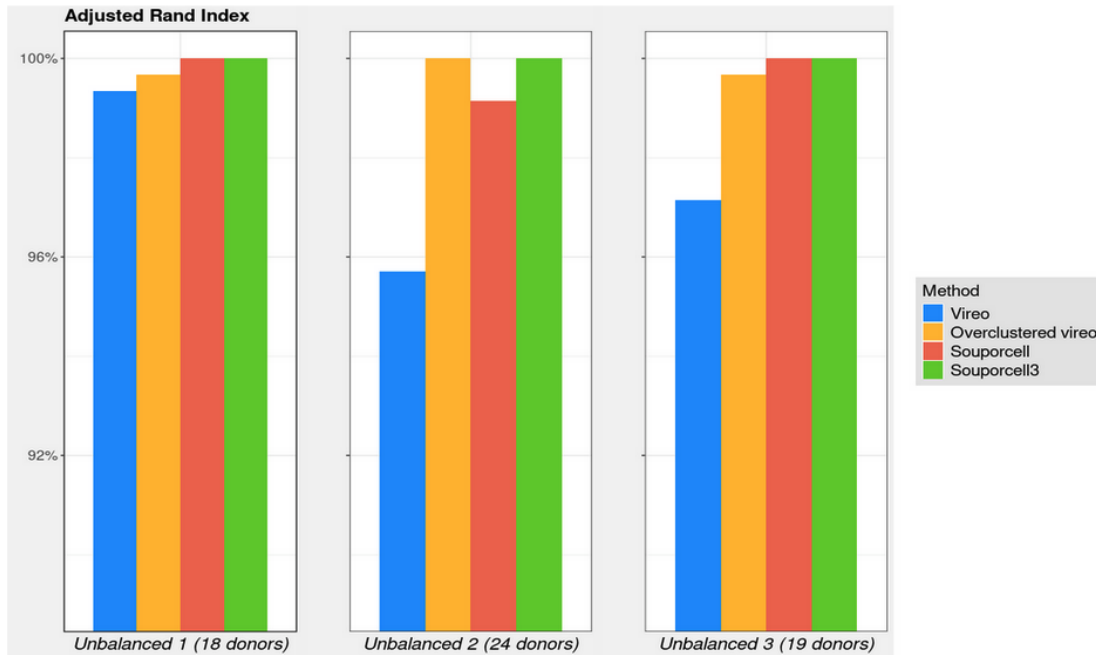


Figure 4.18: Adjusted Rand Index for Vireo, overclustered Vireo, Souporecell, and Souporecell3 on the three unbalanced datasets.

Figure 4.18 shows the ARI of the four methods on the unbalanced datasets. Souporecell3 achieves the highest ARI scores across all datasets, indicating nearly perfect clustering. Overclustered Vireo performs slightly worse, with moderately lower ARI values, while Vireo shows the lowest ARI, reflecting more frequent clustering errors. Notably, Souporecell3 achieves a perfect ARI of 1.0 on all three unbalanced datasets when evaluated against cells labeled as individual donors by Demuxlet, including on datasets with minority clusters as small as 0.84%, 1.4%, and 1.13% of the cells, respectively.

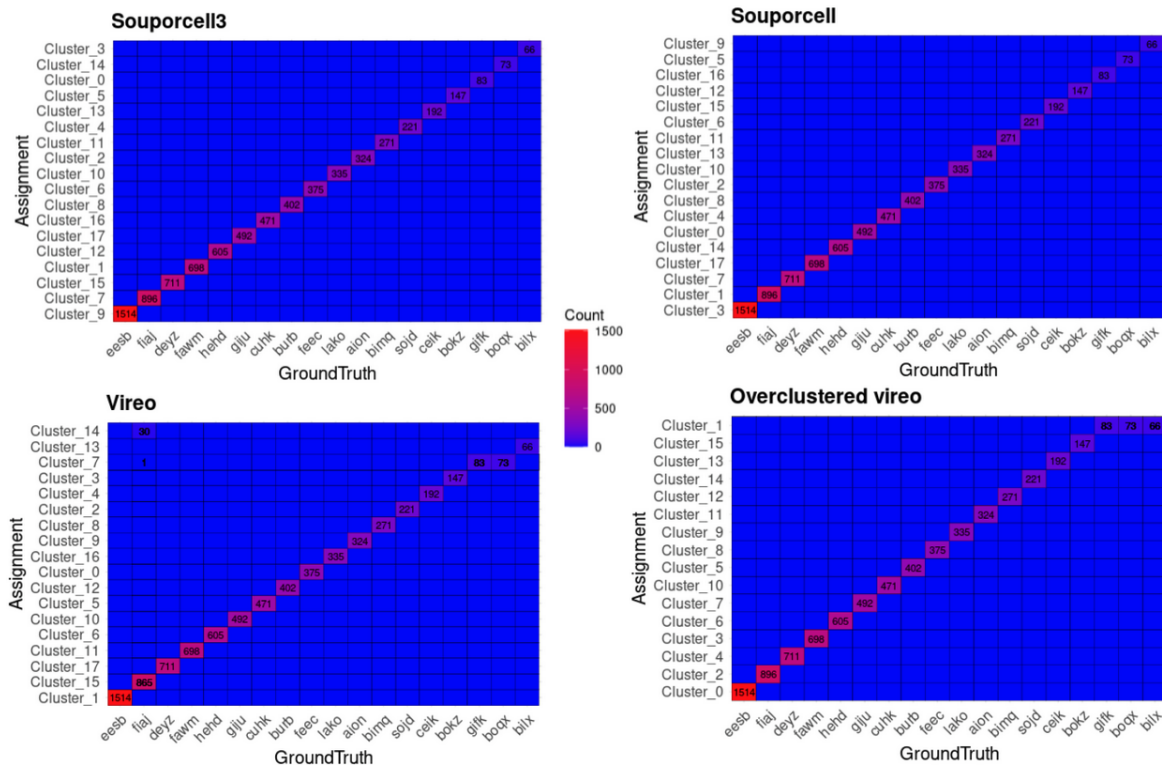


Figure 4.19: Heat maps showing cell counts for assigned cluster versus ground truth for unbalanced dataset 1, after running Souporcell, Souporcell3, Vireo, and overclustered Vireo.

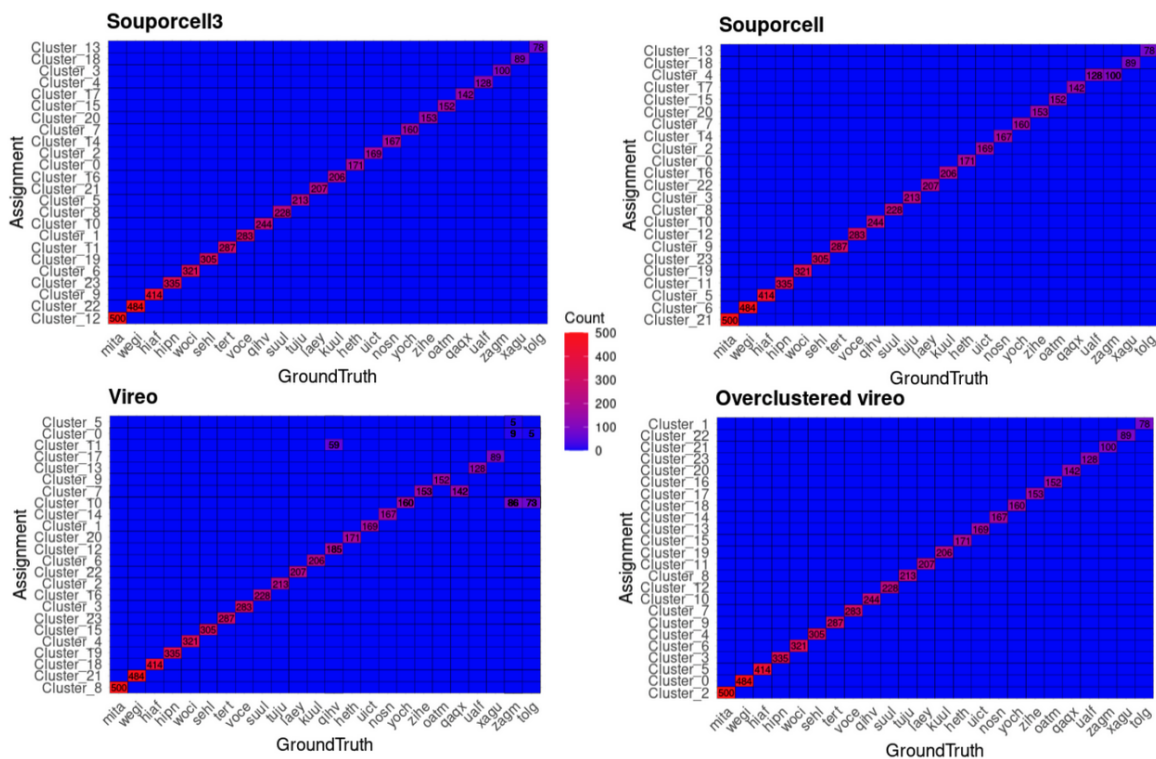


Figure 4.20: Heat maps showing cell counts for assigned cluster versus ground truth for unbalanced dataset 2, after running Souporecell, Souporecell3, Vireo, and overclustered Vireo.

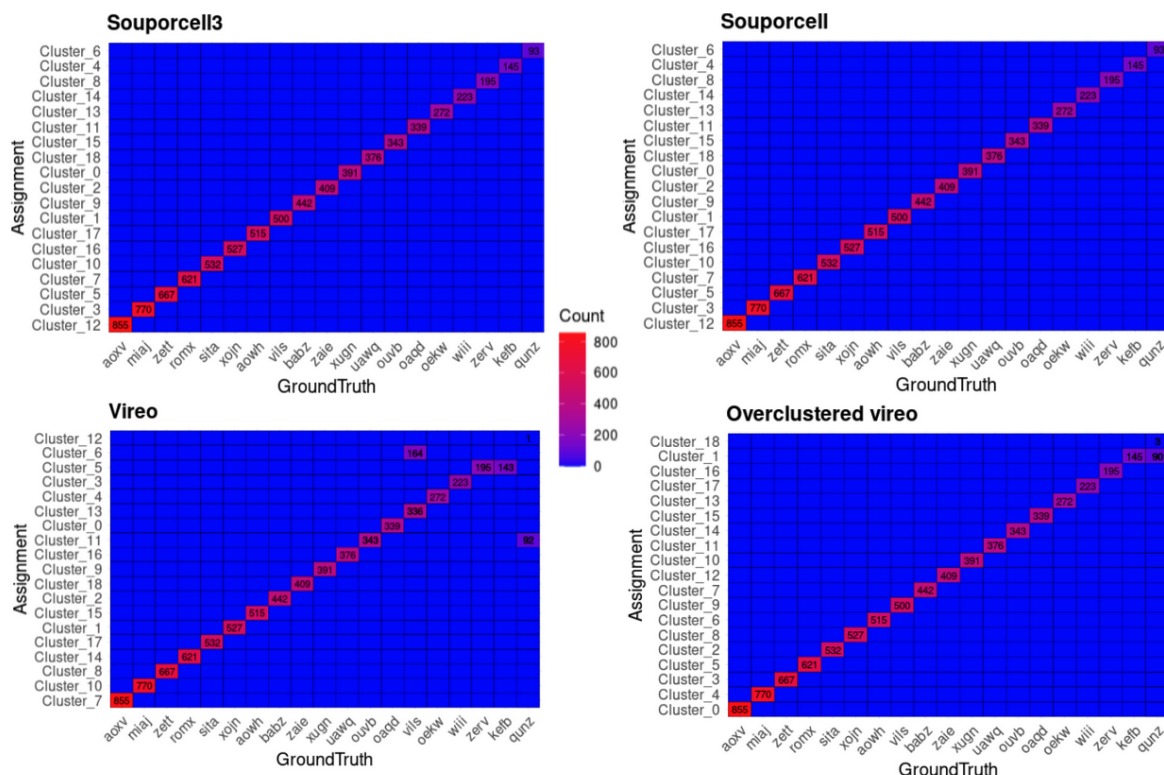


Figure 4.21: Heat maps showing cell counts for assigned cluster versus ground truth for unbalanced dataset 3, after running Souporcell, Souporcell3, Vireo, and overclustered Vireo.

