Università degli Studi di Padova
Department of Biology

PhD Programme in Biosciences:
Genetics, Genomics and Bioinformatics

XXXVIII Cycle

Development of efficient and scalable methods for omic data analyses in cancer biology

Supervisor

Prof. Chiara Romualdi

Ph.D. Student

Ilaria Billato

Co-supervisor

Prof. Davide Risso

Abstract

High-throughput technologies have propelled biology into the big data era. Single-cell RNA sequencing now produces datasets with millions of cells, while digital pathology generates whole-slide images containing billions of pixels. These advances enable unprecedented discovery but create a computational paradox: data are generated faster than they can be processed, and standard workflows often fail to scale. Efficient algorithms and integrative strategies are therefore essential for analyzing massive, heterogeneous datasets.

This PhD thesis addresses these challenges through two complementary aims. First, we benchmark different Singular Value Decomposition (SVD) algorithm for Principal Component Analysis (PCA), a key dimensionality-reduction step in single-cell transcriptomics. Classical PCA becomes prohibitively slow and memory-intensive as data size increases. To overcome these limitations, we evaluate state-of-the-art algorithms and out-of-memory data formats across complete single-cell workflows. The benchmark compares Seurat, OSCA/Bioconductor and scrapper in R and Scanpy, and GPU-enabled frameworks such as rapids_singlecell in Python, leveraging GPU acceleration to reduce runtime and memory usage on datasets with millions of cells. These analyses quantify performance trade-offs and provide reproducible guidance for selecting optimal pipelines for large-scale single-cell studies.

Second, we focus on digital pathology, where histopathological images reveal tissue architecture, cellular morphology, and tumor spatial organization. We processed 11,765 H&E-stained images from 32 TCGA cancer types using deep learning (HoVer-Net) to extract nuclei-level features and Prov-GigaPath to extract slide level embeddings. To bridge the gap between image analysis and the R/Bioconductor ecosystem, we released three packages: TCIAAPI, HistoImageR and imageTCGA, a Shiny application for interactive exploration, filtering, and visualization of extracted features alongside the original images.

By combining scalable computation with cross-modal integration, this work improves the efficiency of single-cell analysis and supports precision medicine through clinically relevant molecular - morphological associations.

Contents

List of Figures

List of Tables

List of Acronyms

Glossary

ADT Antibody-derived tags

ARI Adjusted Rand Index, a metric to evaluate the similarity between two clustering results

ARPACK An algorithm used to compute dominant eigenvalues and eigenvectors, often used with BiocSingular

BCR Biospecimen Core Resource

BE1 Reference database

BH BenjaminiHochberg

BP Biological process

BS Bottom-section

CAFS Cancer-associated fibroblasts

cb Coord Blood data

CC Cellular component

CITE-SEQ Cellular Indexing of Transcriptomes and Epitopes by sequencing

CNN Convolutional Neural Network, a deep learning model for image processing tasks

CNA Copy number alterations

CNS Copy Number Signatures

CPU Central Processing Unit, the main hardware that executes instructions in a computer

CPE Consensus Purity Estimation

CT Computed tomography

DL Deep learning

FFPE Formalin-fixed paraffin-embedded

FIGO International Federation of Gynecology and Obstetrics

geoJSON A geospatial extension of JSON used to store spatial features like nuclei polygons

GO Gene Ontology

GPU Graphics Processing Unit, used to accelerate deep learning and parallel computations

H5 Hierarchical Data Format version 5, used for large-scale scientific datasets

H5AD An AnnData-based HDF5 format for annotated biological data

HE Hematoxylin and Eosin

HGSOC High-grade serous ovarian cancer

HPC High Performance Computing, infrastructure designed for large-scale data analysis

HR Homologous recombination

HVG Highly variable genes, genes with high expression variance used in feature selection

IPCA Incremental Principal Component Analysis

IRLBA Implicitly Restarted Lanczos Bidiagonalization Algorithm

JSON JavaScript Object Notation, a structured data format often used for storing image annotations

KEGG Kyoto Encyclopedia of Genes and Genomes

KNN k-nearest neighbors

LGSOC Low-grade serous ovarian cancer

LUMP Leukocytes Unmethylation for Purity

MF Molecular function

MRI Magnetic resonance imaging

NMF Non-negative matrix factorization

OD Optical density

OSCA Orchestrating Single-Cell Analysis, the Bioconductor book for single-cell workflows

OSTA Orchestrating Spatial Transcriptomics Analysis with Bioconductor

OV Ovarian cancer, a specific tumor type included in TCGA

PC Principal component

PCA Principal Component Analysis

PET Positron emission tomography

PNG Portable Network Graphics, a lossless image format for plots and thumbnails

PPA Point Pattern Analysis

QIA Quantitative image analysis

RAM Random Access Memory, temporary data storage used during computation

scmix Single-cell mixology data

scRNA-seq Single-cell RNA sequencing, used to measure gene expression at single-cell resolution

SNN Shared nearest neighbors

SVD Singular Value Decomposition

SVS Slide Virtual Scan, a TIFF-based format for high-resolution whole-slide images

TCGA The Cancer Genome Atlas, a comprehensive multi-omics cancer dataset

TCIA The Cancer Imaging Archive, a public repository for medical and histopathological images

TME Tumor microenvironment

TIFF Tagged Image File Format, commonly used for storing raster graphics

TS Top-section

T-SNE t-distributed Stochastic Neighbor Embedding

TSS Tissue Source Site

UMAP Uniform Manifold Approximation and Projection

UMI Unique Molecular Identifier, a barcode used to distinguish original RNA molecules

WSI Whole Slide Image, a digital representation of an entire histopathology slide

Introduction

The development of efficient and scalable methods for omic data analysis is becoming increasingly crucial in cancer biology. My PhD project is situated within this broad yet timely theme, addressing the challenges posed by the exponential growth of biological data and the need for optimized computational strategies. In modern biomedical research, time is a precious resource, yet current analytical pipelines often lack automation or optimization, resulting in significant computational and human effort. This inefficiency stems largely from the rapid increase in the size and complexity of available datasets.

Over the past decades, advances in high-throughput technologies have led to the emergence of big data in life sciences. In transcriptomics, for example, the field has evolved from bulk sequencing to single-cell approaches, now encompassing datasets of millions of cells. (Stuart and Satija 2019) Similarly, in digital pathology, a single high-resolution whole-slide image can occupy up to 10 gigabytes of storage, with higher resolution directly translating into a larger number of pixels to process. As the number of sequenced cells or image pixels increases, computational time and memory usage grow exponentially, creating bottlenecks that slow down discovery. (Basak, Ozyoruk, and Demir 2023; Moses and Pachter 2022)

The goal of this thesis is to investigate the efficiency and scalability of computational methods in two major domains single-cell transcriptomics and digital pathology while also exploring strategies to improve performance and facilitate usability for researchers. A central connection between these domains lies in the potential to integrate image-derived features with molecular profiles, advancing multi-omic frameworks that combine complementary layers of biological information. In this context, spatial transcriptomics has emerged as a transformative technology, bridging molecular measurements with tissue architecture to provide unprecedented insights into the spatial organization of gene expression. However, the extremely high costs and limited throughput of spatial transcrip-

tomics currently restrict its widespread adoption. By contrast, digital pathology, while requiring an initial investment in scanning infrastructure, offers a more sustainable solution: once established, thousands of histological images can be digitized and analyzed at relatively low incremental cost. Thus, combining computational pathology with single-cell and spatial transcriptomics provides a promising and more cost-effective strategy for multi-modal cancer research.

The thesis is structured into four chapters. In the first chapter, we present a systematic benchmark on principal component analysis (PCA), a key dimensionality reduction method that often represents a computational bottleneck in single-cell data analysis. Specifically, we evaluate several singular value decomposition (SVD) algorithms across different contexts, implementations in R and Python, and hardware backends including CPU and GPU architectures.

The second chapter extends this analysis by comparing complete single-cell workflows, assessing not only computational efficiency but also biological accuracy at different stages of data processing.

In the third chapter, I shift focus to digital pathology, introducing the methodological choices and computational strategies that led to the development of three R/Bioconductor packages: imageTCGA, TCIAAPI, and HistoImageR, which collectively enable data access, feature extraction, and reproducible analyses of histological images.

Finally, the fourth chapter presents a case study on ovarian cancer, in which extracted image features are integrated into a downstream multi-omic analysis, demonstrating the practical value of combining histopathological and molecular data in translational cancer research.

Chapter 1

Benchmark SVD algorithm in large scRNA-Seq analysis

1.1 Introduction

As single-cell RNA sequencing (scRNA-seq) reaches its teenage years (Stark, Grzelak, and Hadfield 2019), we are witnessing a rapid increase in the size and complexity of experiments and datasets (Jovic et al. 2022). In fact, whereas early scRNA-seq comprised hundreds to a few thousand cells, typically in a single condition and often without biological replicates (Qu, Kao, and Hakonarson 2024), contemporary experiments include cells from multiple individuals measured across conditions (e.g., treatments, genotypes, health states). Often, researchers target $5 - 10{,}000$ cells per replicate and the final dataset easily comprises hundreds of thousands of cells (e.g., Pijuan-Sala et al. 2019; Stephenson et al. 2021). Furthermore, single-cell atlases have become mature, and thanks to programmatic access, it is now straightforward to download and locally analyse datasets made of millions of cells from multiple organs across different labs (Program et al. 2025; Rood et al. 2024).

This increasing complexity poses significant computational challenges throughout the analysis pipeline, from data preprocessing to downstream interpretation. These challenges are exacerbated by the exploratory nature of many scRNA-seq analyses, which need to process large datasets in an interactive way, e.g., trying different analysis paths and exploring the downstream results, often using desktops or laptops rather than high-performance computing (HPC), requiring frugal and efficient workflows. In this context, recent developments have

explored the use of Graphics Processing Units (GPUs) to accelerate core steps of the workflow, offering substantial improvements in scalability and runtime efficiency (Nolet et al. 2022).

A typical scRNA-seq analysis workflow starts with gene expression quantitation, a process that includes assigning reads to barcodes, aligning reads to the appropriate genome or transcriptome, and quantifying gene expression by counting Unique Molecular Identifiers (UMIs) assigned to each gene. This process is typically performed in HPC clusters and is usually carried out with stand-alone software or standardized pipelines. This step is typically not part of the interactive process described above; hence for the sake of this chapter, we will consider the output of this step as the starting point of the analysis. Interested readers can refer to the literature for a benchmark of the different preprocessing pipelines (e.g., (You et al. 2021)).

A typical scRNA-seq analysis workflow comprises several steps, including quality control, gene and cell filtering, normalization, identification of highly variable genes, dimensionality reduction, visualization, clustering, and cell type annotation (Amezquita et al. 2020).

Principal Component Analysis (PCA) and related methods are the cornerstone of dimensionality reduction in scRNA-seq workflows. They are typically applied after gene filtering and normalization to capture the main axes of transcriptional variation before downstream steps such as clustering, visualization (e.g., UMAP, t-SNE), and trajectory inference. However, the increasing scale of datasets highlights the need for algorithms that can balance accuracy, memory usage, and computational efficiency.

In this chapter, we systematically benchmark PCA implementations and related algorithms, assessing how different approaches scale with data size and computational resources, explicitly contrasting CPU- and GPU-based methods. Our goal is to provide practical guidelines for choosing efficient dimensionality reduction strategies in the context of modern large-scale single-cell datasets.

1.2 Methods

1.2.1 An introduction to PCA

Here, we briefly describe PCA and its relation to SVD. We refer the reader to Mardia, Kent, and Taylor 2024 and Hastie et al. 2009 for a more thorough

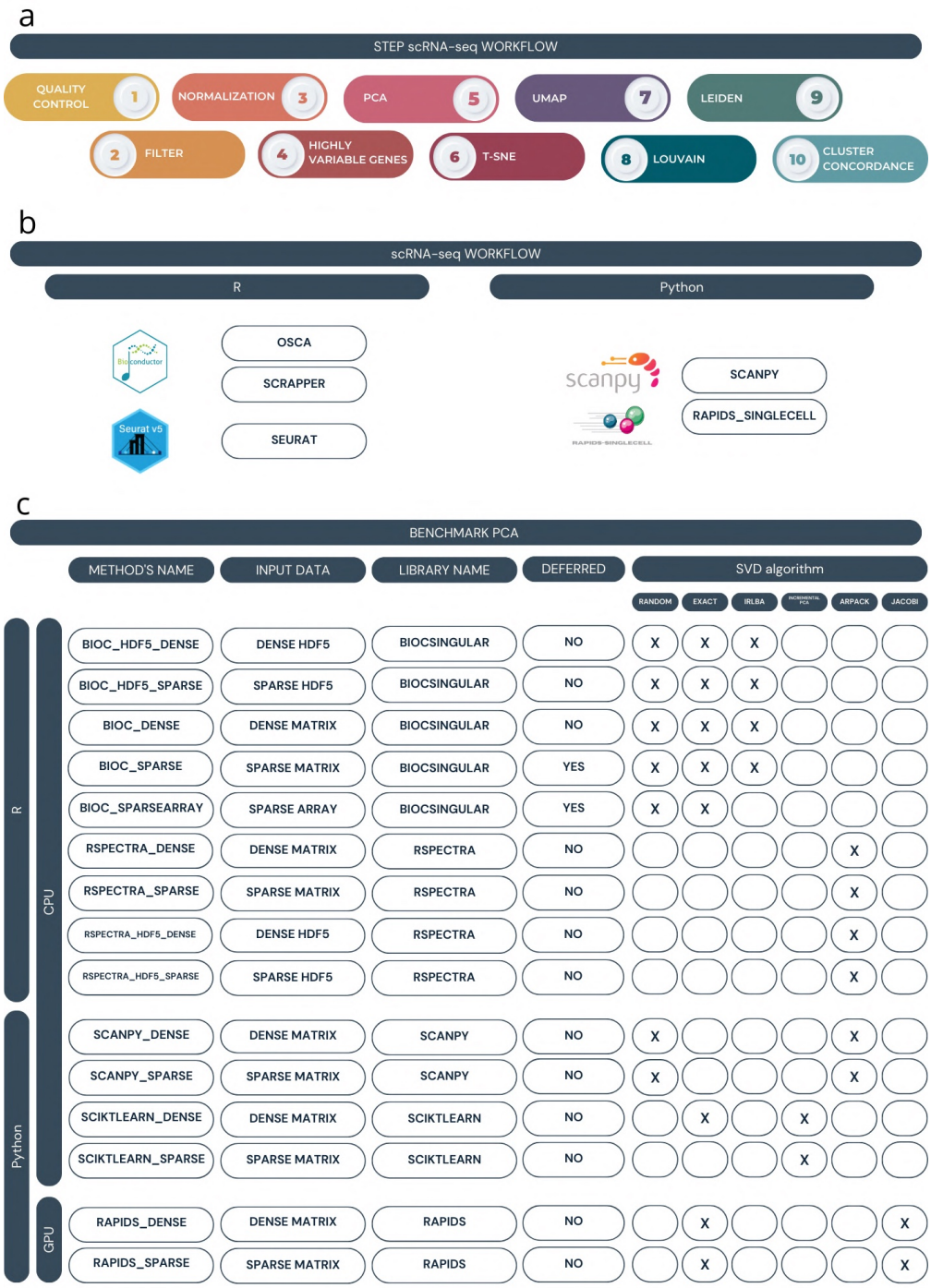

Figure 1.1: **Overview of PCA benchmarking and single-cell workflow comparison.** (a) Schematic of the typical steps in single-cell data processing workflows, including mitochondrial gene detection, filtering, normalization, highly variable genes, selection, dimensionality reduction (PCA, UMAP, t-SNE), clustering (Louvain, Leiden), and cluster concordance assessment. (b) List of R and Python-based single-cell RNA-seq analysis frameworks used for comparing full workflows: OSCA, Scrapper, and Seurat (R); Scanpy and Rapids_singlecell (Python).(c) Summary of PCA implementations evaluated in the benchmark, categorized by programming language (R or Python), computation type (CPU or GPU), input data format (dense matrix, sparse matrix, or HDF5), library name, and support for deferred computation. The table also indicates the supported SVD algorithms for each method (random, exact, irlba, Incremental PCA, arpack, jacobi).

introduction and the mathematical details.

Let X be the $n \times m$ matrix that contains the log-transformed, normalized expression of m genes in n cells. We assume that the matrix X has been centered such that each row has mean 0.

The principal components are the eigenvectors of the covariance matrix and can be computed by its eigendecomposition. Alternatively, the PCs can be computed directly by the Singular Value Decomposition (SVD) of the original matrix X. Specifically, we construct the following decomposition of X:

$$X = UDV^\top, \tag{1.1}$$

where U is a $n \times p$ orthogonal matrix of left singular vectors, V is a $m \times p$ orthogonal matrix of right singular vectors, and D is a $p \times p$ diagonal matrix, whose elements $d_1 \geq d_2 \geq \ldots \geq d_p \geq 0$ are the singular values. The first columns of UD correspond to the principal components, ordered by decreasing explained variance.

1.2.2 Algorithms to compute the SVD

Exact algorithm

In R, an exact SVD algorithm is implemented in the base `svd` function, which is called by the *BiocSingular* package's `runPCA` function when selecting the "exact" method. In Python, *scikit-learn*'s PCA function calls `scipy.linalg.svd` function. Both R's and Python's svd functions internally call the `gesdd` function in the LAPACK (Linear Algebra Package) library in Fortran. This function implements an efficient divide-and-conquer algorithm to compute the SVD (Anderson et al. 1999).

Truncated SVD

Often, one is interested in computing only a few, say k, singular values and vectors. For instance, in scRNA-seq, we are often limiting ourselves in calculating the top 30 or 50 PCs. Specifically, we may solve the problem:

$$X_k = U_k D_k V_k^\top,$$

where U_k and V_k denote the leading k columns of U and V, respectively, D_k denotes the diagonal matrix with the k largest singular values. It can be shown that X_k is the best rank-k approximation of X.

When $k \ll \min\{n, m\}$, as in the case of scRNA-seq analysis, truncated SVD algorithms exist to efficiently calculate the k largest singular values and the corresponding singular vectors. Here, we review a few of these methods.

ARPACK ARPACK Lehoucq, Sorensen, and C. Yang 1998 is a collection of Fortran77 subroutines designed to solve large-scale eigenvalue problems. ARPACK stands for ARnoldi PACKage and is based on the Arnoldi iteration method Arnoldi 1951. This method provides a good approximation of the k largest singular values and the corresponding singular vector of a matrix by constructing an orthonormal basis of its Krylov subspace Arnoldi 1951. Thanks to the relation between eigendecomposition and SVD, the ARPACK algorithm can be used to efficiently calculate the k largest singular values and the corresponding singular vectors.

Consider the decomposition in (1.1), one can show that

$$X^\top X = VD^2V^\top,$$

with $U = XVD^{-1}$. Thus, the singular values and singular vectors can be computed from the singular values and singular vectors of $X^\top X$.

The Arnoldi method is particularly useful when dealing with large sparse matrices, as the algorithm does not require explicitly the whole matrix, but only a matrix-vector multiplication result. Therefore, if the matrix-vector product can be computed efficiently, which is the case when X is sparse, ARPACK will be very efficient for large-scale matrices.

The ARPACK Fortran routine is used by Scanpy's PCA function through Scipy's `svds` function. In R, the *RSpectra* function provides an interface to the Spectra C++ library (Qiu and Mei 2025) that reimplements the ARPACK algorithms. Since Spectra implements the same underlying algorithm, we refer to both methods as ARPACK in the main text.

Random SVD The randomized SVD algorithm (Halko, Martinsson, and Tropp 2011) computes a near-best rank k approximation of a matrix using matrix-vector products with standard Gaussian vectors. The randomized SVD algorithm uses

a random projection matrix to sample the column space of the original matrix, allowing approximation of the SVD of the original matrix by computing SVD on a smaller matrix.

In detail, given a $n \times l$ orthonormal matrix Q, with $k \leq l \leq m$, such that $X \approx QQ^\top X$:

- Form $B = Q^\top X$;
- Compute the SVD of B, i.e., $B = \tilde{U}\Sigma V^\top$;
- Set $U = Q\tilde{U}$.

Since $X \approx QQ^\top X = Q(\tilde{U}\Sigma V^\top)$, it follows that $X \approx U\Sigma V^\top$. This algorithm is efficient when $l \ll m$, because we can efficiently compute the matrix B and then we just need to compute the SVD of a much smaller matrix.

The randomness comes from the construction of the orthonormal matrix Q, which is obtained by concatenating l randomly generated Gaussian random vectors (Halko, Martinsson, and Tropp 2011).

The randomized SVD algorithm is implemented in the *rsvd* R package, internally called by *BiocSingular* and in the `truncatedSVD` function in *scikit-learn*, which is called by Scanpy.

IRLBA The AugImplicitly Restarted Lanczos Bidiagonalization Algorithm (IRLBA) (Baglama and Reichel 2005) is a fast and memory-efficient method for computing truncated SVD of a matrix. It is based on the augmented implicitly restarted Lanczos bidiagonalization algorithm, which is a variant of the Lanczos algorithm that uses a bidiagonalization step to reduce the dimensionality of the matrix.

IRLBA computes sequences of projections of X onto specific low-dimensional subspaces that reduce the dimensionality of the problems. Restarting is implemented to iteratively reduce the approximation error.

More specifically, the algorithm applies l steps of partial Lanczos bidiagonalization to X and yields the decomposition

$$XP = QB,$$

where P and Q are orthonormal matrices of dimension $m \times l$ and $n \times l$, respectively, while B is upper bidiagonal. An approximate SVD of X can be obtained by the SVD of B and the matrices P and Q (Baglama and Reichel 2005).

When X is large, the storage requirement of the partial Lanczos bidiagonalization is large, unless the number of Lanczos bidiagonalization steps is small. However, for a small value of l, the SVD of X may be approximated poorly by the algorithm. To avoid this problem, the algorithm uses a sequence of initial vectors to form P, a technique known as restarted partial Lanczos bidiagonalization. Finally, to avoid numerical instabilities, restarting is carried out by augmentation of Krylov subspaces (see (Baglama and Reichel 2005) for details).

The IRLBA algorithm is implemented in the *irlba* R package, internally called by *BiocSingular*.

Incremental PCA (IPCA)

Incremental PCA (Ross et al. 2008) (IPCA) is a method for computing the principal components of a matrix in an incremental manner. This means that the algorithm processes the data in batches, computing the principal components for each batch and then combining them to obtain the final result. This approach avoids the need to load the entire dataset into memory, making it suitable for large-scale data analysis. The basic idea is to update the principal components incrementally as new data batches become available.

Here, we briefly illustrate the algorithm, assuming two batches of data. Let us denote with X_A and X_B the two batches of data. For the first batch, we compute the SVD in the usual way, say, $X_A = U_A \Sigma_A V_A^\top$. The goal is to find the SVD of the concatenation of X_A and X_B, which can be expressed as

$$[X_A \, X_B] = \left([U_A \, \tilde{X}_B] \tilde{U} \right) \tilde{\Sigma} \left(\tilde{V}^\top \begin{bmatrix} V_A^\top & 0 \\ 0 & I \end{bmatrix} \right),$$

where $\tilde{X}_B$ is the component of X_B orthogonal to U_A and $\tilde{R} = \tilde{U} \tilde{\Sigma} \tilde{V}^\top$ is the SVD of a matrix of size $k + d$, with d the number of observation in the batch X_B, which can be computed efficiently even for large n. The algorithm is slightly more complicated because it needs to take into account the fact that the sample mean of the data will change as more batches are added, the full details are in Ross et al. 2008.

The IPCA algorithm is implemented in the `IncrementalPCA` function in *scikit-learn*.

Jacobi

The Jacobi SVD algorithm (Drma and Veseli 2008) is an iterative algorithm that applies a sequence of plane rotations to a symmetric matrix to bring it to diagonal form. The Jacobi SVD algorithm is particularly known for its numerical stability and accuracy, making it suitable for high-precision applications.

The Jacobi method is an algorithm for the diagonalization of symmetric matrices and can be used to compute the SVD of general matrices. Indeed, applying the Jacobi method to $H = XX^\top$ leads to the sequence $A^{(k+1)} = A^{(k)}V^{(k)}$, whose limit matrix is $U\Sigma$, while the accumulated product of Jacobi rotations is V (Drma and Veseli 2008).

The Jacobi algorithm is implemented in the *RAPIDS* library.

1.2.3 SVD Performance Evaluation

SVD Accuracy

To evaluate the degree of similarity among the principal components obtained from different SVD algorithms, we computed the absolute Pearson correlation coefficient between the PC computed with the "exact" method and the corresponding PC derived from all other methods. The Pearson correlation coefficient measures the linear correlation between two variables, ranging from -1 to 1, where values close to zero indicate weak or no linear correlation, and values close to one indicate a strong linear correlation. By computing the absolute value of the correlation we account for PCA's invariance with respect to rotations. High values of absolute correlation coefficients indicate high similarity in the patterns captured by the principal components.

Computational Time

To measure the computational time for PCA computation in R, we employed the `proc.time` function, capturing the start and end times for each computational step. Conversely, in Python, we imported the time module and used the `time.time` function to capture the start time of the process, then using the `sys.argv` command we calculate the end time of the process.

Memory Usage

To monitor memory usage efficiently, we employed different strategies in R and Python. In R, we used the `Rprof` tool to profile memory consumption during program execution. With Rprof, we were able to track the memory consumption and then identify the peak of memory usage. Conversely, in Python, we used the `/usr/bin/time -v` command within our Slurm job scripts to measure and record the maximum memory usage. By appending this command to our job scripts, we captured memory usage statistics at the process level, allowing us to monitor and analyze resource utilization effectively.

Ranking SVD methods

The Plackett-Luce model (Maystre and Grossglauser 2015), (H. L. Turner et al. 2020) provides a flexible framework for modeling preferences and rankings in various domains. It handles ties of arbitrary order in the ranking. This means that the model can accommodate rankings in which items are tied for a particular rank. To model such rankings, we utilized the *PlackettLuce* R package, which extends the traditional Plackett-Luce model by incorporating methods that accommodate tied rankings. Using the `as.rankings` function, the data were restructured into a format suitable for analysis by converting preferences into rankings. The functions used included the weights argument, which allows for the weighting of observations based on the frequency of preferences, and the `log = FALSE` option, which transforms the scores into normalized probabilities, ensuring that their sum equals one. This step is crucial, as the first item in the ranking is treated as the reference for others, allowing the calculation of relative probabilities for each item to be placed in the top position.

Subsequently, we performed a summary analysis of the results using the `summary` function, which provided detailed information on the estimated coefficients of the model. To visualize the relative importance of each method, we used the `qvcalc` function, which calculates the worth values for each method, enabling the graphical representation of the results.

1.2.4 Infrastructure

PCA

The computational resources employed for the generation of the results presented in this research paper consisted of a dual-processor configuration featuring 2x AMD EPYC 7301 CPUs clocked at 2.7 GHz, specifically provided by DiBio UniPD. Additionally, substantial computational acceleration was achieved through the utilization of four NVIDIA A100 SXM4 GPUs, each equipped with 40GB of memory and integrated into the JetStream platform. These high-performance computing resources played a pivotal role in facilitating the execution of complex calculations and simulations, thereby contributing to the robustness and accuracy of the findings presented in this study.

BLAS/LAPACK version

To optimize the performance of linear algebra operations in R, the default Ubuntu BLAS/LAPACK implementation can be replaced with the OpenBLAS library, which provides substantial computational improvements. On Ubuntu systems, this configuration can be readily modified using the apt package manager. Initially, the desired OpenBLAS version must be installed using `sudo apt install libopenblas0-pthread` for the parallelized version, or `libopenblas0-serial` for the single-threaded implementation. Subsequently, the `update-alternatives` command is employed to register the new library as an available alternative, specifying a priority level (e.g., 110 for the pthread version). The system administrator can then interactively select the desired implementation through `sudo update-alternatives -config`. This procedure must be repeated for both BLAS and LAPACK libraries, replacing the `libblas.so.3` and `liblapack.so.3` files respectively. Proper configuration can be verified by executing `sessionInfo()` in R, which will display the path of the currently utilized library. This approach enables seamless transitions between different BLAS implementations without requiring R recompilation, thereby allowing researchers to select the configuration most suitable for their specific computational requirements.

Code availability

The code for the benchmark for Principal Component Analysis (PCA) at `https://github.com/billila/pca_scwf_paper`. Here the integration of Arpack algorithm in BiocSingular package: `https://github.com/billila/BiocSingular`. The definition file to create the container to run the R based analysis is available here: `https://github.com/billila/spca/blob/main/bioc_3_20_pca_wfsc.def`

Data availability

The 1.3M cell dataset is available as part of the TENxBrainData Bioconductor package at `https://bioconductor.org/packages/TENxBrainData`.

1.3 Data

1.3M cell dataset (1.3M)

This dataset contains the gene expression of 1.3 million brain cells isolated from E18 mice and was generated using the 10x Genomics platform for the investigation of cellular heterogeneity within the developing mouse brain at embryonic day 18 (E18) (G. X. Zheng et al. 2017). Cells from the cortex, hippocampus, and ventricular zone of two embryonic mice were dissociated and used to create 133 scRNA-seq libraries. The samples were sequenced on 11 Illumina Hiseq 4000 flow cells, resulting in a read-depth of approximately 18,500 reads per cell. The sequencing data were processed by Cell Ranger 1.2 to generate single-cell expression profiles of 1,308,421 cells. This dataset lacks a ground truth for the cell-type annotation.

The downsampling was performed at the cellular level using the `sample` function in R, selecting random subsets of 100k, 500k, and 1M cells without replacement. The total number of genes remained constant across all subsets.

1.4 Results

1.4.1 Benchmarking PCA

While quality control, normalization, and clustering are all essential steps in single-cell data analysis, we chose to focus on PCA as an exemplar step to thoroughly benchmark. In fact, PCA represents a key focal point of the entire pipeline as it is a non-trivial, often computationally heavy step, necessary for the subsequent stages of the analysis. Moreover, several algorithms exist to compute PCA and the choice of the algorithm can have profound impact on its computational load (Tsuyuzaki et al. 2020).

First, we give a brief introduction to PCA, reviewing standard methods for computing the principal components of a data matrix, i.e., the Singular Value Decomposition (SVD), as well as some of the most popular approximations, based on truncated SVD.

Briefly, PCA is a multivariate technique that consists of defining a new coordinate system defined by a set of orthogonal vectors, which correspond to the direction of maximal variability in the data. The principal components (PCs) are hence a set of linear combinations of the original variables (in our case the genes) that explains the most variation in the data.

PCA is typically used as a dimensionality reduction techniques, in which the first few components are retained and the data are projected onto their subspace. PCA ensures that this is the "best" linear subspace, in the sense that it preserves the most variation in the data.

In mathematical terms, the principal components are the eigenvectors of the covariance matrix and can be computed by its eigendecomposition. Alternatively, the PCs can be computed directly by the SVD of the original expression matrix. We refer the reader to Mardia, Kent, and Taylor 2024 and Hastie et al. 2009 for a more thorough introduction and the mathematical details.

SVD is a standard matrix factorization technique, and many algorithms exist for its computation (see e.g., Golub and Van Loan 2013). However, when only the first few PCs are needed, approximated methods, collectively known as truncated or partial SVD, are computationally advantageous (see Methods for details).

When choosing the appropriate SVD algorithm, the input format may play an important role. In fact, it is likely that an algorithm that very efficiently computes

the first k principal components when the data are stored in RAM memory, becomes highly inefficient when the data are stored out of RAM memory, as in the case of delayed operations on HDF5-backed datasets (Folk et al. 2011; Pagès 2025a; Virshup, Rybakov, et al. 2021). Similar considerations may be made for dense versus sparse matrix representations.

In our benchmark, we compared the computational time and RAM memory usage of six SVD algorithms, implemented in R and Python, applied to three different input formats on four subsets of the 1.3M cell dataset of increasing size, for a total of 28 combinations (Fig. 1.1c and Supplementary Table A.1). The process was repeated ten times for both time and memory assessments, for a total of 2,240 SVD calculations.

In particular, the SVD algorithms considered include arpack (Lehoucq, Sorensen, and C. Yang 1998), random (Halko, Martinsson, and Tropp 2011), exact Kalman 1996, irlba (Baglama and Reichel 2005), jacobi (Drma and Veseli 2008), and incrementalPCA Ross et al. 2008, with technical details provided in the Methods section. In R, we used the BiocSingular A. Lun 2023 and Rspectra Qiu and Mei 2025 packages, while in Python, we relied on Scanpy (Wolf, Angerer, and Theis 2018), Scikit-Learn (Kramer and Kramer 2016), and Rapids (Nolet et al. 2022) (Fig. 1.1c).

Several input data formats were considered: the data were stored either in an HDF5 (Folk et al. 2011) file or in RAM memory, and in each case they were stored as dense or sparse matrices. We compared these input formats with the *SparseArray* representation (Pagès 2025b) (Fig. 1.1c). This exploration allowed us to discern how the choice of input data influenced the results. For sparse matrices, whenever possible, we employed the "deferred" option, which postpones centering and scaling until after matrix multiplication to leverage the sparsity of the matrix for as long as possible.

To evaluate the scalability, we considered three datasets (random subsamples of the 1.3 Million brain cell dataset (G. X. Zheng et al. 2017)) with the same genes but with an increasing number of cells: 100k, 500k, 1M (see Methods for details). To ensure the robustness of the results, each method was executed 10 times on the same machine, using a single core, mitigating potential biases arising from parallel computation.

Indeed, certain algorithms are optimized for parallel processing, while others are not (Fig.1.2b). In some cases, increasing the number of cores does not enhance computational performance. For example, with the IRLBA algorithm, parallel

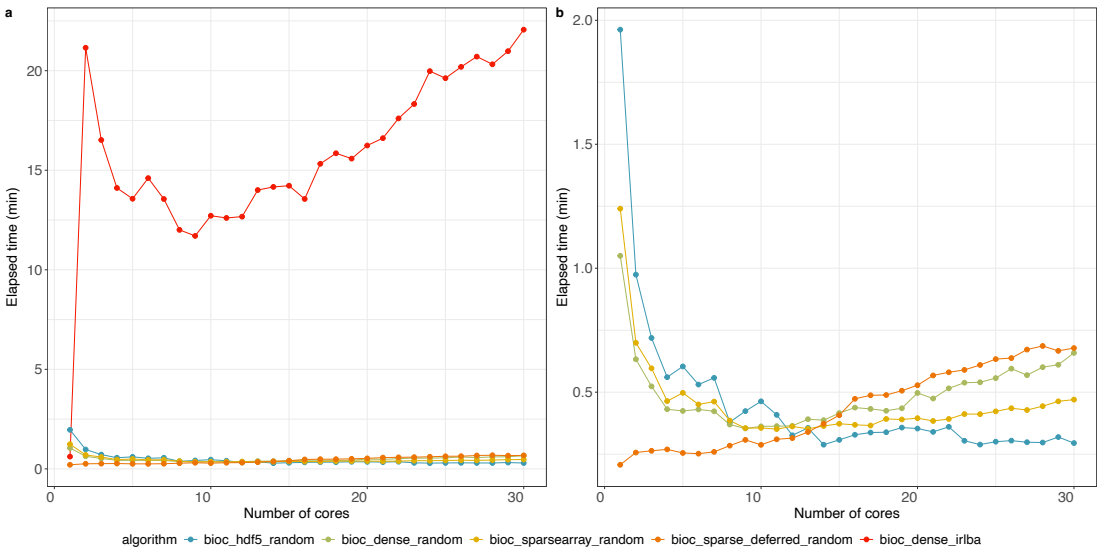

Figure 1.2: a. Elapsed time for increasing number of cores for best performing method for 100k in R. b. Same as a. without *bioc_dense_irlba*.

processing results in a substantial increase in computation time, rising from 0.39 minutes with a single core to 22.56 minutes with two cores (Fig.1.2a).

Hereafter, the term "method", which indicates the combination of algorithm and input type, is used as the comparison unit. The names of the methods follow this structure: package name + input data type + deferred (optional) + SVD algorithm.

Accuracy of truncated SVD

Before analyzing the performance in terms of computational time and memory usage, we evaluate the degree of agreement among the different methods to ensure that the faster algorithms return a good approximation of the true principal components. We measured the consistency across methods by computing the correlation between principal components (PCs). Because PCA is rotation-invariant and the singular vectors are defined only up to multiplication by 1, we report the absolute value of the correlation when comparing components, using the exact SVD algorithm as the ground truth.

In single-cell analysis, the number of selected PCs can vary substantially depending on the dataset and analytical requirements. However, a maximum of 50 PCs are often assumed to capture sufficient variability. Hence, we consider here only the first 50 PCs.

Figure 1.3 shows the correlation between exact SVD (bioc_dense_exact) and

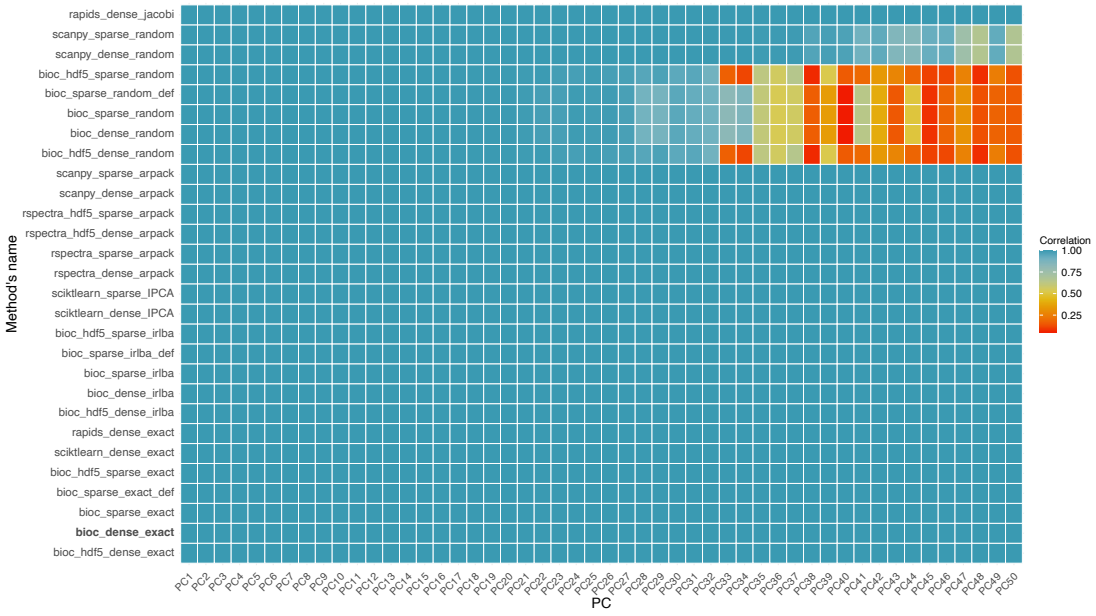

Figure 1.3: **Correlation heatmaps between principal components (PC).** Correlation heatmaps between principal components (PCs) obtained using different PCA algorithms and the Exact Algorithm as reference on 1.3M cell dataset. Each row represents a different combination of PCA algorithm and input matrix, and each column a principal component (PC1 - PC50). The color indicates the Pearson correlation between the PC obtained from each method and the corresponding PC from the exact reference.

all the other methods in random subsets of the 1.3M cell dataset. Strikingly, the correlation between exact SVD and most other methods is exactly equal to 1 for all considered PCs. The only exception is random SVD, which, especially in its R implementation, shows a significant decline in correlation values (up to values close to 0) starting from the 35th PC. This result implies that if one expect important information to be captured by the later PCs, random SVD is a poor algoritmic choice. However, it is important to note that typically the variance explained by the 35th and later PCs is very low. For instance, in the 1.3M cell dataset, the PCs from 35 to 50 in bioc_dense_random explain only 0.2% of the total variability (Table A.5). Hence, this result is unlikely to have a large practical importance for real data analysis.

Computational time and memory usage

Figure 1.4 reports the time (panel a) and memory consumption (panel b) for each combination of SVD algorithm implementation, computational platform,

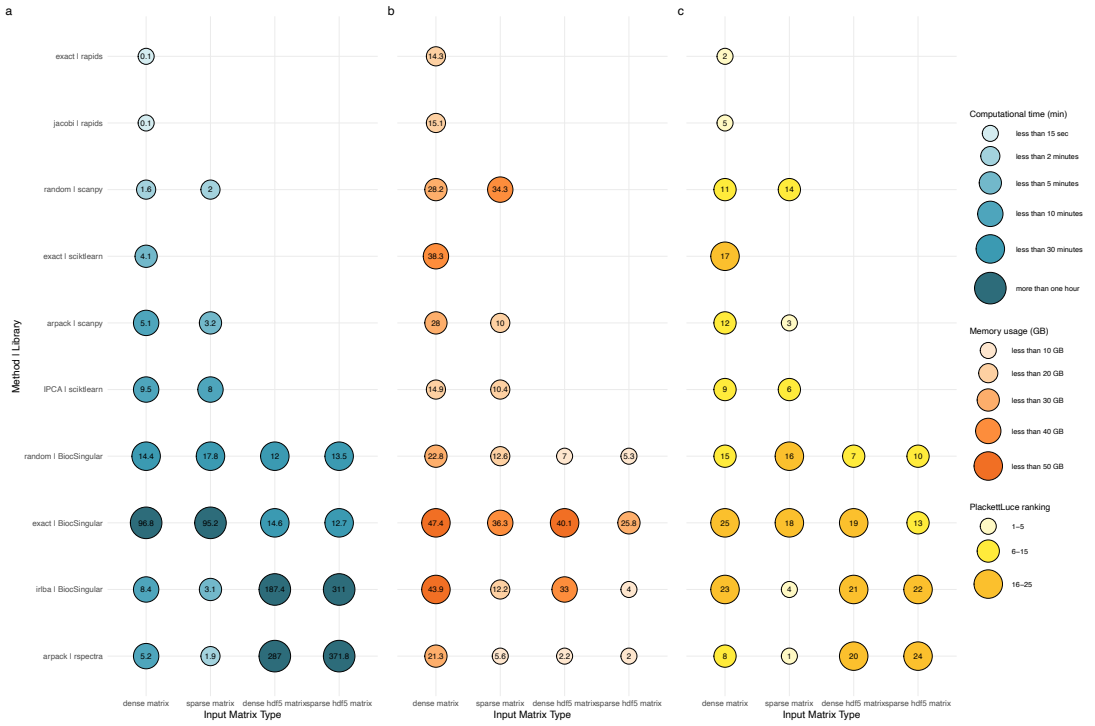

Figure 1.4: **Comparison of PCA method performance across input matrix types for 1.3M cell dataset.** Each panel shows a different evaluation criterion: (a) computational time in minutes, (b) peak memory usage in gigabytes (GB), and (c) PlackettLuce ranking (lower is better). Rows correspond to combinations of dimensionality reduction methods and software libraries, while columns represent different matrix formats (dense, sparse, HDF5-backed). Circle size is proportional to the measured value, and text labels indicate exact values. Color gradients group values into interpretable ranges.

and input type for the 1.3 million cell dataset (see Supplementary TableA.11 for the exact numerical results).

The GPU-aware RAPIDS library is the fastest approach, independent of the SVD algorithm, being at least one order of magnitude faster than CPU methods. Its memory footprint is moderate, even though it works only on dense matrices, providing an excellent choice for those that have a GPU available. In fact, it only takes 7.5 seconds to compute the exact SVD of a matrix with 1.3 million cells (Fig. 1.4a).

The performance of exact SVD using CPU is surprisingly different in stock installations of Python and R. Indeed, Python's scikit-learn implementation of exact SVD is 20 times faster than R's base *svd* function (Fig. 1.4a), despite both functions calling the same underlying LAPACK routine in FORTRAN (*gesdd*).

This surprising result can be explained by the different LAPACK configuration between the two systems: R's default BLAS/LAPACK on many systems is a reference implementation, which is not optimized for performance. Python, especially in environments like Conda (contributors 2025), usually links to highly optimized libraries like OpenBLAS (Xianyi, Qian, and Yunquan 2012) or MKL (Intel Math Kernel Library). These optimized libraries use hardware-specific vectorization and threading to speed up linear algebra. In fact, running R's exact SVD with optimized OpenBlas results in a 15X speed up, leading to computing times similar to those of Python (Supplementary TableA.12)

When the input matrix is large and one does not need the full singular values/vectors, truncated SVD algorithms are advantageous compared to computing the exact SVD, both in terms of speed and memory. The fastest approaches are random SVD as implemented in Scanpy, both applied to dense and sparse matrices (1.6 and 2 minutes, respectively, Fig. 1.4a) and the ARPACK algorithm applied to sparse matrices, especially as implemented in the RSpectra package (1.9 minutes, Fig. 1.4a). Notably, using a sparse matrix representation as input saves RAM memory, using e.g., 5.6GB for RSpectra sparse, compared to 21.3GB for RSpectra dense. Similar considerations can be made for ARPACK in Scanpy and random SVD in BiocSingular, but surprisingly random SVD in Scanpy uses more RAM memory for sparse matrices than for dense matrices, perhaps due to the fact that centering the matrix prior to SVD removes sparsity. Note that BiocSingular applies centering in a deferred way to preserve sparsity for the SVD calculations.