Figure 4.19 Figure 4.20 Figure 4.21 present heat maps for each of the three unbalanced datasets, comparing the performance of the four methods. Across all datasets, Souporcell3 shows the most accurate clustering, with heat maps closely matching the true donor labels and minimal mixing between clusters. Souporcell also performs well, with only minor deviations in a few clusters. In contrast, Vireo and overclustered Vireo show more mixing and less clearly defined clusters. These results demonstrate that Souporcell3 provides the most precise and robust demultiplexing across all unbalanced datasets.

4.8 Discussion

The evaluation of genotype-based demultiplexing using both balanced synthetic and unbalanced experimental datasets demonstrates that the improved Souporcell3 approach

consistently enhances clustering performance. The method scales effectively to 64 donors while maintaining high accuracy under varying levels of doublets and across different dataset complexities. By combining 10x-merge initialization, deterministic annealing-enhanced K-Harmonic Means clustering, and iterative refinement with selective cluster locking, Souporecell3 eliminates common artifacts such as empty or incorrectly merged clusters and achieves higher concordance with ground-truth donor labels than Vireo [10], overclustered Vireo, or the original Souporecell [9].

The ablation studies presented in Sections 4.5 and 4.6 reveal that no single component is sufficient on its own. Even the *optimal* initialization, which uses ground-truth donor information to seed the cluster centers, does not yield perfect clustering on the 64-donor dataset, because subsequent log-loss updates can shift cluster centers toward dense regions of the genotype space and cause cluster mergers. Similarly, deterministic annealing alone (DA-KHM) improves over the vanilla baselines by a wide margin but still leaves a residual of incorrectly merged clusters. The iterative refinement procedure (Section 4.2.5), inspired by node freezing and dropout in deep learning [75, 85, 86], is required to close this gap by reinitializing the small subset of bad clusters while preserving the much larger set of correctly converged ones.

These improvements highlight the advantages of robust initialization and adaptive cluster correction strategies in high-throughput demultiplexing, and they have direct practical implications for the design of large-scale scRNA-seq studies. By raising the practical donor-count limit from approximately 16 to 64 donors per pool, Souporecell3 enables a roughly fourfold reduction in the per-donor cost of multiplexed experiments while simultaneously reducing technical batch effects and improving doublet detection statistics [9, 56–58, 64].

Several limitations of the current method merit discussion. First, the bad-cluster detection procedure described in Section 4.2.5 relies on the assumption that the cell distribution across donors is at least moderately balanced, or that minority clusters are nonetheless

distinguishable from empty clusters via cell-count thresholds. While the unbalanced results in Section 4.7.2 demonstrate robustness down to minority cluster fractions of approximately 0.84%, extreme imbalance regimes may require additional refinement criteria. Second, the negative binomial distance metric and KHM loss assume biallelic variants and treat each variant locus independently; extending the model to handle multi-allelic variants and linkage disequilibrium between nearby variants could further improve performance. Third, the deep-learning-inspired locking strategy is currently applied stochastically; principled criteria for selecting which clusters to lock at each iteration remain an open question for future work.

Future work will continue to refine the method in the context of different data types such as scATAC-seq and long-read scRNA-seq with even larger sample sizes and increasingly complex single-cell datasets, ensuring accurate and scalable demultiplexing for population-scale single-cell studies. The next chapter, Chapter 5, extends the clustering paradigm developed here to spatial single-cell technologies such as Slide-seq and Visium HD, where the assumptions underlying sequence-based demultiplexing do not directly apply and a separate methodological effort is required.

Chapter 5

Clustering Spatial scRNA-seq

This chapter presents *SpaceCell*, a three-stage clustering method for assigning cell types in spatial single-cell transcriptomics data. Section 5.1 motivates the need for spatially aware clustering and reviews current spatial transcriptomics technologies. Section 5.2 analyses two benchmark datasets, the Slide-seqV2 mouse cerebellum and the Visium human dorsolateral prefrontal cortex (DLPFC), comparing the negative binomial and GLM-PCA loss functions alongside k-means and Gaussian mixture model (Gaussian mixture model (GMM)) clustering. Section 5.3 describes the proposed method: an initial GMM clustering on a GLM-PCA embedding, a refinement step that selects high-confidence cells by negative binomial likelihood, and a final assignment step applying Leiden community detection to a graph combining transcriptomic and spatial distances. Section 5.4 presents results on the cerebellum sample, where SpaceCell reaches an adjusted Rand index of 0.72 against 0.38 for CellCharter. Section 5.5 discusses these results and the evaluations currently in progress.

5.1 Introduction

Single-cell RNA sequencing (scRNA-seq) has greatly improved our understanding of cellular diversity by allowing researchers to measure gene expression in individual cells. This has revealed previously unknown cell states, lineages, and rare cell populations across many tissues and diseases [89–91]. However, preparing tissue for scRNA-seq requires cells to be separated from their original environment, which removes important

spatial information. Tissues are highly organised structures where neighbouring cells interact through signalling, mechanical forces, and shared developmental history. Without information about cell location, scRNA-seq alone cannot fully address questions about tissue structure, cellular niches, or spatial signalling patterns [92, 93].

Spatial transcriptomics overcomes this limitation by preserving the spatial location of each cell or transcript while still measuring genome-wide gene expression [94, 95]. Unlike scRNA-seq, spatial transcriptomics includes coordinates for every measurement, allowing researchers to study gene expression together with tissue structure. This changes downstream analysis because methods such as clustering, differential expression, and trajectory inference must account for spatial relationships. It also enables new types of analysis, including identifying tissue boundaries, defining cellular niches, and studying cell–cell communication [96, 97].

5.1.1 Spatial transcriptomics technologies

Several spatial transcriptomics technologies have been developed, each with different trade-offs between spatial resolution, sensitivity, and transcriptome coverage. In this work, we focus on sequencing-based spatial transcriptomics methods, where tissue sections are placed onto surfaces containing spatially barcoded oligonucleotides that capture RNA molecules while preserving their spatial location. Sequencing then produces genome-wide gene expression profiles linked to spatial coordinates. The datasets analysed in this study include human cerebellum data generated using Visium and mouse cerebellum data generated using Slide-seqV2 technologies.

The Visium dataset was obtained from the study by Maynard et al. [98], which used the 10x Genomics Visium platform to generate spatial transcriptomic profiles from human brain tissue. Visium uses barcoded capture spots of approximately 55 μm arranged on a hexagonal grid, with each spot typically capturing transcripts from multiple neighbouring cells.

The Slide-seqV2 cerebellum dataset was obtained from the study by Cable et al. [99], which analysed mouse cerebellum tissue at near-cellular resolution using Slide-seqV2 [100, 101]. Slide-seqV2 uses densely packed $\sim 10\text{ }\mu\text{m}$ beads coated with spatial barcodes, enabling finer spatial resolution than Visium, although with lower transcript capture efficiency per bead.

Both Visium and Slide-seqV2 are untargeted, capture-based methods that measure the whole transcriptome without requiring predefined gene panels. However, both technologies produce sparse data, where many genes have zero observed counts in individual spots or beads. In addition, both spatial technologies share the common downstream challenge of integrating transcriptional and spatial information into a single coherent representation of tissue structure.

5.2 Analysis of Data

To evaluate whether spatial transcriptomics data permit the recovery of annotated tissue domains under standard probabilistic models, we examined two datasets that span the two dominant capture-based platforms and two anatomically layered brain regions. Both were chosen because their ground-truth annotations are derived from well-established histology and have been used extensively as benchmarks in the spatial clustering literature.

5.2.1 Slide-seqV2 mouse cerebellum data

Paper. “Cell type-specific inference of differential expression in spatial transcriptomics,” *Nature Methods* 19, 1076–1087 [99].

Sample. Adult mouse cerebellum. The cerebellum is a deeply stratified structure organised into the molecular layer, the Purkinje cell layer (a thin band of large GABAergic

projection neurons), the densely packed granule cell layer, and the underlying white matter [102]. This stereotyped lamination, combined with the very distinct cell type compositions of each layer, makes the cerebellum a canonical positive control for any spatial clustering or annotation method: a method that cannot recover the cerebellar layers is unlikely to perform well on more diffusely organised tissues.

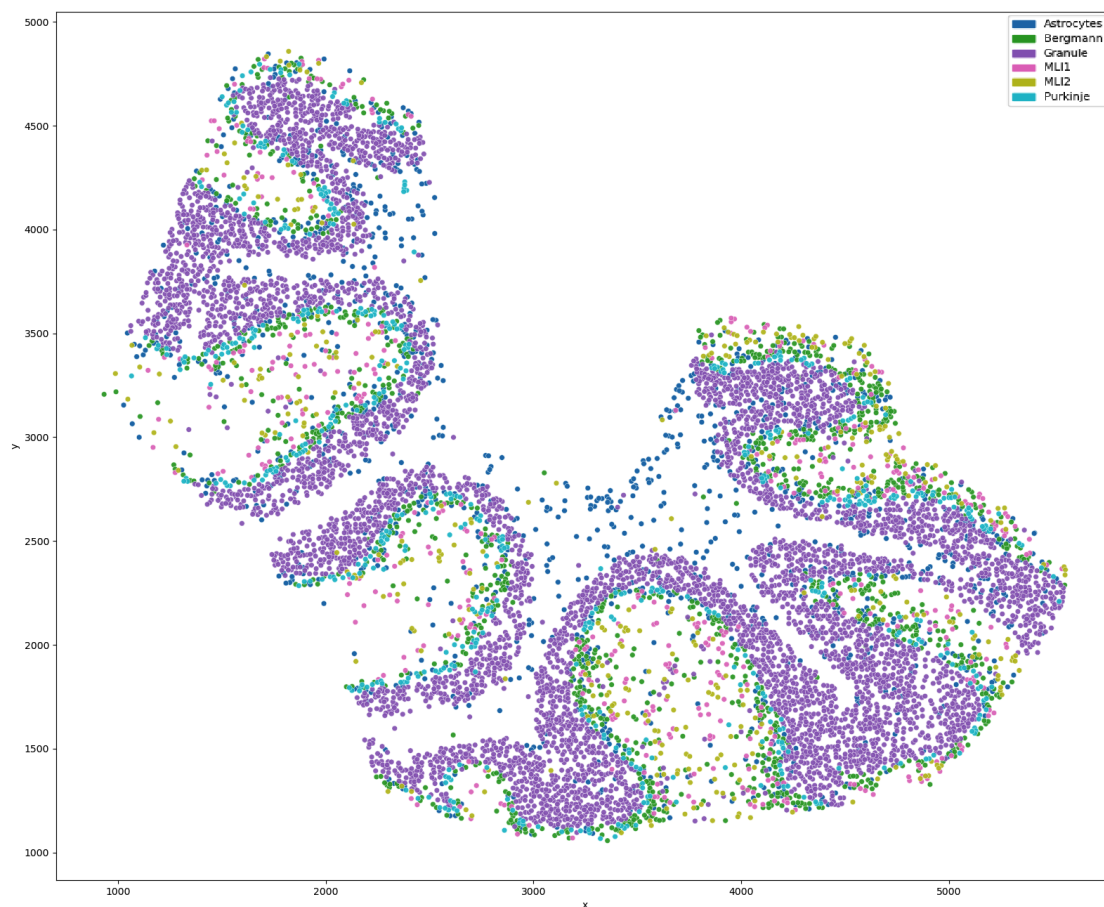


Figure 5.1: Ground-truth annotation of the Slide-seqV2 mouse cerebellum sample, with cells coloured by their assigned cell type.

Annotation. Per-bead annotations were derived by deconvolving each bead into its constituent cell types using RCTD [99] with a comprehensive single-nucleus reference atlas of the mouse cerebellar cortex [103], and assigning beads to laminar domains based on the spatial distribution of the dominant cell types. This yields ground-truth labels for the molecular layer, Purkinje layer, granule layer, and white matter, together with rarer populations such as Bergmann glia and oligodendrocytes.

Goal. Using the annotation as cluster initialisation, we evaluate whether the cerebellar layers are separable in expression space under a suite of probabilistic models, including raw count likelihoods under negative binomial assumptions and reduced-dimensional embeddings such as GLM-PCA [104]. The question we ask is intentionally narrow and pre-clustering: *if a clustering algorithm were given the correct centres, would the data themselves admit separation under each model?*

Negative binomial likelihood-based cell separation

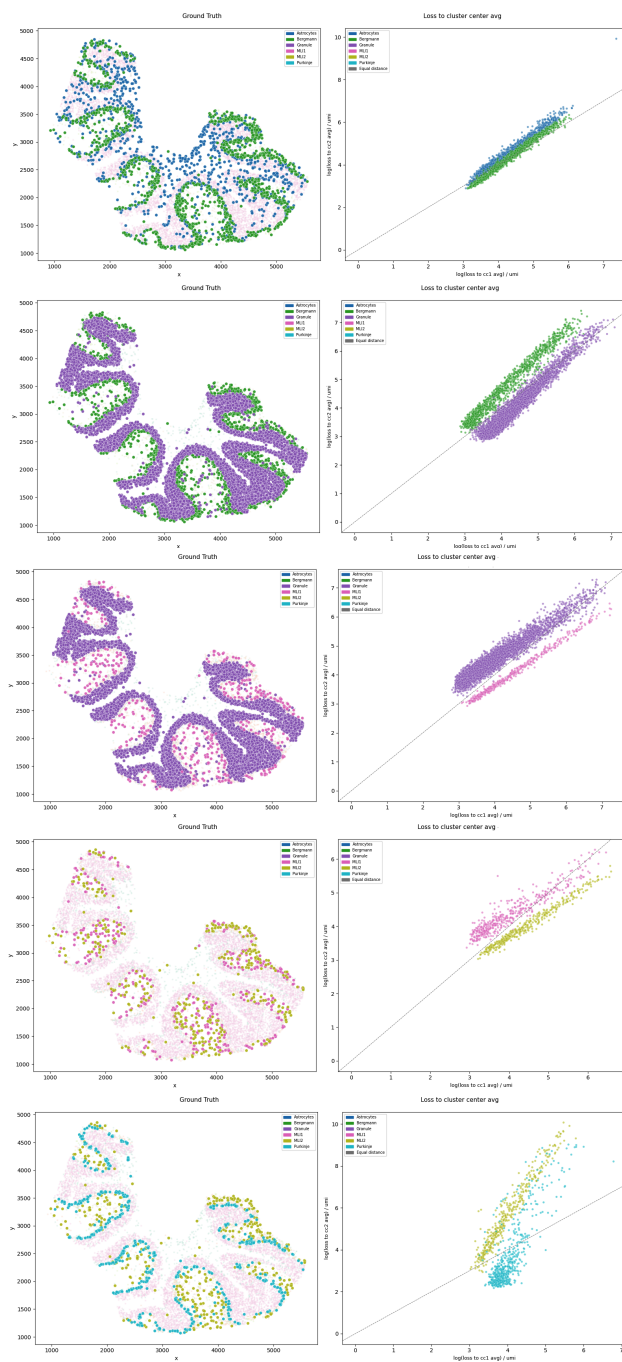


Figure 5.2: Negative binomial likelihood-based cell separation for each cell type in the mouse cerebellum sample. Left: spatial view with cells coloured by ground-truth label. Right: scatter of own-cluster loss against other-cluster loss; cells below the diagonal are closer to their assigned cluster's average.

GLM-PCA based cell separation

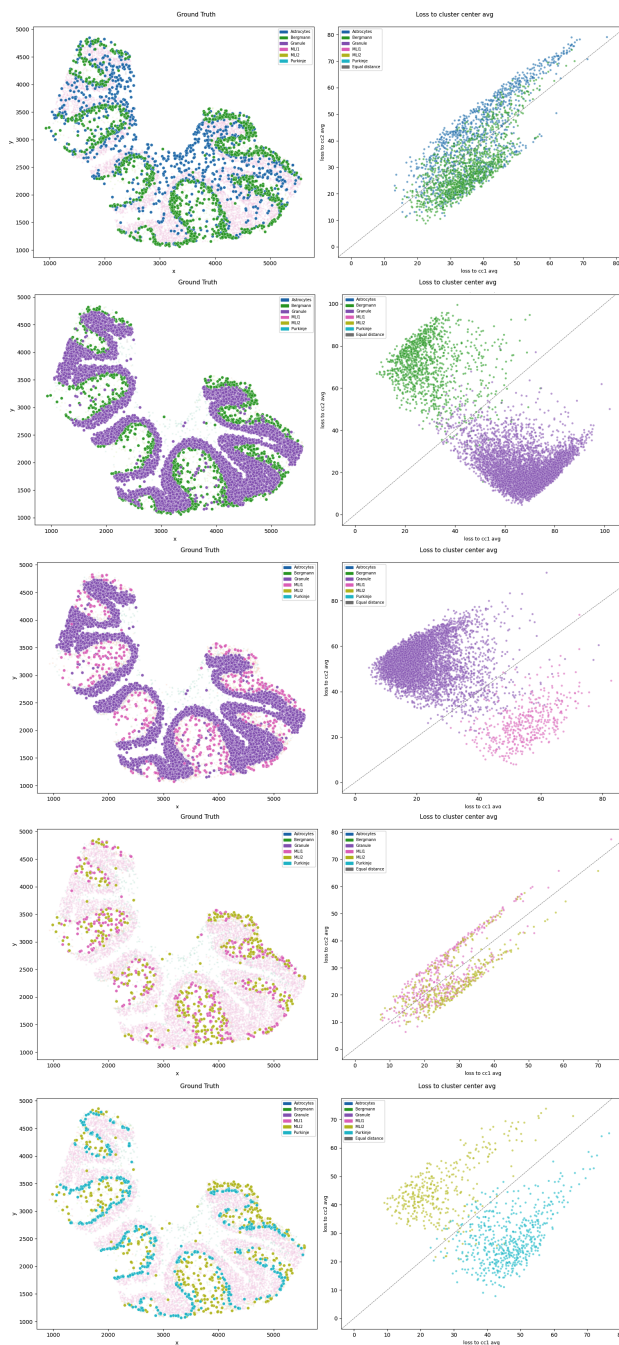


Figure 5.3: GLM-PCA based cell separation for each cell type in the mouse cerebellum sample. Cells more widely dispersed in the loss space than under the NB model, but a meaningful fraction sit on the wrong side of the diagonal.

K-means++ clustering using GLM-PCA

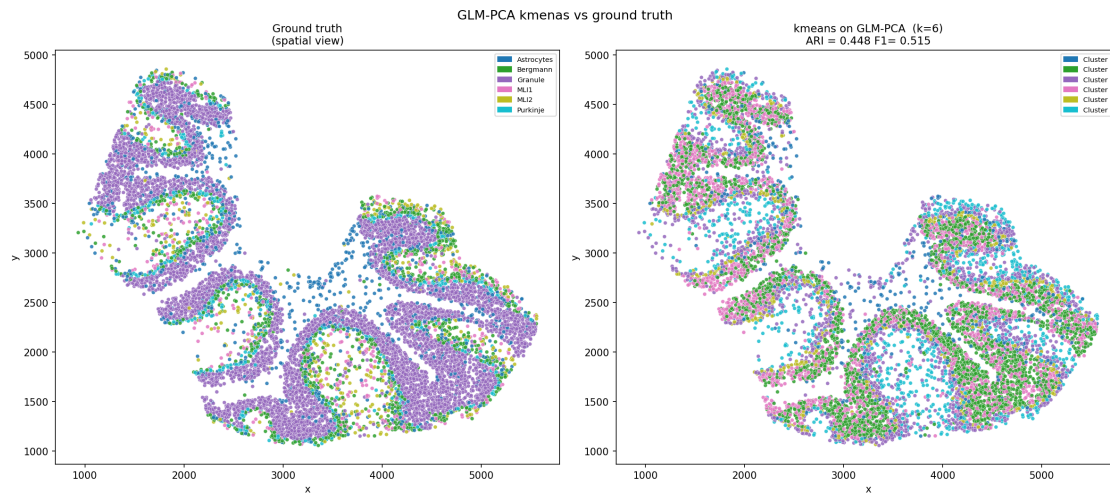


Figure 5.4: K-means++ clustering on the GLM-PCA embedding of the cerebellum data. Ground truth (left) compared to k-means++ cluster assignments (right). ARI = 0.448, F1 = 0.515.

Gaussian mixture model clustering using GLM-PCA

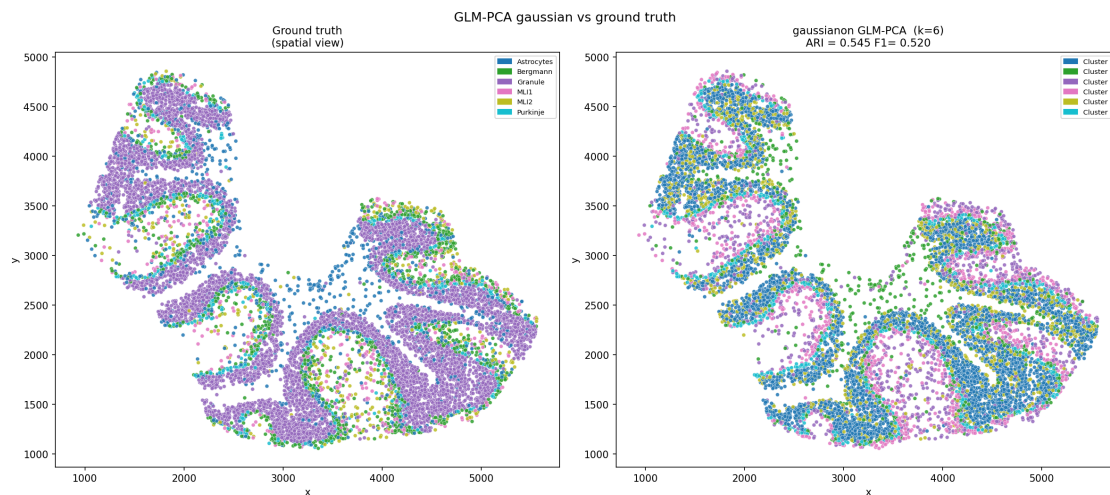


Figure 5.5: Gaussian mixture model clustering on the GLM-PCA embedding of the cerebellum data. Ground truth (left) compared to GMM cluster assignments (right). ARI = 0.545, F1 = 0.520.

5.2.2 Visium human dorsolateral prefrontal cortex data

Paper. “Transcriptome-scale spatial gene expression in the human dorsolateral prefrontal cortex,” *Nature Neuroscience* 24, 425–436 [98].

Sample. The study profiled 12 Visium sections from three neurotypical adult donors (four sections per donor, sampled from the DLPFC, Brodmann area 9/46), each spanning the full cortical depth and capturing all six neocortical laminae (L1–L6) plus the underlying white matter (WM), seven anatomically defined layers in total. For this analysis we use section 151673, the most commonly benchmarked section across recent spatial clustering methods [98].