IRLBA is a fast alternative for truncated SVD for in-memory matrices, especially when represented in a sparse format (3.1 minutes, Fig. 1.4a). However, IRLBA and ARPACK are both very slow when applied to out-of-memory data, exceeding 1.5 hours when applied to HDF5 matrices. This is likely due to the multiple passes on the data needed by these algorithms, which make them suffer from the I/O bottleneck. On the other hand, exact and random SVD perform better for HDF5 than for in-memory inputs, making it the best choice for out-of-memory data: e.g., random SVD applied to dense HDF5 data takes 12 minutes.

We next used the Plackett-Luce model (H. L. Turner et al. 2020; Maystre and Grossglauser 2015) to establish a single, final method ranking, encompassing computational time, memory usage, and the average correlation among the top 50 principal components (Fig. 1.4c). The ranking clearly highlights four methods

that outperform all the other: ARPACK sparse (both implemented in RSpectra and Scanpy), IRLBA sparse, and Rapids dense (both exact and Jacobi).

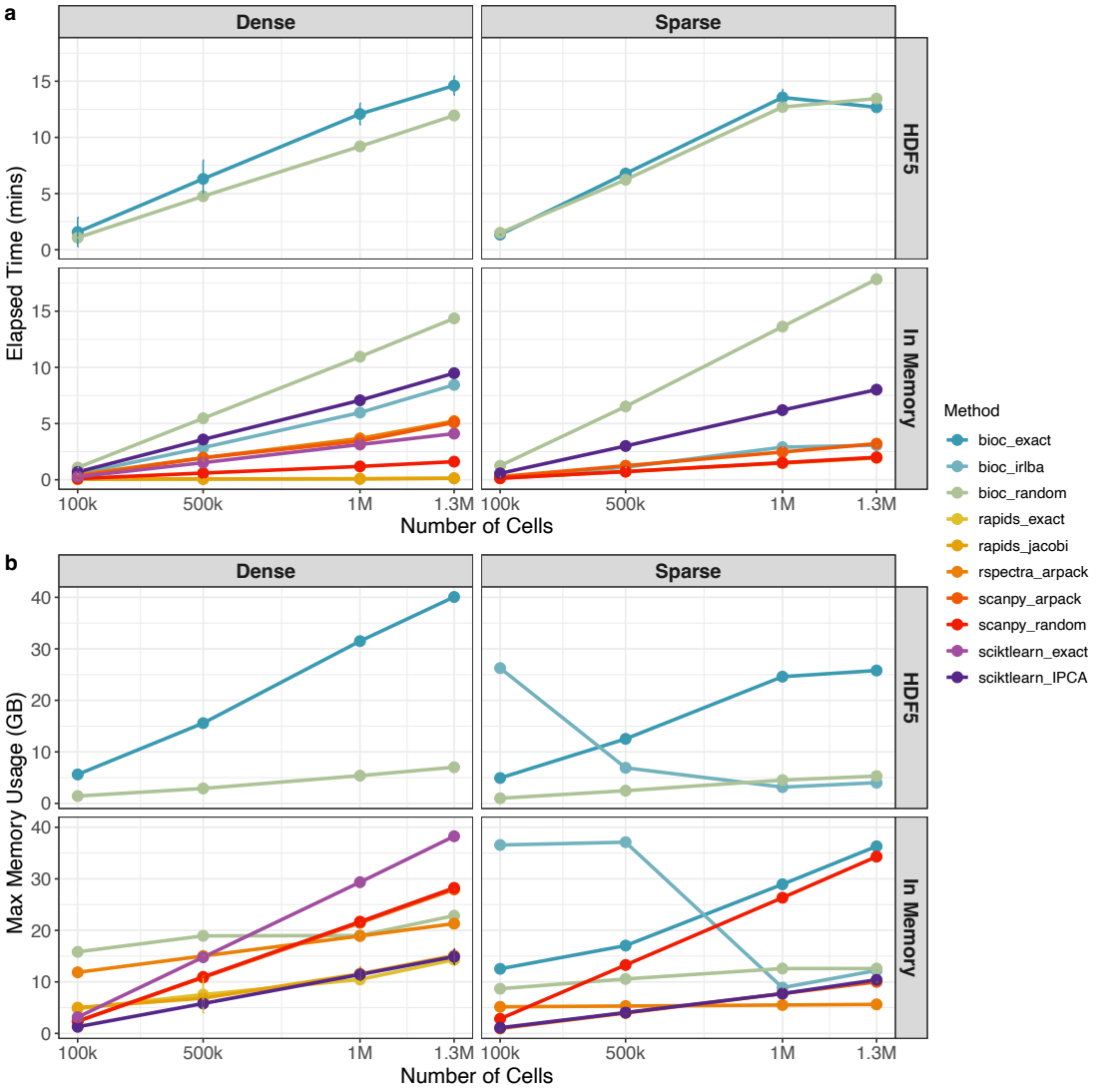

Figure 1.5: **Scalability Assessment of PCA Methods by Input Dimensions, Runtime, and Memory Consumption** (a) Elapsed time (in minutes) required to perform principal component analysis (PCA) across a range of dataset sizes (100k, 500k, 1M, 1.3M cells), using different combinations of methods, matrix formats (dense or sparse), and storage types (in-memory or HDF5). Only methods with a maximum average execution time below 75 minutes are shown. (b) Maximum memory usage (in GB) required to perform principal component analysis (PCA) across a range of dataset sizes (100k, 500k, 1M, 1.3M cells), using different combinations of methods, matrix formats (dense or sparse), and storage types (in-memory or HDF5). Only methods whose maximum memory usage did not exceed 42 GB are shown.

Finally, we assessed the scalability of each algorithm by taking random subsets of the 1.3M cell dataset and evaluate the time and memory consuption of each method across dataset sizes (Fig. 1.5a and Fig. A.1). In terms of time, all methods scale linearly with the number of cells. Strikingly, with dense input matrices, all Python methods are more scalable than R methods: this is largely due to the lack of BLAS optimization in the default R installation. Linking R to an optimized BLAS/LAPACK almost entirely accounts for this difference (see Supplementary TableA.12). On the other hand, R is very efficient at handling sparse matrices, thanks in part to the BiocSingular ability to defer centering and scaling (Fig. 1.5a and Fig. A.1). The IRLBA algorithm is by far the slowest algorithm when the input is out of memory, due to the multiple read accesses (Fig. A.1; see Methods).

In terms of RAM memory, as expected, sparse matrices lead on average to lower usage than dense matrices (Fig. 1.5b). Interestingly, random SVD and ARPACK show better RAM scalability than other methods, as evident from the lower slope of the lines (Fig. 1.5b); this confirms that for larger data sets ARPACK is a good choice. A surprising pattern is observed for IRLBA on sparse matrices (both in and out of memory): RAM usage is much higher for low-dimensional matrices than for much higher dimensions. The IRLBA algorithm involves multiple read access to the input matrix (two per iteration; see Methods) and caching in memory may be employed by the algorithm for smaller matrices to improve computational time.

Taken together, these results show that the choice of the algorithm is critically dependent on the input data and the computing infrastructure. If the data are small enough to fit in the GPU's VRAM, this is by far the most computationally efficient method; if one cannot use a GPU, but the data fit in the CPU's RAM, ARPACK, random SVD, and IRLBA are all good choices. If the data are sparse, RSpectra is the fastest and most memory-efficient implementation, while if the data are dense, Scanpy's random SVD method is the fastest, while incremental PCA is the most memory-efficient. Finally, when the data are out-of-memory, the best approach is random SVD as implemented in BiocSingular, which results in a reasonable time and a low memory footprint.

1.5 Discussion and conclusion

The ever-increasing size of single-cell datasets and the increasing avilability of single-cell atlases has rendered the scalability of workflows, both in terms of speed and memory usage, crucial. Focusing on PCA as an exemplar step, we demonstrated that the type of input, the choice of algorithm, and the software configuration all critically influence the scalability of the methods.

Accelerating single-cell analyses with GPUs is a promising avenue for the near future: while GPUs are routinely used in deep learning, also in the context of single-cell analyses (Lopez et al. 2018), their use to accelerate matrix multiplications and other common steps in the analysis workflows is still not widespread. One notable exception is the rapids single-cell library, that we proved accurate and scalable in terms of both time and memory usage. In fact, their use in PCA calculations results in a speed up of approximately 16X compared to the fastest CPU alternative. More work is needed to make it easy for R/Bioconductor developers to leverage GPU computations in their packages: the GPUMatrix CRAN package (Lobato-Fernandez, A.Ferrer-Bonsoms, and A. Rubio 2025) seems to be a promising starting point to achieve this goal. One limitation of GPU is the amount of VRAM available: while the datasets used in this benchmark were small enough to fit in VRAM, for larger datasets the amount of memory needed is a critical factor for the use of GPU, as larger-than-VRAM datasets may lead to out-of-memory error or result in slower performance due to data transfer between RAM and VRAM.

BLAS/LAPACK optimization has, perhaps not surprisingly, a profound impact on the computational performance of the methods. Indeed, the same exact R code can be sped up by 15X by simply linking to an optimized BLAS/LAPACK instead of using the default reference implementation. In this benchmark, we have decided to keep the default BLAS/LAPACK version in R and Python to mimic the experience of the typical user, who will likely keep the default R installation, perhaps not even realizing that such an important performance gain can be achieved with a different BLAS/LAPACK implementation. Nonetheless, our recommendation is to run R with an optimized BLAS/LAPACK and we provide instructions to do so in Ubuntu with a simple apt call (see Methods).

Chapter 2

Comparison scRNA-Seq workflow

2.1 Introduction

A typical scRNA-seq analysis workflow, after preprocessing, comprises several steps: (Fig. 1.1a): (i) quality control; (ii) gene and cell filtering; (iii) normalization; (iv) identification of highly variable genes; (v) dimensionality reduction, generally employed using principal component analysis (PCA) or similar methods; (vi) data visualization, using methods such as t-SNE and UMAP, which are commonly applied to data reduced through PCA; (vii) clustering for the identification of groups of cells with similar transcriptional profiles; and (viii) cell type annotation, the process by which cells or clusters are labeled using either external reference datasets or the expression of known marker genes (Amezquita et al. 2020).

Each of these steps requires its own processing time and memory usage and, in the last few years, several approaches have been proposed in different programming languages (mostly R and Python).

While there have been attempts in the literature to benchmark individual analysis steps, such as normalization, dimensionality reduction, clustering, trajectory inference, cell-type annotation, data integration (Luecken et al. 2022; L. Yu et al. 2022; Mereu et al. 2020; Saelens et al. 2019; Tran et al. 2020; Tsuyuzaki et al. 2020), the comparison of entire analysis workflows has received less attention. (Rich et al. 2024) explores the impact of parameter choices and their default values in Seurat and Scanpy, but does not thoroughly examine their scalability. On the other hand, (Tian et al. 2019), while benchmarking all the different steps of a typical analysis, does not consider the workflow typically recommended by

the most popular frameworks. Moreover, the dataset used for benchmarking is relatively small and insufficient to tackle scalability issues.

Here, we focus on the three most popular analysis frameworks, namely, Bioconductor (Amezquita et al. 2020), scverse/Scanpy (Wolf, Angerer, and Theis 2018), and Seurat (Y. Hao, Stuart, et al. 2023) (Fig. 1.1b). For Bioconductor, we consider both the standard Orchestrating Single-Cell Analysis (OSCA) workflow and scrapper, a more recent efficient implementation that offloads most computations to C++ (see Methods). To compare these workflows, we rely on their respective documentation, purposely choosing the "quick start" or "basic" tutorials, which comprise the steps that are always present in one's analysis; more advanced methods, such as multi-sample integration, batch-effect reduction, reference-based annotations, are not included in our benchmark.

While traditional approaches rely on CPU architectures, although often allowing for multithreading and parallel computing, recent efforts have explored the use of GPUs to enhance speed and scalability (Nolet et al. 2022). Hence, we include in our benchmark the rapids-singlecell framework (Dicks et al. 2024), an open-source library that implements a standard single-cell analysis workflow leveraging NVIDIA GPUs (Fig. 1.1b). In this context, benchmarking tools and workflows is essential for evaluating the performance of different methods in different experimental settings (such as the number of cells, sequencing depth, or biological complexity) not just in terms of efficiency, but to ensure that scalability does not come at the price of reduced accuracy.

The goal of this chapter is to benchmark these workflows in terms of computational performance and scalability, providing insights into how different frameworks handle increasingly large datasets. By comparing CPU- and GPU-based implementations, we aim to establish practical guidelines for researchers seeking efficient and reliable workflows for single-cell RNA-seq analysis.

2.2 Methods

2.2.1 A typical workflow for Single Cell RNA-seq data analyses

Here, we describe the steps of a typical workflow for single-cell RNA-seq (scRNA-seq) data. These steps are common to all five pipelines that we benchmarked in this chapter. After a general description of all the steps, we describe how each specific workflow differ in the implementations of these steps.

Preprocessing

Low-quality libraries in scRNA-seq data can result from various sources, such as cell damage during dissociation or errors in library preparation (e.g., inefficient reverse transcription or PCR amplification). These typically manifest as "cells" with low total counts, few expressed genes, and high proportions of mitochondrial reads, leading to misleading results in downstream analyses (McCarthy et al. 2017). After excluding empty droplets and identifying potential doublets, droplets containing damaged cells or low-quality reads are filtered out. Droplets are small compartments that capture single cells and barcoded beads during scRNA-seq experiments. Empty droplets lack cells, while doublets occur when two cells are mistakenly captured in the same droplet, creating mixed signals that can compromise data accuracy. The library size, defined as the total sum of counts across all relevant features for each cell, is a commonly used metric for filtering. Cells with small library sizes are more likely to be of low quality due to RNA loss during library preparation, either from cell lysis or inefficient cDNA capture and amplification. Another metric is the number of expressed features per cell, defined as the number of endogenous genes with non-zero counts for that cell. Cells with very few expressed genes are likely of poor quality as the diverse transcript population has not been successfully captured. The proportion of reads mapped to mitochondrial genes can also be used; high proportions suggest possible loss of cytoplasmic RNA due to cell damage, as mitochondria, being larger than individual transcript molecules, are less likely to escape through cell membrane holes.

Normalization

Normalization aims to adjust raw counts for variable sampling effects by scaling the observed counts. This process helps to reduce technical noise and preserve biological variation. Systematic differences in coverage between libraries in scRNA-seq data, often due to variations in cDNA capture or PCR amplification efficiency, can interfere with expression profile comparisons. Normalization removes these differences, ensuring accurate clustering and differential expression analyses (Vallejos et al. 2017).

Feature selection

Feature selection aims at reducing the dimension of the input matrix to identify genes whose expression is variable enough across cells to be able to capture most of the cell heterogeneity filtering out noise. Methods such as dimensionality reduction and clustering are generally performed using the filtered matrix (Yip, Sham, and Junwen Wang 2019; Andrews and Hemberg 2019; Ahlmann-Eltze and Huber 2023).

Dimensionality Reduction: PCA, t-SNE, UMAP

Dimensionality reduction techniques are typically used for the visualization of complex datasets. These methods aim to transform high-dimensional data into a lower-dimensional space preserving the underlying structure and variability inherent in the data.

Principal Component Analysis (PCA) is widely employed in scRNA-seq to identify orthogonal axes (principal components) that capture the maximum variance in the dataset. By projecting cells onto these components, PCA provides a simplified representation that retains the most significant sources of variation.

In contrast, t-Distributed Stochastic Neighbors Embedding (t-SNE) (Maaten and Hinton 2008) focuses on preserving local similarities between cells in the high-dimensional space. It maps cells into a low-dimensional space (often 2D) that aims at placing similar cells together. Uniform Manifold Approximation and Projection (UMAP) (McInnes et al. 2018) is a more computationally efficient alternative that has gain popularity in recent years.

It is a standard practice that both t-SNE and UMAP take as input the first 50 PCs obtained from the pre-processed gene expression matrix, where 50 is a number of PCs that should guarantee to capture most of the cell heterogeneity present in the data.

Clustering: Louvain, Leiden

Clustering approaches such as Louvain (Blondel et al. 2008) and Leiden (Traag, Waltman, and Van Eck 2019) are used to identify discrete cell populations based on gene expression patterns.

These methods operate on a graph representation of the data, often constructed using k-nearest neighbors (KNN) or shared nearest neighbors (SNN)

algorithm, where nodes represent cells, and edges represent similarities or connections based on gene expression profiles.

An important concept to this clustering approaches is modularity, which is used to evaluate and optimize the grouping of cells into meaningful clusters, also referred to as communities. Modularity is a measure used to evaluate how well cells are grouped into clusters based on their gene expression profiles. It quantifies the quality of clustering by comparing the density of connections (or similarities) between cells within the same cluster to those between cells in different clusters. Higher modularity indicates that cells within a cluster are more tightly connected (e.g., share similar gene expression patterns) compared to connections between clusters. Lower modularity implies that the boundaries between clusters are less distinct, potentially reflecting noise or suboptimal clustering. In the single-cell RNA-seq context, communities refer to groups of cells that are more connected or similar to one another based on similarity metrics, such as shared gene expression patterns, than they are to cells in other groups. These communities may represent biologically meaningful structures, such as cell types, subtypes, or states, within scRNA-seq datasets.

2.2.2 Workflows for Single Cell RNA-seq data analyses

Seurat

Seurat, developed and maintained by the Satija Lab, is an R toolkit tailored for single-cell genomics. For our analysis, we utilized Seurat version 5 (Y. Hao, Stuart, et al. 2023).

Specific steps and default values of the Seurat workflow include:

- Find Mitochondrial Genes: This step identifies mitochondrial genes by calculating the proportion of transcripts mapping to mitochondrial genes, using the `PercentageFeatureSet` function with the regex pattern `"^MT-"`or `"^mt-"` from *Seurat*. Cells with high mitochondrial content (less than 5 %) or low total counts are flagged and filtered out.
- Filtering: A filtering step is applied to remove cells and genes with undesired properties. To do the subset is used the `subset` function from *SeuratObject*. Cells are retained if they express between 200 and 5000 features, have mitochondrial content below 5%, and a total count under 25,000 for BE1 dataset. Cells are retained if they express between 200 and 2500 features, have mitochondrial content below 5%, and a total count under 4,000 for cb dataset. Cells are retained if they express between 200 and 6200 features, have mitochondrial content below 5%, and a total count under 60,000 for sc_mixology dataset.

- Normalization: The data is normalized using the `LogNormalize` method, which scales gene expression counts by the total expression for each cell, followed by a log transformation to stabilize variance.
- Highly Variable Genes: Highly variable genes are identified using the variance-stabilizing transformation (vst) method, with the top 1,000 features selected. The function used is `FindVariableFeatures` from *Seurat*
- Scaling: All genes are scaled using the `ScaleData` function, which centers the expression values around zero and normalizes variance.
- PCA: Principal Component Analysis (PCA) is conducted using the highly variable genes and retaining the first 50 PCs. The algorithm used is IRLBA.
- t-SNE: `RunTSNE` from *Seurat* is applied on the first 50 PCs. The perplexity parameter is set to 18.
- UMAP: The `runUMAP` function of the `Seurat` package reduces dimensionality starting from the top 50 PCs using the *uwot* R package implementation, using the cosine metric.
- Louvain: `FindNeighbors` function, a Shared Nearest Neighbor (SNN) Graph is constructed setting k equal to 20. `FindClusters` function is applied setting *algorithm* parameter equal to 1 for Louvain algorithm.. The Louvain algorithm identifies clusters by constructing a k-nearest neighbors graph (based on the top 50 components) and optimizing modularity. Clusters are generated with a resolution of 0.2 for BE1, 0.1 for sc_mix, 0.2 for cb.
- Leiden: `FindClusters` function is applied setting *algorithm* parameter equal to 4 for Leiden algorithm. The Leiden algorithm identifies clusters by constructing a k-nearest neighbors graph (based on the top 50 components) and optimizing modularity. Clusters are generated with a resolution of 0.2 for BE1, 0.08 for sc_mix, 0.2 for cb.

Orchestrating Single-Cell Analysis (OSCA)

Orchestrating Single-Cell Analysis (OSCA) with Bioconductor is a framework and set of tools within the Bioconductor ecosystem for analyzing single-cell RNA seq data. It focuses on integrating multiple R packages to provide a comprehensive workflow for processing, analyzing, and visualizing single-cell datasets. Bioconductor uses the SingleCellExperiment class for storing single-cell assay data and metadata: count matrices, are stored in the assays component, where rows represent features (e.g. genes, transcripts) and columns represent cells. In addition, low-dimensional representations of the primary data, and metadata describing cell or feature characteristics can also be stored in

the SingleCellExperiment object. By standardizing the storage of single-cell data and results, Bioconductor supports interoperability between single-cell analysis packages.

Specific steps and default values of the OSCA workflow include:

- Find Mitochondrial Genes: Mitochondrial genes are identified by mapping gene symbols to their chromosomal location using the *EnsDb.Hsapiens.v75* package. Metrics such as mitochondrial percentage and total counts per cell are calculated using the `perCellQCMetrics` function from *scuttle*. Cells with high mitochondrial content (less than 5%) or low total counts are flagged and filtered out.

- Filtering: Cells are filtered based on sequencing depth and quality control metrics obtained in the previous step.

- Normalization: The `logNormCounts` function of the *scuttle* package normalizes gene expression data by calculating size factors and performing log-transformation.

- Highly Variable Genes: The `modelGeneVar` function of the *scran* Bioconductor package models gene expression variance across cells, and the `getTopHVGs` function selects the top 1,000 most variable genes.

- Scaling: In the OSCA suggestion there isn't a specific function for running the scaling step. In this case, we set `scale = TRUE` in the `runPCA` function to compute the scaling.

- PCA: The `runPCA` function of the *scater* package reduces dimensionality using the previously identified HVGs, by default using the Exact algorithm from `runSVD` in *BioCSingular* package and computing the first 50 PCs. The `runPCA` function performs scaling as part of the PCA computation.

- t-SNE: The `runTSNE` function of the *scater* package reduces dimensionality starting from the top 50 PCs. By default, the function will set a perplexity that scales with the number of cells.

- UMAP: The `runUMAP` function of the *scater* package reduces dimensionality starting from the top 50 PCs.

- Louvain: The `clusterCells` function from *scran*, with the `NNGraphParam` parameter, applies the Louvain algorithm to cluster cells based on the top 50 PCs and using a resolution of 0.5 for all the datasets. This function is a wrapper around `clusterRows` function from the *bluster* package.

- Leiden: The `clusterCells` function, configured with the `NNGraphParam` parameter, applies the Leiden algorithm to to cluster cells based on the top 50 PCs and using a resolution of 0.5 for all the datasets.

scrapper

The Bioconductor package *scrapper* reimplements some of the OSCA function in C++. It provides R bindings to C++ code for the analysis of single-cell expression data, primarily utilizing various *libscran* libraries. Each function within the workflow addresses a specific step in the single-cell analysis pipeline, including tasks such as quality control, clustering, and marker detection.

Specific steps and default values of the scrapper workflow include:

- Find Mitochondrial Genes: Mitochondrial genes are identified by applying a regular expression pattern `"^mt-"` or `"^MT-"` to the row names of the count matrix. This step allowed for the selection of genes associated with mitochondrial function.
- Filtering: RNA quality control metrics are computed using the `computeRnaQcMetrics` function, which included subsets for mitochondrial genes. Suggested thresholds for filtering were determined using the `suggestRnaQcThresholds` function. The `filterRnaQcMetrics` function was then applied to retain cells that met the established quality criteria.
- Normalization: The filtered count matrix was log-normalized using size factors derived from the RNA quality metrics using `centerSizeFactors`. This log-normalization was performed using the `normalizeCounts` function, ensuring that the data were adjusted for sequencing depth and other technical variations.
- Highly Variable Genes: Gene variances were modelled using the `modelGeneVariances` function. Highly variable genes were identified using the `chooseHighlyVariableGenes` function, selecting the top 1000 genes based on their variability.
- Scaling: In this package there isn't a specific function for running the scaling step. In this case, we set `scale = TRUE` in the `runPca` function to compute the scaling.
- PCA: PCA was conducted on the normalized data of the highly variable genes using the `runPca` function, which by default computes the first 25 PCs. To keep the PCA results comparable, we decided to compute the first 50 PCs.
- t-SNE: t-SNE was applied to the fist 50 PCs and a default perplexity of 30. This was executed using the `runTsne` function.
- UMAP: UMAP was performed on the first 50 PCs, and the default number of neighbors set to 15.
- Louvain: not supported in this package.

- Leiden: A shared nearest neighbor (SNN) graph was constructed from the top 50 PCs using the `buildSnnGraph` function. The Leiden algorithm was applied to this graph with a resolution of 0.18 for BE1 dataset, 0.16 for sc_mix, 0.20 for cb.

Scanpy

Scanpy Wolf, Angerer, and Theis 2018 is a scalable Python package for analyzing single-cell gene expression data, encompassing a wide array of functionalities such as preprocessing, visualization, clustering, pseudotime and trajectory inference, differential expression testing, and simulation of gene regulatory networks.

Specific steps and default values of the Scanpy workflow include:

- Find Mitochondrial Genes: This step identifies mitochondrial genes by annotating those whose names start with `"MT-"` or `"mt-"`. Quality control (QC) metrics, including the percentage of mitochondrial counts per cell, are calculated using `sc.pp.calculate_qc_metrics`
- Filtering: A filtering step is applied to remove cells and genes with undesired characteristics using `scanpy.pp.filter_cells` and `scanpy.pp.filter_genes` functions in *scanpy* package. For BE1 and sc_mix dataset cells expressing fewer than 200 genes and genes present in fewer than 3 cells are excluded. Additionally, cells with an excessive number of expressed genes (> 5000) or a high percentage of mitochondrial counts (> 5%) are filtered out. For cb dataset cells expressing fewer than 200 genes and genes present in fewer than 3 cells are excluded. Additionally, cells with an excessive number of expressed genes (> 2500) or a high percentage of mitochondrial counts (> 5%) are filtered out.
- Normalization: `scanpy.pp.normalize_total` function normalizes each cell by total counts over all genes. `scanpy.pp.log1p` perform log-transformation on the normalized gene expression.
- Highly Variable Genes: Highly variable genes are identified using `sc.pp.highly_variable_genes` function using Seurat method. Specific thresholds for mean expression (min_mean=0.0125, max_mean=3) and dispersion (min_disp=0.5). Up to 1,000 highly variable genes are selected for downstream analyses.
- Scaling: All genes are scaled using `scanpy.pp.scale` function, data is scaled to center values around zero and normalize variance, with a maximum absolute value of ś10.
- PCA: Principal Component Analysis (PCA) is performed using the Arpack SVD solver and retaining the first 50 PCs. The function used is `scanpy.tl.pca`

- t-SNE: t-SNE was applied to the first 50 PCs and a default perplexity of 30. This was executed using the `scanpy.tl.tsne` function.
- UMAP: UMAP is applied to the first 50 PCs and the number of neighbors is set by default to 10. The function used is `scanpy.tl.umap`.
- Louvain: The nearest neighbors distance matrix and a neighborhood graph of observations is computed with `scanpy.pp.neighbors` function. The size of local neighborhood (in terms of the number of neighboring data points) used for manifold approximation is 10 on the top 50 PCs. The Louvain clustering algorithm is applied with `scanpy.tl.louvain` function and a resolution of 0.13 for BE1 dataset, sc_mix and for cb.
- Leiden: Similar to Louvain, the Leiden algorithm performs clustering using this `scanpy.tl.leiden` function with a resolution of 0.13 for BE1 dataset, sc_mix and for cb.

rapids_singlecell

rapids_singlecell (Virshup, Bredikhin, et al. 2023) is a Python-based workflow that utilizes GPU computing to enhance the analysis of single-cell data. This workflow is integrated within the *scverse* environment, and its structure is built upon *Scanpy*. By leveraging GPU capabilities through CuPy and NVIDIAs RAPIDS framework, this approach prioritizes computational efficiency, facilitating the rapid analysis of large-scale single-cell datasets.

Specific steps and default values of the rapids_singlecell workflow include:

- Find Mitochondrial Genes: Mitochondrial (MT) and ribosomal (RIBO) genes are identified and annotated based on their respective prefixes (`"MT-"` and `"RPS"`) using *rsc.pp.flag_genefamily*. Quality control (QC) metrics, including percentages of MT and RIBO counts, are calculated to assess sequencing quality using *rsc.pp.calculate_geneqc_genemetrics* function.
- Filtering: A filtering step is applied to remove cells and genes with undesired characteristics using `rsc.pp.filter_cells` and `rsc.pp.filter_genes` functions in *rapids_singlecell* package. For BE1, sc dataset cells expressing fewer than 200 genes and genes present in fewer than 3 cells are excluded. Additionally, cells with an excessive number of expressed genes (> 5000) or a high percentage of mitochondrial counts (> 5%) are filtered out. For cb dataset For sc_mix dataset
- Normalization: `rsc.pp.normalize_total` function normalizes each cell by total counts over all genes. `rsc.pp.log1p` perform log-transformation on the normalized gene expression.
- Highly Variable Genes: Highly variable genes are identified using the Seurat v3 method with `n_top_genes=1000` and the counts layer.

- Scaling: Data is scaled to zero mean and unit variance, with a maximum absolute value of plus/minus 10 to prevent extreme values from dominating the analysis.
- PCA: PCA is performed using 50 components and the auto algorithm could be one between Exact or Jacobi.
- t-SNE: t-SNE is applied to the first 50 PCs and with a perplexity of 30. *rapids_singlecell.tl.tsne* is used to perform t-sne.
- UMAP: UMAP is applied to the first 50 PCs and the number of neighbors is set by default to 10. *rapids_singlecell.tl.umap* is used to perform umap.
- Louvain: the nearest neighbors distance matrix and a neighborhood graph of observations is computed with *rapids_singlecell.pp.neighbors* function. he size of local neighborhood (in terms of number of neighboring data points) used for manifold approximation is 10 on the top 50 PCs. The Louvain clustering algorithm is applied with `rapids_singlecell.tl.louvain` function and a resolution of 0.6 for BE1 dataset, 0.1 for sc_mix, 0.1 for cb.
- Leiden: Similar to Louvain, the Leiden algorithm performs clustering using this `rapids_singlecell.tl.leiden` function with a resolution of 0.6 for BE1 dataset, 0.1 for sc_mix, 0.1 for cb.

2.2.3 Workflow performance evaluation

ARI

The Adjusted Rand Index (ARI) (Rand 1971) is a statistical measure commonly used to assess the similarity between two clustering results. It takes into account both the agreement and disagreement between data points in the original and clustered datasets, providing a normalized index that ranges from 0 to 1. In this benchmark, we opt to employ the ARI for the comparative assessment of various pipelines comparing the known cell type annotation with that obtained after the cluster algorithms.

Cell type purity

Cell type purity is defined as the percentage of neighbors for each cell that belong to the same cell type. Well-separated cell types should exhibit high purity as the cells from different types do not mix. Low purity corresponds to a data representation in which cells of different types are mixed together. We computed the cell type purity using the function `neighborPurity` from

the *bluster* Bioconductor package (A. Lun 2025), using the known cell types as "clusters" and the default value of $k = 50$ nearest neighbors.

Variability explained by cell types

To quantify the proportion of total variability explained by the cell types, we employed a strategy inspired from the *scDiagnostics* Bioconductor package (Christidis et al. 2025). Briefly, for each method and each PC, we multiplied the percentage of variance explained by each PC by the R^2 index of a linear model that uses that PC as a response variable and the cell type as covariate.

2.2.4 Infrastructure

Pipeline Single Cell

In the pursuit of benchmarking the single-cell pipelines presented in this research paper, computational resources from the CAPRI facility at the University of Padova were leveraged. The CPU component of this infrastructure was comprised of 16 Intel(R) Xeon(R) Gold 6130 processors, each operating at a clock speed of 2.10GHz (CAPRI UniPD). Additionally, we utilized 2 NVIDIA Tesla P100 GPUs for GPU-intensive tasks, each equipped with 16GB of memory (CAPRI UniPD). These resources, available through CAPRI, played a crucial role in executing the computational workflows associated with the benchmarking analyses, ensuring the efficiency and accuracy of the results detailed in this study.

Code availability

The code for the Single-cell RNA-seq Workflow benchmark is accessible via the GitHub repository at `https://github.com/billila/pca_scwf_paper`.

The definition file to create the container to run the R based analysis is available here: `https://github.com/billila/spca/blob/main/bioc_3_20_pca_wfsc.def`

Data availability

sc_mixology dataset is available at `https://ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE118767` and more tecnichal details at `https://github.com/`

`LuyiTian/sc_mixology`. The BE1 dataset is available at `https://doi.org/10.6084/m9.figshare.23939481.v1`. The cb dataset is available as parte of the *SingleCellMultiModal* package at `https://bioconductor.org/packages/SingleCellMultiModal`

2.3 Data

Cite-seq coord blood (cb)

CITE-seq (Cellular Indexing of Transcriptomes and Epitopes by sequencing) applied to cord blood samples provides a comprehensive and high-dimensional dataset that captures both transcriptomic and proteomic information at the single-cell level Stoeckius et al. 2017. This innovative technique combines single-cell RNA sequencing with the measurement of surface protein markers, allowing for the simultaneous profiling of gene expression and cell surface protein expression within individual cells. CITE-seq data are a combination of two data types extracted simultaneously from the same cell. The first data type is scRNA-seq data, while the second one consists of up to a hundred antibody-derived tags (ADT). We analyze the transcriptomic measurements paired with abundance estimates for 11 surface proteins, whose levels are quantified with DNA-barcoded antibodies. The dataset contains 20,400 genes and 7,858 cells. Cell type annotation was performed by leveraging ADT surface protein marker data to manually gate and classify cells based on immunological knowledge of marker distributions. This manual gating was informed by established marker-to-cell type mappings, as described in Eckenrode et al. 2023. These annotations were used to generate a reference set of labels independent of scRNA-seq-derived predictions, enabling robust benchmarking of scRNA-seq analysis methods.

BE1

BE1 is a single-cell RNAseq benchmark dataset providing a controlled heterogeneity environment using lung cancer cell lines characterised by expressing seven different driver genes (EGFR, ALK, MET, ERBB2, KRAS, BRAF, ROS1) (Arigoni et al. 2024). Specifically the following lung cancer cell lines are included:

- PC9, 4492 sequenced cells (EGFR Del19, activating mutation (Simonetti et al. 2010))

- A549, 6898 sequenced cells (KRAS p.G12S, growth and proliferation, (Yoon et al. 2010))
- NCI-H596 (HTB178), 2965 sequenced cells (MET Del14 , enhanced protection from apoptosis and cellular migration (Cerqua et al. 2022))
- NCI-H1395 (CRL5868), 2673 sequenced cells (BRAF p.G469A, gain of function, resistant to all tested MEK ± BRAF inhibitors, (Negrao et al. 2020))
- DV90, 2998 sequenced cells (ERBB2 p.V842I, increases kinase activity, (Bose et al. 2013))
- HCC78, 2748 sequenced cells (SLC34A2-ROS1 Fusion, ROS1 inhibitors have antiproliferative effect (Davies et al. 2012))
- CCL.185.IG, 6354 sequenced cells EML4-ALK Fusion-A549 Isogenic Cell.

The experiment was done using CellPlex technology from 10x Genomics allowing multiplexing samples into a single channel and therefore removing unwanted batch effects. BE1 is composed of 36,753 genes and 29,606 cells. This dataset has been used for benchmarking the pipelines and the cell line is used as the ground truth for the cell-type annotation.

sc_mixology

sc_mixology uses three human lung adenocarcinoma cell lines HCC827, A549, H1975, H838 and H2228, which were cultured separately and then processed in three different ways. (Tian et al. 2019) Single cells from each cell line were mixed in equal proportions, with libraries generated using three different protocols: CEL-seq2, Drop-seq (with Dolomite equipment) and 10x Chromium. In this work, we use only single cells from the mixture of five cell lines with 10x Chronium protocol (GSM3618022). sc_mixology containing 11,786 genes and 3,918 cells. This dataset has been used for benchmarking the pipelines and the cell line is used as the ground truth for the cell-type annotation.

1.3M cell dataset (1.3M)

The dataset analyzed here was previously introduced in Chapter 1.3.

2.4 Results

2.4.1 Single-cell RNA-seq Workflow

We now shift our focus to evaluating single-cell analysis pipelines in their entirety.

We assess each pipeline using its standard analysis steps as recommended by the developers.

In addition to computational time and memory usage, we aim at comparing each workflow's accuracy. To do so, we selected three datasets that contain different degrees of ground truth, which allows us to evaluate the ability of the workflows to capture real biological characteristics of the analyzed data.

In particular, we tested the pipelines using the BE1 (Arigoni et al. 2024) and sc_mixology (Tian et al. 2019) datasets, composed of mixtures of cell lines: the cell line of origin of each cell is recorded, making these data useful to evaluate the ability of the workflows to recover known cell clusters. In addition, we use a Coord Blood CITEseq dataset (Stoeckius et al. 2017), which, in addition to scRNA-seq, includes cell surface markers that allow the manual gating of cells for cell typing independent of scRNA-seq; treating these cell types as ground truth, we evaluate the ability of the workflows to recover them using only the scRNA-seq profiles. Specifically, we evaluate the concordance of clustering results, measured by the Adjusted Rand Index (ARI) and the cell-type separation in PCA space (see Methods for details).

While useful for the presence of ground truth, none of the previously described datasets are "large", comprising only a few thousand cells. Hence, to study the computational efficiency and scalability of the methods, we employ the 1.3M cell dataset datasets from 10x Genomics (G. X. Zheng et al. 2017). Taken together, these datasets allow us to test the workflows on different organisms, data throughput, and sample types (Table 2.1).

The time in seconds for each step of the analysis is reported in Supplementary Table B.1 and in Figure 2.1 for each dataset. As expected the computational time is proportional to the dimension of the dataset ranging from a few seconds for sc_mixology to more than 2 hours for the 1.3M cell dataset.

As expected, the RAPIDS single-cell pipeline is the fastest across all datasets, highlighting the potential of leveraging GPU acceleration for single-cell analysis, as already reported (Nolet et al. 2022). On the other hand, Seurat and OSCA

	Number of Genes	Number of Cells	Species	Sample	Technology	Ground Truth
1.3M	**27,998**	**1,306,127**	**Mouse**	**Brain**	**10X Genomics Chromium Protocol**	**No**
BE1	**36,753**	**29,606**	**Human**	**Lung Adenocarcinoma**	**10X Genomics Chromium Protocol**	**Lung Cancer Cell Lines**
cb	**20,400**	**7,858**	**Mouse**	**Cord Blood Mononuclear**	**CITE-seq**	**Manual Gating Antibody Expression Reference**
sc_mix	**11,786**	**3,918**	**Human**	**Lung Adenocarcinoma**	**10X Genomics Chromium Protocol**	**Lung Cancer Cell Lines**

Table 2.1: Summary of datasets used to compare the four workflows of single-cell analysis.

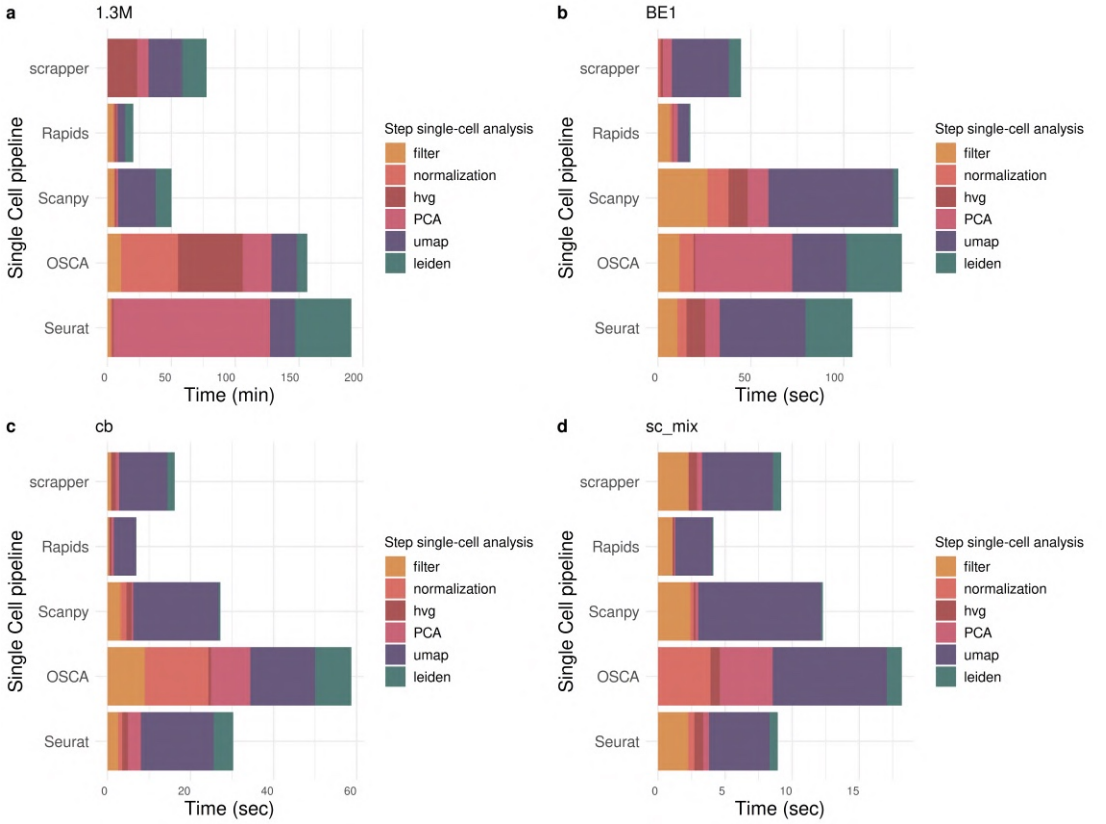

Figure 2.1: **Barplot of elapsed time for each dataset used in the workflow benchmark.** (a) Each panel (ad) displays the computational time required by different single-cell analysis pipelines across various datasets: (a) 1.3M cells, (b) BE1, (c) cb, and (d) sc_mix. Pipelines include Seurat, OSCA, Scanpy, Rapids, and Scrapper. Bars are colored by processing step: filtering, normalization, selection of highly variable genes (HVG), PCA, UMAP, and Leiden clustering. Time is expressed in minutes for the 1.3M cell dataset and in seconds for the others.

exhibit the largest computational times. The color bands in the barplots of Figure 2.1 represent the different steps of the analysis (see Fig. 1.1a). Across all implementations, UMAP is consistently the most computationally demanding step. A notable exception is represented by Seurat in the 1.3M cell dataset, for which PCA is the computational bottleneck (Fig. 2.1a).

We next turned to evaluating the accuracy of the clustering derived from each pipeline: Table 2.2 shows the ARI between the clustering derived from each workflow and the ground truth cell labels. Notably, all workflows perform similarly in the sc_mixology (Fig. 2.2) and cb datasets (Fig. B.1), leading to an average ARI of 0.98 and 0.78, respectively, reflecting the different complexity of the two datasets. We did not observe any major differences between Leiden and

Louvain clustering (Table 2.2).

More interesting is the evaluation of the pipelines on the BE1 dataset, in which OSCA and scrapper achieve an almost perfect concordance with the ground truth, with an ARI of 0.95-0.97, while the other methods stop shy of 0.7 (Table 2.2). The main reason for this difference is the failure to separate two closely related cell lines, A549 and CCL-185-IG, which is derived from A549 cells (Arigoni et al. 2024). From the t-SNE representations it is evident that OSCA and scrapper separate the two cell lines in two distinct clusters, whille Seurat, Scanpy and RAPIDS achieve only partial separation (Fig. 2.2a-e).

ARI	**Algorithm**	**Seurat**	**OSCA**	**Scanpy**	**Rapids**	**scrapper**
cb	louvain	0.88	0.81	0.75	0.78	NA
	leiden	0.7	0.81	0.75	0.77	0.81
sc_mix	louvain	0.97	0.96	0.99	0.99	NA
	leiden	0.98	0.96	0.99	0.99	0.96
BE1	louvain	0.67	0.97	0.68	0.68	NA
	leiden	0.68	0.95	0.68	0.68	0.97

Table 2.2: Adjusted Rand Index (ARI) between the workflow and between single-cell RNA-seq dataset.