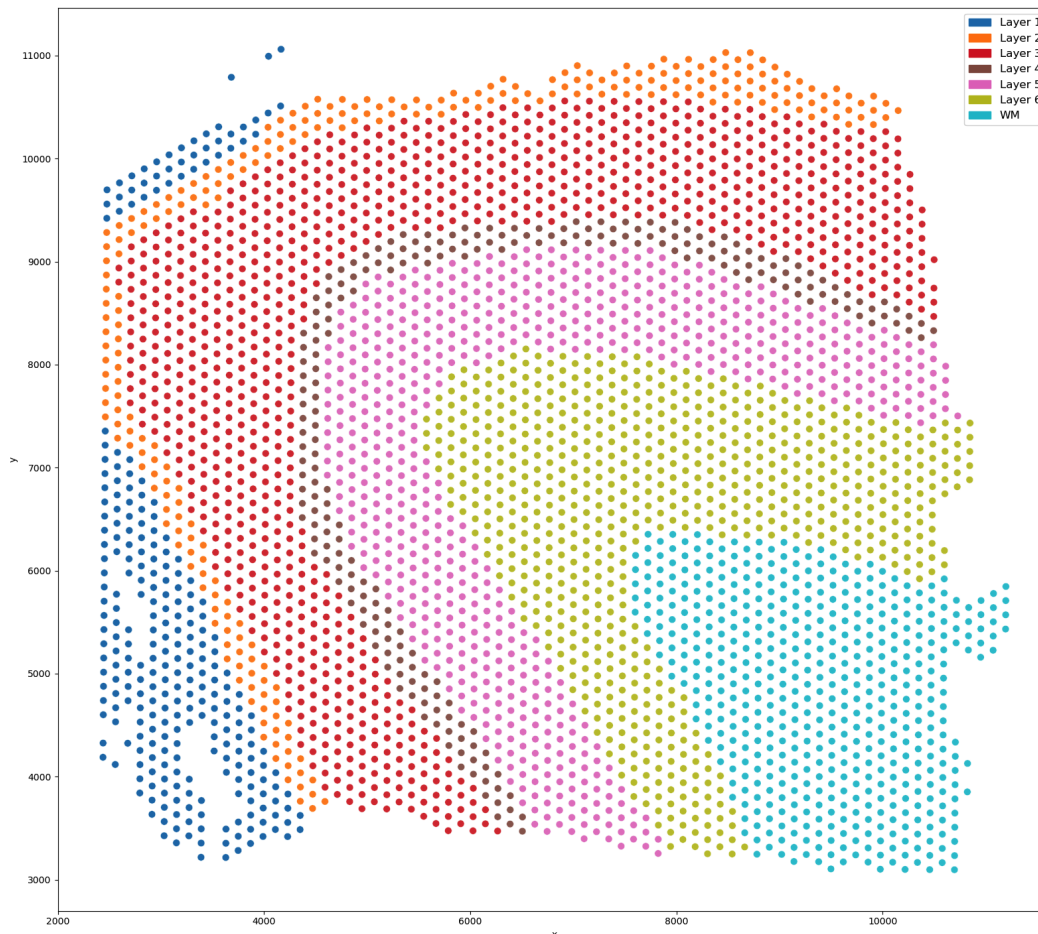


Figure 5.6: Ground-truth annotation of Visium DLPFC section 151673, with spots coloured by manually assigned cortical layer (L1–L6) and white matter (WM).

Annotation. Layers were manually annotated on a per-spot basis by domain experts using the accompanying H&E-stained images and known cytoarchitectural landmarks of the DLPFC [105, 106], guided by canonical laminar marker genes such as *AQP4* (L1), *HPCAL1* (L2/3), *BCL11B* / *CTIP2* (L5), *KRT17* (L6), and *MBP* (WM) [98, 107, 108]. The resulting layer-level ground truth, L1, L2, L3, L4, L5, L6, and WM, has subsequently become the de facto benchmark annotation for spatial clustering methods, used in essentially every recent methodological paper in the field [109–113].

Goal. As with the cerebellum analysis, the DLPFC sections are used to test whether the seven annotated cortical layers are separable in expression space under negative binomial and GLM-PCA representations of the data, with cluster centres initialised from the manual annotations. The DLPFC complements the cerebellum analysis in an important way: the cerebellar layers contain fundamentally distinct cell types (granule cells vs. Purkinje cells vs. Bergmann glia), whereas the six cortical layers contain a more graded continuum of pyramidal projection neurons and interneurons that share large parts of their transcriptional programme [108, 114]. The DLPFC therefore represents the harder case, a layered tissue with transcriptionally similar but spatially distinct domains, and tests the limits of common modelling assumptions.

Negative binomial likelihood-based cell separation

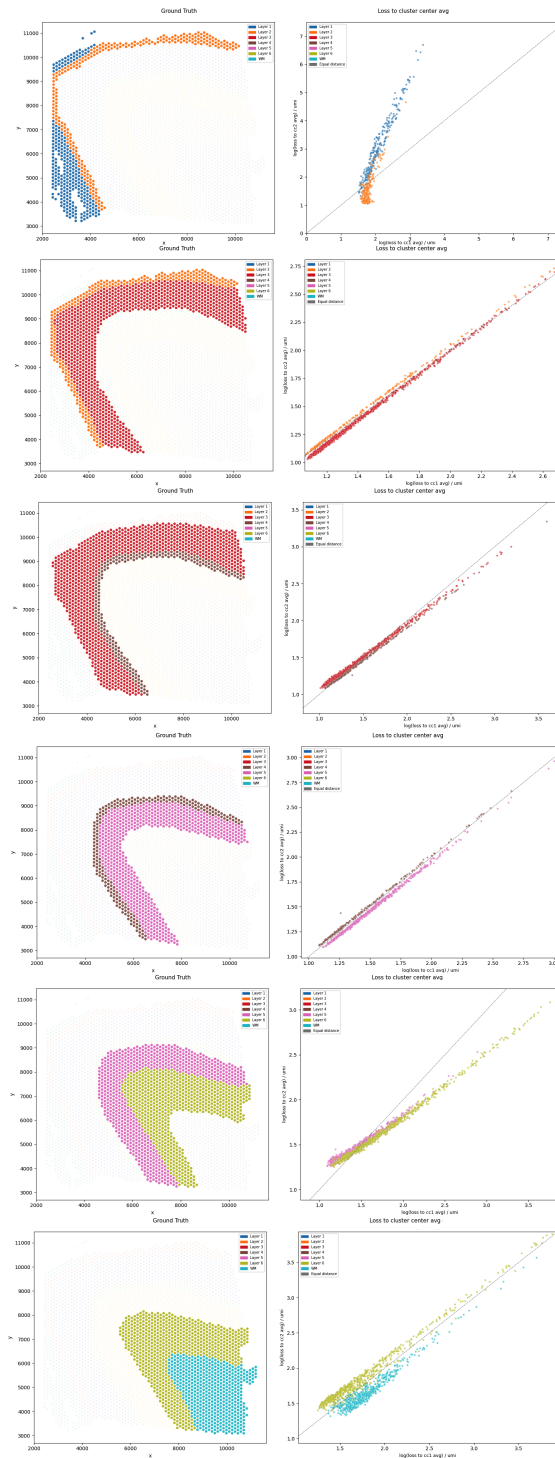


Figure 5.7: Negative binomial likelihood-based cell separation for each layer in the DLPFC sample. Spots fall very close to the diagonal but consistently on the correct side.

GLM-PCA based cell separation

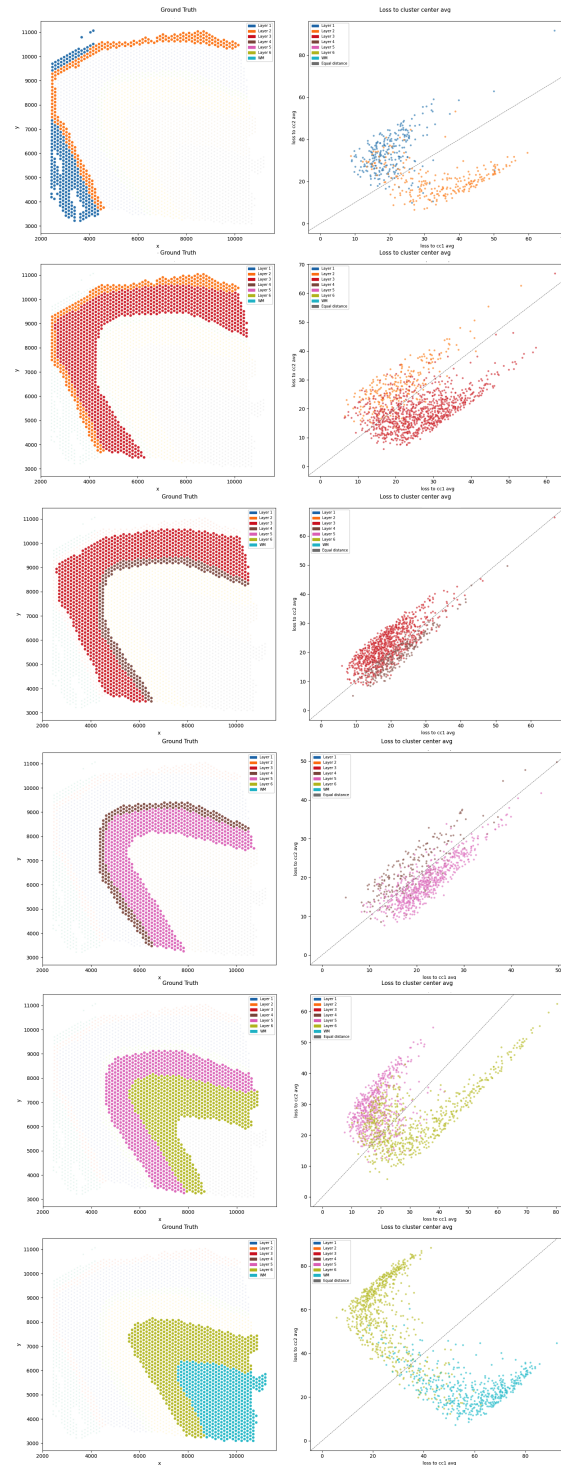


Figure 5.8: GLM-PCA based cell separation for each layer in the DLPFC sample.

K-means++ clustering using GLM-PCA

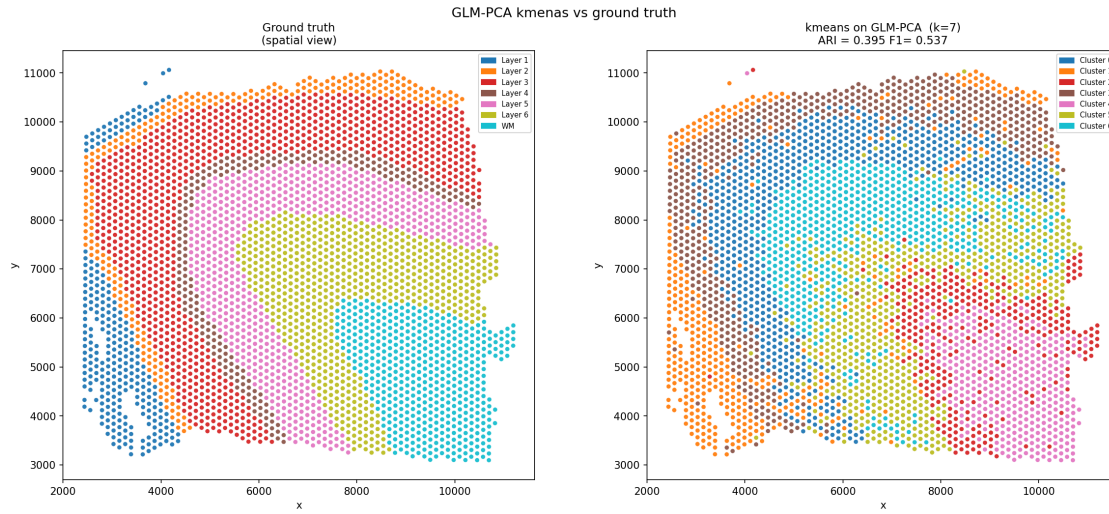


Figure 5.9: K-means++ clustering on the GLM-PCA embedding of the DLPFC data. ARI = 0.395, F1 = 0.537.

Gaussian mixture model clustering using GLM-PCA

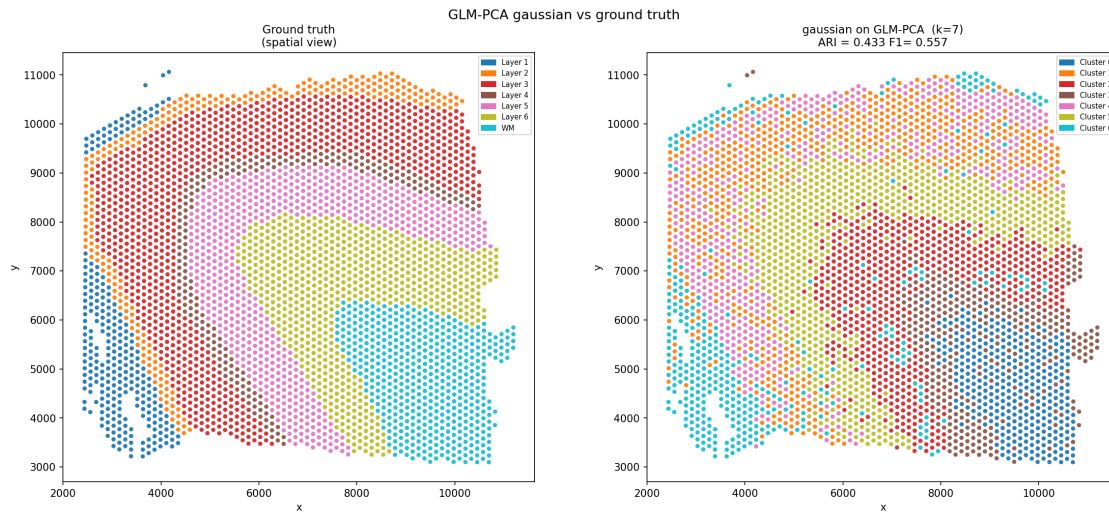


Figure 5.10: Gaussian mixture model clustering on the GLM-PCA embedding of the DLPFC data. ARI = 0.433, F1 = 0.557.