To quantify the degree of separation between the two cell types and to ensure that this is not an artifact of the t-SNE representation, we computed the cell type purity in the space of the 50 PCs for each method (Fig. 2.2f) and the cumulative percentage of variance explained by the cell lines for each PC for each method (Fig. 2.3; see Methods). These analyses confirmed that OSCA and scrapper are better able to separate these two closely-related cell lines.

We next explored the reason why this happens, by exploring the individual steps of the workflows, including QC, normalization, and clustering. Interestingly, the main reason for the difference in performance is the selection of the highly variable genes (HVGs). Indeed each workflow uses different ways to account for the mean-variance relation of the data in selecting HVGs (see Methods), leading to a unique set of genes for each method (Fig. 2.2g). OSCA and scrapper approach to model the gene-variance trend seems to achieve greater performance than the methods employed by the other workflows.

Taken together, these results show how the algorithmic choices of the analysis workflows can impact the downstream results and how the selection of HVGs

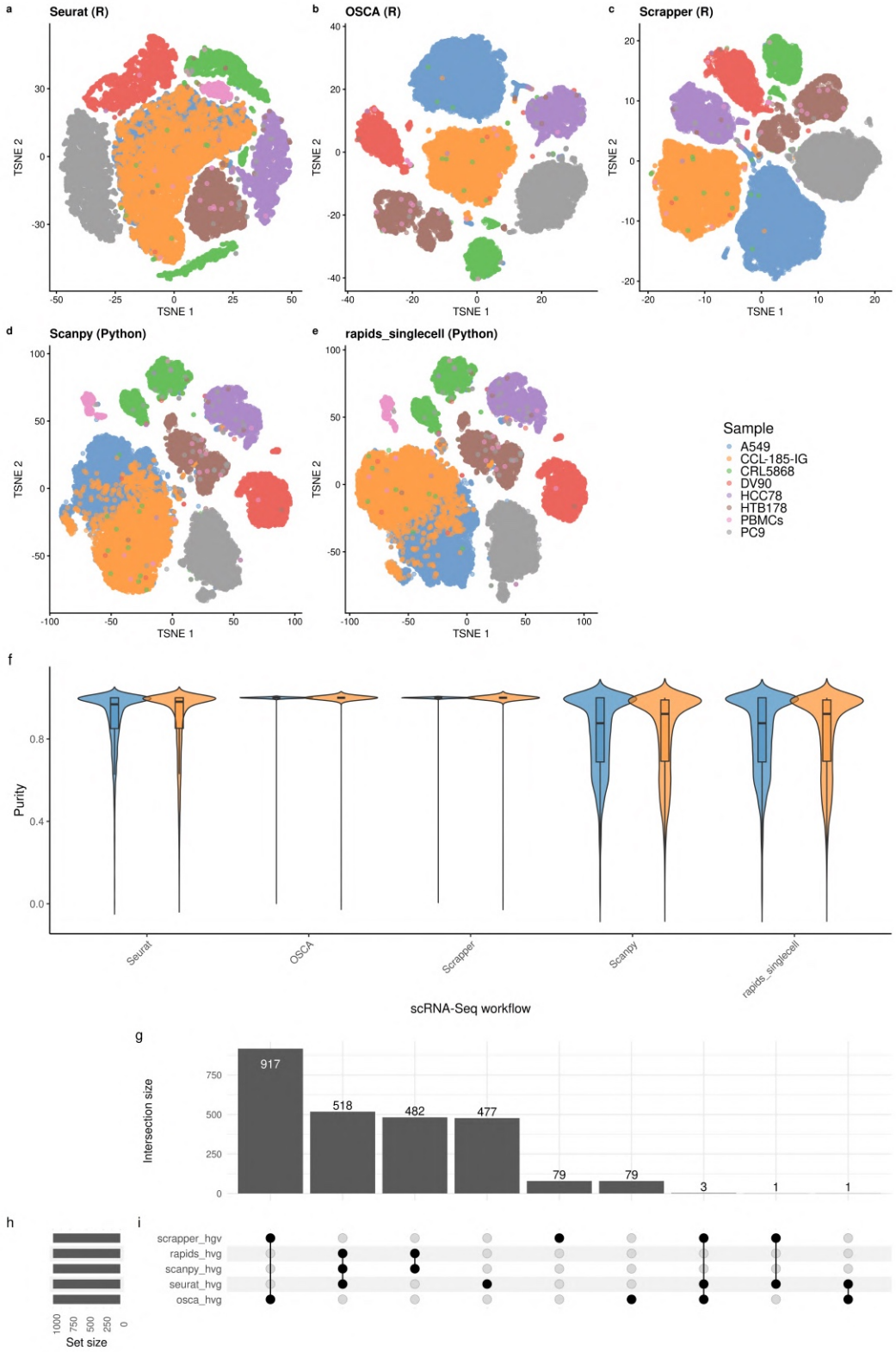

Figure 2.2: **T-SNE plot and HVGs in the BE1 dataset.** (ae) t-SNE embeddings of the BE1 dataset colored by sample identity, generated using five different single-cell workflows: Seurat (a), OSCA (b), Scrapper (c), Scanpy (d), and rapids_singlecell (e). Each workflow applies its own normalization and highly variable gene (HVG) selection procedure prior to dimensionality reduction. (f h) UpSet plot showing the intersection of HVG sets selected by each workflow. Panel (f) indicates the size of each intersection set, (g) shows the number of genes selected per method (set size), and (h) depicts the overlap structure across methods.

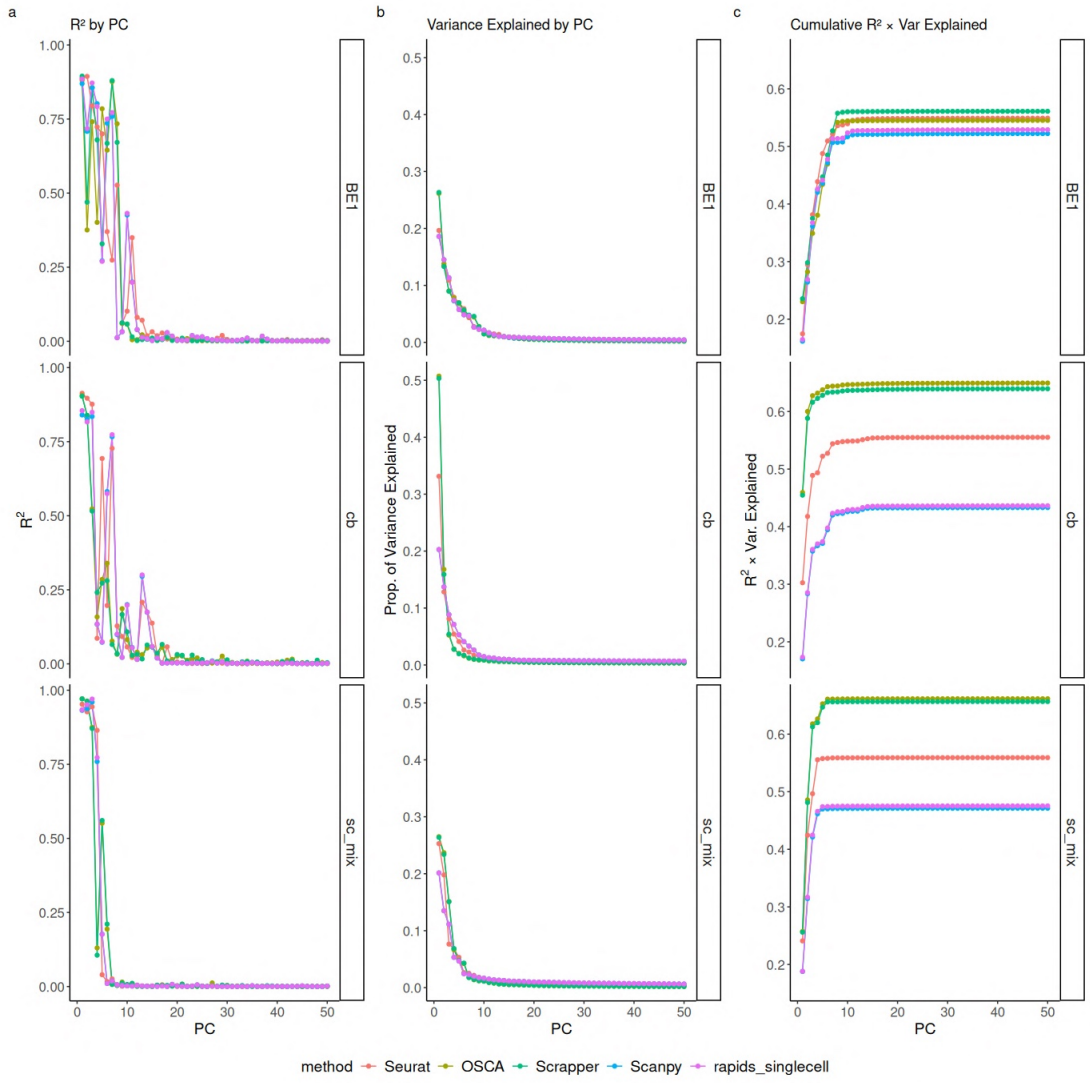

Figure 2.3: **Comparison of dimensionality reduction performance across single-cell workflows.** Line plots show three performance metrics computed across principal components (PCs) for three datasets: BE1, cb, and sc_mix (rows), and five workflows: Seurat, OSCA, Scrapper, Scanpy, and rapids_singlecell (colors). (a) R^2 by PC: the proportion of variance in the cell-type labels explained by each PC. (b) Variance Explained by PC: the proportion of total variance in the gene expression data captured by each PC. (c) Cumulative R^2 x Variance Explained: a composite metric reflecting both biological signal and data structure captured by the PCs.

is a critical step in the pipeline.

2.5 Discussion and conclusion

Finally, we have chosen to benchmark the most basic workflow available from each of the software frameworks, mimicking the "basic tutorials" or "getting started" guides. Obviously, modern single-cell RNA-seq studies required more complex analyses, involving multi-sample comparisons, batch-effect removal (Haghverdi et al. 2018; Korsunsky et al. 2019), reference-based annotation (Dvir Aran, Looney, et al. 2019), and multi-modal integration (Y. Hao, S. Hao, et al. 2021), to name a few. It is likely that the impact of optimized code and GPU acceleration is even more important for these more complex algorithms.

Another important finding concerns the role of parameter choices and default settings across workflows. Differences in the number of highly variable genes selected, the algorithm used for dimensionality reduction, or the resolution parameters for clustering can directly influence biological interpretations such as the number of clusters identified or the ability to detect fine-grained subpopulations. While OSCA and `scrapper` benefit from optimized low-level implementations that improve scalability without altering downstream results, Seurat and Scanpy, despite their popularity and flexibility, show a decrease in efficiency when scaling beyond several hundred thousand cells. The evaluation on datasets with a ground truth, such as BE1, cb and `sc_mixology`, confirms that most workflows can recover biologically meaningful clusters. However, metrics such as ARI and cluster purity reveal non-negligible differences between methods, suggesting that even subtle variations in preprocessing or dimensionality reduction can affect performance. On real-world large datasets, such as the 1.3M brain cell dataset, the absence of a ground truth complicates the evaluation, and benchmarking relies primarily on scalability and stability rather than accuracy.

From a practical perspective, the results suggest that for small- to medium-scale datasets (up to ~100k cells), widely used frameworks such as Seurat and scrapper in R and Scanpy in Python remain reliable and convenient choices due to their extensive documentation, community adoption, and broad functionality. For larger datasets, optimized frameworks such as `scrapper` or GPU-enabled workflows such as `rapids_singlecell` provide clear advantages, and will likely play an increasingly central role as single-cell studies continue to grow in size and complexity. Ultimately, researchers should select workflows not only based

on dataset size, but also on available computational resources and the specific biological questions at hand.

Chapter 3

Digital Pathology

3.1 Introduction

In cancer research, in addition to omics data previously discussed in earlier chapters, an increasing variety of medical imaging modalities can be leveraged for translational research and clinical decision-making. These include computed tomography (CT), magnetic resonance imaging (MRI), positron emission tomography (PET), and histopathological microscopy images, such as hematoxylin and eosin (H&E)-stained whole-slide images (WSIs). This multimodal information contributes to the broader aim of precision medicine, where insights from multiple biological layers guide personalized therapy decisions.

Within the scope of my doctoral project, we have chosen to focus specifically on H&E-stained whole slide images (WSIs). This decision stems from several key considerations: (i) the widespread clinical availability of H&E images as a diagnostic standard; (ii) the relatively lower cost and higher accessibility of image data compared to omics data; and (iii) the increasing number of applications of deep learning techniques, including convolutional neural networks (CNNs) and foundation models such as UNI (R. J. Chen et al. 2024) and Prov-GigaPath (Xu et al. 2024), in extracting meaningful information from pathology images (Bulten et al. 2025; Tizhoosh and Pantanowitz 2021).

Digital pathology offers the possibility to computationally analyze tissue architecture and cellular morphology, providing clinicians with decision support tools that enhance reproducibility and objectivity in diagnosis (Tomaszewski and Bejnordi 2021). Recent studies have demonstrated how histopathological images can be used to predict genomic alterations, transcriptional states, or

even patient outcomes using machine learning (Madabhushi and G. Lee 2020; Schmauch et al. 2020; Bergstrom et al. 2024). Examples include HE2RNA, which predicts RNA-Seq profiles from images (Schmauch et al. 2020), and models that infer spatial transcriptomics from histology (Pizurica et al. 2024; Stefanovska 2025).

Despite these advancements, many of the image analysis tools and pipelines are implemented in Python and remain disconnected from the R/Bioconductor ecosystem, which is the predominant framework for multi-omics data integration. This fragmentation presents a challenge for researchers seeking to perform integrated analyses. Thus, one of the key aims of this project is to develop standardized workflows and software tools that allow researchers to extract both human-readable features (using HoVer-Net, Squidpy, and QuPath) and learned embeddings (via Prov-GigaPath) directly within the R environment.

Portions of the material presented in this chapter were also authored by us for the online book *Orchestrating Spatial Transcriptomics Analysis with Bioconductor (OSTA)*. This open-access resource provides reproducible examples and detailed discussion of computational workflows for spatial omics data using the R/Bioconductor ecosystem. The image analysis chapter, written by us, describes platform-independent approaches and demonstrates how R-based methods can be integrated with Python toolsclosely mirroring the workflows developed and applied in this thesis.

3.2 Image Analysis

3.2.1 Image Analysis in R/Bioconductor and Python

Several tools are available for histopathological image analysis, particularly in the Python ecosystem. Notable Python libraries include OpenSlide (Goode et al. 2013) for reading whole-slide images, scikit-image (Van der Walt et al. 2014) for general-purpose image processing, and Squidpy (Palla et al. 2022), which extends the AnnData framework to spatial omics and tissue imaging. Deep learning frameworks such as PyTorch and TensorFlow support advanced modeling, while domain-specific tools like HoVer-Net (Graham et al. 2019) and Prov-GigaPath (Xu et al. 2024) enable state-of-the-art segmentation and feature extraction from H&E-stained images. In contrast, the Bioconductor ecosystem in R, while well-developed for high-throughput omics data, offers only a limited set

of tools for histopathological image analysis. Packages such as EBImage (Pau et al. 2010) support basic image processing tasks, and SpatialExperiment provides a standardized data infrastructure to represent spatially resolved transcriptomic data, which can be extended to image-derived features. However, beyond these foundational tools, there is a lack of dedicated Bioconductor packages for working directly with pathology images or for integrating image-derived features into multi-omic workflows. This gap highlights a critical limitation for researchers working within the R/Bioconductor environment who seek to incorporate digital pathology into integrative analyses. As a result, users are often forced to rely on external Python-based tools and complex inter-language data transfers, which can hinder reproducibility and accessibility. The aim of this work is therefore to fill this methodological and tool gap by developing and disseminating R-compatible packages and workflows that enable the processing, analysis, and integration of image-derived features within Bioconductor.

3.3 Image Analysis Workflow

3.3.1 Whole Slide Image

Digital pathology workflows rely on high-resolution whole-slide images (WSIs) generated by proprietary scanners from different vendors. These WSIs are saved in specific file formats, each corresponding to a particular scanner type. Understanding these formats is essential for designing interoperable and reproducible computational pipelines.

The scanner brands and their respective file formats commonly encountered in digital pathology include:

- Aperio: .svs, .tif
- DICOM-compatible scanners: .dcm
- Hamamatsu: .vms, .vmu, .ndpi
- Leica: .scn
- MIRAX: .mrxs
- Philips: .tiff
- Sakura: .svslide
- Trestle: .tif

Figure 3.1: The pyramidal structure of WSI, resulting from different levels of magnification.

- Ventana: .bif, .tif
- Zeiss: .czi
- Generic tiled TIFF: .tif

Each scanner uses a unique tiling scheme and metadata structure to support rapid visualization and efficient storage. For instance, Aperio's .svs format uses a pyramidal tiling strategy with multiple image resolutions stored within a single file as shown in Fig.3.1. Some formats, such as DICOM, are standardized and widely used in clinical radiology, while others remain proprietary.

Open-source libraries like OpenSlide provide support for reading many of these formats, abstracting the technical complexities and enabling downstream image analysis. This compatibility ensures that digital pathology data from diverse clinical sources can be processed uniformly, facilitating the development of scalable computational frameworks for precision oncology.

These high-resolution WSIs, while rich in diagnostic content, come with substantial storage requirements. A single WSI can range from several hundred megabytes to multiple gigabytes, depending on the scan resolution, number of channels, and compression method used. For example, uncompressed 40X scans of large tissue sections can exceed $9 - 10$ GB per slide. Managing, storing,

and processing such large datasets necessitates high-performance computing resources and efficient file formats. Furthermore, the image quality can vary depending on scanner type, staining quality, and the presence of artifacts such as tissue folds or background debris. These factors not only affect the visual interpretability for pathologists but also introduce variability in algorithmic feature extraction and model performance. Thus, both hardware capacity and image standardization are critical considerations when designing digital pathology pipelines.

3.3.2 Image Preprocessing

Image preprocessing is a critical step in computational histopathology, as it addresses both technical and biological variability, thereby enhancing the accuracy and reproducibility of downstream analyses. A fundamental component of preprocessing involves stain separation, such as color deconvolution in hematoxylin and eosin (H&E) stained slides, which isolates the hematoxylin channel to highlight nuclear structures and facilitate subsequent segmentation tasks (Yi et al. 2017; Omar et al. 2024).

Morphological operations, specifically opening and closing by reconstruction, are subsequently applied to refine the segmentation masks. Opening by reconstruction removes small spurious objects and noise while preserving the overall shape of nuclei, whereas closing by reconstruction fills small gaps or holes inside nuclear regions, ensuring more compact and homogeneous structures. These post-processing steps are essential to reduce artifacts introduced by staining variability or image acquisition, and to adapt the segmentation to the wide range of nuclear sizes and image resolutions typically encountered in whole-slide imaging (Yi et al. 2017).

Automatic thresholding methods, such as Otsus algorithm, are frequently employed to objectively distinguish between nuclei and background, even in the presence of intensity variations, making them especially effective for bimodal histograms typical of H&E images (Omar et al. 2024; Yi et al. 2017). Furthermore, to address challenges posed by overlapping or touching nuclei, advanced segmentation techniques like the marker-controlled watershed transform are utilized. This approach leverages both internal and external markers to accurately delineate individual nuclear boundaries.

Collectively, these preprocessing steps are essential for ensuring that sub-

sequent quantitative analyses, classification, and predictive modeling in digital pathology are robust, reproducible, and standardized, as underscored by recent comprehensive reviews in the field (Omar et al. 2024) (Schömig-Markiefka et al. 2021).

Several specialized tools are available for preprocessing digital pathology images, each offering unique features tailored to the needs of histopathological analysis and machine learning workflows:

- QuPath: An open-source software specifically designed for digital pathology, QuPath provides comprehensive tools for preprocessing whole-slide images. It supports stain vector estimation (color deconvolution), tissue detection, artifact removal, and batch processing. Its object-based data model and scripting capabilities allow for highly customizable preprocessing pipelines suitable for large datasets and integration with downstream analysis (Bankhead et al. 2017).

- HistoClean: This open-source, user-friendly GUI tool focuses on image preprocessing and augmentation for histological images. Augmentation refers to artificially increasing the size and variability of the dataset through transformations such as rotations, flips, color adjustments, and other perturbations, which improve model robustness and reduce overfitting. HistoClean offers modules for image patching, whitespace thresholding, dataset balancing, normalization, and augmentation. It is designed to help both biomedical scientists and computer scientists prepare robust datasets for deep learning, without requiring advanced programming skills (McCombe et al. 2021).

- PathML: PathML is a Python-based framework that enables the creation of modular preprocessing pipelines for digital pathology. It supports a wide range of file formats and provides domain-specific transformations such as H&E stain deconvolution, tissue detection, and artifact detection, as well as general-purpose operations like blurring and thresholding. PathML is optimized for large-scale, distributed processing of gigapixel whole-slide images, making it suitable for high-throughput research settings (Berman et al. 2021).

- ImageJ: Widely used in the biomedical sciences, ImageJ is an open-source platform that supports a variety of plugins for preprocessing tasks such as color deconvolution, filtering, morphological operations, and batch processing. Its extensibility and compatibility with digital pathology formats make it a versatile choice for custom preprocessing workflows (Abràmoff, Magalhães, and Ram 2004).

- Ilastik: This tool offers an intuitive interface for non-experts to perform segmentation, classification, and preprocessing of pathology images. Ilastik is particularly useful for tasks such as pixel classification and object detection, and can be integrated into broader digital pathology pipelines (Berg et al. 2019).

These tools enable researchers and clinicians to standardize and automate the preprocessing of digital pathology images, ensuring data quality and reproducibility for downstream computational analyses and machine learning applications.

3.3.3 Segmentation and Classification

To extract biologically meaningful information from whole-slide images (WSIs), one of the fundamental tasks is the identification and delineation of individual cell nuclei. This process, referred to as nuclei segmentation, is crucial for downstream analyses such as cell counting, morphology assessment, spatial organization analysis, and tumor microenvironment profiling. Accurate segmentation allows researchers to distinguish between nuclear and non-nuclear regions and to quantify features at the single-cell level.

Deep learning (DL) has revolutionized quantitative image analysis (QIA) in digital pathology. These advances not only improve the accuracy of tissue-level quantification but also ensure that the extracted biomarkers are biologically meaningful and clinically relevant, paving the way for more effective and personalized cancer treatment strategies. (Omar et al. 2024)

For instance, spatial and morphological features from cell nuclei can be extracted from H&E WSIs and used to predict estrogen receptor status in breast cancer (Rawat et al. 2018). One of the most prominent DL models for this task is HoVer-Net (Graham et al. 2019), which performs simultaneous nuclear segmentation and classification using pixel-level statistics. HoVer-Net can be trained on different datasets to classify nuclei into subtypes depending on tissue context. Variants of the model include:

- CoNSep, trained on colorectal samples to classify fibroblasts, normal and tumor epithelial cells, inflammatory, necrotic, and muscle cells;
- PanNuke (Gamper et al. 2019), encompassing 19 tissue types, enabling classification into benign epithelial, tumor, inflammatory, necrotic, and stromal cells;
- MoNuSeg (Kumar et al. 2017), trained on slides from seven organs to segment and classify a broader variety of nuclear instances.

These examples highlight the utility of DL algorithms for deciphering the complex cellular composition of tumors using routine histopathology images. This composition can then be leveraged to extract human-interpretable features

like cellular ratios and densities, which can subsequently be used in diagnostic, prognostic, and integrative multi-omic applications. The next, more molecular step will be to leverage immunohistochemical and transcriptomic annotation of finer-grained subtypes within these broad cellular categories.

Our Approach: Using HoVer-Net

For our analyses, we selected the HoVer-Net deep learning model for nuclei segmentation and classification due to its well-documented performance and reproducibility across diverse tissue types. Unlike models trained on narrow or tissue-specific datasets, HoVer-Net was pretrained on a broad pan-cancer dataset, making it particularly well-suited for use with TCGA images, which span multiple tumor types.

HoVer-Net provides simultaneous segmentation and classification of nuclei at the pixel level. The model outputs include nuclear boundaries, centroid coordinates, and assigned cell-type labels, enabling both spatial and phenotypic analyses at the single-cell level.

The output of HoVer-Net is stored in a structured JSON format that captures detailed information about each segmented nucleus. The structure includes the following fields:

- mag: Magnification level (e.g., 40X)
- nuc: A dictionary of detected nuclei, each indexed by a unique key. Each entry contains:
  - bbox: Bounding box coordinates ([[x1, y1], [x2, y2]])
  - centroid: Coordinates of the nucleus center
  - contour: Polygon representing the nucleus outline
  - type_prob: Probability score for predicted class
  - type: Class label (e.g., benign epithelial, neoplastic, tumor, stromal, inflammatory, necrotic)

This output can be seamlessly integrated into downstream analysis workflows by converting it into an `AnnData` object, a widely adopted format for structured biological data that supports multi-modal annotations. The resulting `AnnData` object can be used both in Python (via `scanpy` or `anndata`) and R (via `zellkonverter` or `SpatialExperiment`), facilitating interoperability with existing pipelines for spatial statistics, visualization, or machine learning. The nuclear-level annotations (coordinates, contours, classifications) populate the

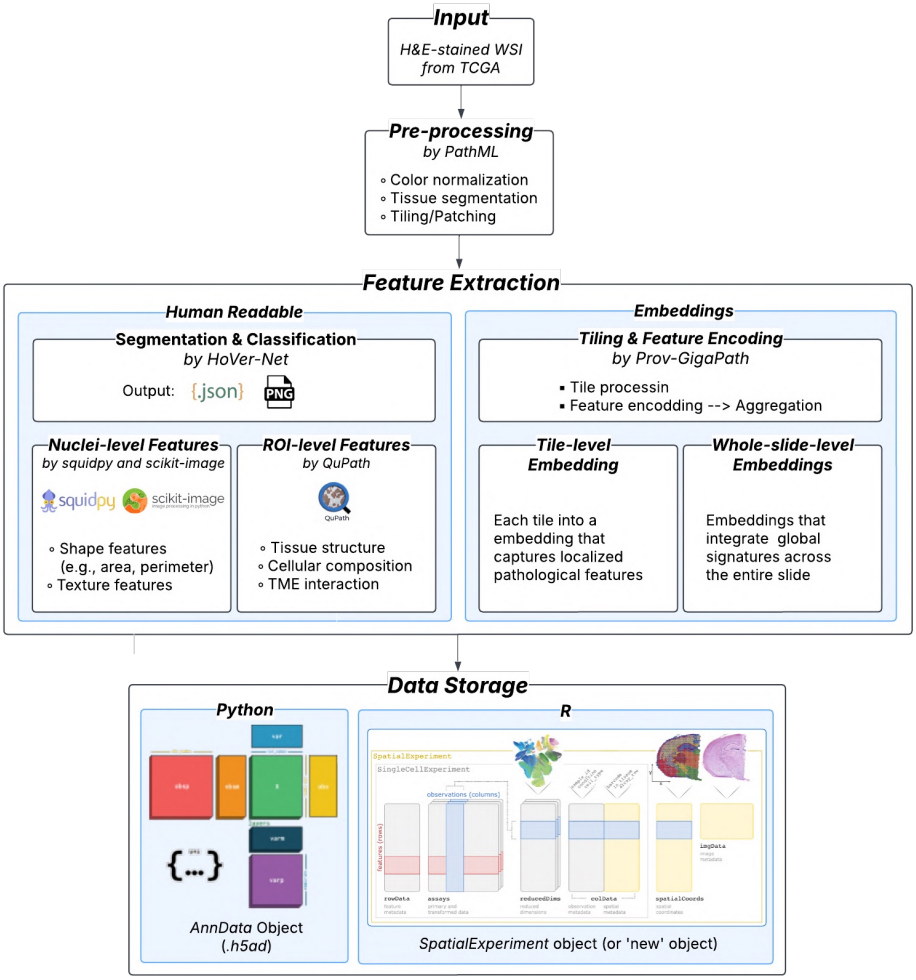

Figure 3.2: Feature Extraction Workflow

`obs` and `obsm` fields, enabling scalable and reproducible single-cell level analyses from histological images.

Using the classification labels output by HoVer-Net, we computed tumor purity for each slide. Specifically, we calculated the ratio of neoplastic nuclei to the total number of nuclei identified in each WSI. This image-derived purity estimate provides a spatially resolved, interpretable measure of tumor cell abundance that can be benchmarked against genomic purity metrics and used for clinical stratification.

$$\text{Purity}_{\text{HoVer-Net}} = \frac{\text{Number of neoplastic nuclei}}{\text{Total number of nuclei in the WSI}}$$

3.3.4 Feature Extraction

A key component of computational pathology is the ability to extract features from histological images that capture both biologically interpretable and latent information. This dual-level characterization supports both mechanistic insights and powerful predictive modeling.

Human-Readable Feature Extraction

To capture human-interpretable features, we implemented Python-based pipelines using libraries such as Squidpy and Scikit-image. These tools allow for the extraction of classical image features:

- Morphological features (e.g., area, perimeter, eccentricity, solidity) that describe nuclear and cellular shape.
- Intensity features (mean, variance, min, max) computed on grayscale representations of the images.
- Spatial features, such as nearest neighbor distances between nuclei (e.g., using KD-Tree indexing), to characterize local cellular environments.

At a higher level, region-of-interest (ROI) based metrics such as neighborhood enrichment and graph-based topology analysis were employed to capture tissue architecture and spatial heterogeneity.

In addition, we used QuPath, a versatile open-source tool, to extract:

- Morphological features: Area, perimeter, circularity, solidity, eccentricity, max/min diameterfor nuclei, cytoplasm, and whole cells.
- Staining intensity statistics in optical density (OD) space across ROIs: max, mean, min, standard deviation, and range.
- Texture features: 13 Haralick descriptors including contrast, correlation, entropy (computed on hematoxylin channel at 2.00 ţm/pixel).
- Derived ratios: nucleus-to-cell area ratio, and counts of nearby detections.

These biologically interpretable features are essential for downstream tasks such as clustering, classification, and statistical correlation with genomic and clinical endpoints.

Latent Feature representation using embeddings

To complement human-interpretable descriptors, we leveraged recent advances in foundation models for computational pathology. These models, typically trained on millions of histological image tiles using self-supervised or contrastive learning strategies, generate high-dimensional embeddings that

capture complex and subtle morphological patterns beyond human visual assessment. Among these, we employed Prov-GigaPath, a large-scale foundation model specifically designed for histopathology, which provides robust and generalizable image representations.

The Prov-GigaPath workflow includes two key stages:

1. Tile processing: WSIs are divided into smaller tiles (typically 224x224 px) at 20X magnification, excluding background regions.
2. Feature encoding: Each tile is passed through the pretrained encoder to produce a 768-dimensional embedding vector (layer 13).

Embeddings can then be aggregated at the tile-level or slide-level using pooling strategies. While these representations are not directly interpretable, they have been shown to perform well in unsupervised learning, patient stratification, and predictive modeling. The pretrained model weights are available on HuggingFace, an open-source platform and model hub widely used in the machine learning community to share, distribute, and benchmark pretrained models, ensuring accessibility and reproducibility.

Combining both feature types enables a comprehensive understanding of tissue morphology: (i) interpretable features anchor findings in biological relevance and (ii) latent embeddings capture high-dimensional structure useful for machine learning.

This integrative approach enhances the utility of histopathological images in multi-omic frameworks for cancer research.

3.4 Data

3.4.1 TCGA Data

The Cancer Genome Atlas (TCGA) includes a collection of 11,765 diagnostic whole-slide images from 9,640 patients across 32 cancer types. (Tomczak, Czerwiska, and Wiznerowicz 2015) These histopathological images represent only one component of TCGA's broader multi-omics repository. Alongside WSIs, TCGA provides a rich array of molecular and clinical data, including gene expression (RNA-Seq), somatic mutation profiles (whole-exome sequencing), DNA methylation, copy number alterations, protein expression (RPPA), and comprehensive clinical annotations. This multidimensional dataset facilitates

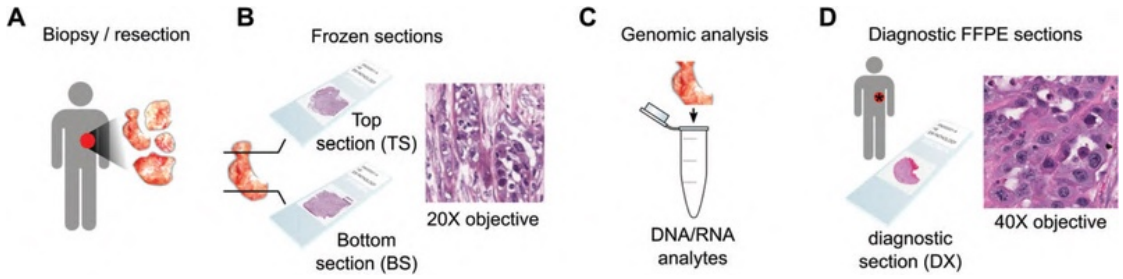

Figure 3.3: Tissue procurement in TCGA. (A) A Tissue Source Site (TSS) obtains samples from surgical resection. (B) A portion of this tissue is selected for submission to TCGA, and the BCR produces top-section (TS) and bottom-section (BS) slides for review to determine that the percentage necrosis and abundance and proportion of tumour cells are adequate for genomic analysis. (C) The middle portion of this tissue is used to extract RNA and DNA analytes for genomic analysis. (D) One or more diagnostic formalin-fixed paraffin-embedded (FFPE) slides are submitted to the BCR by the TSS for confirmation of histological diagnosis. (Cooper et al. 2018)

integrative analyses that connect tissue morphology with molecular alterations and clinical outcomes.

TCGA includes two main types of histological slides: flash frozen and formalin-fixed paraffin-embedded (FFPE). Flash frozen slides are typically produced intraoperatively in a cryolab to help surgeons assess tumor margin status. While this method ensures close proximity to the tissue used for genomic extraction, it often introduces morphological artifacts such as tissue cracking and holes due to freezing, resulting in a "Swiss cheese" appearance that limits their utility for computational analysis. (Cooper et al. 2018)

Conversely, FFPE slides, considered the gold standard in diagnostic histopathology, are created by chemically fixing tissue in formalin and embedding it in paraffin wax before slicing. These slides preserve fine tissue architecture and provide visually high-quality samples, making them more suitable for algorithmic analysis. However, because of spatial heterogeneity in tumors, FFPE samples may not precisely correspond to the regions used for genomic profiling.

As shown in Fig3.3 tissue submitted to TCGA undergoes a structured workflow at the Biospecimen Core Resource (BCR). Two slides-designated top-section (TS) and bottom-section (BS)are reviewed to evaluate tumor content and necrosis percentage. The central portion of the sample is reserved for RNA and DNA extraction. Additionally, one or more diagnostic FFPE slides are submitted to

confirm histopathological diagnosis. These diagnostic slides originate from the same tumor, but the spatial and molecular correspondence to the genomics-extracted tissue is often uncertain. Thus, researchers must consider a trade-off between image quality and genomic adjacency when designing image-based studies using TCGA data.

3.4.2 TCIA Data

The Cancer Imaging Archive (TCIA) is a large-scale open-access repository that provides a comprehensive collection of medical images of cancer, including radiological scans (e.g., CT, MRI, PET) and histopathological images. TCIA is a critical resource for cancer imaging research as it includes richly annotated datasets with accompanying clinical, genomic, and pathological metadata. It supports a wide range of applications, including image-based biomarker discovery, radiogenomics, and multi-modal integration studies. Researchers can access TCIA datasets through its user interface or programmatically via APIs, which facilitate the retrieval and processing of large volumes of image data in a reproducible and automated manner.

3.5 R Packages

The integration of image-derived features into downstream statistical and multi-omic analyses requires software tools that are both reproducible and compatible with existing Bioconductor infrastructures. While the Python ecosystem offers a broad range of libraries for digital pathologyincluding frameworks for whole-slide image reading, nuclei segmentation, and feature extractionthe lack of analogous tools in R creates a barrier for researchers who wish to perform end-to-end analyses within a single computational environment. This limitation is particularly relevant in cancer genomics, where R/Bioconductor remains the predominant ecosystem for RNA-Seq processing, differential expression analysis, survival modeling, and integrative multi-omic workflows.

To address this gap, as part of this doctoral work we developed a collection of R packages designed to streamline (i) programmatic access to large-scale pathology datasets, (ii) standardized extraction of both human-interpretable and latent image features, and (iii) interactive visualization and exploration of histopathological data. These packages `imageTCGA`, `TCIAAPI`, and `HistoImageR`

provide interoperable interfaces that allow users to seamlessly incorporate digital pathology into R-based pipelines without relying on external tools or ad hoc data transformations.

Together, these resources enable researchers to move from raw whole-slide images to analysis-ready features within the R/Bioconductor framework. Table 3.1 summarizes the main functionality, dependencies, and intended use cases of each package, illustrating how they collectively support scalable and reproducible digital pathology workflows.

Package	Description	When to use it?	Link
imageTCGA	Collection of histopathological features from TCGA H&E images with Shiny app for visualization	When working with TCGA image-derived features	`https://github.com/billila/imageTCGA`
TCIAAPI	Interface to The Cancer Imaging Archive API for downloading clinical and imaging data	When accessing TCIA image data programmatically	`https://github.com/billila/TCIAAPI`
HistoImageR	Tools for image processing and feature extraction from histopathological slides	When extracting quantitative image features	`https://github.com/shbrief/HistoImageR`

Table 3.1: Summary of R/Bioconductor Packages for Histopathological Image Analysis

3.5.1 imageTCGA

The `imageTCGA` package (Billato 2025), available in Bioconductor from April 2025, provides a standardized framework to access, visualize, and analyze image-derived features from histopathological slides within the TCGA repository.

Figure 3.4: imageTCGA logo

It includes both a comprehensive R package and an integrated Shiny web application that allow users to interactively explore clinical and morphological features from over 11, 975 diagnostic slides.

One of the key motivations for the development of `imageTCGA` is the high computational and time cost associated with downloading, processing, and analyzing TCGA histopathology images. The following table outlines the estimated processing time for each major step:

Table 3.2: Estimated Time Required for Image Processing Pipeline

Task	**Time Required**
Download TCGA slides	1 month
HoVer-Net segmentation	6 months
Prov-GigaPath embeddings	6 months
QuPath extraction	6 months
Squidpy feature extraction	2 months
JSON to GeoJSON conversion	3 months
Total Time	**22 months (1.8 years)**

By making pre-extracted features accessible in Bioconductor data structures, `imageTCGA` dramatically reduces the entry barrier for researchers aiming to integrate imaging with molecular data.

This package is developed under the auspices of the Multi-Omic Integration of Histopathology Image Analysis working group and serves as a foundation for reproducible, scalable, and interpretable research in digital pathology.

After installing the package, the Shiny application can be launched with:

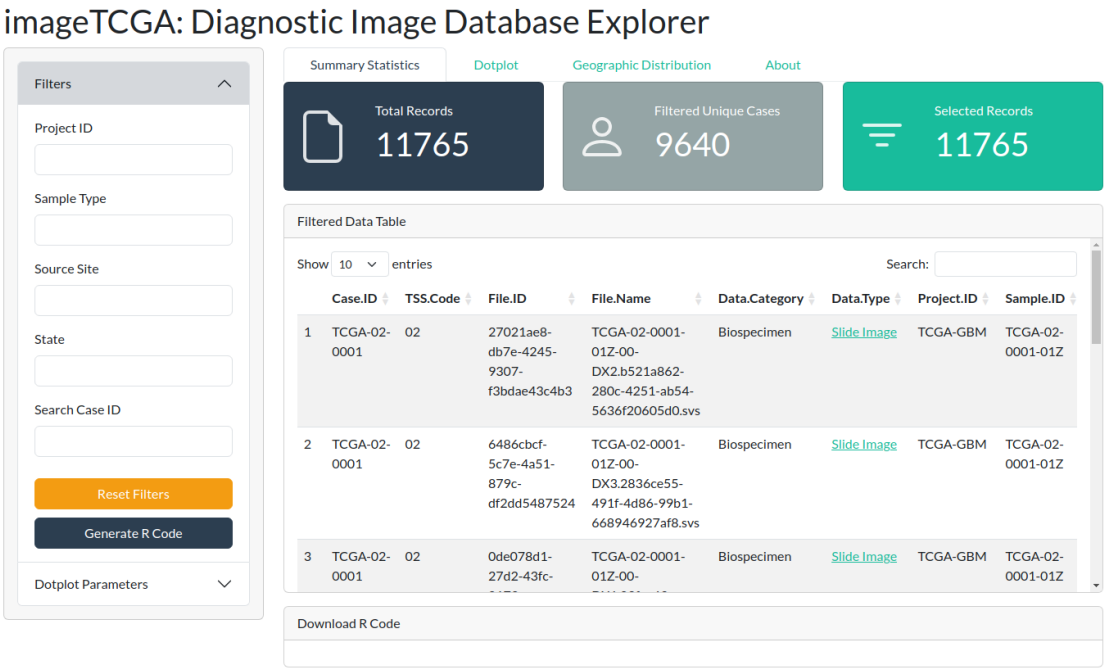

Figure 3.5: Graphical interface of the imageTCGA shiny app

```
library(imageTCGA)
imageTCGA::imageTCGA()
```

This opens a web-based graphical interface that enables exploration of 11,765 diagnostic slides from 9,640 TCGA cases, with filtering options based on clinical and pathological metadata (Fig3.5).

Overview of functionalities

User interface and filtering:

- Filters by Project ID, Sample Type, Source Site, and State.
- Case ID search and reset options.

Visual summaries and metrics:

- Value boxes show total cases, filtered records, and selected subset.
- Interactive data table of filtered samples.
- Downloadable R code for selected data subset.

Feature exploration panels:

- Dedicated panels for HoVer-Net and Prov-GigaPath outputs.
- View segmentation and embedding-based features interactively. (non so vediamo)

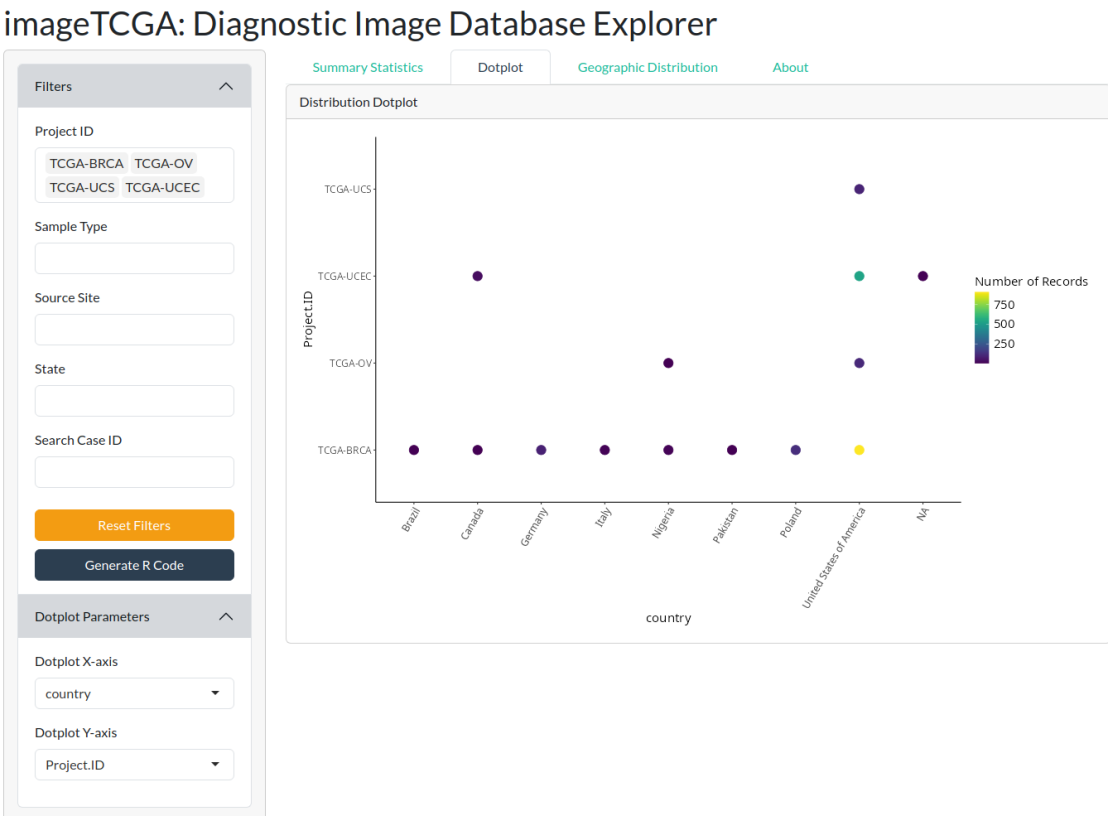

Figure 3.6: DotPlot of the imageTCGA shiny app. The dot plot visualization allows users to explore gynecological tumors (BRCA, OV, UCS, UCEC). On the left panel, you can select which variables to plot on the x-axis and y-axis.

DotPlot visualization:

- Customizable x/y axis selection.
- Visualize categorical data distributions.

Geographic Distribution:

- Interactive Leaflet map of source sites.
- Sample count scaling by marker size.
- Distribution stats and bar plots by state.

Additionally, `imageTCGA` allows users to select images of interest and generate R code to download these images directly to their computer for further analysis. This feature enhances the apps usability, making it easier for researchers to work with the specific slides relevant to their research.

`imageTCGA` provides a user-friendly and computationally efficient solution to interact with large-scale digital pathology datasets like the TCGA, facilitating research in cancer biology, spatial analysis, and machine learning.

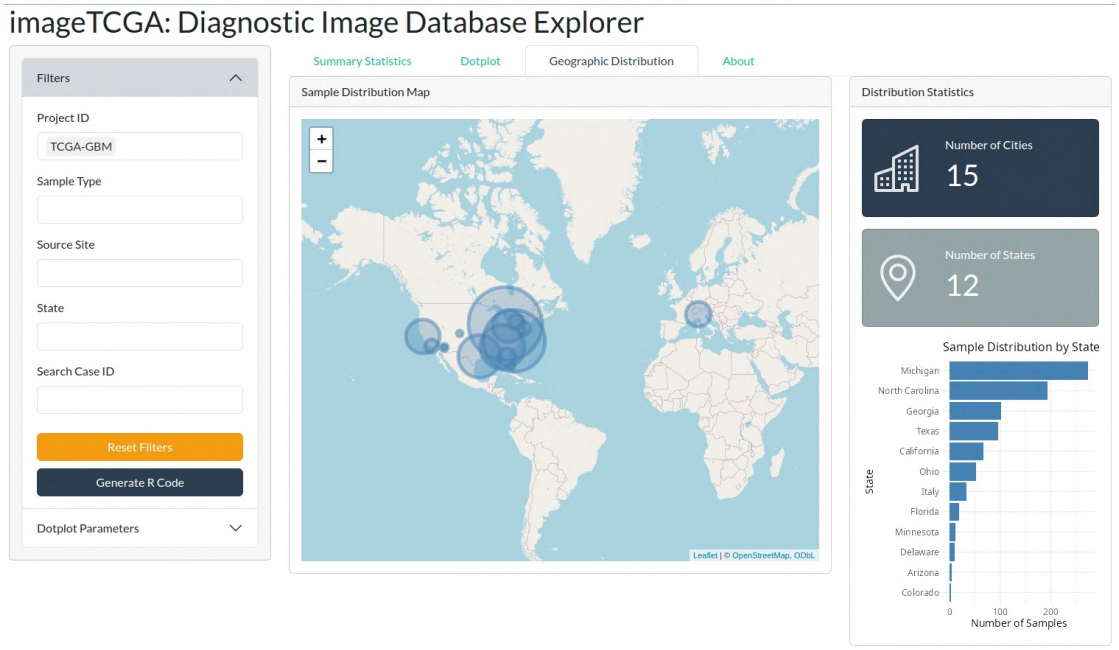

Figure 3.7: Geographic Distribution of the imageTCGA shiny app

3.5.2 TCIAAPI

The TCIAAPI package enables programmatic access to The Cancer Imaging Archive (TCIA) (Freymann et al. 2012). It provides functionality for retrieving image metadata and downloading WSIs, supporting reproducible workflows in digital pathology. This package complements imageTCGA by facilitating access to raw image data that can be processed and analyzed with standardized pipelines.

The `TCIAAPI` package (**billila/TCIAAPI**) provides an R interface to the Cancer Imaging Archive (TCIA) application programming interface (API), enabling programmatic access to histopathological images and associated metadata. Through this package, users can obtain authentication tokens, query metadata on whole-slide images, and download selected image files directly from the TCIA repository.

Since the `TCIAAPI` requires authentication, the package implements the function `tcia_access_token`, which retrieves an access token from the server. By default, the function is configured to obtain a public token, although the token is valid only for a limited period of time and must be periodically refreshed. To ensure security, tokens can be masked in the output using the `httr2::obfuscate` function.

Table 3.3: Main functions of the `TCIAAPI` package and their purpose.

Function	**Input**	**Output / Purpose**
`tcia_access_token`	None (optional arguments for authentication)	Retrieves an access token required to interact with the TCIA API. By default, obtains a public token; must be refreshed periodically.
`tcia_svs_info`	`camic_id` (from TCIA Histopathology Custom Dataset Builder JSON)	Returns metadata about SVS whole-slide images, including image identifiers and download URLs.
`tcia_svs_download`	`camic_id`; optional `destdir` argument	Downloads SVS whole-slide images from the TCIA API. Files are saved to a temporary directory by default or to a user-defined path.

Once authentication is established, the function `tcia_svs_info` allows retrieval of metadata associated with SVS whole-slide images. The function requires a unique `camic_id`, which can be obtained from the TCIA Histopathology Custom Dataset Builder JSON file available on the TCIA portal. The returned metadata include information about the available images as well as the download URL, which can subsequently be used to fetch the corresponding files.

For direct image acquisition, the package provides the function `tcia_svs_download`. Similar to `tcia_svs_info`, this function requires a `camic_id` and enables downloading of SVS files directly from TCIA. By default, images are stored in the temporary directory of the R session, although a user-defined destination folder can be specified via the `destdir` argument. At present, the package does not include functionality to programmatically download the large JSON metadata file (~150 MB) required to obtain the `camic_id`.

Overall, the `TCIAAPI` package simplifies access to TCIA resources within the R environment, facilitating the integration of large-scale histopathological imaging data into reproducible workflows.

3.5.3 HistoImageR

`HistoImageR` provides functions to import extracted H&E image features into R, incorporate them with multi-omics data, and connect to existing R/Bioconductor workflows. The package supports importing feature tables into well-established data structures such as `SpatialExperiment`, `SpatialFeatureExperiment`, and `MultiAssayExperiment`, enabling seamless integration with spatial and multi-modal analyses. In addition, `HistoImageR` includes tools for preprocessing feature tables and generating visualizations that overlay the original H&E image with reconstructed representations derived from the extracted features. This includes per-nucleus annotation maps, where each nucleus can be color-coded according to specific labels or quantitative metrics. Such functionalities facilitate exploratory data analysis, spatial interpretation, and quality control of histopathological features directly within the R environment.

3.6 Conclusion

In this chapter, we have outlined the rationale, methodological framework, and computational strategies for incorporating digital pathology into multi-omic cancer research. Histopathological whole-slide images (WSIs) represent a uniquely rich data source, bridging cellular morphology and tissue architecture with molecular and clinical endpoints. Advances in deep learning and feature engineering now enable the extraction of both interpretable and latent image-derived features, thereby opening new avenues for integrative oncology studies.

A recurring theme in this chapter is the fragmentation of the computational landscape. While Python provides a mature ecosystem of tools for digital pathologyincluding HoVer-Net, Squidpy, and Prov-GigaPathR and Bioconductor remain the dominant environments for multi-omic integration, statistical modeling, and reproducible workflows. To bridge this gap, we have established an R-based infrastructure with packages such as `imageTCGA`, `TCIAAPI`, and `HistoImageR`. These tools provide streamlined access to large repositories, standardized preprocessing routines, and reproducible feature extraction workflows.

By encapsulating complex and time-consuming operations into user-friendly functions, this infrastructure dramatically reduces the computational and technical burden for researchers. Tasks that would otherwise require extensive

scripting, manual data handling, or external software dependencies can now be performed with a few lines of code in a fully reproducible environment. In doing so, these packages not only save time but also lower the barrier for incorporating digital pathology into multi-omic analyses.

Looking ahead, standardized pipelines will be indispensable as datasets continue to increase in scale and as foundation models and deep learning approaches become more pervasive in digital pathology. The convergence of scalable computing, GPU acceleration, and interoperable software ecosystems will further enhance these capabilities, establishing digital pathology as a central component of precision oncology research.

Chapter 4

Case Study: TCGA-OV

4.1 Introduction

In this chapter, we focus on the TCGA-OV cohort (Tomczak, Czerwiska, and Wiznerowicz 2015), which comprises high-grade serous ovarian cancer (HGSOC) cases. Building upon the image-based methods described in the previous chapter, we integrate histopathological features with transcriptomic and clinical data to explore molecular and microenvironmental heterogeneity. Fig. 4.1

Multi-omics integration is increasingly recognized as a key strategy to capture the complex biology of cancer. However, the cost and technical requirements for generating different data types can vary substantially. For instance, whole-slide histopathology imaging is relatively inexpensive and routinely performed in clinical practice, whereas high-throughput assays such as spatial transcriptomics remain costly and resource-intensive. Leveraging the broad availability of histological images therefore offers a scalable opportunity to extract quantitative features that complement transcriptomic, genomic, and clinical information.

The TCGA-OV dataset provides a unique resource for such an integrative analysis. It combines hematoxylineosin (H&E) whole-slide images, RNA-sequencing data, genomic profiles, and detailed clinical annotations, including survival outcomes. By linking image-derived embeddings, nuclear composition, tumor purity metrics, and bulk transcriptomic subtyping, this case study aims to uncover clinically relevant subgroups and to highlight the potential of digital pathology as a cost-effective component of multi-omics research.

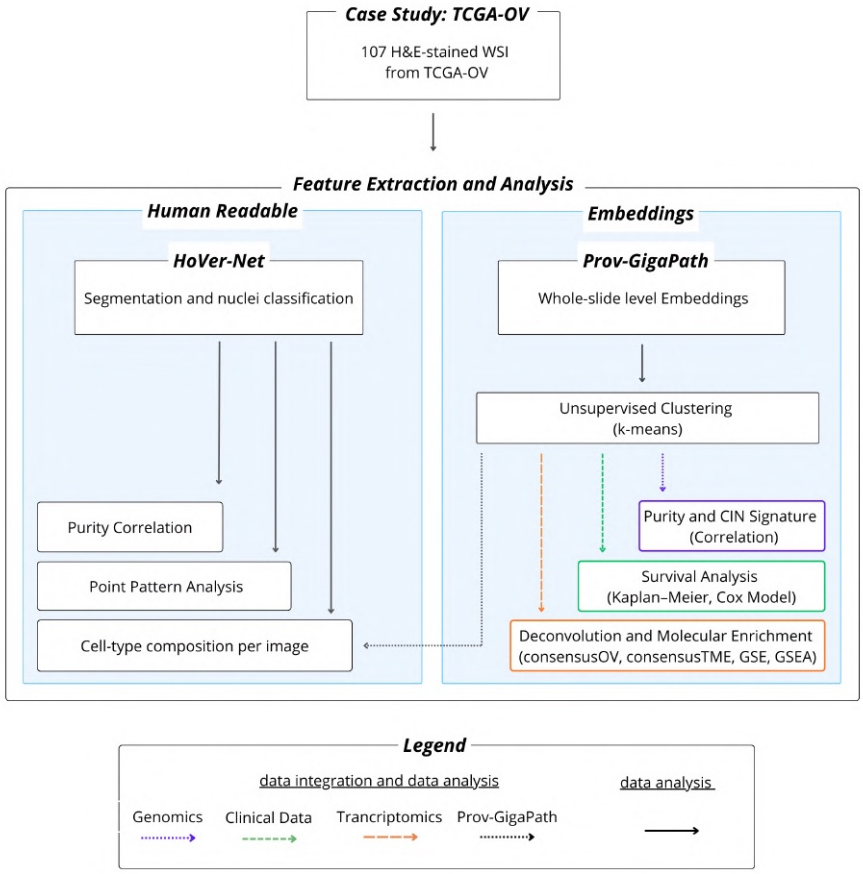

Figure 4.1: Whole-slide H&E-stained images (n = 107) from TCGA-OV were processed through two complementary pipelines for feature extraction and downstream analysis. (Left) Human-readable pipeline: HoVer-Net was used for nuclei segmentation and cell-type classification, enabling purity correlation, point-pattern analysis, and quantification of cell-type composition per image. (Right) Embedding-based pipeline: whole-slide embeddings generated with Prov-GigaPath were clustered using unsupervised k-means. Resulting clusters were evaluated through purity and chromosomal instability (CIN) signature correlation, survival analysis (KaplanMeier and Cox proportional hazards models), and molecular enrichment analyses (consensusOV, consensusTME, GSE, GSEA). Multi-omics datasets (genomic, clinical, and transcriptomic data) were integrated to support downstream interpretation, as illustrated in the legend.

4.2 High-grade serous ovarian cancer (HGSOC)

High-grade serous ovarian cancer (HGSOC) represents the most frequent and aggressive histological subtype of epithelial ovarian cancer (EOC). Epithelial ovarian cancers can be further divided into serous, endometroid, mucinous, and clear-cell based on the tissue of origin, with serous carcinomas arising from the serous epithelial layer. The serous subtype is the most common, comprising arount 75% of all epithelial cancers. (Stewart, Ralyea, and Lockwood 2019) Serous ovarian carcinomas are then classifies as either Low Grade (LGSOC) or High Grade (HGSOC) forms. LGSOC is characterized by low proliferative activity and recurrent point mutations in KRAS and BRAF genes whereas HGSOC is far more common (over 90% of serous cases) and is almost universally driven by TP53 mutations (Hatano et al. 2019) (Kurman and Shih 2016). HGSOC is also responsible for the majority of ovarian cancerrelated deaths, accounting for 70 - 80% of mortality (Lisio et al. 2019), corresponding to an estimated 111,500 - 127,500 deaths worldwide in 2022 corresponding to an estimated 111,500 - 127,500 deaths worldwide in 2022 (Bowtell et al. 2015) (Kim et al. 2018)

Most patients are diagnosed at advanced FIGO stages (III - IV) (Stewart, Ralyea, and Lockwood 2019), when the disease has already disseminated throughout the peritoneal cavity. Although tumors typically respond to first-line platinum-based chemotherapy, relapse is common, and overall prognosis remains poor, with a median overall survival of 40 months and a 5-year survival rate below 50%. (Andreou et al. 2023; L. Yang et al. 2022)

At the molecular level, HGSOC is defined by near-universal TP53 mutations, accompanied by pronounced genomic instability, widespread copy number alterations (P. Smith et al. 2023), and structural rearrangements, but without recurrent oncogenic drivers beyond TP53. This genomic complexity underlies the extensive inter - and intra - tumoral heterogeneity observed both molecularly and histopathologically. (Lynch, Bradford, and Burkard 2024; S. C. Yang et al. 2018; Azzalini et al. 2023)

The tumor microenvironment (TME) further shapes HGSOC progression and patient outcome. The degree of immune infiltration particularly cytotoxic T lymphocytes and tumor - associated macrophagescorrelates with survival, while stromal components such as cancer-associated fibroblasts (CAFs) promote invasion, progression, and chemoresistance. These observations underscore the importance of considering not only the epithelial tumor cells but

also the immune and stromal compartments in understanding ovarian cancer heterogeneity. (Arneth 2019; Hinshaw and Shevde 2019) In the case of ovarian cancer, the TME can range from solid (primary lesions, omentum) to liquid niches (ascites), with a high degree of heterogeneity both between patients and between different sites in the same patient. (Schoutrop et al. 2022)

Integrative analyses from The Cancer Genome Atlas (TCGA) identified four molecular subtypes of HGSOC (immunoreactive, mesenchymal, proliferative, and differentiated), each associated with distinct biological features and survival outcomes. The availability of matched molecular, H&E images and clinical data, including survival endpoints, renders TCGA-OV a valuable resource for linking tumor biology and microenvironmental features with prognosis. Building on this foundation, the case study presented in this chapter integrates image-derived embeddings, nuclear composition, tumor purity metrics, and transcriptomic subtyping to investigate clinically relevant subgroups within TCGA ovarian cancer patients.

4.3 Methods

4.3.1 Cluster Analysis

Clustering of the embedding features from Prov-GigaPath was performed to identify distinct molecular subgroups. First, the optimal number of clusters was estimated using the `clustree` package (Zappia and Oshlack 2018) in R, which evaluates cluster stability across a range of cluster resolutions. Based on these results, *k*-means clustering was applied to the selected embedding space using the `kmeans` function from the `stats` package in R.

To assess whether clinical or molecular variables differed significantly across the identified clusters, a Kruskal-Wallis rank-sum test was performed for continuous variables, as implemented in the `stats` package. When the Kruskal-Wallis test indicated significant differences, pairwise post hoc comparisons were carried out using Dunns test with Benjamini-Hochberg adjustment for multiple testing.

4.3.2 Survival Analysis

To evaluate whether the image-derived clusters were associated with patient outcome, overall survival (OS) analysis was performed using clinical data from TCGA. Kaplan–Meier survival curves were generated with the `survival` and `survminer` packages in R, stratifying patients according to the k-means clusters identified from the embedding features.

Differences in survival across clusters were assessed using the log-rank test (`survdiff` function, `survival` package). When the global test indicated significance, we carried out all pairwise log-rank comparisons between clusters. The pairwise comparisons between clusters were conducted and the resulting p-values were corrected for multiple testing using the Benjamini–Hochberg (BH) method. This procedure allowed us to determine which specific cluster pairs exhibited statistically significant differences in overall survival.

4.3.3 Point Pattern Analysis

Results obtained from HoVer-Net nuclear segmentation were further investigated using *Point Pattern Analysis* (PPA), a branch of spatial statistics that represents the locations of events or objects as points in space (Baddeley, Rubak, and R. Turner 2016). As outlined by Emons et al. (2024), cells can be summarized as points following two main strategies:

1. Treat molecular features (e.g., mRNA molecules) as spatial points.
2. Segment cells and represent the centroids of the segmented objects as spatial points.

Because the TCGA-OV dataset lacks spatially resolved transcriptomics, only the second approach was applicable. HoVer-Net output provides precise nuclear boundaries, enabling representation of each nucleus by its centroid for subsequent point pattern analysis.

The following spatial statistics were applied to quantify tissue architecture and heterogeneity:

- Spatial distribution of nuclei: Plotting the raw centroids of all nuclei over the histological image offers a direct visualization of cell density and tissue organization, distinguishing highly cellular regions from sparse areas.
- Ripleys K- and L-functions: These second-order statistics quantify spatial dependence across scales. Deviations from the expectation under complete spatial randomness (CSR) identify clustering or regularity in nuclear arrangement.

- Pair correlation function ($g(r)$): Complementary to Ripleys analysis, the pair correlation function captures scale-specific attraction or inhibition among nuclei, offering a finer-grained perspective on microstructural patterns.

- Marked point pattern analysis: Incorporating cell-level features (e.g., morphology, intensity, predicted cell type) as *marks* enables stratified analyses, such as comparing tumor versus stromal nuclei or assessing the spatial distribution of highly proliferative cells.

Together, these complementary statistics provide a detailed description of spatial heterogeneity and enable the identification of potential spatial biomarkers. Integrating these metrics with transcriptomic or clinical data can yield biologically and clinically meaningful insights.

Metric	Type	Description	Interpretation in tissue analysis
Morans I	Global	Quantifies overall spatial autocorrelation; > 0 indicates clustering, < 0 dispersion.	Detects whether nuclei with similar properties cluster across the entire image.
Gearys C	Global	Alternative global autocorrelation measure, more sensitive to local variation; $C < 1$ suggests clustering.	Highlights local dissimilarities and validates the presence of nuclear clustering.
LISA (Local Morans I)	Local	Identifies clusters of neighboring nuclei with similar or dissimilar values.	Reveals localized hotspots or coldspots of nuclear similarity.
LOSH (Local Spatial Heteroscedasticity)	Local	Detects spatial heterogeneity in local variance.	Identifies tissue regions with high microenvironmental variability.
GetisOrd G_i^*	Local	Detects statistically significant hotspots (high values) and coldspots (low values).	Highlights tumor regions enriched for dense or sparse nuclear patterns.

Table 4.1: Spatial statistics applied to nuclear centroid coordinates.

4.3.4 CNA

Copy number alterations (CNAs) are genomic changes where segments of DNA are deleted or amplified, leading to an abnormal number of copies of certain genes or genomic regions. They can range from small, focal changes affecting single genes to large-scale chromosomal alterations. CNAs play a critical role in various diseases, especially in cancer activating oncogenes or inactivating

tumor suppressor genes and promoting genomic instability, leading to further mutations and tumor progression. In non-cancerous conditions, CNAs are associated with developmental disorders, neurodegenerative diseases, and immune system dysfunctions. An important piece of work recently has been dedicated to the identification of Copy Number Signatures (CNS) (Drews et al. 2022; Steele et al. 2022; Tao et al. 2023), patterns of CNAs across the genome that reflect the underlying biological processes driving these alterations. It has been shown that these signatures can be used to i) infer the mechanisms of genomic instability (such as defective DNA repair pathways or chromothripsis) (Drews et al. 2022), ii) classify tumor subtypes, or iii) predict clinical outcomes (such as treatment response, or resistance to therapies) (Steele et al. 2022; Tao et al. 2023). The most widely used method for inferring copy number signatures is Non-negative Matrix Factorization (NMF), valued for its flexibility and the interpretability of its results. This approach has been predominantly applied to data from The Cancer Genome Atlas (TCGA), enabling the identification of pan-cancer CNS (Drews et al. 2022). To date, three studies on CNS identification have been published, collectively identifying fifty-eight distinct signatures, two of them published simultaneously in 2022. Drews et al. 2022 presents a compendium of 17 CNS characterizing specific types of genomic instability, with their putative aetiologies with some of them able to predict drug response and to identify new drug targets. A recent follow-up further demonstrated the clinical utility of Drews chromosomal instability signatures by validating their predictive power for chemotherapy resistance through emulated biomarker clinical trials across multiple cancer types (Thompson et al. 2025). Steele et al. 2022 presents a set of 21 CNS that explain the copy number patterns of 97% of samples. Seventeen CNS were attributed to biological phenomena of whole-genome doubling, aneuploidy, loss of heterozygosity, homologous recombination deficiency, chromothripsis and haploidization while four CNS remain unexplained. One year later, in 2023, Tao and colleagues (Tao et al. 2023) on the same data but with a slightly different mechanism-agnostic approach, identified 20 CNS some of which were associated with known biological characteristics and patients prognosis.

4.3.5 consensusOV

Molecular subtype classification of high-grade serous ovarian carcinoma (HGSOC) samples was performed using the consensusOV package (G. M. Chen et al. 2018). This tool integrates four established gene expression-based classifiers derived from prior studies by Helland et al. 2011, Bentink et al. 2012, Verhaak et al. 2012, and Konecny et al. 2014, providing a robust consensus classification framework (G. M. Chen et al. 2018). The package is designed for transcriptome-level gene expression data and utilizes a classification approach based on random forests. This allows for tumor classification with a confidence score, distinguishing between clearly classifiable cases and more ambiguous ones. The consensus classification was developed to increase agreement among different classification methods and represents the most robust standard currently available for the study of molecular subtypes of HGSOC.

4.3.6 consensusTME

The tumor microenvironment (TME) cell-type composition was inferred using the `ConsensusTME` framework (Jiménez-Sánchez, Cast, and Miller 2019), an R package that integrates multiple immune and stromal cell gene signatures derived from large-scale transcriptomic datasets. ConsensusTME computes normalized enrichment scores for a curated panel of immune cell populations (e.g., B cells, T cells, natural killer cells, macrophages, dendritic cells) and stromal components. Normalized expression matrices obtained from bulk RNA-seq data were used as input. Default parameters were applied, and all cell-type scores were scaled across samples to facilitate downstream comparisons.

4.3.7 xCell

Cell-type enrichment analysis was also performed with the `xCell` algorithm (Dvir Aran, Hu, and Atul J Butte 2017), which estimates the relative abundance of 64 immune and stromal cell types from bulk transcriptomic profiles. `xCell` utilizes a gene setbased approach coupled with spillover compensation to distinguish closely related cell populations. Normalized gene expression data were provided as input, and enrichment scores for each cell type were calculated using default settings. The resulting xCell enrichment scores were subsequently used for comparative analyses across the predefined clusters.

4.3.8 Bulk RNA-seq analysis

Differential gene expression analysis was conducted using the `DESeq2` package (Love, Huber, and Anders 2014) in R. Raw count matrices were normalized using the median-of-ratios method, and genes with an adjusted p-value (`padj`) < 0.05 were considered significantly differentially expressed.

Gene set enrichment analysis (GSEA) and over-representation analysis (ORA) were performed with the `clusterProfiler` package (G. Yu et al. 2012), using the Gene Ontology (GO), Kyoto Encyclopedia of Genes and Genomes (KEGG), and Reactome databases as annotation sources. Enrichment results were adjusted for multiple testing using the BenjaminiHochberg procedure, and pathways with an adjusted p-value < 0.15 were considered significant, consistent with the criteria applied throughout the study.

4.4 Data

For this case study, we integrated multiple data modalities from the TCGA-OV cohort and complementary sources, including histopathological images, transcriptomics, tumor purity estimates, molecular subtypes, immune/stromal deconvolution, and clinical annotations.

- Histopathology: A total of 107 H&E whole-slide images (WSIs) from TCGA-OV were processed. Nuclear segmentation and classification were performed using HoVer-Net, generating cell-level annotations. In addition, image embeddings were obtained from Prov-GigaPath at both the tile and slide level.
- Transcriptomics: RNA-seq (STAR counts) data for TCGA-OV were retrieved via the `TCGAbiolinks` R package. Gene identifiers were harmonized (removal of Ensembl version suffixes, collapsing duplicated IDs), and expression data were organized into a `SummarizedExperiment` object (`se_ov`) linking molecular profiles with clinical metadata.
- Tumor purity: Purity estimates were derived from multiple sources. First, HoVer-Netbased estimates were stored as `purity_hovernet`. Second, TCGA clinical data provided mean purity values (`tcga_purity_mean`). Third, independent estimates were obtained from Aran, Sirota, and A. Butte (2017), which systematically compared four computational and pathology-based methods:
  - *ESTIMATE*, based on expression of immune and stromal gene signatures.
  - *ABSOLUTE*, relying on somatic copy-number alterations.

  - *LUMP* (Leukocytes Unmethylation for Purity), based on 44 non-methylated immune-specific CpG sites.

  - *IHC*, derived from visual assessment of H&E slides.

  - *CPE* (Consensus Purity Estimation), integrating the above methods after normalization.

  These complementary purity estimators allowed cross-validation of results and robustness checks.

- Consensus molecular subtypes: Bulk RNA-seq profiles were classified into consensus subtypes using the `consensusOV` Bioconductor package.

- Immune/stromal composition from `xCell` (Dvir Aran, Hu, and Atul J Butte 2017) and `consensusTME` (Jiménez-Sánchez, Cast, and Miller 2019).

- Survival and clinical data: Overall survival and progression-free survival endpoints, as well as demographic and clinical features (e.g., sex, age), were obtained from TCGA clinical records.

Data type	Source	Processing / Usage
Histopathology	107 H&E WSIs (TCGA-OV)	Nuclear segmentation and classification with HoVer-Net; image embeddings from Prov-GigaPath (tile and slide level).
Transcriptomics	RNA-seq STAR counts (TCGA-OV, via `TCGAbiolinks`)	Gene ID harmonization; `SummarizedExperiment` object linking expression with clinical metadata.
Tumor purity	HoVer-Net estimates; TCGA clinical data; computational methods (Aran, Sirota, and A. Butte 2017)	Multiple estimators: HoVer-Net (`purity_hovernet`), TCGA mean purity, ESTIMATE, ABSOLUTE, LUMP, IHC, CPE. Used for cross-validation and robustness analyses.
Consensus subtypes	`consensusOV` Bioconductor package	Classification of bulk RNA-seq profiles into four molecular subtypes.
Immune/stromal composition	xCell (Dvir Aran, Hu, and Atul J Butte 2017)	Gene signaturebased deconvolution into 64 immune and stromal cell types.
Survival & clinical data	TCGA clinical records	Overall/progression-free survival, demographic and clinical variables (sex, age, etc.).

Table 4.2: Summary of data modalities integrated in the TCGA-OV case study.

Code availability

The code used to reproduce the analyses presented in this chapter is available at: `https://github.com/billila/ImageAnalysisR`.

4.5 Results

4.5.1 HoVer-Net data exploration

Nuclei distribution

The first step of the analysis consisted of characterizing the distribution of nuclei annotations across ovarian cancer histopathological slides. Figure 4.2 shows the relative abundance of the six cell categories identified by HoVer-Net segmentation and classification. As expected, neoplastic nuclei represented the predominant population, with more than 65 million cells annotated, followed by stromal nuclei with approximately 35 million cells. Inflammatory cells were also frequent (around 6.8 million), while necrotic and benign epithelial nuclei were less represented (1.6 and 1.2 million, respectively). A smaller fraction of nuclei could not be confidently assigned to any category and were therefore classified as *no label* (about 1.1 million).

This distribution highlights the strong dominance of tumor and stromal components within the ovarian cancer microenvironment, with inflammatory and necrotic elements providing additional layers of heterogeneity. Such a compositional overview serves as the basis for subsequent integrative analyses, where morphological and transcriptomic features will be jointly investigated.

After providing an overview of the global composition of nuclei across the ovarian cancer dataset, we then focused on the distribution at the level of individual images. While the aggregated counts offer a general picture of the tumor microenvironment, examining each slide separately allows us to appreciate the variability in cell type composition and the potential influence of technical or biological factors.

In Figure 4.3 the number of nuclei per slide was not uniformly distributed, indicating substantial variability across the 106 samples. This heterogeneity may reflect differences in image quality, sampling procedures, or the actual size of the tissue area under analysis. The latter aspect is particularly relevant, as variations in the surface area directly influence the number of nuclei detected. For this

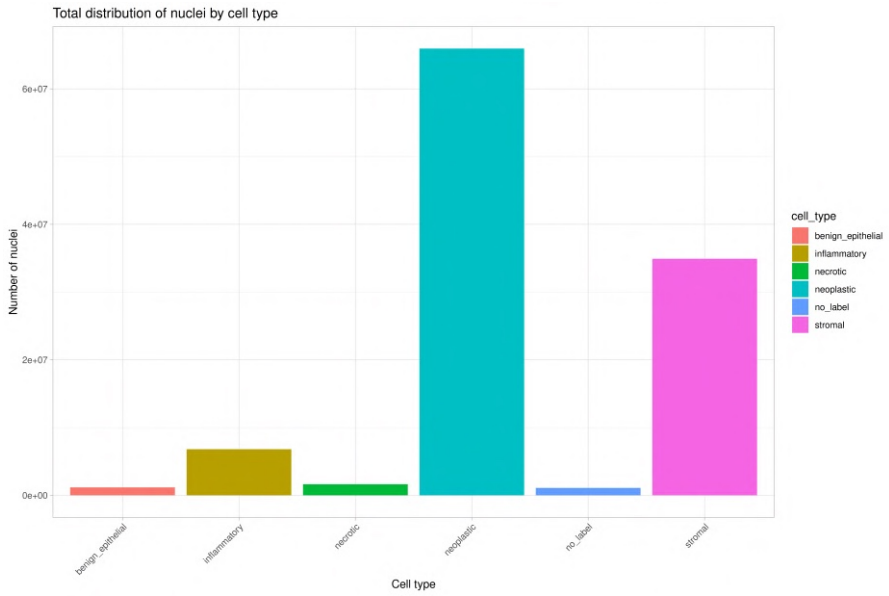

Figure 4.2: Distribution of nuclei annotations across ovarian cancer samples. Neoplastic and stromal nuclei represent the majority of the cellular compartment, followed by inflammatory, necrotic, benign epithelial, and *no label* categories.

reason, a normalization step accounting for tissue size should be considered in order to enable fair comparisons across slides.

To account for differences in the total number of nuclei detected per slide, we next examined the proportional composition of cell types within each image (Figure 4.4). By representing the data as fractions of the total nuclei per slide, we can more directly compare the relative abundance of neoplastic, stromal, inflammatory, necrotic, benign epithelial, and *no label* cells across samples. This analysis reveals that, despite the variability in absolute counts, the relative composition of cell types is generally consistent, with neoplastic and stromal cells dominating most images. Minor variations in the proportions of inflammatory, necrotic, and benign epithelial cells are also apparent, suggesting heterogeneity in the tumor microenvironment that is not solely driven by tissue area or image size.

Given the distinct cellular compositions observed across clusters, we next evaluated whether these differences are reflected in tumor purity estimates derived from multiple methodologies.

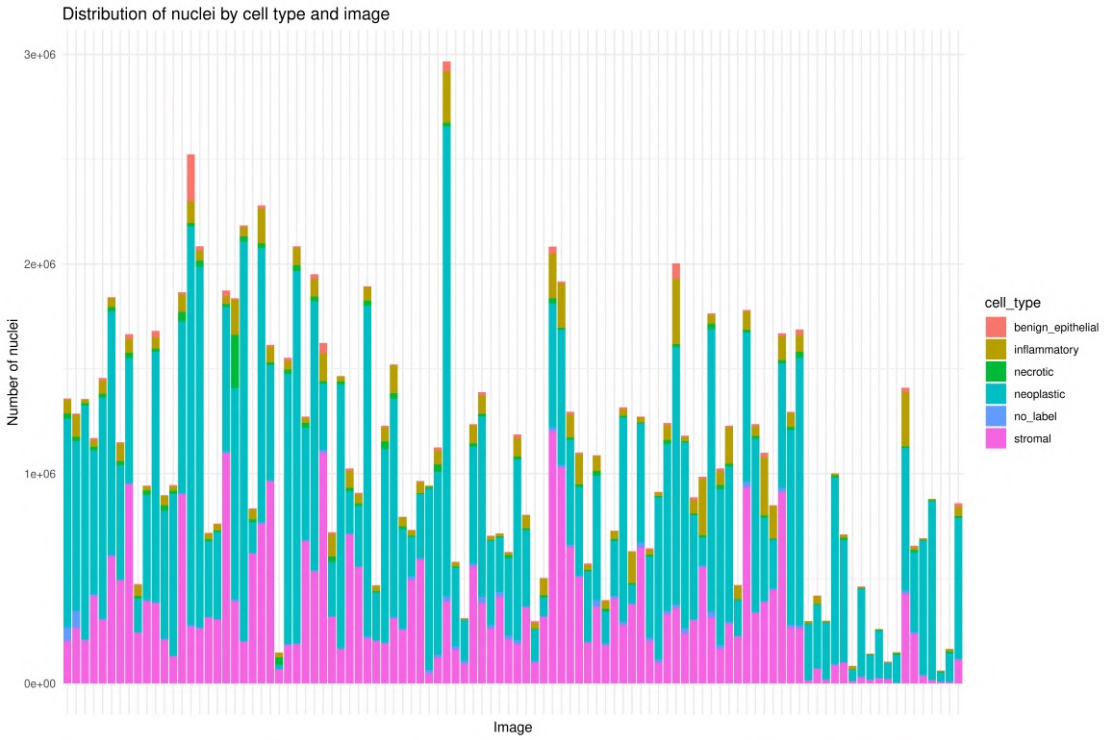

Figure 4.3: Distribution of nuclei annotations by image and cell type in the ovarian cancer dataset. The number of nuclei is not uniformly distributed across the ten slides, highlighting variability potentially due to tissue size, image quality, or sampling heterogeneity.

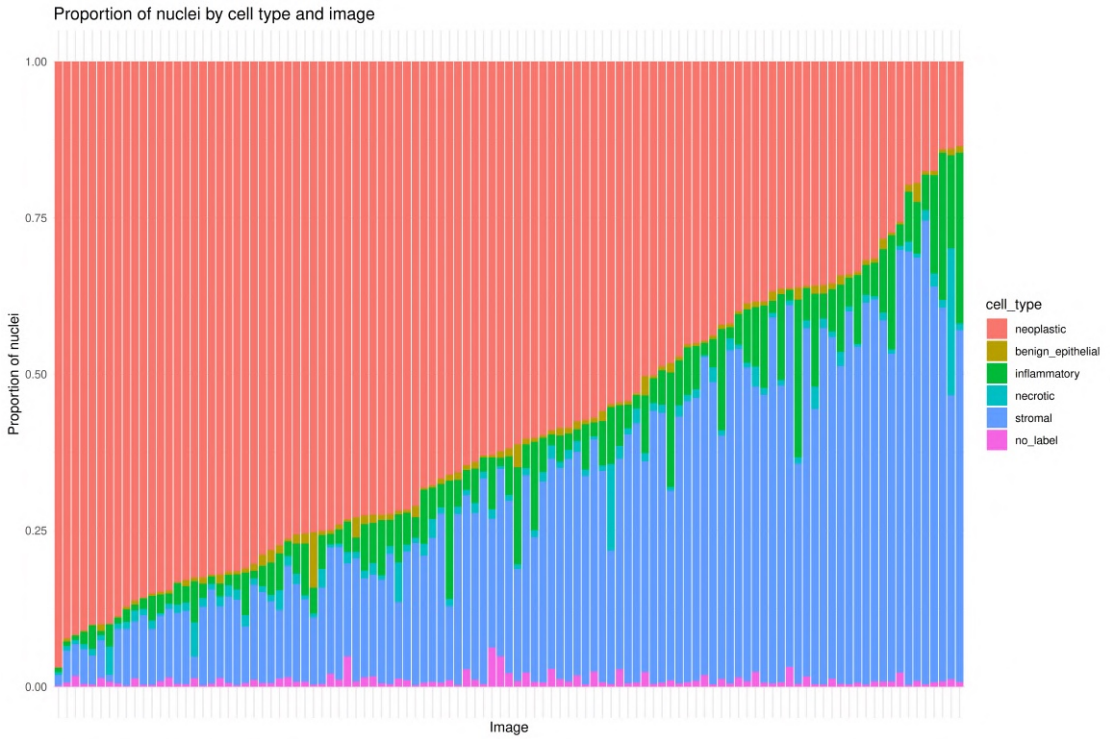

Figure 4.4: Proportional distribution of nuclei annotations by image in the ovarian cancer dataset. Representing cell types as fractions of the total nuclei per slide allows for comparison of relative composition, highlighting consistent dominance of neoplastic and stromal cells across images alongside minor variations in other populations.

Purity metrics

As anticipated in Chapter 3, we also investigated tumor purity estimates derived from HoVer-Net annotations. Purity was computed as the proportion of neoplastic nuclei within each slide and compared against multiple established purity metrics, including ABSOLUTE, LUMP, ESTIMATE, IHC, CPE, and the TCGA-reported mean purity. Correlation analysis (Table 4.3) revealed that HoVer-Net derived purity showed moderate positive correlations with LUMP ($r = 0.689$, $p = 0.028$) and ESTIMATE ($r = 0.355$, $p = 0.009$), and weaker correlations with ABSOLUTE ($r = 0.224$, $p = 0.036$) and CPE ($r = 0.257$, $p = 0.009$). Correlations with IHC ($r = -0.035$, $p = 0.724$) and TCGA mean purity ($r = 0.194$, $p = 0.045$) were negligible or marginally significant.

It is important to note that several studies have reported substantial inconsistencies in TCGA-derived purity estimatesparticularly for CPEwhich in some tumor types can be systematically biased or biologically implausible (Antonello et al. 2024). This limitation should be kept in mind when interpreting correlations involving CPE, as well as when comparing image-derived purity estimates with reference metrics.

Interestingly, HoVer-Net purity was inversely correlated with mean intensity of the neoplastic nuclei ($r = -0.430$, $p < 0.001$) and, to a lesser extent, with maximum intensity ($r = -0.222$, $p = 0.022$), whereas no significant relationships were observed with variance of intensity. These results suggest that HoVer-Net derived purity captures tumor cellularity effectively, but may also be influenced by staining intensity, highlighting the importance of considering image-derived artifacts when integrating digital pathology metrics.

Data completeness of purity metrics. We evaluated the completeness of each purity metric: as shown in Figure 4.5 and Figure 4.6, *purity_hovernet*, TCGA mean purity, and IHC have no missing values, whereas ESTIMATE and LUMP exhibit substantial missingness (50% and 90.2%, respectively). ABSOLUTE and CPE have low proportions of missing values (18.6% and 2.9%, respectively). These differences highlight that some metrics are less consistently available across samples, which may limit their utility in comparative analyses. Importantly, *purity_hovernet* provides complete coverage, supporting its suitability for cluster-based comparisons.

Given that ESTIMATE values were not available for approximately 50% of the

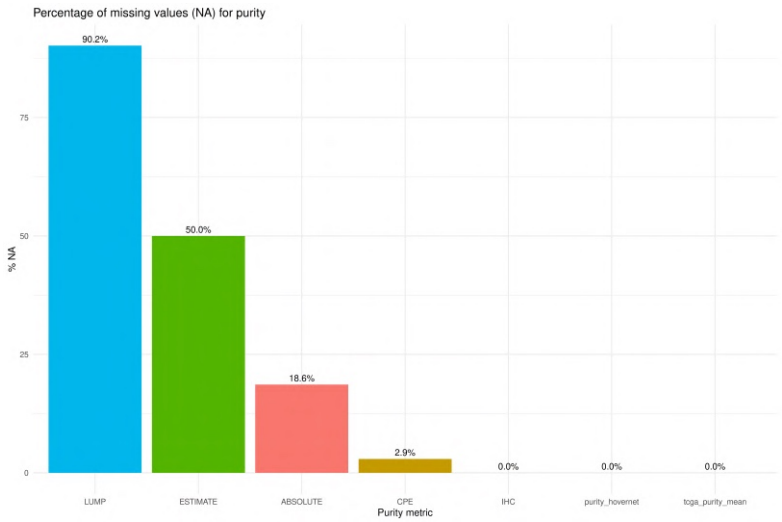

Figure 4.5: Percentage of missing values (NA) for each tumor purity metric. *Purity_hovernet*, TCGA mean, and IHC have complete coverage, whereas ESTIMATE and LUMP show substantial missingness.

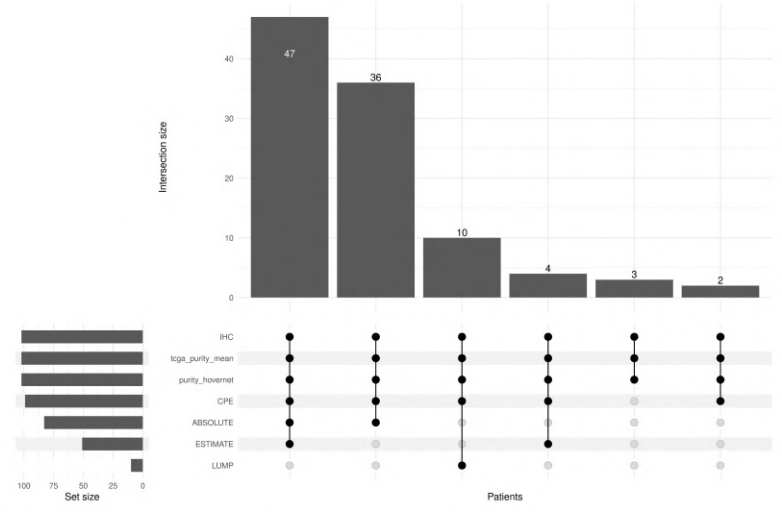

Figure 4.6: UpSet plot showing intersections between tumor purity metrics, i.e., patients for which non-missing values are simultaneously available.

Comparison	Correlation	P-value
purity_hovernet vs ABSOLUTE	0.224	**0.036**
purity_hovernet vs LUMP	0.689	**0.028**
purity_hovernet vs ESTIMATE	0.355	**0.009**
purity_hovernet vs IHC	-0.035	0.724
purity_hovernet vs CPE	0.257	**0.009**
purity_hovernet vs tcga_purity_mean	0.194	**0.045**
purity_hovernet vs percent_tumor_nuclei_mean	-0.032	0.745
purity_hovernet vs mean_intensity	-0.430	**0.001**
purity_hovernet vs variance_intensity	0.042	0.668
purity_hovernet vs max_intensity	-0.222	**0.022**
strom_perc vs percent_stromal_cells_mean	0.134	0.189
benign_perc vs percent_normal_cells_mean	-0.009	0.928
necrot_perc vs percent_necrosis_mean	0.007	0.946

Table 4.3: Correlation between HoVer-Net derived purity and various tumor purity metrics and image features.

samples, we recalculated them using the R package `tidyestimate`. To validate the procedure, we compared the new values with those originally reported by Aran, Sirota, and A. Butte 2017, obtaining a near-perfect correlation ($R = 0.99$). In addition to the purity score, `tidyestimate` also provides a stromal score. We correlated this stromal score with the stromal fraction estimated from histopathological images, defined as the proportion of stromal nuclei relative to the total number of segmented nuclei using Hover-Net. The stromal score derived from `tidyestimate` is positively correlated with the histology-based stromal fraction ($R = 0.36$, $p = 9.2 \times 10^{-4}$), supporting the consistency between transcriptomic- and image-based measures of stromal content.