5.2.3 Discussion of the loss functions and clustering methods

The cell separation analyses of the two datasets in Sections 5.2.1 and 5.2.2 show that the two probabilistic models have complementary strengths and weaknesses.

GLM-PCA loss spreads cells widely across the loss space. Most cells fall on the correct side of the diagonal, closer to their own cluster's average than to a competing cluster's average, but a meaningful fraction sit on the wrong side. These mis-localised cells are usually either at the spatial border between two layers or belong to rarer cell populations. GLM-PCA is therefore good at producing a well-separated global embedding, but it can be insensitive to small gene-level differences between transcriptionally similar layers.

Negative binomial loss behaves in the opposite way. Almost all cells sit very close to the diagonal, meaning the loss to the own cluster and the loss to the competing cluster are numerically similar. However, the direction is almost always correct: cells consistently fall on the right side of the diagonal even if only slightly. This is the expected behaviour of a likelihood-based count model on sparse data; only a few genuinely layer-discriminative genes push the total in the right direction. The negative binomial loss is therefore precise but has a narrow range, making it a poor primary clustering signal but an excellent refinement signal.

This suggests a two-stage strategy: use GLM-PCA first to get good large-scale separation, then refine with the negative binomial loss to capture the residual gene-level signal that distinguishes similar layers.

When comparing the two clustering algorithms on the GLM-PCA embedding, the Gaussian mixture model outperforms k-means++ on both datasets (ARI 0.545 / F1 0.520 vs. 0.448 / 0.515 on the cerebellum; ARI 0.433 / F1 0.557 vs. 0.395 / 0.537 on the DLPFC). This is consistent with the underlying assumptions of each method: k-means++ assumes spherical, equal-variance clusters with hard assignments, whereas GMM allows each cluster to have its own ellipsoidal shape and produces soft probabilistic assignments. For spatial transcriptomic clusters that are typically elongated and overlapping in embedding

space, this flexibility is a noticeable advantage. These observations directly motivate the method developed below.

5.3 Method

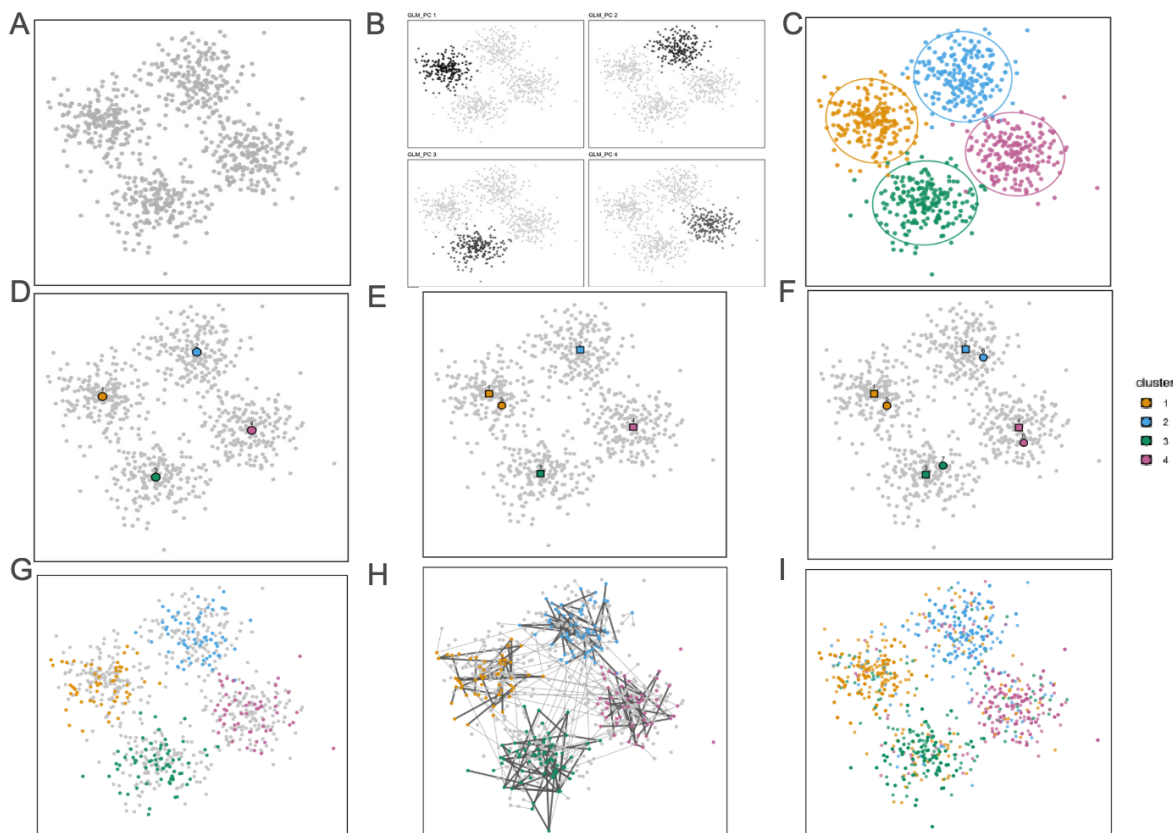


Figure 5.11: Method overview. **(A–C)** Cluster with a Gaussian mixture model on the GLM-PCA embedding of the count matrix and spatial coordinates. **(D–F)** Refine with negative binomial likelihood-based cell selection. **(G–I)** Assign the unassigned cells using Leiden community detection, connecting neighbouring cells with a combined negative binomial and spatial loss.

5.3.1 Initial clustering with a Gaussian mixture model using GLM-PCA

We begin by computing a low-dimensional embedding of the count matrix using GLM-PCA [104], a dimensionality reduction technique designed for sparse count data. Standard PCA assumes Gaussian noise and is typically applied after log-normalisation, which introduces artefacts when the data are sparse and overdispersed, as UMI counts almost

always are [115]. GLM-PCA avoids this by fitting an exponential-family likelihood directly to the raw counts, producing an embedding that better preserves biologically meaningful variation.

On this embedding we fit a Gaussian mixture model [116, 117], which describes the data as a weighted sum of multivariate Gaussian components fit by the expectation-maximisation algorithm. Each cell receives a probability of belonging to each component, and the most likely component is taken as its initial cluster label. Unlike k-means, which assumes spherical equal-variance clusters and produces hard assignments, the GMM allows each cluster its own ellipsoidal shape, a better fit for the overlapping clusters typical of spatial transcriptomic data, as confirmed by the ARI and F1 comparisons in Section 5.2.3.

5.3.2 Refining with negative binomial loss

The negative binomial distribution is the standard model for UMI counts because gene-count variance consistently exceeds the mean, a property known as overdispersion [118–120]. It generalises the Poisson by introducing a per-gene dispersion parameter:

$$P(y \mid \mu_g, \theta_g) = \frac{\Gamma(y + \theta_g)}{\Gamma(\theta_g) y!} \left(\frac{\theta_g}{\theta_g + \mu_g} \right)^{\theta_g} \left(\frac{\mu_g}{\theta_g + \mu_g} \right)^y \quad (5.1)$$

where y is the observed count of a gene in a cell, μ_g is the expected mean count for gene g , θ_g is the dispersion parameter for gene g , and $\Gamma(\cdot)$ is the gamma function.

For each cluster, we compute its average count vector across all member cells and the UMI-normalised negative log-likelihood of each cell's counts under that average:

$$\mathcal{L}_{i \rightarrow k} = -\frac{1}{T_i} \sum_{g=1}^G \log P(C_{ig} \mid \bar{\mu}_g^{(k)}, \theta_g) \quad (5.2)$$

where $\bar{\mu}_g^{(k)}$ is the average count of gene g across cells in cluster k , C_{ig} is the observed count of gene g in cell i , $\mathcal{L}_{i \rightarrow k}$ is the loss of cell i against cluster k 's average, $T_i = \sum_g C_{ig}$ is the total UMI count of cell i , G is the total number of genes, and $P(\cdot)$ is the negative

binomial probability mass function from Equation 5.1.

The UMI normalisation makes losses comparable across cells of different sequencing depth. From this we define an own-cluster loss, evaluated against the cell's currently assigned cluster, and an other-cluster loss, evaluated against the closest competing cluster. The quality of the partition is then summarised by a scale-invariant discriminability score that takes the ratio of mean other-cluster loss to mean own-cluster loss:

$$\mathcal{S} = \frac{\text{mean}(\mathcal{L}^{\text{other}})}{\text{mean}(\mathcal{L}^{\text{own}}) + \varepsilon} \quad (5.3)$$

where $\mathcal{S}$ is the discriminability score of the partition, $\mathcal{L}_i^{\text{own}}$ is the own-cluster loss (the loss of cell i against its currently assigned cluster), $\mathcal{L}_i^{\text{other}}$ is the other-cluster loss (the minimum loss of cell i across all competing clusters), $\text{mean}(\cdot)$ is the average taken across all cells under consideration, and ε is a small constant to avoid division by zero.

Refinement proceeds in two phases. First, we run the GMM with multiple random initialisations and retain the partition with the highest score. Second, starting from this partition, we sample one cell per cluster as a seed and iteratively add cells one at a time, each time selecting the candidate whose addition produces the largest increase in the score and recomputing the cluster averages after each step. Cells that no longer improve the score, or that prefer a different cluster, are left unassigned and passed to the next stage.

5.3.3 Assigning the unassigned cells with the Leiden algorithm

The remaining unassigned cells are placed using Leiden community detection [121], a graph clustering algorithm that improves on the widely-used Louvain method [122] by guaranteeing internally connected communities and producing higher-quality partitions in practice. Leiden has become the standard for community detection in single-cell and spatial transcriptomic analyses.

We construct a graph in which each cell is a node and the edge weight between two cells combines transcriptomic and spatial proximity:

$$w_{ij} = \mathcal{L}_{ij}^{\text{NB}} + \lambda \cdot \|s_i - s_j\|_2 \quad (5.4)$$

where w_{ij} is the edge weight between cell i and cell j , $\mathcal{L}_{ij}^{\text{NB}}$ is the negative binomial loss of cell i 's counts evaluated against cell j 's count vector, s_i and s_j are the 2D spatial coordinates of cells i and j , $\|s_i - s_j\|_2$ is the Euclidean distance between the two cells, and λ is a tunable weight controlling the trade-off between transcriptomic and spatial similarity.

For pairs of cells already assigned to the same cluster in Section 5.3.2, the weight is set to zero so they remain strongly connected and anchor the community structure. The weights are then inverted before being passed to Leiden, since the algorithm treats higher weights as stronger similarity, and the resulting communities define the final cluster assignments.

5.4 Results

We evaluated SpaceCell on the Slide-seqV2 mouse cerebellum dataset described in Section 5.2.1, using the six annotated cell types as ground truth. Performance is reported using the Adjusted Rand index (ARI) against the published annotations. As a baseline we ran CellCharter [123], a recent spatial clustering method that combines a variational autoencoder embedding with neighbourhood aggregation and Gaussian mixture clustering, using the same number of clusters and the same preprocessed count matrix.