Point Pattern Analysis

As introduced in Chapter 3, the output of HoVer-Net not only provides nuclear annotations but also the precise spatial coordinates of each nucleus. This information allows us to reconstruct histopathological images by exploiting the spatial arrangement of nuclei through spatial statistics packages in `R`. Figure 4.7 shows an example of such a reconstruction on TCGA-23-1121-01Z-00-DX1 image, where nuclei are represented according to their positions. In this subsection we focus on a selected images.

The spatial point pattern structure enables the computation of several metrics

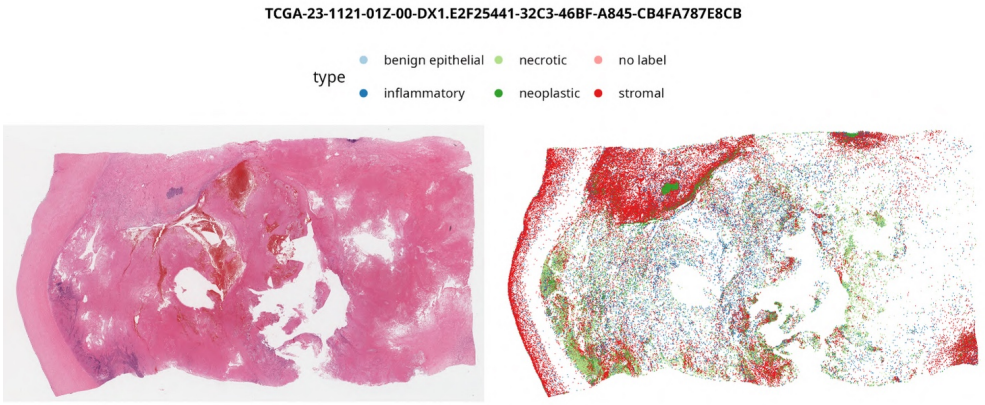

Figure 4.7: Example of reconstructed histopathological image based on nuclear coordinates extracted with HoVer-Net.

that capture local and global dependencies among nuclei. At the global level, spatial autocorrelation was assessed using Morans I and Gearys C. For TCGA-23-1121-01Z-00-DX1 image, the tests yielded the following results:

- Morans I: $I = 0.5565$, expectation = -6.2×10^{-6}, variance = 1.85×10^{-6}, standard deviate = 408.77, $p < 2.2 \times 10^{-16}$. This indicates a strong positive spatial autocorrelation, suggesting that nuclei with similar characteristics tend to cluster together rather than being randomly distributed.

- Gearys C: $C = 0.4475$, expectation = 1, variance = 2.97×10^{-6}, standard deviation = 320.55, $p < 2.2 \times 10^{-16}$. A Monte Carlo test with 1000 simulations confirmed the significance ($p = 0.001$). This result supports the presence of spatial clustering.

To further explore local patterns, we computed Local Indicators of Spatial Association (LISA). The Local Morans I revealed distinct clusters of neighboring nuclei with similar values (Figure 4.8).

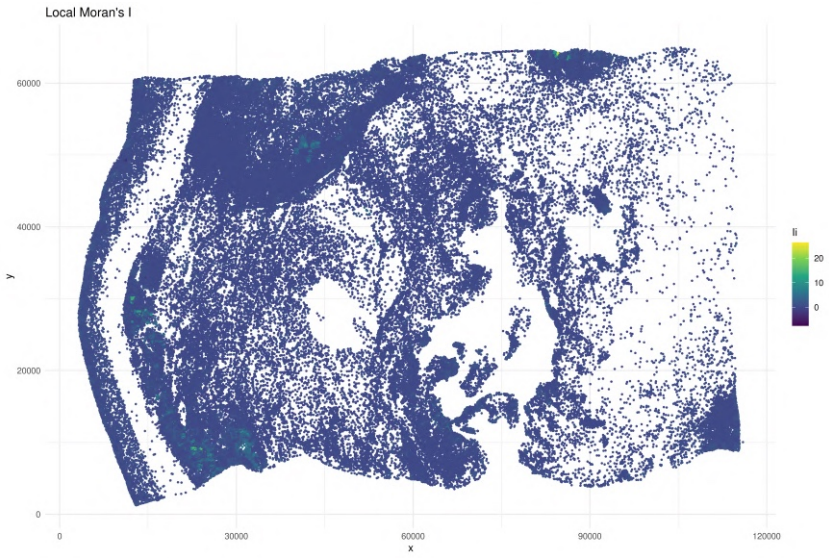

Figure 4.8: Local Morans I computed on nuclear coordinates. Warmer colors indicate higher local autocorrelation.

In addition, the Local Spatial Heteroscedasticity (LOSH) index highlighted areas with significant heterogeneity in nuclear variability. For this image we observed a mean LOSH value of 0.9886 with a mean z-score of 1.3246. Out of 161,251 nuclei, 33,151 (20.5%) were significantly heterogeneous ($p < 0.05$). These results are visualized in Figure 4.9.

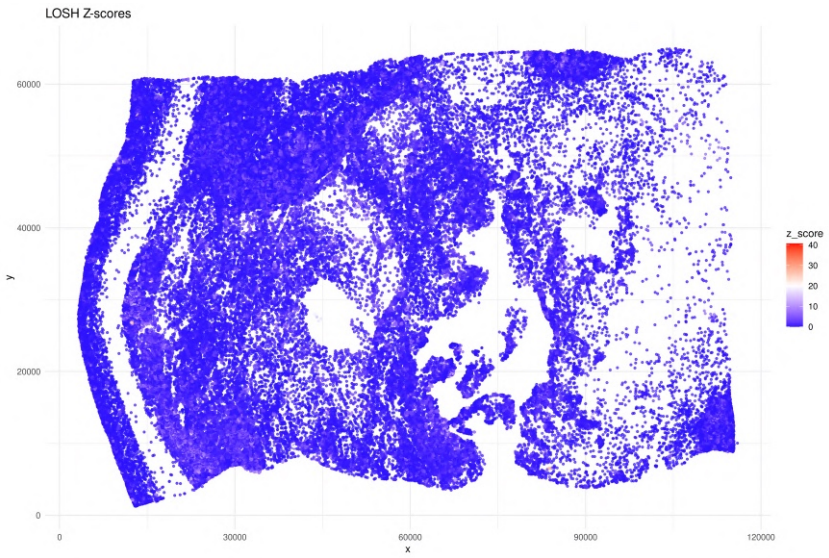

Figure 4.9: LOSH z-scores. Red areas indicate high heteroscedasticity, while blue areas indicate low heteroscedasticity.

Finally, we applied the GetisOrd G_i^* statistic to identify hotspots and coldspots

within the tissue. This analysis revealed the presence of regions enriched in high or low values, reflecting spatially organized microenvironments within the tumor tissue (Figure 4.10).

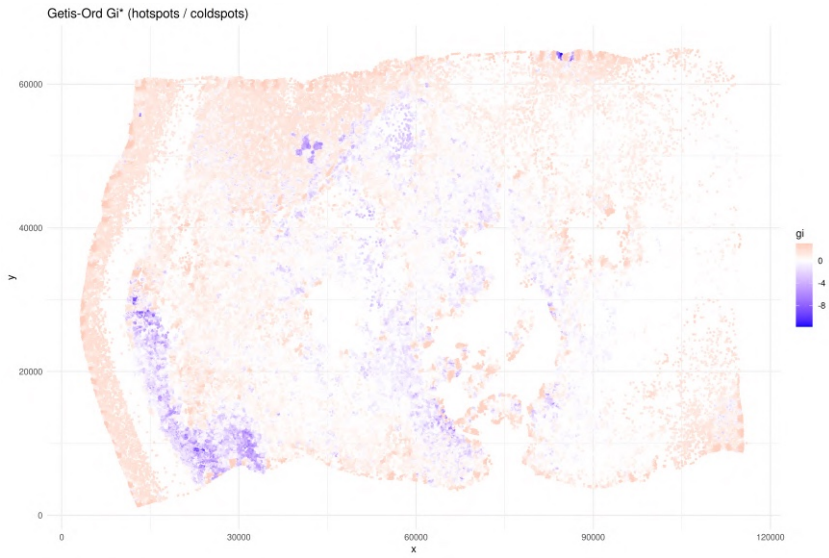

Figure 4.10: GetisOrd G_i^* statistic. Red areas correspond to hotspots, blue areas to coldspots.

Taken together, these results demonstrate that nuclear coordinates extracted from histopathological images can be effectively exploited for point pattern analysis, enabling the quantification of spatial dependencies and the identification of biologically relevant tissue structures.

Moreover, these spatial metrics can serve as the basis for developing methods to evaluate and guide the selection of images exhibiting specific spatial characteristics, and they will be further leveraged in future developments to refine image selection strategies and enhance downstream spatial analyses.

4.5.2 Analysis of Prov-GigaPath embeddings

We extracted embeddings from the first layer (layer 0) of the ProvGigaPath model in order to capture low-level morphological representations of nuclei. To visualize these high-dimensional embeddings, we applied t-distributed stochastic neighbor embedding (t-SNE), projecting the data into two dimensions. Different coloring strategies were evaluated, including (i) nuclei annotation categories from HoVer-Net, (ii) image identity, and (iii) HoVer-Net derived purity values, in order to assess whether the embeddings capture biologically meaningful

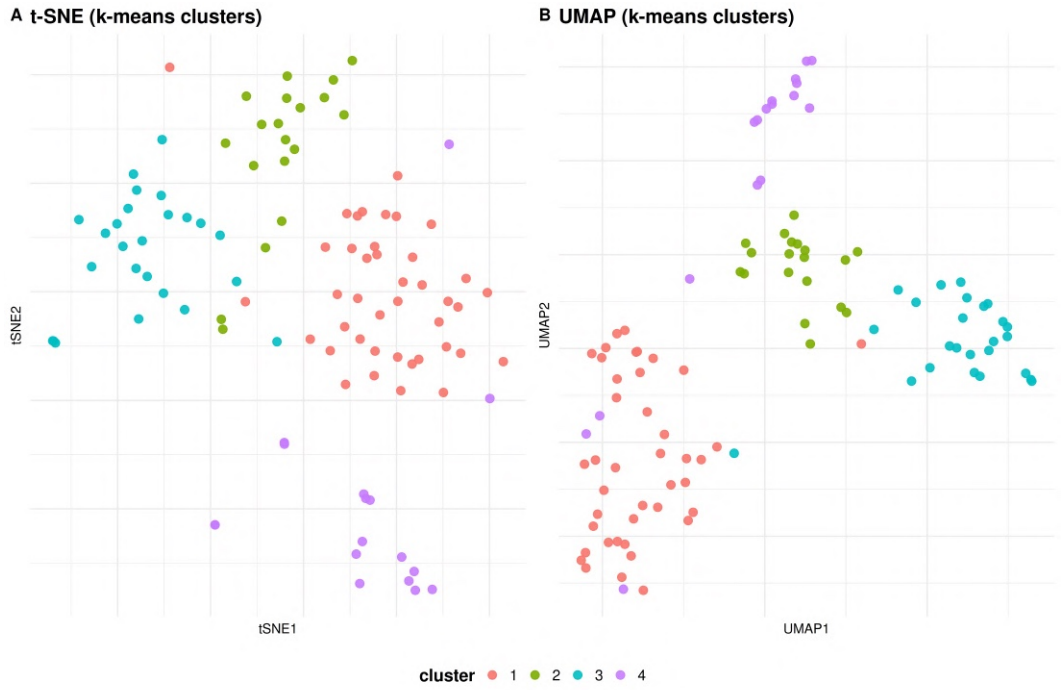

Figure 4.11: Two-dimensional representation of the embeddings extracted from the first layer of Prov-GigaPath using two different dimensionality reduction methods: (A) t-SNE and (B) UMAP. In both representations, samples are colored according to the k-means clustering assignment ($k = 4$), as determined by the clustree analysis. Both methods consistently identify the presence of four distinct groups, providing complementary evidence of the clustering structure in the embedding space.

variation or primarily reflect technical factors. Figure 4.11

To determine the optimal number of clusters in the ProvGigaPath embeddings, we applied `clustree` (Zappia and Oshlack 2018), which evaluates the stability of clustering assignments across different values of k. As shown in Figure 4.12, cluster identities remained stable up to $k = 4$, while higher values resulted in increasing instability and frequent cluster splits. Based on this analysis, we selected $k = 4$ for subsequent k-means clustering.

Survival analysis across embedding-derived clusters. To evaluate the clinical relevance of the unsupervised clustering on ProvGigaPath embeddings, we performed survival analysis using clinical data from the TCGA-OV cohort across the four k-means clusters identified.

Kaplan–Meier curves (Figure 4.13) revealed significant differences in overall survival between clusters ($p < 0.001$, log-rank test). In particular, clusters 1 and 2

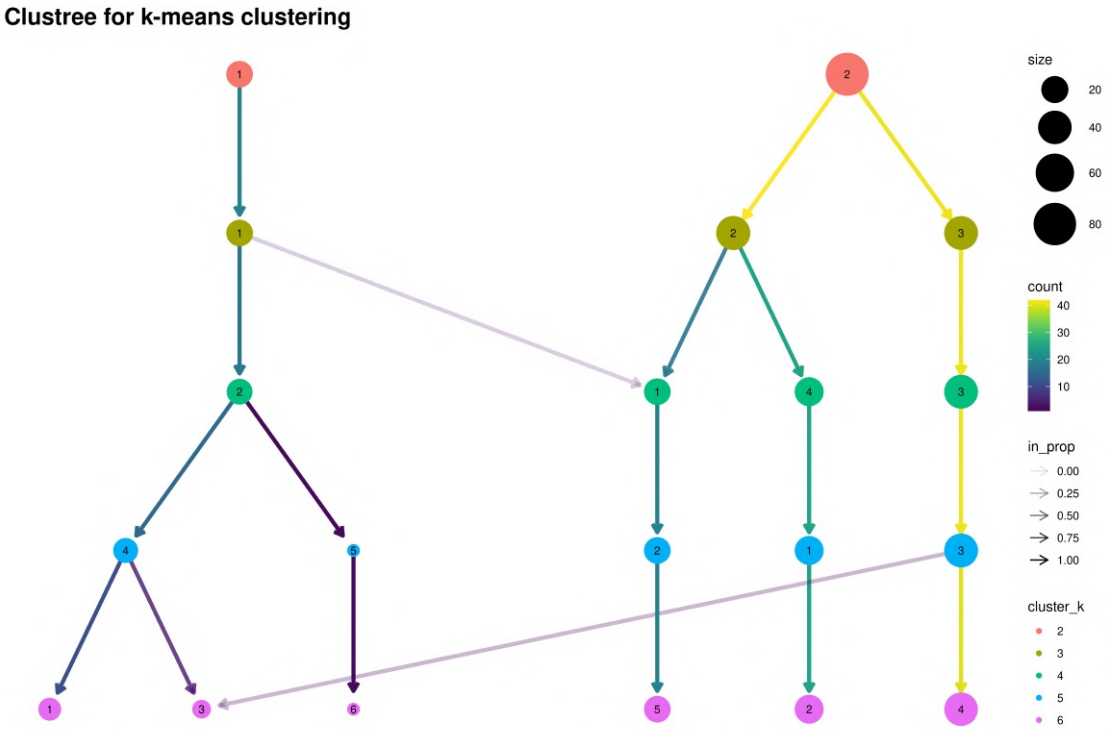

Figure 4.12: Clustree visualization of k-means clustering stability across increasing values of k. The plot indicates that four clusters represent a stable and interpretable partition of the data.

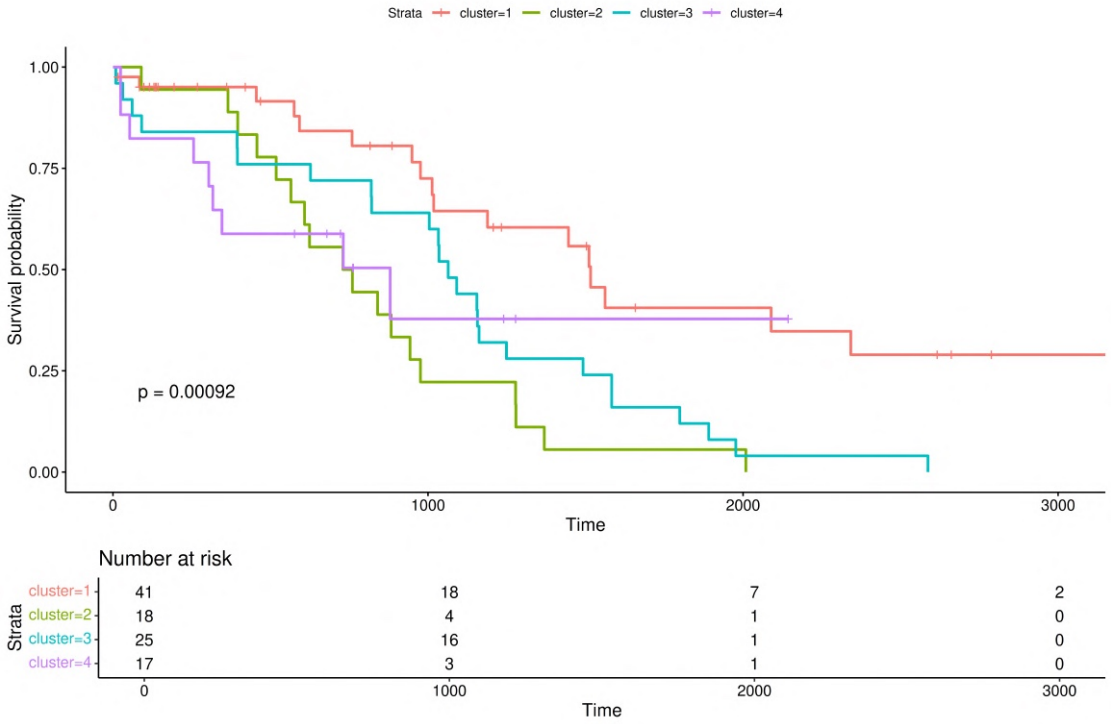

Figure 4.13: Kaplan–Meier survival curves stratified by k-means clusters derived from ProvGigaPath embeddings. Significant differences in survival are observed, particularly between clusters 1 and 2. P-value from log-rank test is reported in the plot.

displayed markedly different survival trajectories, whereas differences between clusters 3 and 4 were less pronounced.

To further assess pairwise differences, we performed log-rank tests between cluster pairs, adjusting p-values using the Benjamini–Hochberg method (Table 4.4). The strongest differences were observed between cluster 1 and both cluster 2 ($p_{adj} = 0.00039$) and cluster 3 ($p_{adj} = 0.01342$), while other comparisons did not reach statistical significance. This suggests that certain embedding-defined subgroups may capture biologically distinct patient populations with differential prognosis.

While KaplanMeier curves provide an initial, unadjusted comparison, clustersurvival associations may be influenced by confounding clinical factors. To determine whether image-derived clusters retain prognostic value after adjustment, we fitted a multivariable Cox proportional hazards model including disease stage, age group, and tumor laterality as covariates.

The results (Table 4.5) indicate that clusters 2, 3, and 4 are all associated with

	Cluster 1	Cluster 2	Cluster 3
Cluster 2	**0.00039**	–	–
Cluster 3	**0.01342**	0.21781	–
Cluster 4	0.12031	0.56775	0.97517

Table 4.4: Pairwise comparisons of Kaplan–Meier survival curves using the log-rank test. P-values adjusted with the Benjamini–Hochberg method.

significantly increased hazard compared to cluster 1, even after adjustment (HR range: 4.776.30, all $p < 0.001$). Disease stage IV was also strongly associated with poorer survival (HR = 3.03, $p < 0.001$), while age showed no significant effect. Unilateral tumors were associated with increased hazard compared to bilateral involvement (HR = 2.36, $p = 0.009$).

These findings suggest that the embedding-derived clusters capture prognostic information that is not fully explained by conventional clinical variables, and may reflect biologically meaningful patterns in tumor morphology.

Characteristic	HR	95% CI	p-value
cluster			
1	—	—	—
2	6.30	2.92, 13.6	< 0.001
3	4.77	2.02, 11.3	< 0.001
4	5.57	2.13, 14.6	< 0.001
stage_group			
IIIC_or_lower	—	—	—
IV	3.03	1.79, 5.13	< 0.001
age_group			
>60	—	—	—
≤ 60	0.95	0.57, 1.60	0.9
ov_full$laterality			
bilateral	—	—	—
unilateral	2.36	1.24, 4.49	0.009

Abbreviations: CI = Confidence Interval, HR = Hazard Ratio

Table 4.5: Multivariable Cox proportional hazards model evaluating the association between embedding-derived clusters and overall survival, adjusted for clinical covariates.

Cell type	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Benign epithelial	0.00987	**0.0113**	0.00810	0.00772
Inflammatory	0.0481	**0.124**	0.0512	0.0465
Necrotic	**0.0232**	0.00972	0.0120	0.0192
Neoplastic	0.582	0.433	0.565	**0.810**
Stromal	0.331	**0.414**	0.344	0.103

Table 4.6: Average proportion of each cell type across the four clusters derived from ProvGigaPath embeddings.

Summary of cluster-specific cellular composition. In order to further characterize the biological meaning of the four clusters identified from the embeddings, we evaluated the distribution of cell type proportions within each group. Figure 4.14 reports the boxplots of the relative abundance of nuclei types across clusters. To complement this visualization, we also computed the average proportion of each cell type per cluster (Table 4.6).

Overall, we observed that Cluster 4 was strongly enriched in neoplastic nuclei, with an average proportion of 81.0%, and showed the lowest stromal content (10.3%). Conversely, Cluster 2 displayed a markedly higher proportion of inflammatory cells (12.4%) compared to the other clusters (which ranged from 4.6% to 5.1%). Clusters 1 and 3 showed intermediate profiles, with relatively balanced contributions of stromal and neoplastic nuclei. Notably, Cluster 1 was characterized by the highest stromal proportion (33.1%) together with a substantial neoplastic fraction (58.2%).

These results indicate that the four clusters, obtained in an unsupervised manner from histopathological embeddings, reflect some degree of heterogeneity within the tumor microenvironment. While benign components are expected to be scarce and tumor cells predominant, stroma is present to varying extents across clusters, and cluster 2 shows a relatively higher immune component.

Statistical comparison of cell type proportions across clusters. To formally assess differences in cell type composition among the four clusters, we performed Kruskal-Wallis tests for each cell type (Table 4.7). The analysis revealed statistically significant differences for inflammatory, necrotic, neoplastic, stromal, and no label nuclei (all $p < 0.01$), indicating that these cell types are unevenly distributed across clusters. In contrast, the proportion of benign epithelial nuclei did not differ significantly between clusters ($p = 0.107$). These results support

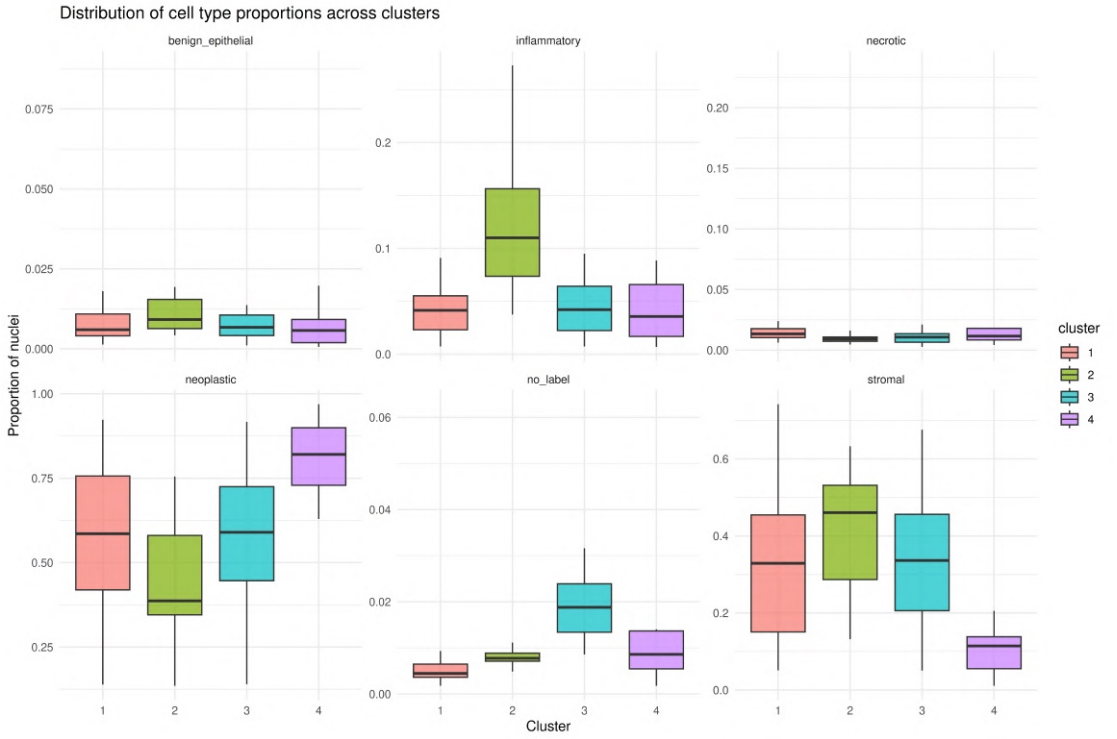

Figure 4.14: Proportions of HoVer-Net derived cell types across the four clusters.

the observation from the boxplots that the clusters capture biologically meaningful heterogeneity in tumor composition.

Cell type	Kruskal-Wallis statistic	p-value
Benign epithelial	6.09	0.107
Inflammatory	25.9	< 0.001
Necrotic	12.5	0.00582
Neoplastic	29.1	< 0.001
Stromal	30.9	< 0.001

Table 4.7: Kruskal-Wallis test for differences in cell type proportions across clusters.

Following the significant Kruskal-Wallis results, we performed Dunn's post-hoc tests with Benjamini-Hochberg correction to identify which clusters differed for each cell type. Significant pairwise differences were observed in several cases (Table 4.8). For example, inflammatory nuclei proportions differed significantly between Clusters 1 and 2, Clusters 2 and 3, and Clusters 2 and 4 ($p_{adj} < 0.001$). Neoplastic nuclei showed significant differences between multiple cluster pairs, notably Clusters 1 vs 4, 2 vs 4, and 3 vs 4 ($p_{adj} < 0.001$). Stromal and no-label

nuclei also exhibited significant differences across several cluster comparisons, indicating that the embedding-derived clusters capture distinct microenvironmental compositions.

Cell type	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Inflammatory	1 vs 2	2 vs 3	2 vs 4	-
Neoplastic	1 vs 2	-	3 vs 4	1 vs 4, 2 vs 4
No label	1 vs 2	1 vs 3	2 vs 3	1 vs 4, 3 vs 4
Stromal	1 vs 4	2 vs 4	3 vs 4	-

Table 4.8: Significant pairwise comparisons (Dunn's test with BH correction) for cell type proportions across clusters. Only comparisons with $p_{adj} < 0.05$ are shown.

Cell type	Cluster pair	Statistic	Adjusted p-value
Inflammatory	1 vs 2	4.48	< 0.001
Inflammatory	2 vs 3	-4.08	< 0.001
Inflammatory	2 vs 4	-4.29	< 0.001
Necrotic	1 vs 2	-3.17	0.00921
Neoplastic	1 vs 2	-2.48	0.0197
Neoplastic	1 vs 4	3.75	< 0.001
Neoplastic	2 vs 4	5.30	< 0.001
Neoplastic	3 vs 4	3.78	< 0.001
No label	1 vs 2	3.05	0.00274
No label	1 vs 3	7.44	< 0.001
No label	1 vs 4	3.16	0.00274
No label	2 vs 3	3.42	0.00185
No label	3 vs 4	-3.10	0.00274
Stromal	1 vs 4	-4.33	< 0.001
Stromal	2 vs 4	-5.23	< 0.001
Stromal	3 vs 4	-4.40	< 0.001

Table 4.9: Significant pairwise comparisons of cell type proportions across clusters (Dunn's test, BH correction). Only comparisons with $p_{adj} < 0.05$ are shown.

Summary of tumor purity across clusters. Consistent with the cell type composition, analysis of tumor purity revealed that Cluster 4, which is dominated by neoplastic nuclei, exhibits the highest purity values according to the *purity_hovernet* metric. Post-hoc Dunn tests confirmed that Cluster 4 differs significantly from all other clusters, while Cluster 2, enriched in inflammatory and

stromal cells, shows the lowest purity values. Clusters 1 and 3 display intermediate purity levels, reflecting their mixed composition of neoplastic, stromal, and inflammatory nuclei. Notably, other purity metrics (TCGA mean, ESTIMATE, ABSOLUTE, LUMP, IHC, and CPE) did not show significant differences among clusters, suggesting that *purity_hovernet*, derived directly from histopathological images, is more sensitive in capturing microenvironmental heterogeneity.

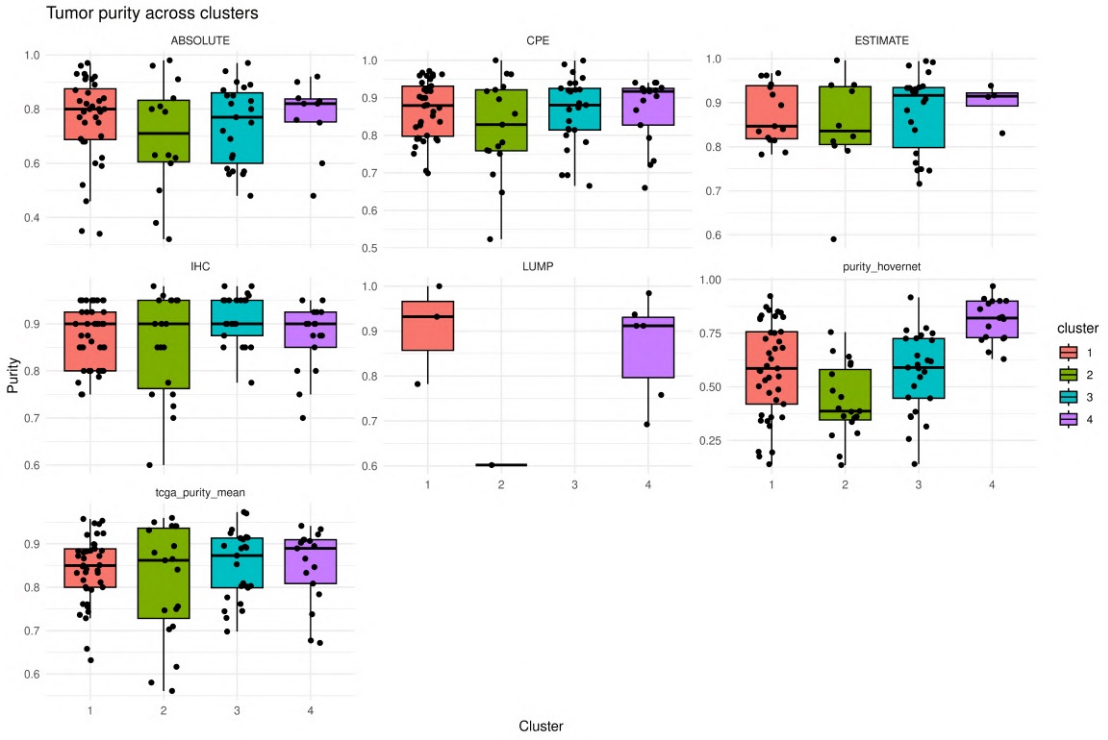

Figure 4.15: Tumor purity across the four clusters. Only *purity_hovernet* shows significant differences among clusters.

Statistical comparison of tumor purity across clusters. We evaluated whether the four clusters differed in tumor purity using multiple metrics, including purity estimated by HoVer-Net, TCGA mean purity, ESTIMATE, ABSOLUTE, LUMP, IHC, and CPE. Kruskal-Wallis tests revealed that only *purity_hovernet* showed significant differences across clusters ($\chi^2 = 29.1$, $p < 0.001$), while all other metrics were non-significant (Table 4.10).

As expected, Dunn's post-hoc tests with BH correction confirmed significant differences in tumor purity between multiple cluster pairs for *purity_hovernet* (Table 4.11). This finding is consistent with the previously observed differences in cell-type proportions, whereas the alternative purity estimates did not show

Purity metric	Kruskal-Wallis statistic	p-value
ABSOLUTE	1.11	0.776
CPE	1.68	0.641
ESTIMATE	0.349	0.951
IHC	6.69	0.0823
LUMP	2.96	0.227
purity_hovernet	29.1	< 0.001
tcga_purity_mean	0.756	0.860

Table 4.10: Kruskal-Wallis tests for differences in tumor purity across clusters.

comparable cluster-specific differences.

Cluster pair	Statistic	Adjusted p-value
1 vs 2	-2.48	0.0197
1 vs 4	3.75	< 0.001
2 vs 4	5.30	< 0.001
3 vs 4	3.78	< 0.001

Table 4.11: Significant pairwise comparisons of tumor purity (HoVer-Net) across clusters (Dunn's test, BH correction).

4.5.3 Copy number alteration signatures

To further characterize copy number alterations (CNA) across tumor clusters, we evaluated three independent sets of CNA signatures: Steele(Steele et al. 2022), Tao (Tao et al. 2023), and Drews (Drews et al. 2022).

For each signature set, we computed the mean activity per cluster and visualized the results in heatmaps (Figure 4.16). The heatmaps highlighted distinct patterns of CNA activity, with specific clusters consistently enriched for particular signatures.

In parallel, we compared the distribution of signature activities at the sample level using boxplots (Figure C.1).

Overall, the integration of multiple CNA signature sets consistently revealed heterogeneity across clusters and provided complementary perspectives on the underlying genomic architecture of ovarian cancer.

Within the Tao repertoire (Figure 4.16A), we observed a marked activation of Sig3 and Sig4. Sig3 is associated with whole-genome doubling (WGD) and sub-

sequent chromosomal fragmentation and amplification, whereas Sig4 reflects extensive chromosomal fragmentation accompanied by multiple amplification events. These patterns are consistent with the widespread structural instability that characterizes high-grade serous ovarian cancer.

In the Steele compendium (Figure 4.16B), we observed a consistent activation of CN17 across all image-derived clusters. This signature has been previously associated with homologous recombination deficiency (HRD). In particular, CN17 is enriched in the tandem duplicator phenotype, which frequently arises in tumors harboring concurrent *BRCA1* and *TP53* mutations, and has been reported to show positive associations with *TP53* mutations across multiple cancer types (Menghi et al. 2018). The activation of CN17 is therefore in line with the biological context of ovarian tumors analyzed in this study. In addition to CN17, we also detected elevated activity of CN1, CN2, and CN9, which are indicative of diploidy, tetraploidy, and focal loss of heterozygosity (LOH), respectively. Together, these signatures point to a combination of chromosomal instability and copy-number heterogeneity within the tumor samples.

Regarding the Drews compendium (Figure 4.16C), we identified activation of several signatures linked to defective DNA repair mechanisms. Specifically, CX3 is associated with impaired homologous recombination under replication stress and defective damage sensing; CX1 reflects chromosome missegregation caused by abnormal mitosis and/or telomere dysfunction; while CX2 and CX5 are both linked to impaired homologous recombination, with CX5 also capturing replication stressinduced defects. The activity of these signatures highlights the pervasive role of genomic instability in driving the molecular heterogeneity observed in the image-based clusters.

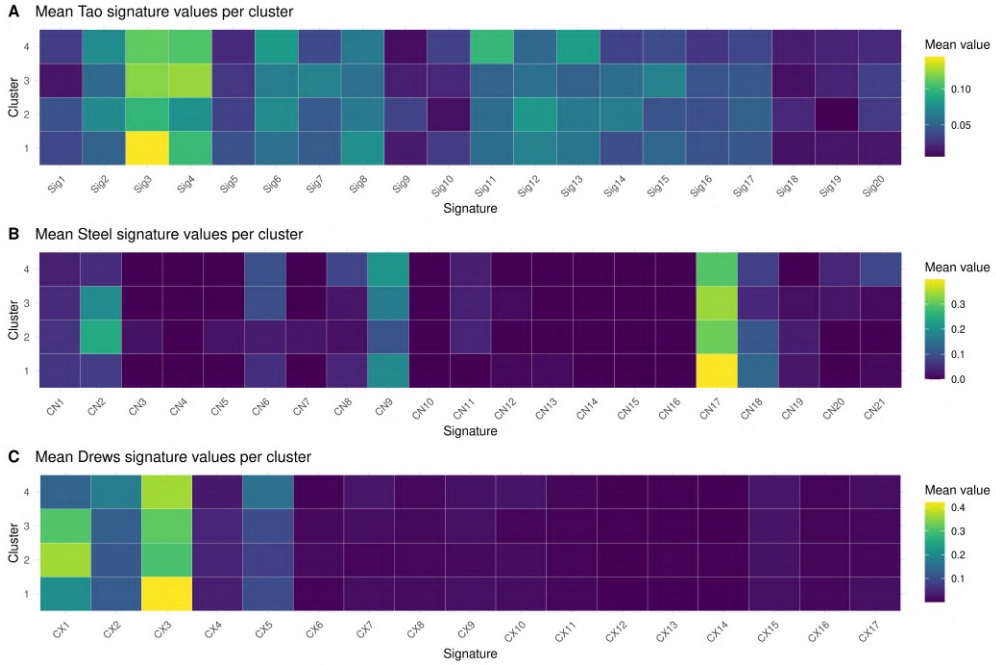

Figure 4.16: Heatmaps showing mean CNA signature activities per cluster for Steel, Tao, and Drews signature sets. Clusters were obtained from histopathological image embeddings.

4.5.4 Transcriptomic subtyping

We then decided to integrate the information extracted from histopathological images with bulk transcriptomic data from TCGA. The TCGA-OV project has released RNA-seq data for 478 ovarian cancer samples. After matching these profiles with the available images, our study cohort was reduced to 82 samples corresponding to 81 patients.

We investigated how these transcriptomic data relate to image embeddings obtained with Prov-GigaPath. By comparing RNA-seq expression patterns with the previously defined image-based clusters, we aimed to uncover biological signals linking histological structures to molecular profiles.

consensusOV ConsensusOV classification revealed that our cohort was distributed across the four canonical subtypes: Differentiated (DIF, $n = 26$), Immunoreactive (IMR, $n = 18$), Mesenchymal (MES, $n = 18$), and Proliferative (PRO, $n = 20$). When comparing these subtypes with the image-derived clusters, we observed a heterogeneous distribution (Figure 4.17). For example, cluster 1 was enriched in Differentiated and Immunoreactive cases, whereas cluster 2

contained a higher proportion of Mesenchymal samples. Clusters 3 and 4 exhibited mixed subtype compositions, reflecting the biological heterogeneity of high-grade serous ovarian cancer.

To further investigate this relationship, we compared the distribution of ConsensusOV assignment probabilities across the clusters (Figure 4.18). Kruskal-Wallis tests (Table 4.12) revealed significant differences for the Mesenchymal subtype ($p = 0.043$), with post-hoc analyses indicating that clusters 1 and 2 differed significantly in their Mesenchymal probability ($p_{adj} = 0.043$). No significant differences were observed for the other subtypes, although Differentiated showed a trend toward significance ($p = 0.060$). These results suggest that image-derived clusters partially capture transcriptional subtype-specific patterns, particularly for the Mesenchymal phenotype.

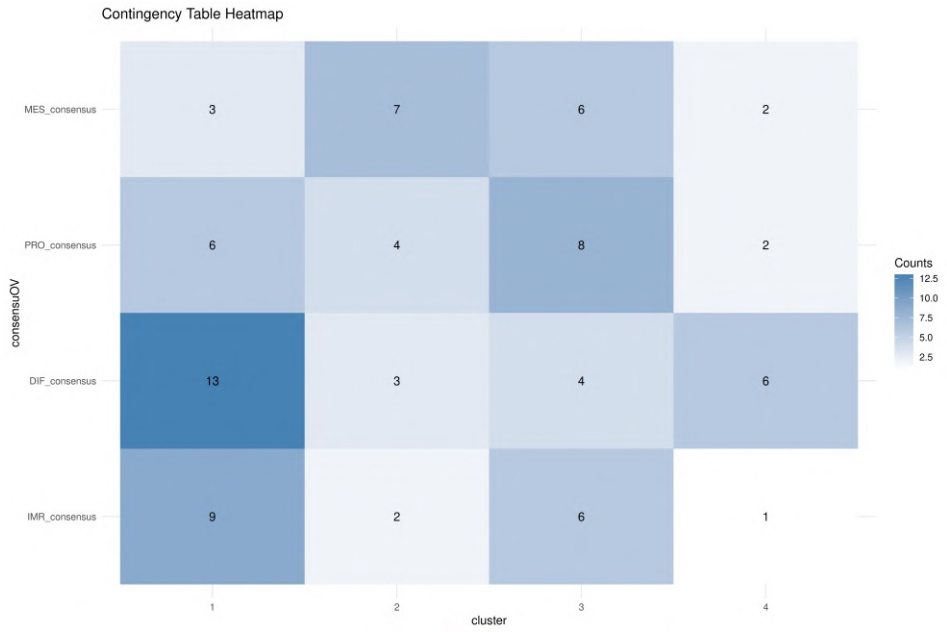

Figure 4.17: Contingency table heatmap showing the distribution of ConsensusOV subtypes across image-derived clusters. Counts are displayed within each tile.

xCell on ovarian signature To further characterize the tumor microenvironment, we applied the `xCell` framework using a customized gene signature specifically developed for ovarian cancer. This collection included 11 major cell types (B cells, CAFs, cancer cells, dendritic cells, endothelial cells, granulocytes, macrophages/monocytes, mesenchymal cells, NK cells, stromal cells, and T cells).

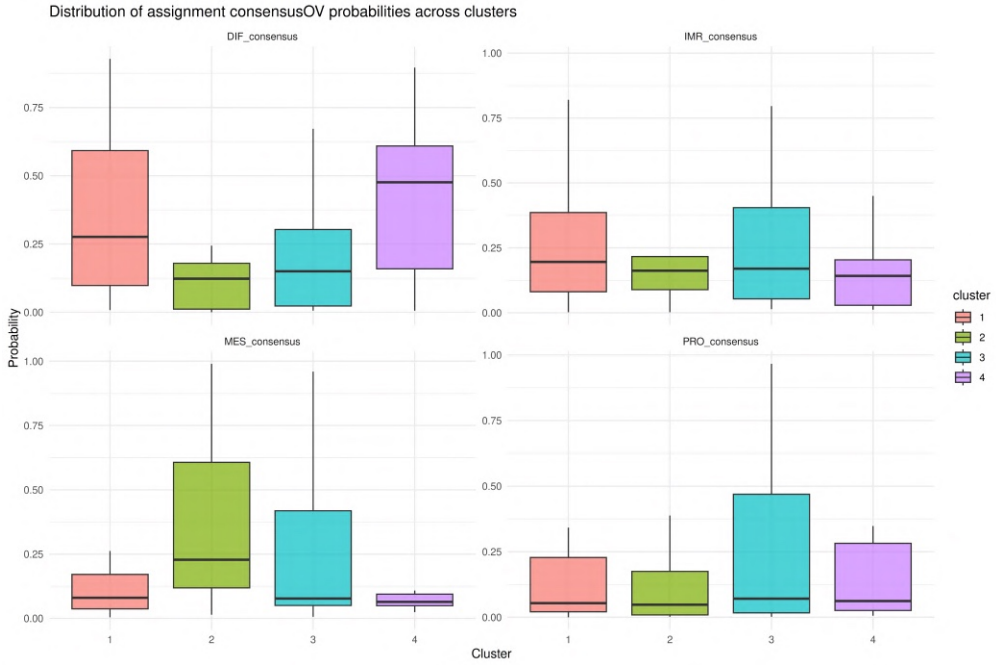

Figure 4.18: Distribution of ConsensusOV subtype probabilities across image-derived clusters. Boxplots show variation in assignment confidence.

Subtype	Variable	n	Statistic	df	p-value	Method
DIF_consensus	probability	102	7.40	3	**0.0603**	Kruskal-Wallis
IMR_consensus	probability	102	1.68	3	0.641	Kruskal-Wallis
MES_consensus	probability	102	8.16	3	**0.0429**	Kruskal-Wallis
PRO_consensus	probability	102	0.99	3	0.803	Kruskal-Wallis

Table 4.12: Kruskal-Wallis test results for ConsensusOV subtype probabilities across image-derived clusters.

For each sample, enrichment scores were computed and normalized to relative proportions. The distribution of these scores was examined across the molecular clusters using stacked barplots (Figure 4.19) and boxplots (Figures 4.20.

Statistical testing with the Kruskal–Wallis test identified a significant difference in B cell infiltration among clusters ($p = 0.0216$), whereas other cell types did not reach statistical significance (Table 4.13). Post-hoc Dunns test further revealed that B cell scores were significantly lower in cluster 4 compared with clusters 1 and 3 (adjusted $p < 0.05$). These results suggest that B cell abundance may represent a distinguishing feature of specific molecular subgroups within ovarian cancer, while other stromal and immune cell populations appeared more

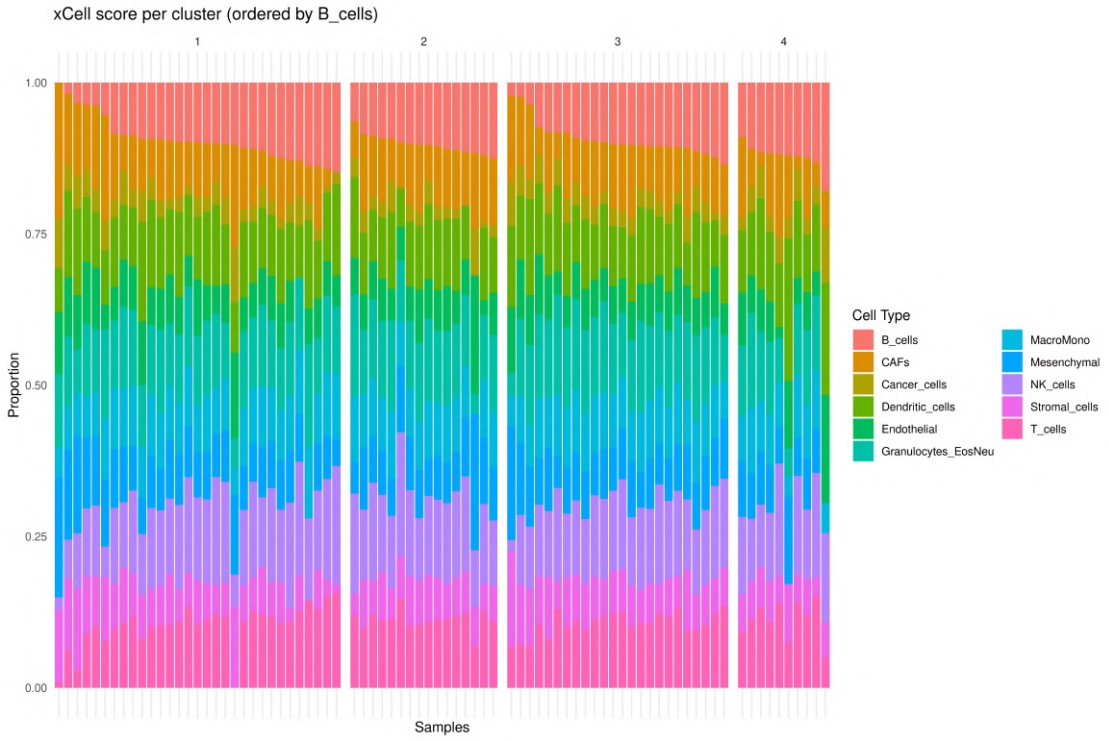

Figure 4.19: Stacked barplots of xCell enrichment scores across samples, grouped by cluster and ordered by B cells.

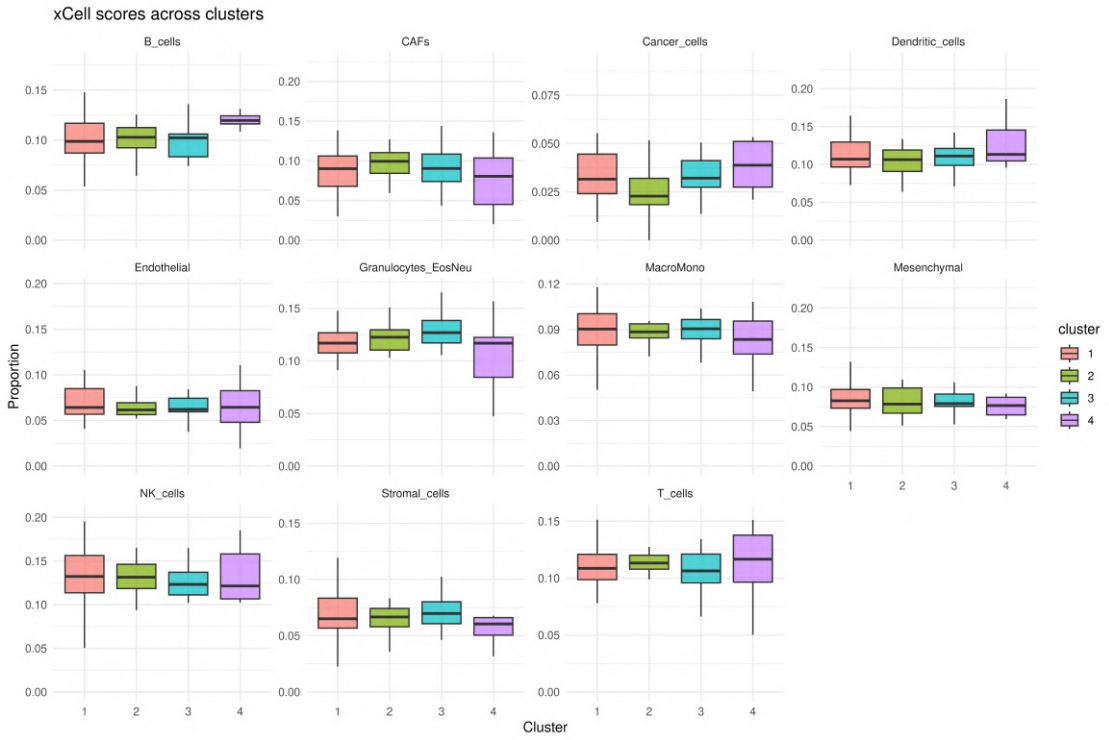

Figure 4.20: Distribution of xCell scores across clusters, stratified by cell type.

uniformly distributed.

Cell type	p-value	Significant post-hoc comparisons
B cells	0.0216	Cluster 1 vs 4 ($p_{adj} = 0.017$); Cluster 3 vs 4 ($p_{adj} = 0.017$)
CAFs	0.654	–
Cancer cells	0.0697	–
Dendritic cells	0.423	–
Endothelial	0.905	–
Granulocytes (Eos/Neu)	0.0763	–
Macrophages/Monocytes	0.779	–
Mesenchymal	0.748	–
NK cells	0.736	–
Stromal cells	0.293	–
T cells	0.639	–

Table 4.13: Kruskal–Wallis test results for xCell enrichment scores across clusters. Significant comparisons from Dunns post-hoc test are also reported.

consensusTME We further evaluated the tumor microenvironment composition using `consensusTME`, a framework that integrates multiple deconvolution methods to estimate immune and stromal cell abundances from bulk RNA-seq data. Scores were computed for 19 immune and stromal cell types, as well as a global Immune Score. The relative proportions of these cell types across samples are visualized in Figure 4.21, while Figures 4.22 illustrate their distribution across clusters.

Kruskal–Wallis tests revealed significant differences across clusters for macrophages M1 ($p = 0.0084$), neutrophils ($p = 0.0062$), and plasma cells ($p = 0.0268$), while the global Immune Score was close to significance ($p = 0.0548$) (Table 4.14). Post-hoc Dunns tests showed that M1 macrophage infiltration was higher in cluster 1 compared to cluster 2, and lower in cluster 4 compared to cluster 2. Neutrophils were enriched in clusters 2 and 3 compared to cluster 4, while plasma cells were significantly reduced in cluster 4 compared to cluster 2. Although B-cell infiltration appeared higher in cluster 4, consistent with the trend observed in the previous xCell analysis, this difference did not reach statistical significance in the consensusTME framework. These findings suggest that consensusTME captures immunological differences across molecular clusters, with neutrophils, plasma cells, and macrophages M1 emerging as key discriminating populations.

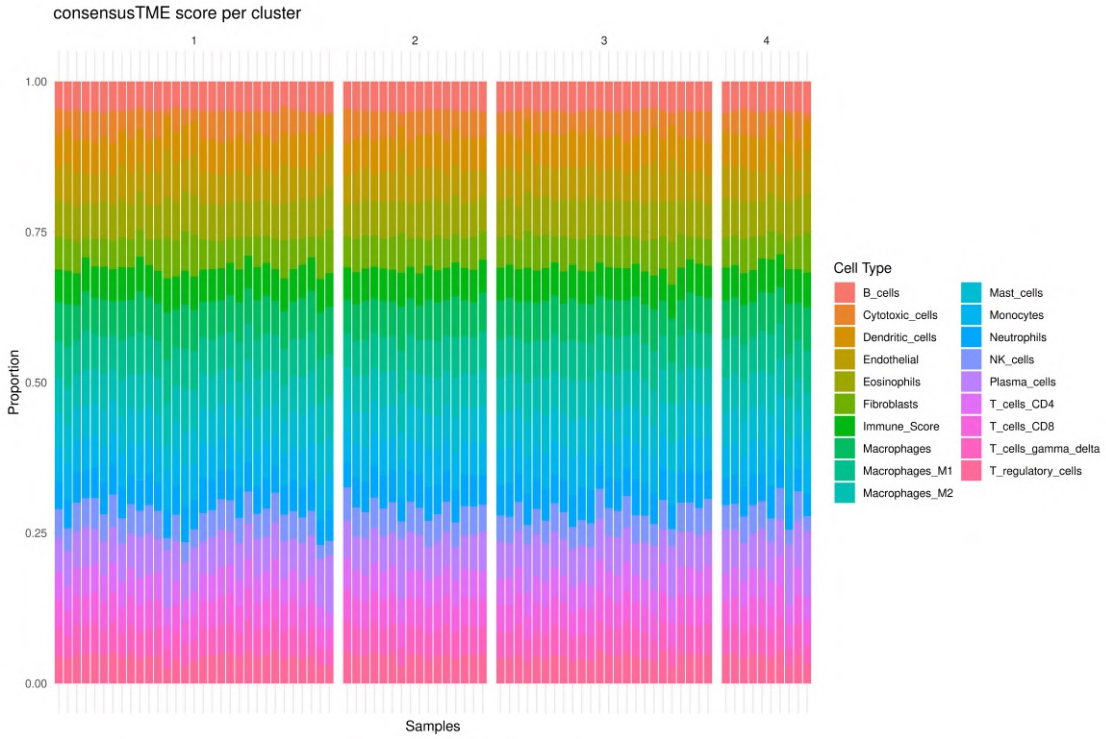

Figure 4.21: Stacked barplots of consensusTME scores across samples, grouped by molecular cluster.

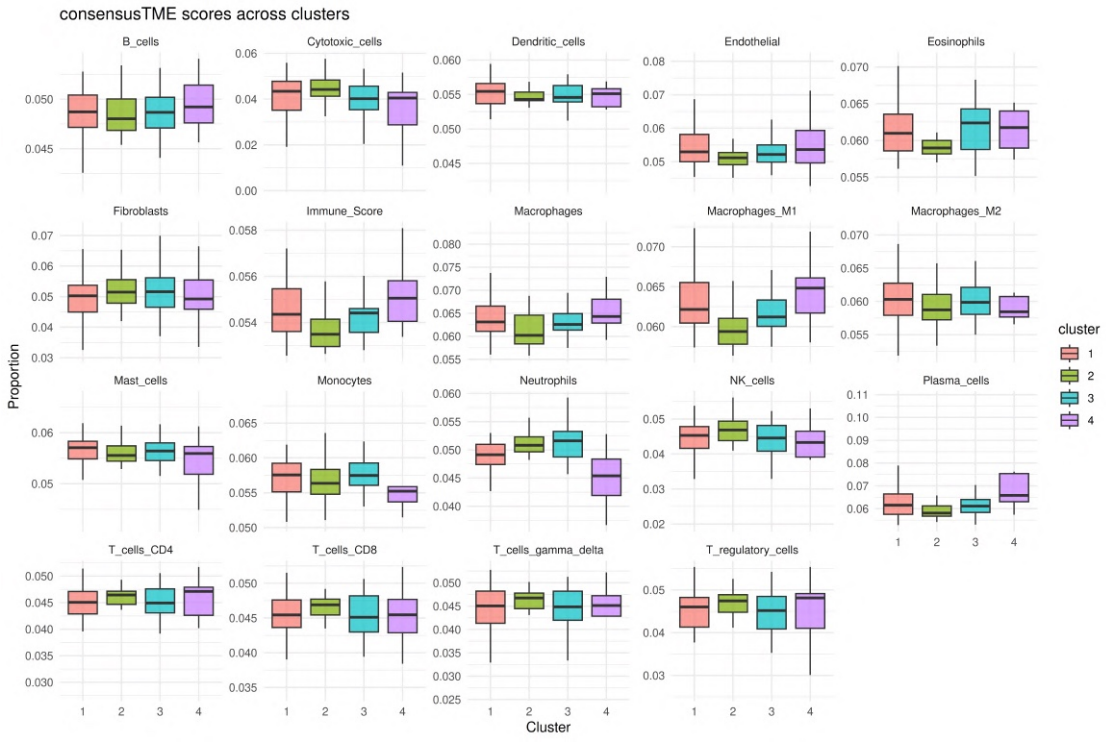

Figure 4.22: Distribution of consensusTME scores across clusters, stratified by cell type.

Cell type	p-value	Significant post-hoc comparisons
B cells	0.879	–
Cytotoxic cells	0.280	–
Dendritic cells	0.707	–
Endothelial	0.495	–
Eosinophils	0.216	–
Fibroblasts	0.775	–
Immune Score	0.0548	–
Macrophages	0.177	–
Macrophages M1	0.0084	Cl1 vs Cl2 ($p_{\text{adj}} = 0.0107$); Cl2 vs Cl4 ($p_{\text{adj}} = 0.0166$)
Macrophages M2	0.699	–
Mast cells	0.597	–
Monocytes	0.301	–
NK cells	0.448	–
Neutrophils	0.0062	Cl2 vs Cl4 ($p_{\text{adj}} = 0.0201$); Cl3 vs Cl4 ($p_{\text{adj}} = 0.0098$)
Plasma cells	0.0268	Cl2 vs Cl4 ($p_{\text{adj}} = 0.0147$)
T cells CD4	0.504	–
T cells CD8	0.497	–
T cells $\gamma\delta$	0.591	–
T regulatory cells	0.644	–

Table 4.14: Kruskal–Wallis test results for consensusTME enrichment scores across clusters. Significant post-hoc comparisons from Dunns test are reported.

Standard bulk RNA-seq analysis To investigate transcriptional differences across clusters derived from the embedding space, we performed a standard bulk RNA-seq analysis comparing all possible cluster pairs. For each comparison, we evaluated whether homologous recombination deficiency (HRD)-related pathways or immune-related pathways were enriched. Two complementary approaches were employed: (i) gene set enrichment (GSE), where differentially expressed (DE) genes were provided as input, and (ii) gene set enrichment analysis (GSEA), where the ranked list of genes was used based on the statistics of the DE test.

Cluster 1 vs Cluster 2 Differential expression analysis between Cluster 1 and Cluster 2 identified 743 upregulated and 1,069 downregulated genes. We next performed gene set enrichment (GSE) analysis using the list of significantly differentially expressed genes. Within the Gene Ontology (GO) framework, no homologous recombination (HR)-related biological processes were detected, whereas a strong immune-related signal emerged. Specifically, 16 terms as-

sociated with immune activation were significantly enriched within the Biological Process category, including B cell mediated immunity (GO:0019724, $p-adjust < 0.001$), immunoglobulin mediated immune response (GO:0016064, adjusted $p-adjust < 0.001$), and multiple pathways linked to leukocyte and lymphocyte migration and differentiation (see Table C.1). In contrast, no HR-related terms were found across Biological Process, Cellular Component, or Molecular Function categories. Consistent results were observed in KEGG analysis, which highlighted enrichment of the pathways *Phagosome* (hsa04145, adjusted $p = 0.117$) and *Neutrophil extracellular trap formation* (hsa04613, adjusted $p = 0.117$) (see Table C.6).

Gene set enrichment analysis (GSEA) was performed using the ranked list of differential expression statistics. No homologous recombination (HR)-related pathways were identified. In contrast, a robust enrichment of immune-related pathways was observed, with a total of 94 significant immune terms across Gene Ontology categories. These included adaptive immune processes such as *B cell mediated immunity* (GO:0019724, NES = -1.809, adjusted $p-adjust < 0.001$) and *immunoglobulin mediated immune response* (GO:0016064, NES = -1.827, adjusted $p-adjust < 0.001$), as well as several pathways related to leukocyte and lymphocyte migration and differentiation. Overall, these results confirm a strong activation of immune programs between Cluster 1 and Cluster 2, while no evidence of HR pathway involvement was detected (Table C.3). These results indicate a strong activation of immune-related programs between Cluster 1 and Cluster 2.

Consistent with GO analysis, KEGG pathways also highlighted immune-related processes, including *Cytokine-cytokine receptor interaction* (hsa04060, NES = -2.031, adjusted $p < 0.001$), *T cell receptor signaling pathway* (hsa04660, NES = -2.049, adjusted $p < 0.001$), and *B cell receptor signaling pathway* (hsa04662, NES = -1.952, adjusted $p < 0.001$). Similarly, Reactome analysis identified key immune-related pathways, such as *Innate Immune System* (R-HSA-168249, NES = -1.60, adjusted $p < 0.001$) and *Adaptive Immune System* (R-HSA-1280218, NES = -1.64, adjusted $p < 0.001$), supporting the GO results and underscoring the activation of both innate and adaptive immune programs.

In addition, a single homologous recombination (HR)-related pathway was identified: *double-strand break repair via homologous recombination* (GO:0000724, NES = 1.275, adjusted $p = 0.045$), suggesting a moderate involvement of DNA repair mechanisms in this contrast (Table C.4). HR-related pathways were also

detected in both KEGG and Reactome analyses. In KEGG (Table C.5), *Homologous recombination* (hsa03440, NES = 1.38, adjusted $p = 0.126$) was moderately enriched, while Reactome (Table C.8) highlighted processes such as *DNA Double-Strand Break Repair* (R-HSA-5693533, NES = 3.234, adjusted $p < 0.001$) and *HDR through Homologous Recombination* (R-HSA-5693568, NES = 1.64, adjusted $p = 0.031$).

Cluster 1 vs Cluster 3 In the comparison between groups 1 and 3, a total of 635 genes were upregulated (LFC > 0, 2.5%) and 293 genes were downregulated (LFC < 0, 1.2%). Functional enrichment analysis using Gene Ontology (GO) terms revealed that, at the level of biological processes (BP), no significant enrichment was observed for immune-related or homologous recombination (HR)-related terms in the set of differentially expressed genes (DEGs). Similarly, cellular component (CC) and molecular function (MF) categories did not show significant enrichment for these terms.

Analysis of KEGG pathways confirmed this trend, as immune-related and HR-related pathways were not significantly enriched among DEGs. However, when GSEA was applied using the ranked gene list, several immune-related processes were significantly enriched. In particular, BP terms associated with immune response, including *immunoglobulin mediated immune response* (GO:0016064), *B cell mediated immunity* (GO:0019724), and *leukocyte mediated immunity* (GO:0002443), displayed negative normalized enrichment scores (NES), indicating downregulation in group 1 relative to group 3 (Table C.9). No HR-related GO terms were significantly enriched in GSEA. Similarly, no significant enrichment was detected in CC and MF categories for either immune- or HR-related terms.

At the pathway level, KEGG GSEA identified several downregulated immune-related pathways, including *T cell receptor signaling pathway* (hsa04660), *Leukocyte transendothelial migration* (hsa04670), and *B cell receptor signaling pathway* (hsa04662), whereas the HR-related *Fanconi anemia pathway* (hsa03460) showed positive enrichment, consistent with mild upregulation (Tables C.10, C.11). Reactome GSEA confirmed enrichment of immune-related processes, such as *Antigen processing: Ub, ATP-independent proteasomal degradation* (R-HSA-9912633) and *Activation of NF-kappaB in B cells* (R-HSA-1169091), while HR-related pathways including *HDR through Homologous Recombination* (R-HSA-5685942) and *DNA Double-Strand Break Repair* (R-HSA-5693532) were upregulated (Tables C.12,

C.13). Overall, these results indicate a predominant downregulation of immune-related processes and a modest upregulation of HR-related pathways in group 1 relative to group 3.

Cluster 1 vs Cluster 4 In the comparison between groups 1 and 4, a total of 250 genes (0.99%) were significantly upregulated and 209 genes (0.83%) were downregulated.

Gene set enrichment analysis (GSEA) revealed significant enrichment in immune-related biological processes, KEGG pathways, and Reactome pathways, whereas homologous recombination (HR)-related enrichment was detected mainly in Reactome pathways. Specifically, several immune-related GO biological processes, including leukocyte migration, chemotaxis, and humoral immune response, were significantly enriched (see Table C.16). Immune-related KEGG pathways, such as *cytokine-cytokine receptor interaction* and *antigen processing and presentation*, were also significantly enriched (see Table C.17). Reactome analysis further highlighted innate immune system activation, cytokine signaling, and antigen cross-presentation as significantly enriched pathways (see Table C.18).

Regarding homologous recombination, several DNA repair and recombination pathways were enriched in Reactome, including inhibition of DNA recombination at telomeres, meiotic recombination, and HDR through HRR or SSA (see Table C.15). No significant HR-related enrichment was observed in GO or KEGG categories.

Overall, these results suggest a strong activation of immune-related pathways in Group 1 compared to Group 4, accompanied by selective enrichment of DNA repair mechanisms related to homologous recombination.

Cluster 2 vs Cluster 3 In the comparison between Group 2 and Group 3, differential expression analysis revealed a total of 111 genes (0.44%) that were significantly upregulated and 52 genes (0.21%) that were downregulated). Gene set enrichment analysis (GSEA) demonstrated significant enrichment in multiple immune-related pathways across different databases, while no homologous recombination (HR)-related pathways showed significant enrichment. KEGG pathway analysis identified 13 significantly enriched immune-related pathways, with the most prominent being Cytokine-cytokine receptor interaction (NES = 1.81, p-adjust < 0.001), Intestinal immune network for IgA production

(NES = 2.10, p-adjust < 0.001), and Inflammatory bowel disease (NES = 1.92, p-adjust < 0.001). Additional significantly enriched pathways included Phagosome, Leukocyte transendothelial migration, and T cell receptor signaling pathway (Table C.21). Gene Ontology biological process analysis revealed extensive enrichment in immune-related processes, with 87 significantly enriched terms related to adaptive immunity, lymphocyte function, and immune cell migration. The top enriched processes included Adaptive immune response based on somatic recombination (NES = 2.16, p-adjust < 0.001), Lymphocyte mediated immunity (NES = 2.10, p-adjust p-adjust < 0.001), and Leukocyte migration (NES = 1.95, p-adjust p-adjust < 0.001) (Table C.20). Reactome pathway analysis identified 6 significantly enriched immune-related pathways, including Adaptive Immune System (NES = 1.43, p-adjust < 0.001) and Cytokine Signaling in Immune system (NES = 1.43, p-adjust < 0.001), confirming the robust activation of immune-related processes (Table C.22). Notably, comprehensive screening across all databases (GO, KEGG, and Reactome) failed to identify any significantly enriched pathways related to homologous recombination or DNA repair mechanisms. This finding suggests that the transcriptomic differences between Group 2 and Group 3 are primarily driven by immune system activation rather than alterations in DNA repair pathways. Overall, these results indicate a pronounced activation of adaptive and innate immune responses in Group 2 compared to Group 3, with particular emphasis on lymphocyte-mediated immunity, cytokine signaling, and immune cell migration processes.

Cluster 2 vs Cluster 4 In the comparison between Group 2 and Group 4, differential expression analysis revealed substantial transcriptomic differences with 590 genes (2.3%) significantly upregulated and 321 genes (1.3%) significantly downregulated, representing the most extensive gene expression changes observed among the group comparisons.

Gene set enrichment analysis demonstrated significant enrichment across multiple pathway databases. Gene Ontology biological process analysis identified 87 significantly enriched immune-related pathways, with the most prominent being Myeloid leukocyte migration (NES = 2.61, p-adjust < 0.001), Leukocyte chemotaxis (NES = 2.59, p-adjust < 0.001), and Regulation of leukocyte migration (NES = 2.59, p-adjust < 0.001). Additional highly enriched processes included cellular activation, immune effector processes, and lymphocyte functions (Table C.23). Notably, one homologous recombination-related path-

way, Double-strand break repair via homologous recombination (GO:0000724), showed significant negative enrichment (NES = -1.48, p-adjust < 0.001). KEGG pathway analysis revealed 13 significantly enriched immune-related pathways, with Cytokine-cytokine receptor interaction being the most significantly enriched (NES = 2.230, p-adjust < 0.001), followed by Phagosome (NES = 2.161, p-adjust < 0.001) and Leukocyte transendothelial migration (NES = 2.224, p-adjust < 0.001). Importantly, two HR-related pathways showed significant negative enrichment: Homologous recombination (NES = -1.673, p-adjust < 0.001) and Fanconi anemia pathway (NES = -1.489, p-adjust < 0.001) (Table C.28). Reactome analysis identified 24 significantly enriched immune-related pathways, including broad categories such as Innate Immune System (NES = 2.01, p-adjust < 0.001), Cytokine Signaling in Immune system (NES = 1.82, p-adjust < 0.001), and Adaptive Immune System (NES = 1.77, p-adjust < 0.001). Additionally, 8 HR-related pathways showed significant enrichment, predominantly with negative enrichment scores, including multiple Defective homologous recombination repair (HRR) pathways due to BRCA1, BRCA2, and PALB2 loss of function (Table C.29). The comparison between Group 2 and Group 4 represents a unique pattern among all group comparisons, being the only one to demonstrate significant enrichment of DNA repair pathways alongside extensive immune system activation. The negative enrichment scores for HR pathways suggest downregulation of homologous recombination repair mechanisms in Group 2 compared to Group 4, potentially indicating compromised DNA repair capacity. This finding, combined with the pronounced immune activation, suggests that Group 2 may represent a distinct molecular phenotype characterized by both immune system upregulation and DNA repair deficiency.

Cluster 3 vs Cluster 4 In the comparison between cluster 3 and cluster 4, a total of 37 genes (0.15 %) were significantly upregulated and 84 genes (0.33 %) were downregulated. Conventional over-representation analyses of Gene Ontology (GO), KEGG, and Reactome categories did not reveal any significant immune- or homologous recombination (HR)related terms.

However, gene set enrichment analysis (GSEA) highlighted a strong immune signature. Numerous immune-related GO Biological Process (BP) terms were significantly enriched, including myeloid leukocyte migration, myeloid leukocyte activation, regulation of leukocyte migration, leukocyte chemotaxis, and leukocyte migration. KEGG pathway analysis also revealed significant enrich-

ment of immune signaling, with prominent pathways such as Toll-like receptor signaling, Phagosome, Leukocyte transendothelial migration, Cytokinecytokine receptor interaction, and both B cell receptor and T cell receptor signaling pathways. (see Table C.31, Table C.34)

For homologous recombination, GSEA detected a single significantly enriched GO BP term, double-strand break repair via homologous recombination, which displayed a negative normalized enrichment score (NES), indicating relative downregulation. (see Table C.32)

4.6 Discussion and conclusion

Our integrative analysis highlights the potential of combining histopathological embeddings, nuclear composition, and transcriptomics for ovarian cancer subtyping. Embedding-derived clusters stratified patients by prognosis and reflected biological processes such as immune infiltration and tumor purity. Alignment between embedding clusters and consensus transcriptomic subtypes supports multimodal stratification in HGSOC.

In future perspectives, we aim to extend our analysis to *tile-level image embeddings*. As previously introduced, tile-level embeddings can capture spatially resolved information, where each tissue region may carry distinct biological meaning. Exploring these patterns in relation to spatial metrics could provide important insights, as spatial context plays a pivotal role in cancer biology.

Histopathological images contain far more information than can be detected by the human eye. Computational approaches allow us to extract and quantify these hidden features, enabling a deeper understanding of tumor heterogeneity. Each small image region may present unique characteristics, and some regions may be particularly relevant for studying cancer progression and treatment response. Accessing such region-specific information could therefore open the way to a more refined level of image-based cancer research.

Correlating tile-level embeddings with external molecular data, such as CNA profiles or xCell estimates, represents a promising direction. Furthermore, the availability of datasets where omics data are matched with spatial resolution would provide a unique opportunity to explore novel integrative methods. This could help leverage the power of image analysis, which is often more cost-effective than omics approaches, while still capturing complementary biological information.

Chapter 5

Conclusions

In this thesis we presented a series of works spanning single-cell transcriptomics and digital pathology, with a common focus on computational efficiency, scalability, and reproducibility within the Bioconductor ecosystem.

The first part addressed the challenges of large-scale single-cell RNA-seq analysis. In Chapter 1 we benchmarked multiple Singular Value Decomposition (SVD) algorithms to compute the top 50 principal components, comparing R and Python implementations across dense, sparse, and HDF5-backed data. The results show that several R methods integrate naturally in Bioconductor workflows (in particular ARPACK SVD), while Python implementations exploiting GPU acceleration achieve remarkable speed ($\approx$ 7.5 s for a 1.3M cell dataset) with comparable accuracy. Chapter 2 extended this work to complete single-cell workflows, highlighting how differences in normalization, highly variable gene selection, and dimensionality reduction can influence both biological conclusions and computational costs.

Looking forward, maintaining this benchmark as methods evolve will be essential to keep Bioconductor competitive. A natural direction is the integration of GPU-enabled algorithms directly in Bioconductor, providing a scalable and user-friendly infrastructure for ever larger datasets. Expanding the benchmark to include cell–cell communication analyses will add a biologically meaningful dimension to these comparisons.

The second part of the thesis focused on histopathological image analysis, where the effort shifted from benchmarking to genuine *software development and infrastructure building*. Chapter 3 describes the substantial work required to design, implement, and document a first Bioconductor framework for H&E

whole-slide images. This included defining reproducible workflows for image preprocessing, segmentation, and feature extraction, andcruciallycreating three new R/Bioconductor packages: `imageTCGA`, `TCIAAPI`, and `HistoImageR`. These packages involved significant original code, extensive testing, and integration of external resources (TCGA and TCIA repositories, QuPath/HoVer-Net outputs, and multi-omic metadata). They fill an important gap by enabling users to programmatically access images, extract thousands of features, and perform interactive exploration entirely within the Bioconductor ecosystem, thereby lowering the barrier for large-scale computational pathology studies.

Chapter 4 then applied this infrastructure in a multi-omic case study on TCGA ovarian cancer, integrating image-derived features, copy-number signatures, and transcriptomic subtypes. This analysis illustrates the power of combining histology, genomics, and clinical data to reveal biologically meaningful heterogeneity in the tumor microenvironment.

Future perspectives for the imaging component include refining methods to identify statistically and biologically relevant representations among the vast number of possible features and deep-learning embeddings, as well as deeper multi-omic integration to link morphology with molecular states. Equally important will be the long-term maintenance and community adoption of the developed packages, ensuring that they remain robust, interoperable, and compatible with future Bioconductor releases.

Overall, this work demonstrates how rigorous benchmarking, GPU-aware computing, and open-source software development can advance both single-cell and image-based cancer research. By continuously updating the benchmark, incorporating GPU support, and sustaining the Bioconductor image infrastructure created here, these resources will help the research community meet the demands of increasingly complex biomedical datasets and foster reproducible, large-scale computational pathology.

References

Abràmoff, Michael D, Paulo J Magalhães, and Sunanda J Ram (2004). "Image processing with ImageJ". In: *Biophotonics international* 11.7, pp. 36–42.

Ahlmann-Eltze, Constantin and Wolfgang Huber (2023). "Comparison of transformations for single-cell RNA-seq data". In: *Nature Methods* 20.5, pp. 665–672.

Amezquita, Robert A et al. (2020). "Orchestrating single-cell analysis with Bioconductor". In: *Nature methods* 17.2, pp. 137–145.

Anderson, E. et al. (1999). *LAPACK users' guide*. Third ed. URL: `https://www.netlib.org/lapack/lug/`.

Andreou, Maria et al. (2023). "Prognostic factors influencing survival in ovarian cancer patients: a 10-year retrospective study". In: *Cancers* 15.24, p. 5710.

Andrews, Tallulah S and Martin Hemberg (2019). "M3Drop: dropout-based feature selection for scRNASeq". In: *Bioinformatics* 35.16, pp. 2865–2867.

Antonello, Alice et al. (2024). "Computational validation of clonal and subclonal copy number alterations from bulk tumor sequencing using CNAqc". In: *Genome Biology* 25.1, p. 38.

Aran, D, M Sirota, and AJ Butte (2017). *Systematic pan-cancer analysis of tumour purity. Nat Commun. 2015; 6: 8971*.

Aran, Dvir, Zicheng Hu, and Atul J Butte (2017). "xCell: digitally portraying the tissue cellular heterogeneity landscape". In: *Genome biology* 18.1, p. 220.

Aran, Dvir, Agnieszka P Looney, et al. (2019). "Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage". In: *Nature immunology* 20.2, pp. 163–172.

Arigoni, Maddalena et al. (2024). "A single cell RNAseq benchmark experiment embedding controlled cancer heterogeneity". In: *Scientific Data* 11.1, p. 159.

Arneth, Borros (2019). "Tumor microenvironment". In: *Medicina* 56.1, p. 15.

Arnoldi, Walter Edwin (1951). "The principle of minimized iterations in the solution of the matrix eigenvalue problem". In: *Quarterly of applied mathematics* 9.1, pp. 17–29.

Azzalini, Eros et al. (2023). "Overview of tumor heterogeneity in high-grade serous ovarian cancers". In: *International Journal of Molecular Sciences* 24.20, p. 15077.

Baddeley, Adrian, Ege Rubak, and Rolf Turner (2016). *Spatial point patterns: methodology and applications with R*. Vol. 1. CRC press Boca Raton.

Baglama, James (2016). *IRLBA: Fast Partial Singular Value Decomposition Method.*

Baglama, James and Lothar Reichel (2005). "Augmented implicitly restarted Lanczos bidiagonalization methods". In: *SIAM Journal on Scientific Computing* 27.1, pp. 19–42.

Bankhead, Peter et al. (2017). "QuPath: Open source software for digital pathology image analysis". In: *Scientific reports* 7.1, pp. 1–7.

Basak, Kayhan, Kutsev Bengisu Ozyoruk, and Derya Demir (2023). "Whole slide images in artificial intelligence applications in digital pathology: challenges and pitfalls". In: *Turkish Journal of Pathology* 39.2, p. 101.

Bentink, Stefan et al. (2012). "Angiogenic mRNA and microRNA gene expression signature predicts a novel subtype of serous ovarian cancer". In: *PloS one* 7.2, e30269.

Berg, Stuart et al. (2019). "Ilastik: interactive machine learning for (bio) image analysis". In: *Nature methods* 16.12, pp. 1226–1232.

Bergstrom, Eric N et al. (2024). "Deep Learning Artificial Intelligence Predicts Homologous Recombination Deficiency and Platinum Response From Histologic Slides". In: *Journal of Clinical Oncology* 42.30, pp. 3550–3560. DOI: 10.1200/JCO.23.02641.

Berman, Adam G et al. (2021). "PathML: a unified framework for whole-slide image analysis with deep learning". In: *MedRxiv*, pp. 2021–07.

Billato, Ilaria (2025). *imageTCGA: TCGA Diagnostic Image Database Explorer*. R package version 1.0.0. DOI: 10.18129/B9.bioc.imageTCGA. URL: https://bioconductor.org/packages/imageTCGA.

Blondel, Vincent D et al. (2008). "Fast unfolding of communities in large networks". In: *Journal of statistical mechanics: theory and experiment* 2008.10, P10008.

Bose, Ron et al. (2013). "Activating HER2 mutations in HER2 gene amplification negative breast cancer". In: *Cancer discovery* 3.2, pp. 224–237.

Bowtell, David D et al. (2015). "Rethinking ovarian cancer II: reducing mortality from high-grade serous ovarian cancer". In: *Nature reviews Cancer* 15.11, pp. 668–679.

Bulten, Wouter et al. (2025). "Artificial intelligence in digital pathologytime for a reality check". In: *Nature Reviews Clinical Oncology*. DOI: 10.1038/s41571-025-00789-6.

Cerqua, Marina et al. (2022). "MET 14 promotes a ligand-dependent, AKT-driven invasive growth". In: *Life science alliance* 5.10.

Chen, Gregory M et al. (2018). "Consensus on molecular subtypes of high-grade serous ovarian carcinoma". In: *Clinical Cancer Research* 24.20, pp. 5037–5047.

Chen, Richard J et al. (2024). "Towards a general-purpose foundation model for computational pathology". In: *Nature Medicine* 30.3, pp. 850–862. DOI: 10.1038/s41591-024-02857-3.

Christidis, A et al. (2025). *scDiagnostics: Cell type annotation diagnostics*. Version 1.2.0. DOI: 10.18129/B9.bioc.scDiagnostics.

contributors, conda (2025). *conda: A system-level, binary package and environment manager running on all major operating systems and platforms*. Version 1.2.0. URL: https://docs.conda.io/projects/conda/.

Cooper, Lee AD et al. (2018). "PanCancer insights from The Cancer Genome Atlas: the pathologist's perspective". In: *The Journal of pathology* 244.5, pp. 512–524.

Davies, Kurtis D et al. (2012). "Identifying and targeting ROS1 gene fusions in non–small cell lung cancer". In: *Clinical Cancer Research* 18.17, pp. 4570–4579.

Dicks, Severin et al. (June 2024). *scverse/rapids_singlecell: v0.10.6*. Version v0.10.6. DOI: 10.5281/zenodo.12533399.

Drews, Ruben M et al. (2022). "A pan-cancer compendium of chromosomal instability". In: *Nature* 606.7916, pp. 976–983.

Drma, Zlatko and Kreimir Veseli (2008). "New fast and accurate Jacobi SVD algorithm. I". In: *SIAM Journal on matrix analysis and applications* 29.4, pp. 1322–1342.

Eckenrode, Kelly B et al. (2023). "Curated single cell multimodal landmark datasets for R/Bioconductor". In: *PLOS Computational Biology* 19.8, e1011324.

Emons, Martin et al. (2024). "pasta: Pattern Analysis for Spatial Omics Data". In: *arXiv preprint arXiv:2412.01561*.

Folk, Mike et al. (2011). "An overview of the HDF5 technology suite and its applications". In: *Proceedings of the EDBT/ICDT 2011 workshop on array databases*, pp. 36–47.

Freymann, John B et al. (2012). "Image data sharing for biomedical researchmeeting HIPAA requirements for de-identification". In: *Journal of digital imaging* 25.1, pp. 14–24.

Gamper, Jevgenij et al. (2019). "Pannuke: an open pan-cancer histology dataset for nuclei instance segmentation and classification". In: *Digital Pathology: 15th European Congress, ECDP 2019, Warwick, UK, April 10–13, 2019, Proceedings 15*. Springer, pp. 11–19.

Golub, Gene H and Charles F Van Loan (2013). *Matrix computations*. JHU press.

Goode, Adam et al. (2013). "OpenSlide: A vendor-neutral software foundation for digital pathology". In: *Journal of pathology informatics* 4.1, p. 27.

Graham, Simon et al. (2019). "Hover-net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images". In: *Medical image analysis* 58, p. 101563.

Haghverdi, Laleh et al. (2018). "Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors". In: *Nature biotechnology* 36.5, pp. 421–427.

Halko, Nathan, Per-Gunnar Martinsson, and Joel A Tropp (2011). "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions". In: *SIAM review* 53.2, pp. 217–288.

Hao, Yuhan, Stephanie Hao, et al. (2021). "Integrated analysis of multimodal single-cell data". In: *Cell* 184.13, pp. 3573–3587.

Hao, Yuhan, Tim Stuart, et al. (2023). "Dictionary learning for integrative, multimodal and scalable single-cell analysis". In: *Nature Biotechnology*. DOI: 10.1038/s41587-023-01767-y. URL: https://doi.org/10.1038/s41587-023-01767-y.

Hastie, Trevor et al. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer.

Hatano, Yuichiro et al. (2019). "A comprehensive review of ovarian serous carcinoma". In: *Advances in anatomic pathology* 26.5, pp. 329–339.

Helland, Åslaug et al. (2011). "Deregulation of MYCN, LIN28B and LET7 in a molecular subtype of aggressive high-grade serous ovarian cancers". In: *PloS one* 6.4, e18064.

Hinshaw, Dominique C and Lalita A Shevde (2019). "The tumor microenvironment innately modulates cancer progression". In: *Cancer research* 79.18, pp. 4557–4566.

Jiménez-Sánchez, Alejandro, Oliver Cast, and Martin L Miller (2019). "Comprehensive benchmarking and integration of tumor microenvironment cell estimation methods". In: *Cancer Research* 79.24, pp. 6238–6246.

Jovic, Dragomirka et al. (2022). "Single-cell RNA sequencing technologies and applications: A brief overview". In: *Clinical and translational medicine* 12.3, e694.

Kalman, Dan (1996). "A singularly valuable decomposition: the SVD of a matrix". In: *The college mathematics journal* 27.1, pp. 2–23.

Kim, Jaeyeon et al. (2018). "Cell origins of high-grade serous ovarian cancer". In: *Cancers* 10.11, p. 433.

Konecny, Gottfried E et al. (2014). "Prognostic and therapeutic relevance of molecular subtypes in high-grade serous ovarian cancer". In: *Journal of the National Cancer Institute* 106.10, dju249.

Korsunsky, Ilya et al. (2019). "Fast, sensitive and accurate integration of single-cell data with Harmony". In: *Nature methods* 16.12, pp. 1289–1296.

Kramer, Oliver and Oliver Kramer (2016). "Scikit-learn". In: *Machine learning for evolution strategies*, pp. 45–53.

Kumar, Neeraj et al. (2017). "A dataset and a technique for generalized nuclear segmentation for computational pathology". In: *IEEE transactions on medical imaging* 36.7, pp. 1550–1560.

Kurman, Robert J and Ie-Ming Shih (2016). "The dualistic model of ovarian carcinogenesis: revisited, revised, and expanded". In: *The American journal of pathology* 186.4, pp. 733–747.

Lehoucq, Richard B, Danny C Sorensen, and Chao Yang (1998). *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*. SIAM. URL: https://epubs.siam.org/doi/10.1137/1.9780898719628.

Lisio, Michael-Antony et al. (2019). "High-grade serous ovarian cancer: basic sciences, clinical and therapeutic standpoints". In: *International journal of molecular sciences* 20.4, p. 952.

Lobato-Fernandez, Cesar, Juan A.Ferrer-Bonsoms, and Angel Rubio (2025). *GPUmatrix: Basic Linear Algebra with GPU*. Version 1.0.2. DOI: 10.32614/CRAN.package.GPUmatrix.

Lopez, Romain et al. (2018). "Deep generative modeling for single-cell transcriptomics". In: *Nature methods* 15.12, pp. 1053–1058.

Love, Michael I, Wolfgang Huber, and Simon Anders (2014). "Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2". In: *Genome biology* 15.12, p. 550.

Luecken, Malte D et al. (2022). "Benchmarking atlas-level data integration in single-cell genomics". In: *Nature methods* 19.1, pp. 41–50.

Lun, Aaron (2023). *BiocSingular: Singular Value Decomposition for Bioconductor Packages*. R package version 1.16.0. DOI: 10.18129/B9.bioc.BiocSingular. URL: https://bioconductor.org/packages/BiocSingular.

— (2025). *bluster: Clustering Algorithms for Bioconductor*. Version 1.18.0. DOI: 10.18129/B9.bioc.bluster.

Lynch, Andrew, Shermineh Bradford, and Mark E Burkard (2024). "The reckoning of chromosomal instability: past, present, future". In: *Chromosome Research* 32.1, p. 2.

Maaten, Laurens van der and Geoffrey Hinton (2008). "Visualizing data using t-SNE". In: *Journal of machine learning research* 9.Nov, pp. 2579–2605.