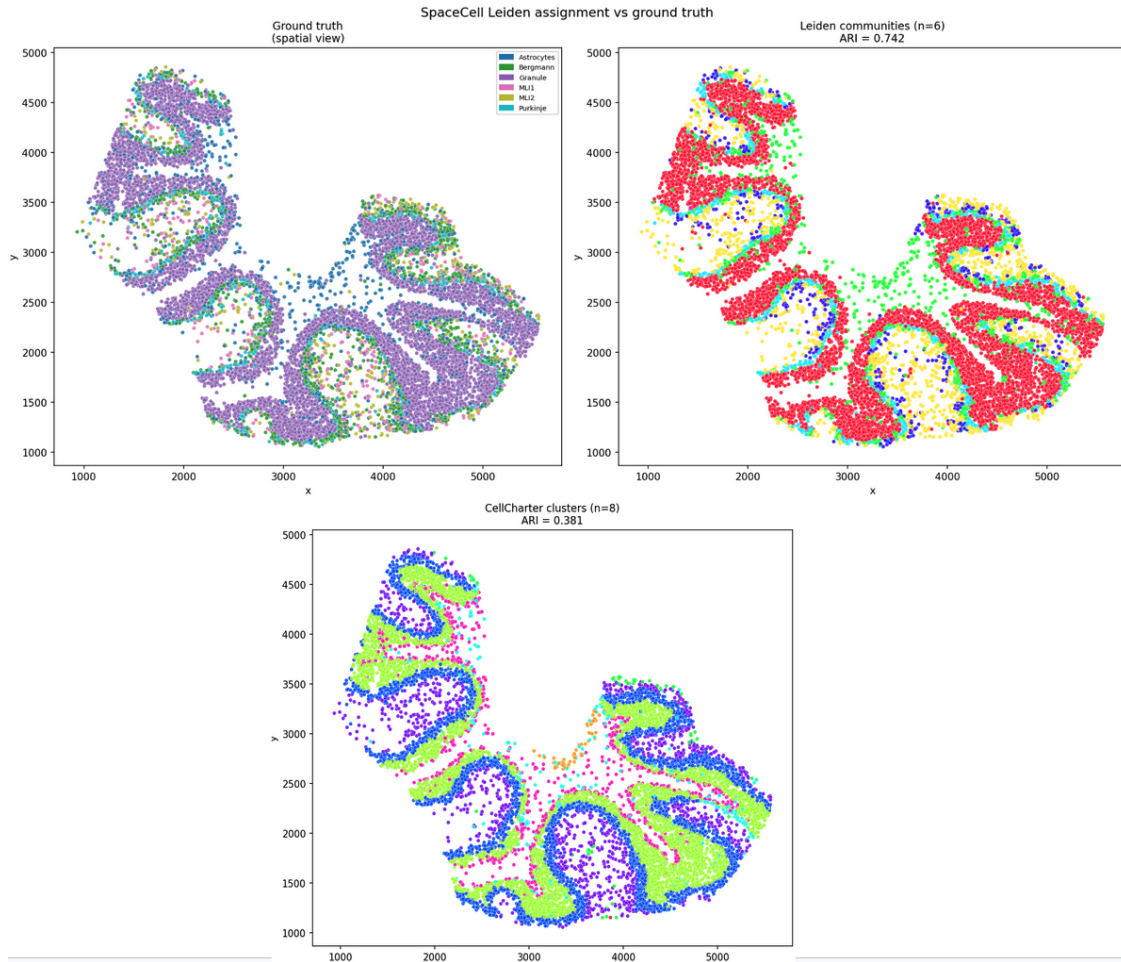


Figure 5.12: Clustering of the Slide-seqV2 mouse cerebellum sample. SpaceCell recovers spatially coherent layers that closely follow the annotated cerebellar architecture, whereas the CellCharter assignments are more fragmented across layer boundaries.

SpaceCell achieved an ARI of 0.74 on the cerebellum sample, compared with 0.38 for CellCharter, an improvement of 0.34 in absolute terms and close to a two-fold increase relative to the baseline.

5.5 Discussion

The cerebellum result supports the main design choice of SpaceCell: the negative binomial and GLM-PCA losses work better at different stages than combined into one objective. As shown in Section 5.2, GLM-PCA separates cells widely but coarsely, so it

suits the initial clustering, while the negative binomial loss is precise over a narrow range, so it suits refinement. Applying each loss where it performs best, and adding the spatial term only at the final assignment step, gives a clearly better partition than the baseline.

This evaluation has limits. It uses one Slide-seqV2 sample and one baseline, so broader testing is needed before drawing general conclusions. Work on the remaining samples is in progress, including the Visium human dorsolateral prefrontal cortex section from Section 5.2.2 and further cerebellum replicates. The cortex is expected to be harder, since its layers change gradually rather than forming distinct cell types, so the margin between methods may be smaller there.

More baseline comparisons are also in progress. Alongside CellCharter, we are running GASTON [113], GraphST [112], STAGATE [111], BayesSpace [110], and SpaGCN [109] on the same data with matched preprocessing and cluster counts. These cover the main approaches in the field, from Bayesian models to graph neural networks, and will show more clearly how the proposed method compares to the state of the art.

Chapter 6

Conclusion

This dissertation has presented four algorithmic contributions that target distinct but complementary problems in modern sequence analysis, spanning long-read consensus sequencing and droplet-based and spatial single-cell transcriptomics. Together, the four methods, *TopoQual* [2], *LCSkPOA* [3], *Souporcell3* [12], and *SpaceCell*, improve the accuracy, computational efficiency, and scalability of pipelines that turn raw sequencing measurements into the variant calls and cell-type assignments that drive downstream biological and clinical inference.

6.1 Summary of Contributions

Chapter 2: TopoQual. The per-base quality scores produced by PacBio HiFi circular consensus sequencing were shown to be systematically overestimated at the high end of the Phred scale, with practical consequences for downstream somatic variant calling. We introduced TopoQual, a method that combines a topological analysis of the partial order alignment graph (TopoCut) with a deep learning model trained against empirical error rates on an umbilical cord blood sample. The resulting recalibrated quality scores track validated error rates more closely than either the original PacBio scores or those produced by DeepConsensus [39], while simultaneously polishing approximately 32% of the errors present in the input CCS reads. TopoQual reaches a maximum validated quality of Q54, compared to Q46 for PacBio and Q49 for DeepConsensus, and provides better-calibrated inputs to variant callers such as Strelka2 [7], Mutect2 [8], ClairS [40], and DeepSomatic [41].

Chapter 3: LCSkPOA. Because TopoQual, and any POA-based consensus method, depends on multiple sequence alignment of subreads, the quadratic time and memory complexity of unbanded POA [19] is a major bottleneck on long reads. We extended the LCSk++ algorithm [48, 49] to graph structures and introduced LCSkPOA, a banded POA method that achieves up to a $60\times$ speedup over unbanded SPOA [43] on long synthetic sequences and maintains alignment accuracy above 99% on real HiFi data at $k = 12$. LCSkPOA also uses roughly tenfold less memory than ABPOA [42] on the longest sequences tested. Together, LCSkPOA and TopoQual form a tractable consensus-plus-polishing pipeline for HiFi sequencing at genome scale.

Chapter 4: Souporcell3. On the single-cell side, existing genotype-based demultiplexing tools such as Souporcell [9], Vireo [10], and scSplit [11] were primarily designed and benchmarked on pools of 16 or fewer donors, beyond which clustering quality degrades sharply. We introduced Souporcell3, which combines a $10\times$ -merge cluster initialization strategy, deterministic annealing-enhanced K-Harmonic Means clustering with L_p -norm normalisation [14, 15], and an iterative refinement procedure that reinitialises bad clusters while locking good ones in place. Souporcell3 achieves near-perfect adjusted Rand index scores and zero incorrectly merged clusters at donor counts of up to 64, substantially outperforming the existing baselines. This roughly fourfold extension of the practical donor-count limit translates directly into a corresponding reduction in the per-donor cost of multiplexed scRNA-seq studies and reduces technical batch effects in large pooled experiments.

Chapter 5: SpaceCell. Finally, we extended the clustering paradigm to spatial single-cell transcriptomics. We showed that on both Slide-seqV2 mouse cerebellum data [99] and Visium human DLPFC data [98], standard probabilistic models exhibit complementary failure modes: the negative binomial likelihood is precise but has narrow range, while GLM-PCA [104] provides wide separation but is insensitive to small gene-level differences

between transcriptionally similar layers. This observation directly motivated SpaceCell, a three-stage method that first clusters cells with a Gaussian mixture model [116] on the GLM-PCA embedding, then refines the partition by iteratively selecting the most representative cells under a negative-binomial discriminability score, and finally assigns the remaining cells using Leiden community detection [121] on a graph whose edge weights combine negative binomial dissimilarity with spatial distance.

6.2 Closing Remarks

Sequencing technologies are now capable of measuring the genome and transcriptome at resolutions that were unimaginable a decade ago, but the value of those measurements depends on the algorithms that turn raw signals into accurate, interpretable biological inferences. The four methods presented in this dissertation address specific bottlenecks in that translation: from base-level quality calibration and efficient alignment in long-read sequencing, to scalable demultiplexing and spatially-aware clustering in single-cell genomics. Each contribution stands alone as a practical improvement to a widely-used pipeline, and together they illustrate the broader principle that careful diagnosis of failure modes, paired with modest, targeted algorithmic changes, can yield substantial gains across the modern sequence analysis stack.

References

- [1] A. M. Wenger, P. Peluso, W. J. Rowell, P.-C. Chang, R. J. Hall, G. T. Concepcion, et al., “Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome”, *Nature Biotechnology* **37**, 1155–1162 (2019).
- [2] M. Weerakoon, S. Lee, E. Mitchell, and H. Heaton, “TopoQual polishes circular consensus sequencing data and accurately predicts quality scores”, *BMC Bioinformatics* **26**, 17 (2025).
- [3] M. Weerakoon, C. T. Saunders, and H. Heaton, “LCSkPOA: enabling banded semi-global partial order alignments via efficient and accurate backbone generation through extended LCSk++”, *BMC Bioinformatics* **26**, 284 (2025).
- [4] B. Vogelstein, N. Papadopoulos, V. E. Velculescu, S. Zhou, L. A. Diaz, and K. W. Kinzler, “Cancer genome landscapes”, *Science* **339**, 1546–1558 (2013).
- [5] I. Martincorena and P. J. Campbell, “Somatic mutation in cancer and normal cells”, *Science* **349**, 1483–1489 (2015).
- [6] D. Hanahan and R. A. Weinberg, “The hallmarks of cancer”, *Cell* **100**, 57–70 (2000).
- [7] S. Kim, K. Scheffler, A. L. Halpern, M. A. Bekritsky, E. Noh, M. Källberg, et al., “Strelka2: fast and accurate calling of germline and somatic variants”, *Nature Methods* **15**, 591–594 (2018).
- [8] D. Benjamin, T. Sato, K. Cibulskis, G. Getz, C. Stewart, and L. Lichtenstein, “Calling somatic SNVs and indels with Mutect2”, *bioRxiv*, 861054 (2019).

- [9] H. Heaton, A. M. Talman, A. Knights, M. Imaz, D. J. Gaffney, R. Durbin, M. Hemberg, and M. K. N. Lawniczak, “Souporecell: robust clustering of single-cell RNA-seq data by genotype without reference genotypes”, *Nature Methods* **17**, 615–620 (2020).
- [10] Y. Huang, D. J. McCarthy, and O. Stegle, “Vireo: Bayesian demultiplexing of pooled single-cell RNA-seq data without genotype reference”, *Genome Biology* **20**, 273 (2019).
- [11] J. Xu, C. Falconer, Q. Nguyen, J. Crawford, B. D. McKinnon, S. Mortlock, et al., “Genotype-free demultiplexing of pooled single-cell RNA-seq”, *Genome Biology* **20**, 290 (2019).
- [12] M. Weerakoon, H. Vu, R. Behboudi, and H. Heaton, “Souporecell3: robust demultiplexing for high-donor single-cell RNA-seq datasets”, *Bioinformatics* **42**, btg117 (2026).
- [13] D. Aloise, A. Deshpande, P. Hansen, and P. Popat, “NP-hardness of Euclidean sum-of-squares clustering”, *Machine Learning* **75**, 245–248 (2009).
- [14] B. Zhang, M. Hsu, and U. Dayal, *K-harmonic means—a data clustering algorithm*, tech. rep. HPL-1999-124 (Hewlett-Packard Labs, 1999).
- [15] B. Zhang, “Generalized K-harmonic means—dynamic weighting of data in unsupervised learning”, in *Proceedings of the 2001 siam international conference on data mining* (2001).
- [16] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, “Biological sequence analysis: probabilistic models of proteins and nucleic acids”, (1998).
- [17] S. B. Needleman and C. D. Wunsch, “A general method applicable to the search for similarities in the amino acid sequence of two proteins”, *Journal of Molecular Biology* **48**, 443–453 (1970).