Madabhushi, Anant and George Lee (2020). "Precision medicine in digital pathology via image analysis and machine learning". In: *Annual Review of Biomedical Engineering* 22, pp. 487–515. DOI: 10.1146/annurev-bioeng-062117-121019.

Mardia, Kanti V, John T Kent, and Charles C Taylor (2024). *Multivariate analysis*. John Wiley & Sons.

Maystre, Lucas and Matthias Grossglauser (2015). "Fast and accurate inference of Plackett–Luce models". In: *Advances in neural information processing systems* 28.

McCarthy, Davis J et al. (2017). "Scater: pre-processing, quality control, normalization and visualization of single-cell RNA-seq data in R". In: *Bioinformatics* 33.8, pp. 1179–1186.

McCombe, Kris D et al. (2021). "HistoClean: Open-source software for histological image pre-processing and augmentation to improve development of robust convolutional neural networks". In: *Computational and Structural Biotechnology Journal* 19, pp. 4840–4853.

McInnes, Leland et al. (2018). "UMAP: Uniform Manifold Approximation and Projection". In: *Journal of Open Source Software* 3.29.

Menghi, Francesca et al. (2018). "The tandem duplicator phenotype is a prevalent genome-wide cancer configuration driven by distinct gene mutations". In: *Cancer cell* 34.2, pp. 197–210.

Mereu, Elisabetta et al. (2020). "Benchmarking single-cell RNA-sequencing protocols for cell atlas projects". In: *Nature biotechnology* 38.6, pp. 747–755.

Moses, Lambda and Lior Pachter (2022). "Museum of spatial transcriptomics". In: *Nature methods* 19.5, pp. 534–546.

Negrao, Marcelo V et al. (2020). "Molecular landscape of BRAF-mutant NSCLC reveals an association between clonality and driver mutations and identifies targetable non-V600 driver mutations". In: *Journal of Thoracic Oncology* 15.10, pp. 1611–1623.

Nolet, Corey et al. (2022). "Accelerating single-cell genomic analysis with gpus". In: *bioRxiv*, pp. 2022–05.

Omar, Mohamed et al. (2024). "Applications of Digital Pathology in Cancer: A Comprehensive Review". In: *Annual Review of Cancer Biology* 8.

Pagès, Hervé (2025a). *DelayedArray: A unified framework for working transparently with on-disk and in-memory array-like datasets*. Version 0.34.1. DOI: 10.18129/B9.bioc.DelayedArray.

— (2025b). *SparseArray: High-performance sparse data representation and manipulation in R*. Version 1.8.0. DOI: 10.18129/B9.bioc.SparseArray.

Palla, Giovanni et al. (2022). "Squidpy: a scalable framework for spatial omics analysis". In: *Nature methods* 19.2, pp. 171–178.

Pau, Grégoire et al. (2010). "EBImagean R package for image processing with applications to cellular phenotypes". In: *Bioinformatics* 26.7, pp. 979–981.

Pijuan-Sala, Blanca et al. (2019). "A single-cell molecular map of mouse gastrulation and early organogenesis". In: *Nature* 566.7745, pp. 490–495.

Pizurica, Marija et al. (2024). "Digital profiling of gene expression from histology images with linearized attention". In: *Nature Communications* 15.1, p. 9886. DOI: 10.1038/s41467-024-54182-5.

Program, CZI Cell Science et al. (2025). "CZ CELLxGENE Discover: a single-cell data platform for scalable exploration, analysis and modeling of aggregated data". In: *Nucleic Acids Research* 53.D1, pp. D886–D900.

Qiu, Yixuan and Jiali Mei (2025). *RSpectra: Solvers for Large-Scale Eigenvalue and SVD Problems*. Version 0.16-2. DOI: 10.32614/CRAN.package.RSpectra.

Qu, Hui-Qi, Charlly Kao, and Hakon Hakonarson (2024). "Single-cell RNA sequencing technology landscape in 2023". In: *Stem Cells* 42.1, pp. 1–12.

Rand, William M (1971). "Objective criteria for the evaluation of clustering methods". In: *Journal of the American Statistical association* 66.336, pp. 846–850.

Rawat, Rishi R et al. (2018). "Correlating nuclear morphometric patterns with estrogen receptor status in breast cancer pathologic specimens". In: *NPJ Breast Cancer* 4.1, p. 32.

Rich, Joseph M et al. (2024). "The impact of package selection and versioning on single-cell RNA-seq analysis". In: *bioRxiv*, pp. 2024–04.

Rood, Jennifer E et al. (2024). "The Human Cell Atlas from a cell census to a unified foundation model". In: *Nature*, pp. 1–2.

Ross, David A et al. (2008). "Incremental learning for robust visual tracking". In: *International journal of computer vision* 77, pp. 125–141.

Saelens, Wouter et al. (2019). "A comparison of single-cell trajectory inference methods". In: *Nature biotechnology* 37.5, pp. 547–554.

Schmauch, Benot et al. (2020). "A deep learning model to predict RNA-Seq expression of tumours from whole slide images". In: *Nature communications* 11.1, p. 3877.

Schömig-Markiefka, Birgid et al. (2021). "Quality control stress test for deep learning-based diagnostic model in digital pathology". In: *Modern Pathology* 34.12, pp. 2098–2108.

Schoutrop, Esther et al. (2022). "Molecular, cellular and systemic aspects of epithelial ovarian cancer and its tumor microenvironment". In: *Seminars in cancer biology*. Vol. 86. Elsevier, pp. 207–223.

Simonetti, Sara et al. (2010). "Detection of EGFR mutations with mutation-specific antibodies in stage IV non-small-cell lung cancer". In: *Journal of translational medicine* 8, pp. 1–8.

Smith, Philip et al. (2023). "The copy number and mutational landscape of recurrent ovarian high-grade serous carcinoma". In: *Nature communications* 14.1, p. 4387.

Stark, Rory, Marta Grzelak, and James Hadfield (2019). "RNA sequencing: the teenage years". In: *Nature Reviews Genetics* 20.11, pp. 631–656.

Steele, Christopher D et al. (2022). "Signatures of copy number alterations in human cancer". In: *Nature* 606.7916, pp. 984–991.

Stefanovska, Elena (2025). "Self-supervised deep learning for H&E-stained histopathology in a spatial omics context". In: *University of Padova Thesis*. Available at: `https://thesis.unipd.it/handle/20.500.12608/81810`.

Stephenson, Emily et al. (2021). "Single-cell multi-omics analysis of the immune response in COVID-19". In: *Nature medicine* 27.5, pp. 904–916.

Stewart, Christine, Christine Ralyea, and Suzy Lockwood (2019). "Ovarian cancer: an integrated review". In: *Seminars in oncology nursing*. Vol. 35. 2. Elsevier, pp. 151–156.

Stoeckius, Marlon et al. (2017). "Simultaneous epitope and transcriptome measurement in single cells". In: *Nature methods* 14.9, pp. 865–868.

Stuart, Tim and Rahul Satija (2019). "Integrative single-cell analysis". In: *Nature reviews genetics* 20.5, pp. 257–272.

Tao, Ziyu et al. (2023). "The repertoire of copy number alteration signatures in human cancer". In: *Briefings in Bioinformatics* 24.2, bbad053.

Thompson, Joe Sneath et al. (2025). "Predicting resistance to chemotherapy using chromosomal instability signatures". In: *Nature Genetics*, pp. 1–10.

Tian, Luyi et al. (2019). "Benchmarking single cell RNA-sequencing analysis pipelines using mixture control experiments". In: *Nature methods* 16.6, pp. 479–487.

Tizhoosh, Hamid R and Liron Pantanowitz (2021). "Artificial Intelligence in Pathology". In: *Journal of Pathology Informatics* 12, pp. 1–9. DOI: 10.4103/jpi.jpi_79_20.

Tomaszewski, John E and Babak E Bejnordi (2021). "Overview of the role of artificial intelligence in pathology: The computer as a pathology digital assistant". In: *Clinics in Laboratory Medicine* 41.1, pp. 1–11. DOI: 10.1016/j.cll.2020.10.001.

Tomczak, Katarzyna, Patrycja Czerwiska, and Maciej Wiznerowicz (2015). "Review The Cancer Genome Atlas (TCGA): an immeasurable source of knowledge". In: *Contemporary Oncology/Współczesna Onkologia* 2015.1, pp. 68–77.

Traag, Vincent A, Ludo Waltman, and Nees Jan Van Eck (2019). "From Louvain to Leiden: guaranteeing well-connected communities". In: *Scientific reports* 9.1, p. 5233.

Tran, Hoa Thi Nhu et al. (2020). "A benchmark of batch-effect correction methods for single-cell RNA sequencing data". In: *Genome biology* 21, pp. 1–32.

Tsuyuzaki, Koki et al. (2020). "Benchmarking principal component analysis for large-scale single-cell RNA-sequencing". In: *Genome biology* 21.1, p. 9.

Turner, Heather L et al. (2020). "Modelling rankings in R: the PlackettLuce package". In: *Computational Statistics* 35.3, pp. 1027–1057.

Vallejos, Catalina A et al. (2017). "Normalizing single-cell RNA sequencing data: challenges and opportunities". In: *Nature methods* 14.6, pp. 565–571.

Van der Walt, Stefan et al. (2014). "scikit-image: image processing in Python". In: *PeerJ* 2, e453.

Verhaak, Roel GW et al. (2012). "Prognostically relevant gene signatures of high-grade serous ovarian carcinoma". In: *The Journal of clinical investigation* 123.1.

Virshup, Isaac, Danila Bredikhin, et al. (2023). "The scverse project provides a computational ecosystem for single-cell omics data analysis". In: *Nature biotechnology* 41.5, pp. 604–606.

Virshup, Isaac, Sergei Rybakov, et al. (2021). "anndata: Annotated data". In: *BioRxiv*, pp. 2021–12.

Wolf, F Alexander, Philipp Angerer, and Fabian J Theis (2018). "SCANPY: large-scale single-cell gene expression data analysis". In: *Genome biology* 19, pp. 1–5.

Xianyi, Zhang, Wang Qian, and Zhang Yunquan (2012). "Model-driven level 3 BLAS performance optimization on Loongson 3A processor". In: *2012 IEEE 18th international conference on parallel and distributed systems*. IEEE, pp. 684–691.

Xu, Hanwen et al. (2024). "A whole-slide foundation model for digital pathology from real-world data". In: *Nature* 630.8015, pp. 181–188.

Yang, Ling et al. (2022). "Molecular mechanisms of platinum-based chemotherapy resistance in ovarian cancer". In: *Oncology reports* 47.4, pp. 1–11.

Yang, SY Cindy et al. (2018). "Landscape of genomic alterations in high-grade serous ovarian cancer from exceptional long-and short-term survivors". In: *Genome medicine* 10.1, p. 81.

Yi, Faliu et al. (2017). "Automatic extraction of cell nuclei from H&E-stained histopathological images". In: *Journal of Medical Imaging* 4.2, pp. 027502–027502.

Yip, Shun H, Pak Chung Sham, and Junwen Wang (2019). "Evaluation of tools for highly variable gene discovery from single-cell RNA-seq data". In: *Briefings in bioinformatics* 20.4, pp. 1583–1589.

Yoon, Young-Kwang et al. (2010). "KRAS mutant lung cancer cells are differentially responsive to MEK inhibitor due to AKT or STAT3 activation: implication for combinatorial approach". In: *Molecular Carcinogenesis: Published in cooperation with the University of Texas MD Anderson Cancer Center* 49.4, pp. 353–362.

You, Yue et al. (2021). "Benchmarking UMI-based single-cell RNA-seq preprocessing workflows". In: *Genome Biology* 22.1, p. 339.

Yu, Guangchuang et al. (2012). "clusterProfiler: an R package for comparing biological themes among gene clusters". In: *Omics: a journal of integrative biology* 16.5, pp. 284–287.

Yu, Lijia et al. (2022). "Benchmarking clustering algorithms on estimating the number of cell types from single-cell RNA-sequencing data". In: *Genome biology* 23.1, p. 49.

Zappia, Luke and Alicia Oshlack (2018). "Clustering trees: a visualization for evaluating clusterings at multiple resolutions". In: *Gigascience* 7.7, giy083.

Zheng, Grace XY et al. (2017). "Massively parallel digital transcriptional profiling of single cells". In: *Nature communications* 8.1, p. 14049.

Acknowledgments

CAPRI: Calcolo ad Alte Prestazioni per la Ricerca e lInnovazione is a project funded by the University of Padua with reference to the BANDO INFRASTRUTTURE STRATEGICHE DI RICERCA (ISR) of 2017.

We acknowledge the CINECA award under the ISCRA initiative, for the availability of high performance computing resources and support.

Appendix

Appendix A

Benchmark SVD algorithm in large scRNA-Seq analysis

Method's Name	SVD Algorithm	Mat type	Library	Software	CPU/GPU	Deferred	Reference
rspectra_sparse_arpack	arpack	sparse	RSpectra	R	CPU	no	Lehoucq, Sorensen, and C. Yang 1998
rapids_dense_exact	exact	dense	Rapids	Python	GPU	no	Kalman 1996
scanpy_sparse_arpack	arpack	sparse	Scanpy	Python	CPU	no	Lehoucq, Sorensen, and C. Yang 1998
rapids_dense_jacobi	jacobi	dense	Rapids	Python	GPU	no	Drma and Veseli 2008
sciktlearn_sparse_IPCA	IPCA	sparse	SciktLearn	Python	CPU	no	Ross et al. 2008
scanpy_dense_arpack	arpack	dense	scanpy	Python	CPU	no	Lehoucq, Sorensen, and C. Yang 1998
bioc_sparse_def_random	random	sparse	BiocSingular	R	CPU	yes	Halko, Martinsson, and Tropp 2011
bioc_sparsearray_def_random	random	sparse	BiocSingular	R	CPU	yes	Halko, Martinsson, and Tropp 2011
scanpy_dense_random	random	dense	Scanpy	Python	CPU	no	Halko, Martinsson, and Tropp 2011
scanpy_sparse_random	random	sparse	Scanpy	Python	CPU	no	Halko, Martinsson, and Tropp 2011
sciktlearn_dense_IPCA	IPCA	dense	SciktLearn	Python	CPU	no	Ross et al. 2008
rspectra_dense_arpack	arpack	dense	RSpectra	R	CPU	no	Lehoucq, Sorensen, and C. Yang 1998
bioc_sparse_def_irlba	irlba	sparse	BiocSingular	R	CPU	yes	Baglama 2016
sciktlearn_dense_exact	exact	dense	SciktLearn	Python	CPU	no	Kalman 1996
bioc_dense_random	random	dense	BiocSingular	R	CPU	no	Halko, Martinsson, and Tropp 2011
bioc_sparse_random	random	dense	biocsingular	R	CPU	no	Halko, Martinsson, and Tropp 2011
bioc_sparsearray_random	random	sparse	BioCsingular	R	CPU	no	Halko, Martinsson, and Tropp 2011
bioc_hdf5_dense_random	random	hdf5	BioCsingular	R	CPU	no	Halko, Martinsson, and Tropp 2011
bioc_sparse_irlba	irlba	sparse	biocsingular	R	CPU	no	Baglama 2016
bioc_sparse_def_exact	exact	sparse	biocsingular	R	CPU	yes	Kalman 1996
bioc_dense_irlba	irlba	dense	BiocSingular	R	CPU	no	Baglama 2016
bioc_sparse_exact	exact	sparse	BiocSingular	R	CPU	no	Kalman 1996
bioc_hdf5_dense_exact	exact	dense	BiocSingular	R	CPU	no	Kalman 1996
bioc_hdf5_dense_irlba	irlba	hdf5	BiocSingular	R	CPU	no	Baglama 2016
bioc_dense_exact	exact	dense	BiocSingular	R	CPU	no	Kalman 1996
bioc_sparsearray_exact	exact	sparse	BiocSingular	R	CPU	no	Kalman 1996
rspectra_hdf5_dense_arpack	arpack	dense	RSpectra	R	CPU	no	Lehoucq, Sorensen, and C. Yang 1998
rspectra_hdf5_sparse_arpack	arpack	sparse	RSpectra	R	CPU	no	Lehoucq, Sorensen, and C. Yang 1998

Table A.1: Comparison of various singular value decomposition (SVD) methods applied to dense or sparse data, specifying the library, programming language, computation type (CPU/GPU), use of deferred computation, and the ranking estimated using the PlackettLuce model. Method names follow a naming convention that encodes the library, data type, and specific SVD algorithm used.

Table A.2: Explained variance percentages for the bioc_dense_random method. Each value represents the proportion of total variance captured by the corresponding singular value component, computed using a randomized SVD algorithm on dense matrices within the BiocSingular framework. (PC1 – PC10)

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
100k	75.28	17.13	1.71	1.15	1.04	0.87	0.51	0.30	0.29	0.22
500k	74.36	17.75	1.78	1.21	1.04	0.92	0.53	0.32	0.29	0.23
1M	74.47	17.72	1.77	1.20	1.03	0.92	0.53	0.32	0.28	0.23
1.3M	74.40	17.76	1.78	1.21	1.03	0.92	0.53	0.32	0.29	0.23

Table A.3: Explained variance percentages for the bioc_dense_random method. Each value represents the proportion of total variance captured by the corresponding singular value component, computed using a randomized SVD algorithm on dense matrices within the BiocSingular framework. (PC11 – PC20)

	PC11	PC12	PC13	PC14	PC15	PC16	PC17	PC18	PC19	PC20
100k	0.19	0.14	0.13	0.08	0.07	0.07	0.06	0.05	0.05	0.04
500k	0.20	0.15	0.13	0.08	0.07	0.07	0.06	0.06	0.05	0.05
1M	0.20	0.15	0.13	0.09	0.08	0.07	0.06	0.06	0.05	0.05
1.3M	0.20	0.15	0.13	0.09	0.08	0.07	0.06	0.06	0.05	0.05

Table A.4: Explained variance percentages for the bioc_dense_random method. Each value represents the proportion of total variance captured by the corresponding singular value component, computed using a randomized SVD algorithm on dense matrices within the BiocSingular framework. (PC21 – PC30)

	PC21	PC22	PC23	PC24	PC25	PC26	PC27	PC28	PC29	PC30
100k	0,04	0,04	0,03	0,03	0,03	0,03	0,03	0,03	0,02	0,02
500k	0,05	0,04	0,04	0,04	0,04	0,03	0,03	0,03	0,03	0,03
1M	0,04	0,04	0,04	0,03	0,03	0,03	0,03	0,03	0,03	0,03
1.3M	0,04	0,04	0,04	0,04	0,03	0,03	0,03	0,03	0,03	0,03

Table A.5: Explained variance percentages for the bioc_dense_random method. Each value represents the proportion of total variance captured by the corresponding singular value component, computed using a randomized SVD algorithm on dense matrices within the BiocSingular framework. (PC31 – PC40)

	PC31	PC32	PC33	PC34	PC35	PC36	PC37	PC38	PC39	PC40
100k	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.01	0.01	0.01
500k	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.01
1M	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.01
1.3M	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.01

Table A.6: Explained variance percentages for the bioc_dense_random method. Each value represents the proportion of total variance captured by the corresponding singular value component, computed using a randomized SVD algorithm on dense matrices within the BiocSingular framework. (PC41 – PC50)

	PC41	PC42	PC43	PC44	PC45	PC46	PC47	PC48	PC49	PC50
100k	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
500k	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
1M	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
1.3M	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01

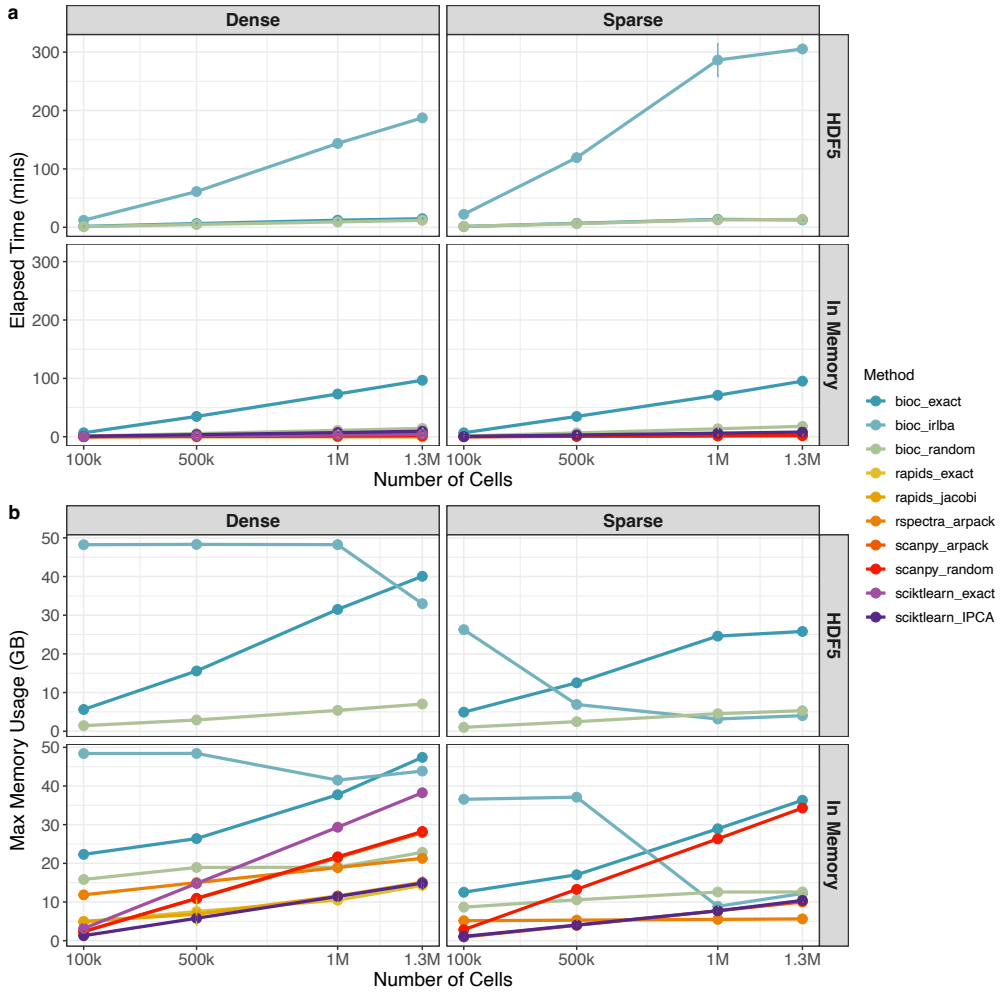

Figure A.1: **Scalability Assessment of PCA Methods by Input Dimensions, Runtime, and Memory Consumption** (a) Elapsed time (in minutes) required to perform principal component analysis (PCA) across a range of dataset sizes (100k, 500k, 1M, 1.3M cells), using different combinations of methods, matrix formats (dense or sparse), and storage types (in-memory or HDF5). (b) Maximum memory usage (in GB) required to perform principal component analysis (PCA) across a range of dataset sizes (100k, 500k, 1M, 1.3M cells), using different combinations of methods, matrix formats (dense or sparse), and storage types (in-memory or HDF5).

Table results PCA benchmark

Table A.7: Computational Time for PCA benchmark for all methods presented in the work.

Methods	n cells	Algorithms	media_time	sd_time	Matrix type	core_eng	soft
bioc_hdf5_dense	100k	random	108.35	12.91	dense matrix	CPU	R
bioc_hdf5_dense	100k	exact	333.85	9.01	dense matrix	CPU	R
bioc_hdf5_dense	100k	irlba	676.50	32.81	dense matrix	CPU	R
rapids_dense	100k	exact	3.00	0.00	dense matrix	GPU	python
rapids_dense	100k	jacobi	3.00	0.00	dense matrix	GPU	python
scanpy_sparse	100k	random	6.20	0.42	sparse matrix	CPU	python
scanpy_sparse	100k	arpack	8.10	0.32	sparse matrix	CPU	python
sciktlearn_sparse	100k	IPCA	23.90	1.10	sparse matrix	CPU	python
rspectra_dense	100k	arpack	33.81	2.64	dense matrix	CPU	R
rspectra_sparse	100k	arpack	4.76	0.05	sparse matrix	CPU	R
sciktlearn_dense	100k	exact	35.80	2.35	dense matrix	CPU	python
bioc_dense	100k	random	62.59	0.41	dense matrix	CPU	R
bioc_dense	100k	exact	320.42	5.15	dense matrix	CPU	R
bioc_dense	100k	irlba	23.47	0.11	dense matrix	CPU	R
bioc_sparse_def	100k	random	8.65	0.03	sparse matrix	CPU	R
bioc_sparse_def	100k	exact	315.40	3.18	sparse matrix	CPU	R
bioc_sparse_def	100k	irlba	11.59	0.08	sparse matrix	CPU	R
bioc_sparse	100k	random	69.36	0.57	sparse matrix	CPU	R
bioc_sparse	100k	exact	319.23	5.57	sparse matrix	CPU	R
bioc_sparse	100k	irlba	11.72	0.08	sparse matrix	CPU	R
scanpy_dense	100k	random	7.10	0.57	dense matrix	CPU	python
scanpy_dense	100k	arpack	14.90	2.42	dense matrix	CPU	python
sciktlearn_dense	100k	IPCA	37.60	1.51	dense matrix	CPU	python
bioc_hdf5_sparse	100k	random	92.71	4.27	sparse matrix	CPU	R
bioc_hdf5_sparse	100k	exact	80.33	2.37	sparse matrix	CPU	R
bioc_hdf5_sparse	100k	irlba	1346.71	68.95	sparse matrix	CPU	R
rspectra_hdf5_dense	100k	arpack	1259.35	0.52	dense matrix	CPU	R
rspectra_hdf5_sparse	100k	arpack	2000.26	0.00	sparse matrix	CPU	R

Table A.8: Computational Time for PCA benchmark for all methods presented in the work. (100k)

Methods	n cells	algorithm	media_time	sd_time	matrix type	core_eng	soft
bioc_hdf5_dense	500k	random	548.66	12.59	dense matrix	CPU	R
bioc_hdf5_dense	500k	exact	1687.71	72.66	dense matrix	CPU	R
bioc_hdf5_dense	500k	irlba	5506.23	214.87	dense matrix	CPU	R
rapids_dense	500k	exact	4.10	0.32	dense matrix	GPU	python
rapids_dense	500k	jacobi	4.00	0.00	dense matrix	GPU	python
scanpy_sparse	500k	random	33.70	1.95	sparse matrix	CPU	python
scanpy_sparse	500k	arpack	40.20	0.42	sparse matrix	CPU	python
sciktlearn_sparse	500k	IPCA	117.00	5.73	sparse matrix	CPU	python
rspectra_dense	500k	arpack	179.49	15.27	dense matrix	CPU	R
rspectra_sparse	500k	arpack	23.63	0.20	sparse matrix	CPU	R
sciktlearn_dense	500k	exact	192.00	8.07	dense matrix	CPU	python
bioc_dense	500k	random	309.98	1.37	dense matrix	CPU	R
bioc_dense	500k	exact	1664.83	9.19	dense matrix	CPU	R
bioc_dense	500k	irlba	126.88	0.60	dense matrix	CPU	R
bioc_sparse_def	500k	random	45.13	0.21	sparse matrix	CPU	R
bioc_sparse_def	500k	exact	1596.32	8.62	sparse matrix	CPU	R
bioc_sparse_def	500k	irlba	51.33	0.34	sparse matrix	CPU	R
bioc_sparse	500k	random	395.89	3.38	sparse matrix	CPU	R
bioc_sparse	500k	exact	1654.41	15.08	sparse matrix	CPU	R
bioc_sparse	500k	irlba	52.10	0.39	sparse matrix	CPU	R
scanpy_dense	500k	random	39.30	1.83	dense matrix	CPU	python
scanpy_dense	500k	arpack	85.80	11.79	dense matrix	CPU	python
sciktlearn_dense	500k	IPCA	197.80	1.99	dense matrix	CPU	python
bioc_hdf5_sparse	500k	random	381.18	17.72	sparse matrix	CPU	R
bioc_hdf5_sparse	500k	exact	400.42	12.33	sparse matrix	CPU	R
bioc_hdf5_sparse	500k	irlba	7142.68	182.73	sparse matrix	CPU	R
rspectra_hdf5_dense	500k	arpack	6213.26	7.51	dense matrix	CPU	R
rspectra_hdf5_sparse	500k	arpack	10754.67	0.00	sparse matrix	CPU	R

Table A.9: Computational Time for PCA benchmark for all methods presented in the work. (500k)

Methods	n cells	Algorithm	media_time	sd_time	MAtrix type	core_eng	soft
bioc_hdf5_dense	1M	random	1147.51	30.52	dense matrix	CPU	R
bioc_hdf5_dense	1M	exact	3440.95	174.00	dense matrix	CPU	R
bioc_hdf5_dense	1M	irlba	12680.44	335.62	dense matrix	CPU	R
rapids_dense	1M	exact	5.00	0.00	dense matrix	GPU	python
rapids_dense	1M	jacobi	5.20	0.42	dense matrix	GPU	python
scanpy_sparse	1M	random	69.00	3.74	sparse matrix	CPU	python
scanpy_sparse	1M	arpack	82.00	0.47	sparse matrix	CPU	python
sciktlearn_sparse	1M	IPCA	222.30	7.01	sparse matrix	CPU	python
rspectra_dense	1M	arpack	376.11	31.68	dense matrix	CPU	R
rspectra_sparse	1M	arpack	47.41	0.46	sparse matrix	CPU	R
sciktlearn_dense	1M	exact	411.50	69.64	dense matrix	CPU	python
bioc_dense	1M	random	631.87	1.97	dense matrix	CPU	R
bioc_dense	1M	exact	3356.71	5.58	dense matrix	CPU	R
bioc_dense	1M	irlba	271.36	1.51	dense matrix	CPU	R
bioc_sparse_def	1M	random	94.47	0.35	sparse matrix	CPU	R
bioc_sparse_def	1M	exact	3226.88	22.44	sparse matrix	CPU	R
bioc_sparse_def	1M	irlba	163.69	0.95	sparse matrix	CPU	R
bioc_sparse	1M	random	925.95	5.16	sparse matrix	CPU	R
bioc_sparse	1M	exact	3343.25	22.82	sparse matrix	CPU	R
bioc_sparse	1M	irlba	163.28	0.31	sparse matrix	CPU	R
scanpy_dense	1M	random	81.70	2.87	dense matrix	CPU	python
scanpy_dense	1M	arpack	205.80	13.26	dense matrix	CPU	python
sciktlearn_dense	1M	IPCA	369.10	16.38	dense matrix	CPU	python
bioc_hdf5_sparse	1M	random	771.28	25.33	sparse matrix	CPU	R
bioc_hdf5_sparse	1M	exact	785.08	33.18	sparse matrix	CPU	R
bioc_hdf5_sparse	1M	irlba	17663.29	961.17	sparse matrix	CPU	R
rspectra_hdf5_dense	1M	arpack	13034.39	126.77	dense matrix	CPU	R
rspectra_hdf5_sparse	1M	arpack	22256.76	0.00	sparse matrix	CPU	R

Table A.10: Computational Time for PCA benchmark for all methods presented in the work. (1M)

Methods	n cells	Algorithm	media_time	sd_time	Matrix type	core_eng	soft
bioc_hdf5_dense	1.3M	random	1515.55	47.78	dense matrix	CPU	R
bioc_hdf5_dense	1.3M	exact	4513.47	250.32	dense matrix	CPU	R
bioc_hdf5_dense	1.3M	irlba	17995.83	436.47	dense matrix	CPU	R
rapids_dense	1.3M	exact	7.50	0.53	dense matrix	GPU	python
rapids_dense	1.3M	jacobi	8.00	0.00	dense matrix	GPU	python
scanpy_sparse	1.3M	random	93.40	4.70	sparse matrix	CPU	python
scanpy_sparse	1.3M	arpack	106.60	0.97	sparse matrix	CPU	python
sciktlearn_sparse	1.3M	IPCA	292.40	9.88	sparse matrix	CPU	python
rspectra_dense	1.3M	arpack	467.73	25.02	dense matrix	CPU	R
rspectra_sparse	1.3M	arpack	61.96	0.92	sparse matrix	CPU	R
sciktlearn_dense	1.3M	exact	546.20	69.81	dense matrix	CPU	python
bioc_dense	1.3M	random	855.92	2.42	dense matrix	CPU	R
bioc_dense	1.3M	exact	4423.35	6.07	dense matrix	CPU	R
bioc_dense	1.3M	irlba	372.98	1.05	dense matrix	CPU	R
bioc_sparse_def	1.3M	random	122.01	0.63	sparse matrix	CPU	R
bioc_sparse_def	1.3M	exact	4281.73	22.22	sparse matrix	CPU	R
bioc_sparse_def	1.3M	irlba	173.29	1.22	sparse matrix	CPU	R
bioc_sparse	1.3M	random	1293.50	6.83	sparse matrix	CPU	R
bioc_sparse	1.3M	exact	4379.43	22.55	sparse matrix	CPU	R
bioc_sparse	1.3M	irlba	174.36	0.39	sparse matrix	CPU	R
scanpy_dense	1.3M	random	109.40	4.86	dense matrix	CPU	python
scanpy_dense	1.3M	arpack	263.70	16.61	dense matrix	CPU	python
sciktlearn_dense	1.3M	IPCA	520.80	18.94	dense matrix	CPU	python
bioc_hdf5_sparse	1.3M	random	811.56	11.06	sparse matrix	CPU	R
bioc_hdf5_sparse	1.3M	exact	763.67	18.25	sparse matrix	CPU	R
bioc_hdf5_sparse	1.3M	irlba	18659.81	1360.73	sparse matrix	CPU	R
rspectra_hdf5_dense	1.3M	arpack	17218.16	176.35	dense matrix	CPU	R
rspectra_hdf5_sparse	1.3M	arpack	22308.97	0.00	sparse matrix	CPU	R

Table A.11: Computational Time for PCA benchmark for all methods presented in the work. (1.3M)

Size	Methods	Unoptimized Elapsed Time Medium03 R	Optimized Elapsed Time Medium03 R	xen6 Unoptimized	xen6 Optimized
100k	bioc_exact_dense	156.14	16.255	408	11.876
500k	bioc_exact_dense	858.491	88.155	2088	76.858
1M	bioc_exact_dense	1655.495	165.451	4392	151.293
1.3M	bioc_exact_dense	2125.143	233.62	5916	197.526

Table A.12: Computation times (in seconds) for exact SVD using different LAPACK/BLAS configurations in R.

Appendix B

Comparison scRNA-Seq workflow

Table B.1: Computational time for each scRNA-seq workflow and each database in input

	step	Seurat	Bioc	Scanpy	Rapids	scrapper
1.3M	find mitochondrial gene	15,544	0,47	98,8152	4,685	1010,838
	filter	139,8	637,84	209,1486	272,8392	
	normalization	62,4	2.674,48	47,0995	60,63093333	0,1137
	hgv	69	3.038,39	55,6003	90,9464	1393,488
	scaling	67,44	0	15,2298	11,978	0
	PCA	7344	1.350,96	85,1788	43,69	536,052
	t-sne	5472	0,00	4513,9915	101,72	13638,96
	umap	1180,8	1.200,00	1766,2184	350,452	1581,12
	louvain	2657,4		572,534	388,746	
	leiden	2636,16	480,00	745,631	392,798	1139,4
cb	find mitochondrial gene	0,9603	8,9845	1,4657	0	0,8605
	filter	1,6499		1,7012	0,455	0
	normalization	0,9406	15,3187	1,4122	0,006	0,0613
	hgv	1,3899	0,6372	1,2622	0,488	1,1154
	scaling	9,4327		0,4977	0,043	0
	PCA	3,0979	9,4448	0,365	0,565	0,763
	t-sne	24,4497	48,0701	24,1626	0,952	17,6854
	umap	17,5386	15,6646	20,581	5,36	11,6143
	louvain	4,5575	14,8174	0,5714	2,052	0
	leiden	4,7349	8,7005	0,4191	0,097	1,7955
sc_mixology	find mitochondrial gene	0,7783	9,211	0,8896	0	2,2526
	filter	1,4675		1,5183	1,087	0
	normalization	0,4666	3,9137	0,2358	0,006	0,0584
	hgv	0,6548	0,6936	0,1535	0,1178	0,596
	scaling	0,9108	0	0,2961	0,04	
	PCA	0,4304	3,9362	0,2058	0,075	0,3807
	t-sne	2,8912	22,3044	2,105	0,045	7,0366
	umap	4,5184	8,5114	9,1539	2,7556	5,2801
	louvain	0,7204	2,0218	0,0268	2,965	0
	leiden	0,6141	1,1177	0,1284	0,089	0,6106
BE1	find mitochondrial gene	4,4057	11,53065	6,819	1,8119	0,6141
	filter	6,1036	0,049	19,8166	4,8015	0
	normalization	4,8742	7,5891	11,3122	1,1635	1,3935
	hgv	10,0936	0,8263	10,3254	0,375	1,263
	scaling	93,576	0	7,7532	1,865	
	PCA	7,7419	52,245	11,1524	2,5294	4,8703
	t-sne	128,982	182,4	102,7969	3,8564	61,326
	umap	46,1377	29,2787	67,1117	6,1052	30,6672
	louvain	24,0327	103,956	5,1166	1,0125	0
	leiden	25,3245	29,7326	2,8551	0,7038	6,4761

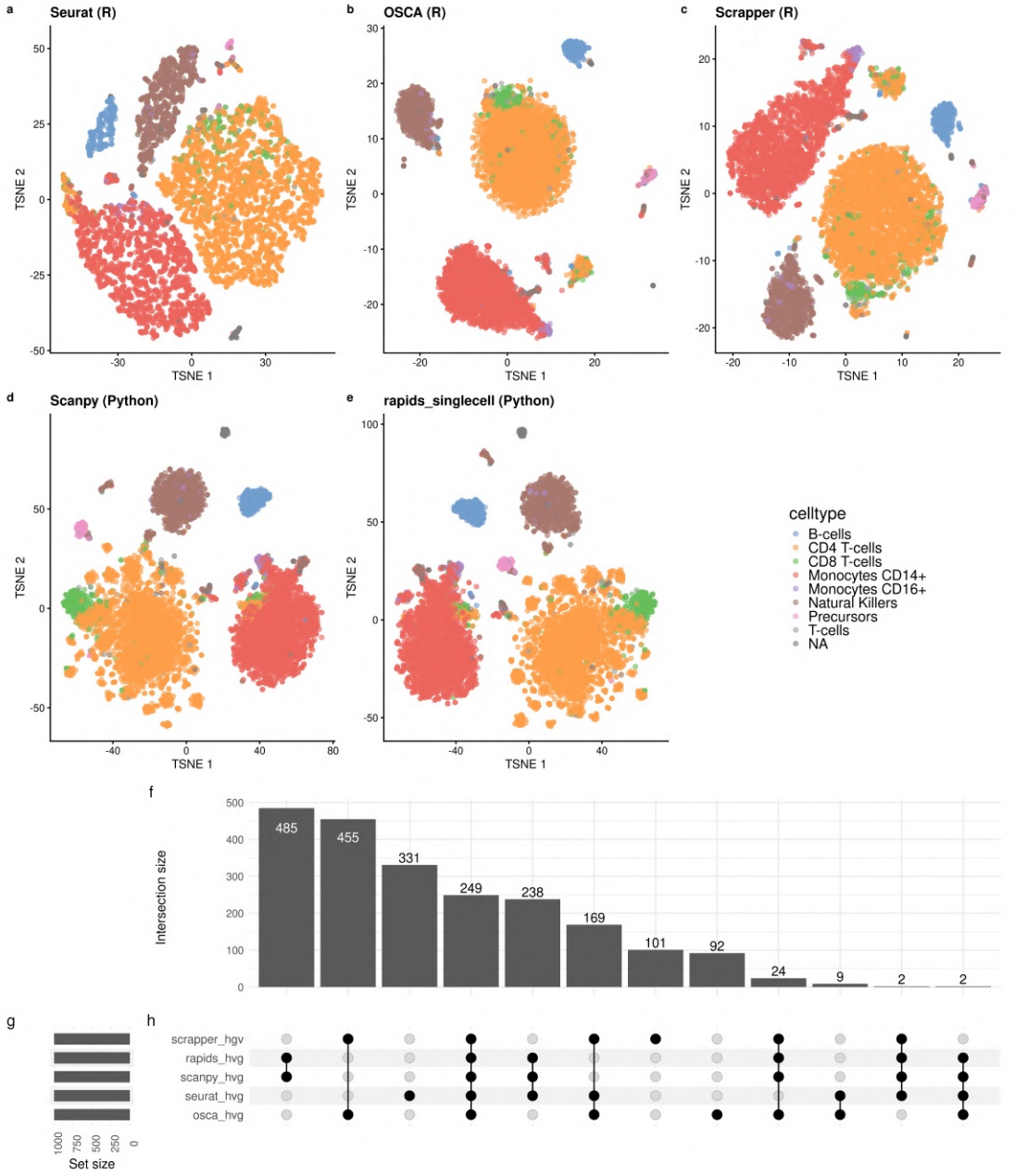

Figure B.1: **T-SNE plot and HVGs in the cb dataset.** (ae) t-SNE embeddings of the cb dataset colored by sample identity, generated using five different single-cell workflows: Seurat (a), OSCA (b), Scrapper (c), Scanpy (d), and rapids_singlecell (e). Each workflow applies its own normalization and highly variable gene (HVG) selection procedure prior to dimensionality reduction. (fh) UpSet plot showing the intersection of HVG sets selected by each workflow. Panel (f) indicates the size of each intersection set, (g) shows the number of genes selected per method (set size), and (h) depicts the overlap structure across methods.

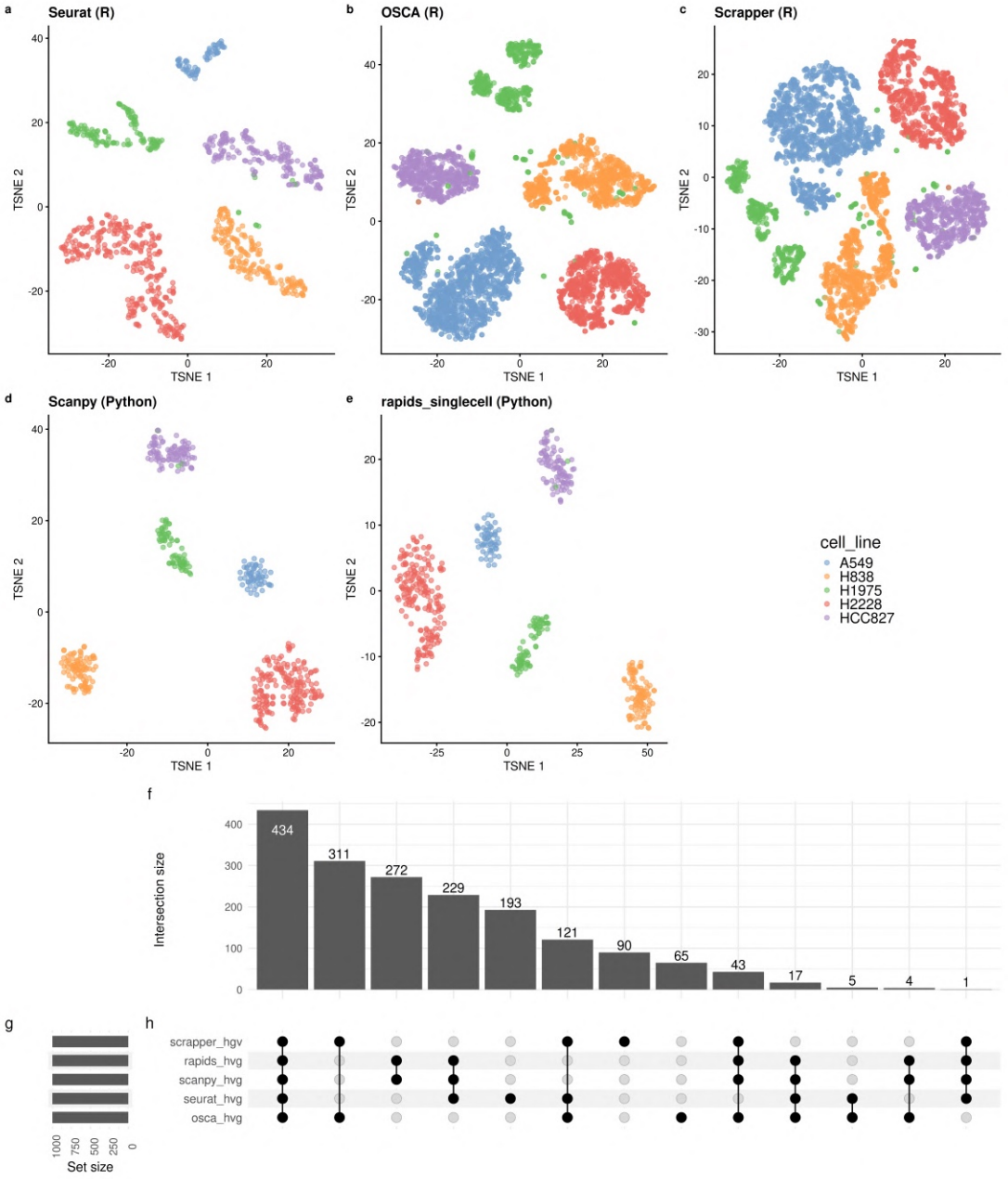

Figure B.2: **T-SNE plot and HVGs in the sc_mixology dataset.** (ae) t-SNE embeddings of the BE1 dataset colored by cell line, generated using five different single-cell workflows: Seurat (a), OSCA (b), Scrapper (c), Scanpy (d), and rapids_singlecell (e). Each workflow applies its own normalization and highly variable gene (HVG) selection procedure prior to dimensionality reduction. (fh) UpSet plot showing the intersection of HVG sets selected by each workflow. Panel (f) indicates the size of each intersection set, (g) shows the number of genes selected per method (set size), and (h) depicts the overlap structure across methods.