- [18] T. F. Smith and M. S. Waterman, "Identification of common molecular subsequences", *Journal of Molecular Biology* **147**, 195–197 (1981).
- [19] C. Lee, C. Grasso, and M. F. Sharlow, "Multiple sequence alignment using partial order graphs", *Bioinformatics* **18**, 452–464 (2002).
- [20] J. Felsenstein, "Evolutionary trees from DNA sequences: a maximum likelihood approach", *Journal of Molecular Evolution* **17**, 368–376 (1981).
- [21] W. Huber, V. J. Carey, R. Gentleman, S. Anders, M. Carlson, B. S. Carvalho, et al., "Orchestrating high-throughput genomic analysis with Bioconductor", *Nature Methods* **12**, 115–121 (2015).
- [22] J. Eid, A. Fehr, J. Gray, K. Luong, J. Lyle, G. Otto, et al., "Real-time DNA sequencing from single polymerase molecules", *Science* **323**, 133–138 (2009).
- [23] J. Korlach, P. J. Marks, R. L. Cicero, J. J. Gray, D. L. Murphy, D. B. Roitman, et al., "Selective aluminum passivation for targeted immobilization of single DNA polymerase molecules in zero-mode waveguide nanostructures", *Proceedings of the National Academy of Sciences* **105**, 1176–1181 (2008).
- [24] D. R. Bentley, S. Balasubramanian, H. P. Swerdlow, G. P. Smith, J. Milton, C. G. Brown, et al., "Accurate whole human genome sequencing using reversible terminator chemistry", *Nature* **456**, 53–59 (2008).
- [25] V. Potapov and J. L. Ong, "Correction: examining sources of error in PCR by single-molecule sequencing", *PLoS One* **12**, e0181128 (2017).
- [26] P. Ren, X. Dong, and J. Vijg, "Age-related somatic mutation burden in human tissues", *Frontiers in Aging* **3**, 1018119 (2022).
- [27] F. G. Osorio, A. Rosendahl Huber, R. Oka, M. Verheul, S. H. Patel, K. Hasaart, et al., "Somatic mutations reveal lineage relationships and age-related mutagenesis in human hematopoiesis", *Cell Reports* **25**, 2308–2316.e4 (2018).

- [28] S. R. Kennedy, M. W. Schmitt, E. J. Fox, B. F. Kohn, J. J. Salk, E. H. Ahn, et al., “Detecting ultralow-frequency mutations by Duplex Sequencing”, *Nature Protocols* **9**, 2586–2606 (2014).
- [29] M. Jain, H. E. Olsen, B. Paten, and M. Akeson, “The Oxford Nanopore MinION: delivery of nanopore sequencing to the genomics community”, *Genome Biology* **17**, 239 (2016).
- [30] A. Rhoads and K. F. Au, “PacBio sequencing and its applications”, *Genomics, Proteomics & Bioinformatics* **13**, 278–289 (2015).
- [31] J. Korlach, K. P. Bjornson, B. P. Chaudhuri, R. L. Cicero, B. A. Flusberg, J. J. Gray, et al., “Real-time DNA sequencing from single polymerase molecules”, *Methods in Enzymology* **472**, 431–455 (2010).
- [32] C.-S. Chin, P. Peluso, F. J. Sedlazeck, M. Nattestad, G. T. Concepcion, A. Clum, et al., “Phased diploid genome assembly with single-molecule real-time sequencing”, *Nature Methods* **13**, 1050–1054 (2016).
- [33] H. Cheng, G. T. Concepcion, X. Feng, H. Zhang, and H. Li, “Haplotype-resolved de novo assembly using phased assembly graphs with hifiasm”, *Nature Methods* **18**, 170–175 (2021).
- [34] S. Nurk, B. P. Walenz, A. Rhie, M. R. Vollger, G. A. Logsdon, R. Grothe, et al., “Hi-Canu: accurate assembly of segmental duplications, satellites, and allelic variants from high-fidelity long reads”, *Genome Research* **30**, 1291–1305 (2020).
- [35] M. Rautiainen, S. Nurk, B. P. Walenz, G. A. Logsdon, D. Porubsky, A. Rhie, et al., “Telomere-to-telomere assembly of diploid chromosomes with Verkko”, *Nature Biotechnology* **41**, 1474–1482 (2023).
- [36] Z. Zhang, J. Zhang, L. Kang, X. Qiu, B. Niu, A. Bi, et al., “Genotyping of structural variation using PacBio high-fidelity sequencing”, *bioRxiv*, 2021.10.28.466362 (2021).

- [37] S. Garg, “Computational methods for chromosome-scale haplotype reconstruction”, *Genome Biology* **22**, 101 (2021).
- [38] S. Garg, “Towards routine chromosome-scale haplotype-resolved reconstruction in cancer genomics”, *Nature Communications* **14**, 1358 (2023).
- [39] G. Baid, D. E. Cook, K. Shafin, T. Yun, F. Llinares-López, Q. Berthet, et al., “Deep-Consensus improves the accuracy of sequences with a gap-aware sequence transformer”, *Nature Biotechnology* **41**, 232–238 (2023).
- [40] Z. Zheng, J. Su, L. Chen, Y.-L. Lee, T.-W. Lam, and R. Luo, “ClairS: a deep-learning method for long-read somatic small variant calling”, *bioRxiv* (2023).
- [41] J. Park, D. E. Cook, P.-C. Chang, A. Kolesnikov, L. Brambrink, J. C. Mier, et al., “DeepSomatic: accurate somatic small variant discovery for multiple sequencing technologies”, *bioRxiv* (2024).
- [42] Y. Gao, Y. Liu, Y. Ma, B. Liu, Y. Wang, and Y. Xing, “abPOA: an SIMD-based C library for fast partial order alignment using adaptive band”, *Bioinformatics* **37**, 2209–2211 (2020).
- [43] R. Vaser, I. Sović, N. Nagarajan, and M. Šikić, “Fast and accurate de novo genome assembly from long uncorrected reads”, *Genome Research* **27**, 737–746 (2017).
- [44] H. Li and R. Durbin, “Fast and accurate short read alignment with Burrows–Wheeler transform”, *Bioinformatics* **25**, 1754–1760 (2009).
- [45] H. Li, “Minimap2: pairwise alignment for nucleotide sequences”, *Bioinformatics* **34**, 3094–3100 (2018).
- [46] Broad Institute, *Base quality score recalibration (BQSR)*, <https://gatk.broadinstitute.org/hc/en-us/articles/360035890531-Base-Quality-Score-Recalibration-BQSR>, Accessed: 2024-10-17, 2024.

- [47] B. Ewing and P. Green, “Base-calling of automated sequencer traces using phred. II. error probabilities”, *Genome Research* **8**, 186–194 (1998).
- [48] F. Pavetić, G. Žužić, and M. Šikić, “LCSk++: practical similarity metric for long strings”, *arXiv preprint arXiv:1407.2407* (2014).
- [49] F. Pavetić, I. Katanić, G. Matula, G. Žužić, and M. Šikić, “Fast and simple algorithms for computing both LCSk and LCSk+”, *arXiv preprint arXiv:1705.07279* (2017).
- [50] B. S. Baker and R. Giancarlo, “Sparse dynamic programming for longest common subsequence from fragments”, *Journal of Algorithms* **42**, 231–254 (2002).
- [51] Y. Ono, K. Asai, and M. Hamada, “PBSIM: PacBio reads simulator—toward accurate genome assembly”, *Bioinformatics* **29**, 119–121 (2013).
- [52] G. X. Y. Zheng, J. M. Terry, P. Belgrader, P. Ryvkin, Z. W. Bent, R. Wilson, et al., “Massively parallel digital transcriptional profiling of single cells”, *Nature Communications* **8**, 14049 (2017).
- [53] 10x Genomics, *What are the benefits of single-cell sequencing?*, <https://www.10xgenomics.com/blog/what-are-the-benefits-of-single-cell-sequencing>, Accessed: 2025-07-15, 2025.
- [54] A. R. Cillo, C. H. L. Kürten, T. Tabib, Z. Qi, S. Onkar, T. Wang, et al., “Immune landscape of viral- and carcinogen-driven head and neck cancer”, *Immunity* **52**, 183–199.e6 (2020).
- [55] L. Jardine, S. Wiscombe, G. Reynolds, D. McDonald, A. Fuller, K. Green, et al., “Mapping the developing human immune system across organs”, *Science* **373**, eabh1516 (2021).
- [56] S. L. Wolock, R. Lopez, and A. M. Klein, “Scrublet: computational identification of cell doublets in single-cell transcriptomic data”, *Cell Systems* **8**, 281–291.e9 (2019).

- [57] C. S. McGinnis, L. M. Murrow, and Z. J. Gartner, “DoubletFinder: doublet detection in single-cell RNA sequencing data using artificial nearest neighbors”, *Cell Systems* **8**, 329–337.e4 (2019).
- [58] M. D. Young and S. Behjati, “SoupX removes ambient RNA contamination from droplet-based single-cell RNA sequencing data”, *GigaScience* **9** (2020).
- [59] J. Jerber, D. D. Seaton, A. S. E. Cuomo, N. Kumasaka, J. Haldane, J. Steer, et al., “Population-scale single-cell RNA-seq profiling across dopaminergic neuron differentiation”, *Nature Genetics* **53**, 304–312 (2021).
- [60] M. Mahajan, P. Nimbhorkar, and K. Varadarajan, “The planar k -means problem is NP-hard”, *Theoretical Computer Science* **442**, 13–21 (2012).
- [61] M. Inaba, N. Katoh, and H. Imai, “Applications of weighted Voronoi diagrams and randomization to variance-based k -clustering”, in *Proceedings of the tenth annual symposium on computational geometry (scg ’94)* (1994), pp. 332–339.
- [62] W. M. Rand, “Objective criteria for the evaluation of clustering methods”, *Journal of the American Statistical Association* **66**, 846–850 (1971).
- [63] L. Hubert and P. Arabie, “Comparing partitions”, *Journal of Classification* **2**, 193–218 (1985).
- [64] 10x Genomics, *What is the maximum number of cells that can be profiled?*, <https://kb.10xgenomics.com/hc/en-us/articles/360001378811>, Accessed: 2025-06-03, 2020.
- [65] N. Ortolano, *The next generation of single cell RNA-seq: an introduction to GEM-X technology*, 10x Genomics, 2024.
- [66] V. Y. Kiselev, K. Kirschner, M. T. Schaub, T. Andrews, A. Yiu, T. Chandra, et al., “SC3: consensus clustering of single-cell RNA-seq data”, *Nature Methods* **14**, 483–486 (2017).

- [67] D. Arthur and S. Vassilvitskii, “ k -means++: the advantages of careful seeding”, in Proceedings of the eighteenth annual acm-siam symposium on discrete algorithms (soda) (2007), pp. 1027–1035.
- [68] Z. Güngör and A. Ünler, “K-harmonic means data clustering with simulated annealing heuristic”, Applied Mathematics and Computation **184**, 199–209 (2007).
- [69] Z. Güngör and A. Ünler, “K-harmonic means data clustering with tabu-search method”, Applied Mathematical Modelling **32**, 1115–1125 (2008).
- [70] Human Cell Atlas, *Immune landscape of viral- and carcinogen-driven head and neck cancer project*, <https://explore.data.humancellatlas.org/projects/3089d311-f9ed-44dd-bb10-397059bad4dc>, 2020.
- [71] Human Cell Atlas, *Mapping the developing human immune system across organs project*, <https://explore.data.humancellatlas.org/projects/fcaa53cd-ba57-4bfe-af9c-eaa958f95c1a>, 2021.
- [72] European Nucleotide Archive, *Sample accessions ERS2630502–ERS2630507 for individual cell lines euts, nufh, babz, oaqd, and ieki*, <https://www.ebi.ac.uk/ena/browser/view/ERS2630502>, 2020.
- [73] J. N. Wells and C. Feschotte, “A field guide to eukaryotic transposable elements”, Annual Review of Genetics **54**, 539–561 (2020).
- [74] S. S. Arcot, Z. Wang, J. L. Weber, P. L. Deininger, and M. A. Batzer, “Alu repeats: a source for the genesis of primate microsatellites”, Genomics **29**, 136–144 (1995).
- [75] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting”, Journal of Machine Learning Research **15**, 1929–1958 (2014).
- [76] D. P. Kingma and J. Ba, “Adam: a method for stochastic optimization”, arXiv preprint arXiv:1412.6980 (2014).

- [77] K. Xiong, D. Shea, J. Rhoades, T. Blewett, R. Liu, J. H. Bae, et al., “Duplex-Repair enables highly accurate sequencing, despite DNA damage”, *Nucleic Acids Research* **50**, e1 (2022).
- [78] M. Rautiainen and T. Marschall, “GraphAligner: rapid and versatile sequence-to-graph alignment”, *Genome Biology* **21** (2020).
- [79] Y. Zhang, C. Park, C. Bennett, M. Thornton, and D. Kim, “Rapid and accurate alignment of nucleotide conversion sequencing reads with HISAT-3N”, *Genome Research* **31**, 1290–1295 (2021).
- [80] R. Mott, “Smith–Waterman algorithm”, *Encyclopedia of Life Sciences* (2005).
- [81] G. Benson, A. Levy, and B. R. Shalom, “Longest common subsequence in k-length substrings”, in *Similarity search and applications* (2013), pp. 257–265.
- [82] J. Köster, “Rust-Bio: a fast and safe bioinformatics library”, *Bioinformatics* **32**, 444–446 (2015).
- [83] H. M. Chung and Y. Huang, “SNPmanifold: detecting single-cell clonality and lineages from single-nucleotide variants using binomial variational autoencoder”, *Genome Biology* **26**, 309 (2025).
- [84] W. Peizhuang, “Pattern recognition with fuzzy objective function algorithms (James C. Bezdek)”, Philadelphia, 25 (1983).
- [85] Y. Liu, S. Agarwal, and S. Venkataraman, “AutoFreeze: automatically freezing model blocks to accelerate fine-tuning”, *arXiv preprint arXiv:2102.01386* (2021).
- [86] X. Ma, Z. Li, X. Zhang, H. Wang, and X. Wang, “Few-shot learning for named entity recognition based on BERT and two-level model fusion”, *ACM Transactions on Asian and Low-Resource Language Information Processing* **20**, 1–19 (2021).

- [87] Human Cell Atlas, *Human Cell Atlas project: cc95ff89-2e68-4a08-a234-480eca21ce79*, <https://explore.data.humancellatlas.org/projects/cc95ff89-2e68-4a08-a234-480eca21ce79>, 2020.
- [88] N. X. Vinh, J. Epps, and J. Bailey, “Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance”, *Properties, Normalization and Correction for Chance* (2009).
- [89] F. Tang, C. Barbacioru, Y. Wang, E. Nordman, C. Lee, N. Xu, X. Wang, J. Bodeau, B. B. Tuch, A. Siddiqui, K. Lao, and M. A. Surani, “mRNA-Seq whole-transcriptome analysis of a single cell”, *Nature Methods* **6**, 377–382 (2009).
- [90] E. Z. Macosko, A. Basu, R. Satija, J. Nemesh, K. Shekhar, M. Goldman, I. Tirosh, A. R. Bialas, N. Kamitaki, E. M. Martersteck, et al., “Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets”, *Cell* **161**, 1202–1214 (2015).
- [91] G. X. Y. Zheng, J. M. Terry, P. Belgrader, P. Ryvkin, Z. W. Bent, R. Wilson, S. B. Ziraldo, T. D. Wheeler, G. P. McDermott, J. Zhu, et al., “Massively parallel digital transcriptional profiling of single cells”, *Nature Communications* **8**, 14049 (2017).
- [92] L. Moses and L. Pachter, “Museum of spatial transcriptomics”, *Nature Methods* **19**, 534–546 (2022).
- [93] V. Marx, “Method of the year: spatially resolved transcriptomics”, *Nature Methods* **18**, 9–14 (2021).
- [94] P. L. Ståhl, F. Salmén, S. Vickovic, A. Lundmark, J. F. Navarro, J. Magnusson, S. Giacomello, M. Asp, J. O. Westholm, M. Huss, et al., “Visualization and analysis of gene expression in tissue sections by spatial transcriptomics”, *Science* **353**, 78–82 (2016).
- [95] A. Rao, D. Barkley, G. S. França, and I. Yanai, “Exploring tissue architecture using spatial transcriptomics”, *Nature* **596**, 211–220 (2021).

- [96] G. Palla, H. Spitzer, M. Klein, D. Fischer, A. C. Schaar, L. B. Kuemmerle, S. Rybakov, I. L. Ibarra, O. Holmberg, I. Virshup, et al., “Squidpy: a scalable framework for spatial omics analysis”, *Nature Methods* **19**, 171–178 (2022).
- [97] S. K. Longo, M. G. Guo, A. L. Ji, and P. A. Khavari, “Integrating single-cell and spatial transcriptomics to elucidate intercellular tissue dynamics”, *Nature Reviews Genetics* **22**, 627–644 (2021).
- [98] K. R. Maynard, L. Collado-Torres, L. M. Weber, C. Uytingco, B. K. Barry, S. R. Williams, J. L. Catallini, M. N. Tran, Z. Besich, M. Tippi, et al., “Transcriptome-scale spatial gene expression in the human dorsolateral prefrontal cortex”, *Nature Neuroscience* **24**, 425–436 (2021).
- [99] D. M. Cable, E. Murray, V. Shanmugam, S. Zhang, L. S. Zou, M. Diao, H. Chen, E. Z. Macosko, R. A. Irizarry, and F. Chen, “Cell type-specific inference of differential expression in spatial transcriptomics”, *Nature Methods* **19**, 1076–1087 (2022).
- [100] R. R. Stickels, E. Murray, P. Kumar, J. Li, J. L. Marshall, D. J. Di Bella, P. Arlotta, E. Z. Macosko, and F. Chen, “Highly sensitive spatial transcriptomics at near-cellular resolution with Slide-seqV2”, *Nature Biotechnology* **39**, 313–319 (2021).
- [101] S. G. Rodriques, R. R. Stickels, A. Goeva, C. A. Martin, E. Murray, C. R. Vanderbilt, J. Welch, L. M. Chen, F. Chen, and E. Z. Macosko, “Slide-seq: a scalable technology for measuring genome-wide expression at high spatial resolution”, *Science* **363**, 1463–1467 (2019).
- [102] K. Leto, M. Arancillo, E. B. E. Becker, A. Buffo, C. Chiang, B. Ding, W. B. Dobyns, I. Dusart, P. Haldipur, M. E. Hatten, et al., “Consensus paper: cerebellar development”, *Cerebellum* **15**, 789–828 (2016).
- [103] V. Kozareva, C. Martin, T. Osorno, S. Rudolph, C. Guo, C. Vanderbilt, N. Nadaf, A. Regev, W. G. Regehr, and E. Macosko, “A transcriptomic atlas of mouse cerebellar cortex comprehensively defines cell types”, *Nature* **598**, 214–219 (2021).

- [104] F. W. Townes, S. C. Hicks, M. J. Aryee, and R. A. Irizarry, “Feature selection and dimension reduction for single-cell RNA-Seq based on a multinomial model”, *Genome Biology* **20**, 295 (2019).
- [105] G. Rajkowska and P. S. Goldman-Rakic, “Cytoarchitectonic definition of prefrontal areas in the normal human cortex: i. Remapping of areas 9 and 46 using quantitative criteria”, *Cerebral Cortex* **5**, 307–322 (1995).
- [106] G. Rajkowska and P. S. Goldman-Rakic, “Cytoarchitectonic definition of prefrontal areas in the normal human cortex: ii. Variability in locations of areas 9 and 46 and relationship to the Talairach coordinate system”, *Cerebral Cortex* **5**, 323–337 (1995).
- [107] H. Zeng, E. H. Shen, J. G. Hohmann, S. W. Oh, A. Bernard, J. J. Royall, K. J. Glattfelder, S. M. Sunkin, J. A. Morris, A. L. Guillozet-Bongaarts, et al., “Large-scale cellular-resolution gene profiling in human neocortex reveals species-specific molecular signatures”, *Cell* **149**, 483–496 (2012).
- [108] B. J. Molyneaux, P. Arlotta, J. R. L. Menezes, and J. D. Macklis, “Neuronal subtype specification in the cerebral cortex”, *Nature Reviews Neuroscience* **8**, 427–437 (2007).
- [109] J. Hu, X. Li, K. Coleman, A. Schroeder, N. Ma, D. J. Irwin, E. B. Lee, R. T. Shino-hara, and M. Li, “SpaGCN: integrating gene expression, spatial location and histology to identify spatial domains and spatially variable genes by graph convolutional network”, *Nature Methods* **18**, 1342–1351 (2021).
- [110] E. Zhao, M. R. Stone, X. Ren, J. Guenthoer, K. S. Smythe, T. Pulliam, S. R. Williams, C. R. Uytingco, S. E. B. Taylor, P. Nghiem, J. H. Bielas, and R. Gottardo, “Spatial transcriptomics at subspot resolution with BayesSpace”, *Nature Biotechnology* **39**, 1375–1384 (2021).

- [111] K. Dong and S. Zhang, “Deciphering spatial domains from spatially resolved transcriptomics with an adaptive graph attention auto-encoder”, *Nature Communications* **13**, 1739 (2022).
- [112] Y. Long, K. S. Ang, M. Li, K. L. K. Chong, R. Sethi, C. Zhong, H. Xu, Z. Ong, K. Sachaphibulkij, A. Chen, et al., “Spatially informed clustering, integration, and deconvolution of spatial transcriptomics with GraphST”, *Nature Communications* **14**, 1155 (2023).
- [113] U. Chitra, B. J. Arnold, H. Sarkar, C. Ma, S. Lopez-Darwin, K. Sanno, and B. J. Raphael, “Mapping the topography of spatial gene expression with interpretable deep learning”, *Nature Methods* **22**, 298–309 (2025).
- [114] R. D. Hodge, T. E. Bakken, J. A. Miller, K. A. Smith, E. R. Barkan, L. T. Graybuck, J. L. Close, B. Long, N. Johansen, O. Penn, et al., “Conserved cell types with divergent features in human versus mouse cortex”, *Nature* **573**, 61–68 (2019).
- [115] S. C. Hicks, F. W. Townes, M. Teng, and R. A. Irizarry, “Missing data and technical variability in single-cell RNA-sequencing experiments”, *Biostatistics* **19**, 562–578 (2018).
- [116] G. J. McLachlan and D. Peel, *Finite mixture models* (Wiley, 2000).
- [117] C. M. Bishop, *Pattern recognition and machine learning* (Springer, 2006).
- [118] M. I. Love, W. Huber, and S. Anders, “Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2”, *Genome Biology* **15**, 550 (2014).
- [119] M. D. Robinson, D. J. McCarthy, and G. K. Smyth, “edgeR: a Bioconductor package for differential expression analysis of digital gene expression data”, *Bioinformatics* **26**, 139–140 (2010).
- [120] C. Hafemeister and R. Satija, “Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression”, *Genome Biology* **20**, 296 (2019).

- [121] V. A. Traag, L. Waltman, and N. J. van Eck, “From Louvain to Leiden: guaranteeing well-connected communities”, *Scientific Reports* **9**, 5233 (2019).
- [122] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks”, *Journal of Statistical Mechanics: Theory and Experiment* **2008**, P10008 (2008).
- [123] M. Varrone, D. Tavernari, A. Santamaria-Martínez, L. A. Walsh, and G. Ciriello, “CellCharter reveals spatial cell niches associated with tissue remodeling and patient survival”, *Nature Genetics* **56**, 74–84 (2023).