Appendix C

Case Study TCGA-OV

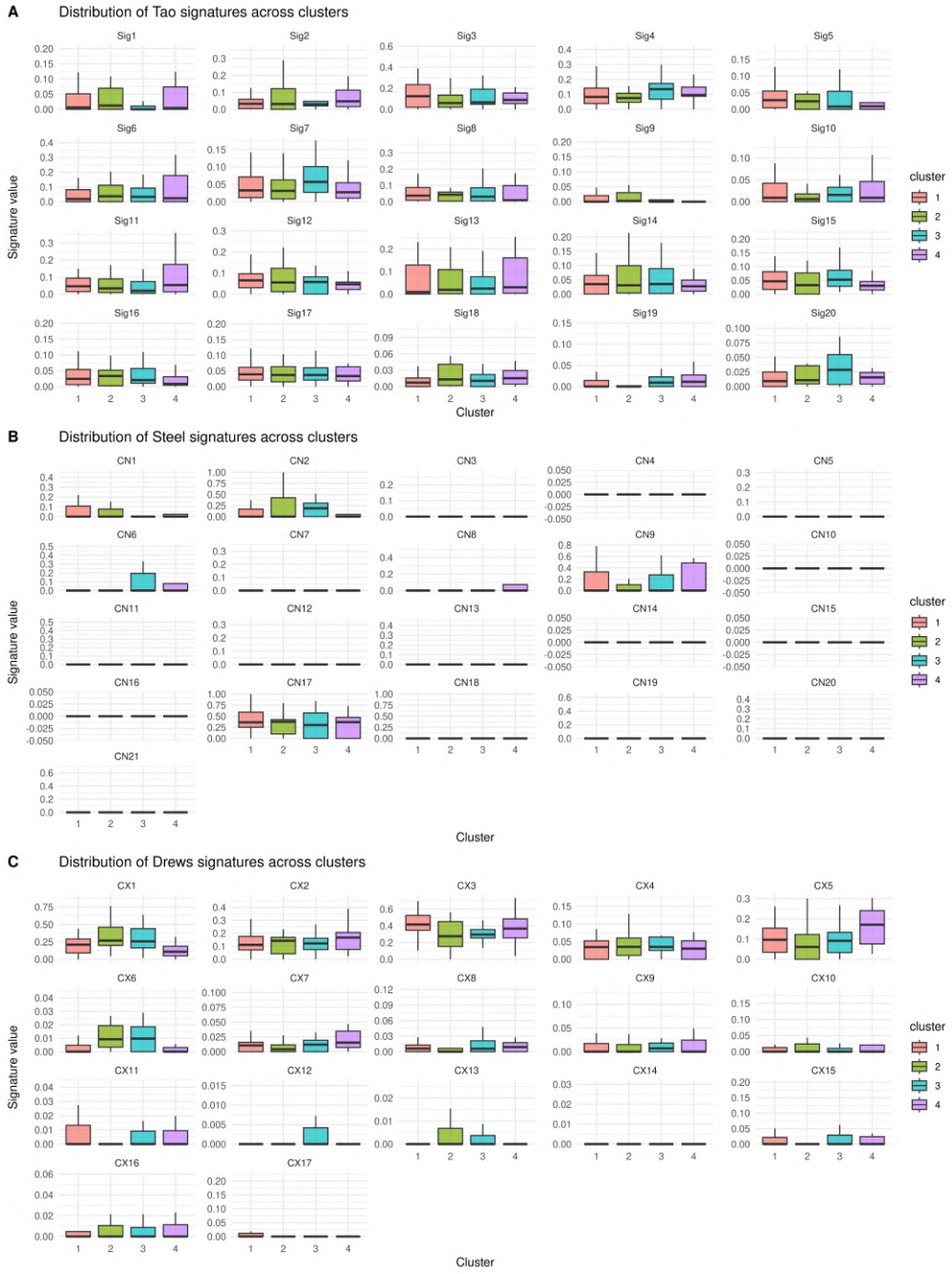

Figure C.1: Distribution of CNA signature activities across clusters, shown as boxplots for Steele, Tao, and Drews signature sets. Clusters were obtained from histopathological image embeddings.

ID	Description	GeneRatio	BgRatio	RichFactor	FoldEnrichmentScore	pvalue	p.adjust	qvalue	Count
GO:0019724	B cell mediated immunity	33/1143	8/19519	0.185	2.858	<0.001	0.001	0.001	33
GO:0002455	immunoglobulin mediated immune response	31/1143	7/19519	0.174	2.984	<0.001	0.001	0.001	31
GO:0072676	lymphocyte migration	17/1143	5/19519	0.149	2.957	0.002	0.009	0.008	17
GO:0002449	lymphocyte mediated immunity	15/1143	4/19519	0.137	3.019	0.005	0.019	0.015	15
GO:0002459	histamine transport	15/1143	4/19519	0.137	3.678	0.005	0.020	0.015	15
GO:0002447	histamine transport	14/1143	3/19519	0.124	3.196	0.007	0.022	0.017	14
GO:0002448	humoral immune response	14/1143	2/19519	0.124	3.438	0.008	0.023	0.018	14
GO:0001888	impulsivity control	14/1143	3/19519	0.124	3.043	0.009	0.024	0.019	14
GO:0045071	negative regulation of immune response	12/1143	5/19519	0.105	2.619	0.012	0.032	0.024	12
GO:0030266	histamine transport	11/1143	2/19519	0.096	2.494	0.019	0.033	0.026	11
GO:0050900	histamine transport	11/1143	1/19519	0.096	2.978	0.021	0.034	0.027	11
GO:0005159	somatotropin receptor binding	10/1143	3/19519	0.087	2.654	0.023	0.036	0.028	10

Table C.1: Immune-related GO Biological Process terms significantly enriched in Cluster 1 vs Cluster 2. GSE Biological Process.

category	subcategory	ID	Description	GeneRatio	BgRatio	RichFactor	FoldEnrich	mzScore	pvalue	p.adjust	qvalue	Count
Cellular Processes	Transport and catabolism	hsa04145	Phagosome	39/1278	146/7129	0.267	1.490	2.796	0.005	0.117	0.105	39
Organismal Systems	Immune system	hsa04613	Neutrophil extracellular trap formation	42/1278	161/7129	0.261	1.455	2.730	0.006	0.117	0.105	42

Table C.2: Immune-related KEGG pathways significantly enriched in Cluster 1 vs Cluster 2 (GSE analysis).

GO ID	Description	setSize	enrichmentScore	NES	pvalue	p.adjust	qvalue
GO:0016064	immunoglobulin mediated immune response	173	-0.3625	-1.8267	<0.001	<0.001	<0.001
GO:0019724	B cell mediated immunity	176	-0.3576	-1.8085	<0.001	<0.001	<0.001
GO:0002443	leukocyte mediated immunity	417	-0.2730	-1.5435	<0.001	<0.001	<0.001
GO:0071674	mononuclear cell migration	202	-0.3087	-1.5876	<0.001	<0.001	<0.001
GO:0002768	immune response-regulating cell surface receptor signaling pathway	335	-0.2586	-1.4228	<0.001	<0.001	<0.001
GO:0002449	lymphocyte mediated immunity	328	-0.2614	-1.4369	<0.001	<0.001	<0.001
GO:0050900	leukocyte migration	353	-0.2567	-1.4225	<0.001	<0.001	<0.001
GO:0002460	adaptive immune response (somatic recombination)	346	-0.2615	-1.4516	<0.001	<0.001	<0.001
GO:0002761	regulation of myeloid leukocyte differentiation	111	-0.3390	-1.5748	<0.001	<0.001	<0.001
GO:0071675	regulation of mononuclear cell migration	134	-0.3184	-1.5262	<0.001	<0.001	<0.001
GO:0097529	myeloid leukocyte migration	211	-0.2875	-1.4803	<0.001	<0.001	<0.001
GO:1902105	regulation of leukocyte differentiation	308	-0.2523	-1.3728	<0.001	0.0107	<0.001
GO:0002274	myeloid leukocyte activation	223	-0.2766	-1.4384	<0.001	0.0111	<0.001
GO:0030217	T cell differentiation	302	-0.2482	-1.3484	<0.001	0.0155	<0.001
GO:0030595	leukocyte chemotaxis	206	-0.2693	-1.3893	<0.001	0.0234	<0.001
GO:0002520	immune system development	187	-0.2734	-1.3911	<0.001	0.0244	<0.001
GO:0002685	regulation of leukocyte migration	220	-0.2598	-1.3429	<0.001	0.0306	<0.001
GO:0001776	leukocyte homeostasis	108	-0.3130	-1.4448	<0.001	0.0309	<0.001
GO:0002688	regulation of leukocyte chemotaxis	118	-0.3015	-1.4113	<0.001	0.0340	<0.001
GO:0030098	lymphocyte differentiation	403	-0.2228	-1.2523	<0.001	0.0402	<0.001
GO:0097530	granulocyte migration	136	-0.2792	-1.3409	<0.001	0.0451	<0.001

Table C.3: GSEA results for immune-related biological processes (BP). Cluster 1 vs Cluster 2

ID	Description	setSize	enrichment	NES	pvalue	p.adjust	qvalue	rank
GO:0000724	double-strand break repair via homologous recombination	182	0.251	1.275	0.033	0.045	0.011	5241

Table C.4: GSEA results for HR-related biological processes (BP). Cluster 1 vs Cluster 2

ID	Description	setSize	enrichmentNES	pvalue	p.adjust	qvalue	rank	score
hsa04060	Cytokine-cytokine receptor interaction	228	-2.031	<0.001	<0.001	<0.001	1205	4663
hsa01434	Leukocyte transendothelial migration	104	-2.037	<0.001	<0.001	<0.001	1087	3247
hsa04145	Phagosome	134	-2.049	<0.001	<0.001	<0.001	1015	3156
hsa04660	T-cell receptor signaling pathway	111	-2.049	<0.001	<0.001	<0.001	1006	5175
hsa04750	Inflammatory mediator regulation of TRP channels	94	-1.984	<0.001	<0.001	<0.001	1832	5212
hsa04662	B cell receptor signaling pathway	101	-1.952	<0.001	<0.001	<0.001	2313	5735
hsa04072	Phospholipase D signaling pathway	146	-1.922	<0.001	<0.001	<0.001	2780	4416
hsa04390	Hippo signaling pathway	139	-1.892	<0.001	<0.001	<0.001	3232	5231
hsa04920	Adipocytokine signaling pathway	66	-1.857	<0.001	<0.001	<0.001	3683	5214
hsa04610	Complement and coagulation cascades	67	-1.875	<0.001	<0.001	<0.001	3412	4913
hsa04512	ECM-receptor interaction	84	-1.826	<0.001	<0.001	<0.001	4171	4855
hsa05131	Shigellosis	64	-1.804	<0.001	<0.001	<0.001	4284	4522
hsa05321	Inflammatory bowel disease	64	-1.929	<0.001	<0.001	<0.001	2715	4733
hsa04064	Viral protein interaction with cytokine and cytokine receptor	113	-1.832	<0.001	<0.001	<0.001	4058	3760
hsa04672	Intestinal immune network for IgA production	43	-1.617	0.007	0.032	0.021	4501	4354
hsa05320	Autoimmune thyroid disease	33	-1.477	0.047	0.087	0.061	4303	4534

Table C.5: KEGG pathways enriched for immune-related pathway (GSEA). Cluster 1 vs Cluster 2

ID	Description	setSize	enrichment	NES	pvalue	p.adjust	qvalue	rank
hsa03440	Homologous recombination	40	0.363	1.379	0.072	0.126	0.061	4868

Table C.6: KEGG pathways enriched for HR-related pathway (GSEA). Cluster 1 vs Cluster 2

ID	Description	setSize	enrichment	NES	p-value	p.adjust	qvalue	rank
R-HSA-168249	Innate Immune System	914	3.046	<0.001	<0.001	<0.001	4801	4801
R-HSA-1280215	Adaptive Immune System	714	3.045	<0.001	<0.001	<0.001	4800	4800
R-HSA-198933	Antigen Presentation: folding	37	-2.317	<0.001	0.013	0.008	4232	4232
hsa046268	TNF receptor by EDA2R family	7	-3.442	0.007	0.013	0.011	1494	1503
R-HSA-190837	Anti-inflammatory response	5	1.752	0.007	0.011	0.009	823	833
R-HSA-6806664	Cytokine Signaling in Immune system	10	-2.843	0.006	0.009	0.009	1	9
hsa04672	sumoylation	18	2.199	0.009	0.020	0.017	8	17
R-HSA-6785807	Antigen processing, Ub-proteasome	13	1.888	0.014	0.022	0.020	22	31
R-HSA-1280218	TLR3/7/8 cascade	5	2.156	0.013	0.023	0.017	7	16
R-HSA-1280219	TLR4 cascade	20	2.673	0.013	0.023	0.017	8	17
R-HSA-168938	TLR-12 cascade	16	-4.142	0.015	0.024	0.019	850	859
hsa04640	SAR-CoV virion	23	0.547	0.016	0.025	0.020	869	878
REAC:678-3108	Antigen Presentation	9	0.015	0.016	0.026	0.021	870	880
reac:2028941	CD209 signaling	9	0.679	0.019	0.031	0.027	81	90
REAC:R-HSA-198935	ACOD1 mediates	39	0.277	0.019	0.033	0.028	117	126

Table C.7: Reactome pathways enriched for immune-related processes. Cluster 1 vs Cluster 2

ID	Description	setSize	enrichmentNES	pValue	pAdjust	qValue	rank
R-HSA-5693533	DNA Double-Strand Break Repair	39	3.234	<0.001	<0.001	<0.001	1880
R-HSA-174417	Inhibition of DNA recombination at telomeres	13	3.182	<0.001	<0.001	<0.001	6886
R-HSA-5693568	HDR through Homologous Recombination (HRR) or Single Strand Annealing (SSA)	28	3.155	<0.001	<0.001	<0.001	1972
R-HSA-5334118	Homologous DNA DSB repair	21	3.147	<0.001	<0.001	<0.001	589
R-HSA-5693538	Processing of DNA double-strand break ends	37	3.123	<0.001	<0.001	<0.001	557
R-HSA-5633007	HDR through Homologous Recombination (HRR)	31	3.120	<0.001	<0.001	<0.001	1679
R-HSA-73933	DNA Repair	203	2.976	<0.001	<0.001	<0.001	3475
R-HSA-5693548	HDR through Single Strand Annealing (SSA)	12	2.805	<0.001	<0.001	<0.001	5875
R-HSA-5655862	Fanconi Anemia Pathway	53	2.728	<0.001	<0.001	<0.001	5876
R-HSA-67428	Gap-filling DNA repair synthesis and ligation in GG-NER	25	2.619	0.026	0.038	0.036	7356

Table C.8: Reactome pathways enriched for HR-related processes. Cluster 1 vs Cluster 2

ID	Description	setSize	enrichmentScore	NES	pvalue	p.adjust	qvalue	rank
GO:0016064	immunoglobulin mediated immune response	173	-0.362	-1.827	<0.001	<0.001	<0.001	4004
GO:0019724	B cell mediated immunity	176	-0.358	-1.809	<0.001	<0.001	<0.001	4004
GO:0002443	leukocyte mediated immunity	417	-0.273	-1.544	<0.001	<0.001	<0.001	4004
GO:0071674	mononuclear cell migration	202	-0.309	-1.588	<0.001	0.002	0.001	3500
GO:0002768	immune response-regulating cell surface receptor signaling pathway	335	-0.259	-1.423	<0.001	0.002	0.001	3771
GO:0002449	lymphocyte mediated immunity	328	-0.261	-1.437	0.001	0.003	0.002	4004
GO:0050900	leukocyte migration	353	-0.257	-1.423	0.001	0.003	0.002	4757
GO:0002460	adaptive immune response based on somatic recombination of immune receptors built from immunoglobulin superfamily domains	346	-0.261	-1.452	0.001	0.006	0.003	4004
GO:0002761	regulation of myeloid leukocyte differentiation	111	-0.339	-1.575	0.002	0.007	0.004	3463
GO:0071675	regulation of mononuclear cell migration	134	-0.318	-1.526	0.002	0.008	0.004	3017
GO:0097529	myeloid leukocyte migration	211	-0.287	-1.480	0.002	0.009	0.005	4670
GO:1902105	regulation of leukocyte differentiation	308	-0.252	-1.373	0.003	0.011	0.006	4600
GO:0002274	myeloid leukocyte activation	223	-0.277	-1.438	0.003	0.011	0.007	4480
GO:0030217	T cell differentiation	302	-0.248	-1.348	0.004	0.016	0.009	2986
GO:0030595	leukocyte chemotaxis	206	-0.269	-1.389	0.007	0.023	0.014	3017
GO:0002520	immune system development	187	-0.273	-1.391	0.007	0.024	0.014	4009
GO:0002685	regulation of leukocyte migration	220	-0.260	-1.343	0.010	0.031	0.018	3017
GO:0001776	leukocyte homeostasis	108	-0.313	-1.445	0.010	0.031	0.018	3995
GO:0002688	regulation of leukocyte chemotaxis	118	-0.302	-1.411	0.011	0.034	0.020	3017
GO:0030098	lymphocyte differentiation	403	-0.223	-1.252	0.014	0.040	0.024	4579
GO:0097530	granulocyte migration	136	-0.279	-1.341	0.016	0.045	0.026	4465

Table C.9: Immune-related GO Biological Process terms significantly enriched in Cluster 1 vs Cluster 3. GSEA Biological Process.

ID	Description	setSize	enrichmentScore	NES	pvalue	p.adjust	qvalue	rank
hsa04920	Adipocytokine signaling pathway	67	-0.411	-1.772	0.001	0.005	0.003	3771
hsa04660	T cell receptor signaling pathway	114	-0.347	-1.671	0.001	0.005	0.004	3939
hsa04670	Leukocyte transendothelial migration	104	-0.348	-1.631	0.001	0.010	0.007	4633
hsa04662	B cell receptor signaling pathway	80	-0.375	-1.677	0.002	0.012	0.008	4151
hsa04145	Phagosome	146	-0.279	-1.387	0.012	0.052	0.037	3670
hsa04750	Inflammatory mediator regulation of TRP channels	93	-0.298	-1.375	0.027	0.096	0.068	4151
hsa04620	Toll-like receptor signaling pathway	90	-0.311	-1.429	0.030	0.104	0.074	4151

Table C.10: KEGG pathways enriched for Immune-related pathway (GSEA). Cluster 1 vs Cluster 3

ID	Description	setSize	enrichmentScore	NES	pvalue	p.adjust	qvalue	rank
hsa03460	Fanconi anemia pathway	50	0.389	1.458	0.041	0.132	0.094	5991

Table C.11: KEGG pathways enriched for HR-related pathway (GSEA). Cluster 1 vs Cluster 3

ID	Description	setSize	enrichmentScore	NES	pvalue	p.adjust	qvalue	rank
R-HSA-9912633	Antigen processing: Ub, ATP-independent proteasomal degradation	19	0.778	2.310	<0.001	<0.001	<0.001	3063
R-HSA-1236978	Cross-presentation of soluble exogenous antigens (endosomes)	41	0.579	2.070	<0.001	<0.001	<0.001	3063
R-HSA-1236974	ER-Phagosome pathway	79	0.427	1.740	0.001	0.005	0.004	3506
R-HSA-168249	Innate Immune System	914	-0.206	-1.230	0.001	0.010	0.007	4461
R-HSA-1169091	Activation of NF-kappaB in B cells	51	0.456	1.710	0.003	0.020	0.014	4058
R-HSA-168898	Toll-like Receptor Cascades	163	-0.289	-1.460	0.003	0.021	0.015	3989
R-HSA-9692916	SARS-CoV-1 activates/modulates innate immune responses	39	0.491	1.740	0.004	0.021	0.016	3617
R-HSA-1168372	Downstream signaling events of B Cell Receptor (BCR)	67	0.413	1.630	0.004	0.025	0.018	4254
R-HSA-1236975	Antigen processing-Cross presentation	94	0.378	1.580	0.006	0.034	0.025	3506
R-HSA-9662851	Anti-inflammatory response favouring Leishmania parasite infection	74	-0.347	-1.540	0.008	0.041	0.030	3901
R-HSA-5260271	Diseases of Immune System	29	-0.469	-1.680	0.010	0.052	0.038	3983
R-HSA-168188	Toll Like Receptor TLR6:TLR2 Cascade	109	-0.293	-1.390	0.018	0.079	0.058	3983
R-HSA-168179	Toll Like Receptor TLR1:TLR2 Cascade	110	-0.291	-1.380	0.021	0.090	0.066	3983
R-HSA-181438	Toll Like Receptor 2 (TLR2) Cascade	110	-0.291	-1.380	0.021	0.090	0.066	3983
R-HSA-168138	Toll Like Receptor 9 (TLR9) Cascade	107	-0.292	-1.380	0.021	0.090	0.066	3934
R-HSA-1280218	Adaptive Immune System	714	-0.197	-1.160	0.030	0.115	0.084	4081
R-HSA-168181	Toll Like Receptor 7/8 (TLR7/8) Cascade	104	-0.286	-1.350	0.032	0.122	0.089	3934

Table C.12: Reactome pathways enriched for immune-related pathway (GSEA). Cluster 1 vs Cluster 3

ID	Description	setSize	enrichmentScore	NES	pvalue	p.adjust	qvalue	rank
R-HSA-5693607	Processing of DNA double-strand break ends	89	0.476	1.970	<0.001	<0.001	<0.001	4530
R-HSA-9670095	Inhibition of DNA recombination at telomere	58	0.546	2.099	<0.001	<0.001	<0.001	4530
R-HSA-5693567	HDR through Homologous Recombination (HRR) or Single Strand Annealing (SSA)	123	0.438	1.907	<0.001	<0.001	<0.001	4170
R-HSA-5693532	DNA Double-Strand Break Repair	159	0.389	1.754	<0.001	<0.001	<0.001	4530
R-HSA-912446	Meiotic recombination	71	0.482	1.921	<0.001	0.001	<0.001	4530
R-HSA-73894	DNA Repair	321	0.298	1.455	0.001	0.009	0.007	4772
R-HSA-5685942	HDR through Homologous Recombination (HRR)	67	0.403	1.586	0.007	0.040	0.030	4410
R-HSA-5696397	Gap-filling DNA repair synthesis and ligation in GG-NER	25	0.519	1.653	0.012	0.058	0.043	4772
R-HSA-6782210	Gap-filling DNA repair synthesis and ligation in TC-NER	62	0.384	1.492	0.021	0.090	0.066	4772

Table C.13: Reactome pathways enriched for HR-related pathway (GSEA). Cluster 1 vs Cluster 3

Category	Subcategory	ID	Description	GeneRatio	BgRatio	RichFactor	FoldEnrichment	zScore	pvalue	p.adjust	qvalue	Count
Human Diseases	Immune disease	hsa05322	Systemic lupus erythematosus	16/448	109/7129	0.147	2.336	3.639	0.001	0.027	0.026	16
Organismal Systems	Immune system	hsa04613	Neutrophil extracellular trap formation	20/448	161/7129	0.124	1.977	3.246	0.003	0.047	0.046	20

Table C.14: Immune-related KEGG pathways significantly enriched in Cluster 1 vs Cluster 4 (GSE analysis).

ID	Description	GeneRatio	BgRatio	RichFactor	FoldEnrichment	zScore	pvalue	p.adjust	qvalue	Count
R-HSA-912446	Meiotic recombination	16/618	87/11214	0.184	3.340	5.280	<0.001	0.001	0.001	16
R-HSA-9670095	Inhibition of DNA recombination at telomere	12/618	67/11214	0.179	3.250	4.460	<0.001	0.005	0.004	12

Table C.15: HR-related KEGG pathways significantly enriched in Cluster 1 vs Cluster 4 (GSE analysis).

ID	Description	setSize	enrichmentScore	NES	pvalue	p.adjust	qvalue	rank
GO:0019730	antimicrobial humoral response	127	0.520	2.316	<0.001	<0.001	<0.001	4897
GO:0006959	humoral immune response	216	0.428	2.069	<0.001	<0.001	<0.001	4897
GO:0097529	myeloid leukocyte migration	211	0.379	1.829	<0.001	<0.001	<0.001	5630
GO:0002274	myeloid leukocyte activation	223	0.362	1.754	<0.001	<0.001	<0.001	6228
GO:0030595	leukocyte chemotaxis	206	0.372	1.788	<0.001	<0.001	<0.001	5670
GO:0050900	leukocyte migration	353	0.327	1.669	<0.001	<0.001	<0.001	5670
GO:0002685	regulation of leukocyte migration	220	0.360	1.742	<0.001	<0.001	<0.001	5630
GO:0002687	positive regulation of leukocyte migration	149	0.384	1.759	<0.001	0.001	0.001	6171
GO:0097530	granulocyte migration	136	0.392	1.773	<0.001	0.001	0.001	7078
GO:0002683	negative regulation of immune system process	495	0.274	1.441	<0.001	0.004	0.003	4959
GO:0002688	regulation of leukocyte chemotaxis	118	0.387	1.719	<0.001	0.005	0.004	4900
GO:0071621	granulocyte chemotaxis	113	0.387	1.713	0.001	0.007	0.006	7078
GO:0019882	antigen processing and presentation	106	0.386	1.695	0.001	0.009	0.007	6076
GO:0002697	regulation of immune effector process	367	0.281	1.437	0.001	0.014	0.012	6085
GO:0071674	mononuclear cell migration	202	0.312	1.497	0.002	0.024	0.020	6171
GO:0034341	response to type II interferon	112	0.360	1.590	0.003	0.028	0.023	6186
GO:0050670	regulation of lymphocyte proliferation	223	0.300	1.453	0.003	0.030	0.025	6143
GO:0002695	negative regulation of leukocyte activation	182	0.310	1.463	0.004	0.038	0.031	5576
GO:0002696	positive regulation of leukocyte activation	343	0.269	1.370	0.005	0.045	0.037	6173
GO:0051250	negative regulation of lymphocyte activation	155	0.315	1.454	0.005	0.045	0.037	5576
GO:0046651	lymphocyte proliferation	277	0.278	1.379	0.005	0.046	0.037	6143
GO:0002699	positive regulation of immune effector process	238	0.289	1.410	0.006	0.046	0.038	6132

Table C.16: GSEA results for HR-related biological processes (BP). Cluster 1 vs Cluster 4

ID	Description	setSize	enrichmentScore	NES	pvalue	p.adjust	qvalue	rank
hsa04061	Viral protein interaction with cytokine and cytokine receptor	84	0.484	2.048	<0.001	<0.001	<0.001	4897
hsa04060	Cytokine-cytokine receptor interaction	228	0.350	1.718	<0.001	<0.001	<0.001	5376
hsa05320	Autoimmune thyroid disease	33	0.571	1.975	<0.001	0.002	0.001	6132
hsa05321	Inflammatory bowel disease	54	0.434	1.702	0.004	0.028	0.022	6100
hsa04612	Antigen processing and presentation	62	0.396	1.587	0.010	0.053	0.041	6230
hsa04670	Leukocyte transendothelial migration	104	0.330	1.455	0.013	0.068	0.053	4748
hsa04920	Adipocytokine signaling pathway	67	-0.343	-1.477	0.018	0.082	0.064	5630
hsa04145	Phagosome	146	0.309	1.436	0.023	0.102	0.079	4567
hsa04672	Intestinal immune network for IgA production	40	0.423	1.540	0.023	0.102	0.080	6439

Table C.17: Immune-related KEGG pathways significantly enriched in Cluster 1 vs Cluster 4 (GSEA analysis).

ID	Description	setSize	enrichmentScore	NES	pvalue	p.adjust	qvalue
R-HSA-168249	Innate Immune System	914	0.322	1.779	<0.001	<0.001	<0.001
R-HSA-1280215	Cytokine Signaling in Immune system	712	0.293	1.592	<0.001	<0.001	<0.001
R-HSA-1236975	Antigen processing-Cross presentation	94	0.477	2.058	<0.001	<0.001	<0.001
R-HSA-1236974	ER-Phagosome pathway	79	0.473	1.980	<0.001	<0.001	<0.001
R-HSA-9912633	Antigen processing: Ub, ATP-independent proteasomal degradation	19	0.628	1.907	0.002	0.016	0.012
R-HSA-1236978	Cross-presentation of soluble exogenous antigens (endosomes)	41	0.478	1.756	0.003	0.023	0.017
R-HSA-9692916	SARS-CoV-1 activates/modulates innate immune responses	39	0.474	1.720	0.005	0.030	0.022
R-HSA-68884	Mitotic Telophase/Cytokinesis	13	-0.648	-1.843	0.005	0.034	0.025
R-HSA-3134975	Regulation of innate immune responses to cytosolic DNA	14	0.624	1.747	0.008	0.045	0.033
R-HSA-177243	Interactions of Rev with host cellular proteins	36	-0.444	-1.674	0.009	0.048	0.036
R-HSA-176033	Interactions of Vpr with host cellular proteins	36	-0.406	-1.531	0.024	0.097	0.071
R-HSA-977606	Regulation of Complement cascade	35	0.421	1.484	0.035	0.132	0.097

Table C.18: Reactome pathways enriched for immune-related pathway (GSEA). Cluster 1 vs Cluster 4

ID	Description	setSize	enrichmentScore	NES	pvalue	p.adjust	qvalue	rank
R-HSA-9670095	Inhibition of DNA recombination at telomere	58	0.629	2.489	<0.001	<0.001	<0.001	4492
R-HSA-912446	Meiotic recombination	71	0.569	2.337	<0.001	<0.001	<0.001	3617
R-HSA-5693607	Processing of DNA double-strand break ends	89	0.437	1.869	<0.001	0.001	0.001	4488
R-HSA-5693567	HDR through Homologous Recombination (HRR) or SSA	123	0.385	1.734	<0.001	0.003	0.002	4488
R-HSA-5696397	Gap-filling DNA repair synthesis and ligation in GG-NER	25	0.560	1.822	0.003	0.024	0.018	3401
R-HSA-6782210	Gap-filling DNA repair synthesis and ligation in TC-NER	62	0.405	1.625	0.004	0.029	0.021	2615
R-HSA-73894	DNA Repair	321	0.271	1.375	0.006	0.036	0.026	2780
R-HSA-5693532	DNA Double-Strand Break Repair	159	0.305	1.425	0.011	0.055	0.041	4488
R-HSA-5685939	HDR through MMEJ (alt-NHEJ)	12	-0.583	-1.615	0.029	0.114	0.084	4867

Table C.19: Reactome pathways enriched for HR-related pathway (GSEA). Cluster 1 vs Cluster 4

ID	Description	setSize	enrichmentScore	NES	pvalue	p.adjust	qvalue	rank
GO:0002460	adaptive immune response...	346	0.478	2.162	<0.001	<0.001	<0.001	6080
GO:0002449	lymphocyte mediated immunity	328	0.468	2.103	<0.001	<0.001	<0.001	6163
GO:0050900	leukocyte migration	353	0.430	1.953	<0.001	<0.001	<0.001	5936
GO:0030098	lymphocyte differentiation	403	0.428	1.951	<0.001	<0.001	<0.001	6242
GO:0002768	immune response-regulating...	335	0.432	1.946	<0.001	<0.001	<0.001	5978
GO:0002443	leukocyte mediated immunity	417	0.425	1.945	<0.001	<0.001	<0.001	6187
GO:0002757	immune response-activating...	492	0.405	1.867	<0.001	<0.001	<0.001	5970
GO:0051249	regulation of lymphocyte...	479	0.388	1.788	<0.001	<0.001	<0.001	6064
GO:0002429	immune response-activating...	308	0.428	1.917	<0.001	<0.001	<0.001	5970
GO:0071674	mononuclear cell migration	202	0.474	2.048	<0.001	<0.001	<0.001	5924

Table C.20: Immune-related GO Biological Process term sig enriched in Cluster 2 vs Cluster 3. (GSE)

ID	Description	setSize	enrichmentScore	NES	pvalue	p.adjust	qvalue	rank
hsa04060	Cytokine-cytokine receptor interaction	228	0.410	1.813	<0.001	<0.001	<0.001	5322
hsa04672	Intestinal immune network for IgA production	40	0.615	2.098	<0.001	<0.001	<0.001	5148
hsa05321	Inflammatory bowel disease	54	0.531	1.920	<0.001	0.001	0.001	5090
hsa04145	Phagosome	146	0.399	1.681	<0.001	0.002	0.001	6059
hsa04750	Inflammatory mediator regulation of TRP channels	93	0.426	1.685	0.001	0.006	0.004	4873
hsa04670	Leukocyte transendothelial migration	104	0.411	1.651	0.002	0.010	0.006	6086
hsa05320	Autoimmune thyroid disease	33	0.524	1.712	0.004	0.021	0.013	5322
hsa04061	Viral protein interaction with cytokine and cytokine receptor	84	0.408	1.581	0.007	0.031	0.020	5015
hsa04610	Complement and coagulation cascades	68	0.411	1.546	0.008	0.032	0.021	6216
hsa04660	T cell receptor signaling pathway	114	0.357	1.455	0.012	0.044	0.029	5128
hsa04620	Toll-like receptor signaling pathway	90	0.352	1.381	0.036	0.099	0.064	4875
hsa04612	Antigen processing and presentation	62	0.383	1.408	0.050	0.128	0.083	6570
hsa04662	B cell receptor signaling pathway	80	0.362	1.388	0.052	0.130	0.084	4237

Table C.21: Immune-related KEGG pathways significantly enriched in Cluster 2 vs Cluster 3 (GSEA analysis).

ID	Description	setSize	enrichmentScore	NES	pvalue	p.adjust	qvalue
R-HSA-1280218	Adaptive Immune System	714	0.303	1.427	<0.001	0.002	0.001
R-HSA-1280215	Cytokine Signaling in Immune system	712	0.303	1.426	<0.001	0.002	0.002
R-HSA-388841	Regulation of T cell activation by CD28 family	72	0.429	1.628	0.004	0.048	0.042
R-HSA-168249	Innate Immune System	914	0.265	1.256	0.006	0.066	0.057
R-HSA-2132295	MHC class II antigen presentation	118	0.358	1.461	0.014	0.114	0.099
R-HSA-9662851	Anti-inflammatory response favouring Leishmania parasite infection	74	0.394	1.497	0.016	0.128	0.111

Table C.22: Reactome pathways enriched for immune-related pathway (GSEA). Cluster 2 vs Cluster 3

ID	Description	GeneRatio	BgRatio	RichFactor	FoldEnrichment	zScore	pvalue	p.adjust	qvalue	Count
GO:0048245	eosinophil chemotaxis	6/629	18/15169	0.333	8.039	6.215	<0.001	0.005	0.004	6
GO:0002686	negative regulation of leukocyte migration	9/629	47/15169	0.191	4.618	5.167	<0.001	0.006	0.006	9
GO:0071675	regulation of mononuclear cell migration	16/629	134/15169	0.119	2.880	4.545	<0.001	0.007	0.006	16
GO:0048247	lymphocyte chemotaxis	8/629	39/15169	0.205	4.947	5.133	<0.001	0.008	0.007	8
GO:0097529	myeloid leukocyte migration	21/629	211/15169	0.100	2.400	4.260	<0.001	0.009	0.008	21
GO:0072677	eosinophil migration	6/629	22/15169	0.273	6.577	5.445	<0.001	0.009	0.008	6
GO:0071674	mononuclear cell migration	20/629	202/15169	0.099	2.388	4.130	<0.001	0.012	0.010	20
GO:0002685	regulation of leukocyte migration	21/629	220/15169	0.095	2.302	4.046	<0.001	0.013	0.012	21
GO:0002688	regulation of leukocyte chemotaxis	14/629	118/15169	0.119	2.861	4.221	<0.001	0.014	0.012	14
GO:0002689	negative regulation of leukocyte chemotaxis	5/629	17/15169	0.294	7.093	5.228	<0.001	0.017	0.015	5
GO:0050900	leukocyte migration	28/629	353/15169	0.079	1.913	3.609	0.001	0.023	0.020	28
GO:0071676	negative regulation of mononuclear cell migration	6/629	28/15169	0.214	5.168	4.591	0.001	0.024	0.021	6
GO:0030595	leukocyte chemotaxis	19/629	206/15169	0.092	2.224	3.680	0.001	0.026	0.023	19
GO:0045649	regulation of macrophage differentiation	5/629	23/15169	0.217	5.243	4.235	0.002	0.044	0.039	5
GO:1901623	regulation of lymphocyte chemotaxis	5/629	23/15169	0.217	5.243	4.235	0.002	0.044	0.039	5

Table C.23: Immune-related GO Biological Process terms significantly enriched in Cluster 2 vs Cluster 4. GSE Biological Process.

Category	Subcategory	ID	Description	GeneRatio	BgRatio	RichFactor	FoldEnrichment	zScore	pvalue	p.adjust	qvalue	Count
Organismal Systems	Immune system	hsa04610	Complement and coagulation cascades	19/796	68/7129	0.279	2.502	4.413	<0.001	0.004	0.003	19
Organismal Systems	Immune system	hsa04640	Hematopoietic cell lineage	19/796	85/7129	0.224	2.002	3.294	0.002	0.029	0.026	19
Cellular Processes	Transport and catabolism	hsa04145	Phagosome	28/796	146/7129	0.192	1.718	3.106	0.003	0.036	0.031	28
Organismal Systems	Immune system	hsa04650	Natural killer cell mediated cytotoxicity	20/796	95/7129	0.211	1.885	3.080	0.004	0.043	0.038	20
Organismal Systems	Immune system	hsa04620	Toll-like receptor signaling pathway	19/796	90/7129	0.211	1.891	3.015	0.004	0.048	0.042	19
Environmental Information Processing	Signaling molecules and interaction	hsa04060	Cytokine-cytokine receptor interaction	38/796	228/7129	0.167	1.493	2.680	0.007	0.074	0.065	38

Table C.24: KEGG pathways enriched for Immune-related pathway (GSE). Cluster 2 vs Cluster 4

ID	Description	setSize	enrichmentScore	NES	pvalue	p.adjust	qvalue	rank
GO:0097529	myeloid leukocyte migration	211	0.579	2.612	<0.001	<0.001	<0.001	3415
GO:0030595	leukocyte chemotaxis	206	0.576	2.593	<0.001	<0.001	<0.001	3479
GO:0002685	regulation of leukocyte migration	220	0.572	2.586	<0.001	<0.001	<0.001	4992
GO:0002274	myeloid leukocyte activation	223	0.570	2.578	<0.001	<0.001	<0.001	3415
GO:0050900	leukocyte migration	353	0.535	2.526	<0.001	<0.001	<0.001	3663
GO:0002688	regulation of leukocyte chemotaxis	118	0.603	2.518	<0.001	<0.001	<0.001	3228
GO:0071674	mononuclear cell migration	202	0.561	2.516	<0.001	<0.001	<0.001	4992
GO:0071675	regulation of mononuclear cell migration	134	0.587	2.488	<0.001	<0.001	<0.001	4992
GO:0002687	positive regulation of leukocyte migration	149	0.575	2.477	<0.001	<0.001	<0.001	5145
GO:0097530	granulocyte migration	136	0.562	2.386	<0.001	<0.001	<0.001	3663
GO:0071621	granulocyte chemotaxis	113	0.564	2.345	<0.001	<0.001	<0.001	3663
GO:0002695	negative regulation of leukocyte activation	182	0.514	2.271	<0.001	<0.001	<0.001	4991
GO:0002768	immune response-regulating cell surface receptor signaling pathway	335	0.462	2.166	<0.001	<0.001	<0.001	5354
GO:0002683	negative regulation of immune system process	495	0.444	2.146	<0.001	<0.001	<0.001	4991
GO:0002697	regulation of immune effector process	367	0.447	2.123	<0.001	<0.001	<0.001	5354
GO:0002366	leukocyte activation involved in immune response	280	0.457	2.110	<0.001	<0.001	<0.001	4779
GO:0002263	cell activation involved in immune response	284	0.454	2.108	<0.001	<0.001	<0.001	4779
GO:0002696	positive regulation of leukocyte activation	343	0.442	2.081	<0.001	<0.001	<0.001	5732
GO:0002429	immune response-activating cell surface receptor signaling pathway	308	0.446	2.076	<0.001	<0.001	<0.001	5346
GO:0046651	lymphocyte proliferation	277	0.444	2.050	<0.001	<0.001	<0.001	5013
GO:0002443	leukocyte mediated immunity	417	0.425	2.037	<0.001	<0.001	<0.001	4779

Table C.25: Immune-related GO Biological Process terms significantly enriched in Cluster 2 vs Cluster 4. GSEA Biological Process.

ID	Description	setSize	enrichmentScore	NES	pvalue	p.adjust	qvalue	rank
GO:0000724	double-strand break repair via homologous recombination	182	-0.2893749869	-1.482	0.001	0.002	0.001	5501

Table C.26: HR-related GO Biological Process terms significantly enriched in Cluster 2 vs Cluster 4. GSEA Biological Process.

ID	Description	setSize	enrichmentScore	NES	pvalue	p.adjust	qvalue	rank
hsa04060	Cytokine-cytokine receptor interaction	228	0.486	2.230	<0.001	<0.001	<0.001	4021
hsa04145	Phagosome	146	0.495	2.161	<0.001	<0.001	<0.001	5175
hsa04670	Leukocyte transendothelial migration	104	0.536	2.224	<0.001	<0.001	<0.001	5206
hsa04061	Viral protein interaction with cytokine & cytokine receptor	84	0.567	2.261	<0.001	<0.001	<0.001	4015
hsa04620	Toll-like receptor signaling pathway	90	0.512	2.068	<0.001	<0.001	<0.001	4305
hsa04662	B cell receptor signaling pathway	80	0.507	2.009	<0.001	<0.001	<0.001	4764
hsa04610	Complement and coagulation cascades	68	0.521	2.011	<0.001	<0.001	<0.001	2461
hsa05321	Inflammatory bowel disease	54	0.550	2.034	<0.001	<0.001	<0.001	5211
hsa04660	T cell receptor signaling pathway	114	0.448	1.882	<0.001	<0.001	<0.001	6611
hsa05320	Autoimmune thyroid disease	33	0.626	2.083	<0.001	<0.001	<0.001	5640
hsa04672	Intestinal immune network for IgA production	40	0.537	1.880	0.001	0.002	0.001	4880
hsa04612	Antigen processing and presentation	62	0.401	1.524	0.020	0.042	0.020	3656
hsa04750	Inflammatory mediator regulation of TRP channels	93	0.347	1.418	0.030	0.061	0.029	6722

Table C.27: KEGG pathways enriched for Immune-related pathway (GSEA). Cluster 2 vs Cluster 4

ID	Description	Set Size	Enrichment Score	NES	p-value	p-adjust	q-value	Rank
hsa03440	Homologous recombination	40	-0.425	-1.673	0.009	0.021	0.010	3785
hsa03460	Fanconi anemia pathway	50	-0.364	-1.489	0.021	0.044	0.021	3735

Table C.28: KEGG pathways enriched for HR-related pathway (GSEA). Cluster 2 vs Cluster 4

ID	Description	setSize	enrichmentScore	NES	pvalue	p.adjust	qvalue	rank
R-HSA-168249	Innate Immune System	914	0.401	2.010	<0.001	<0.001	<0.001	6423
R-HSA-1280215	Cytokine Signaling in Immune system	712	0.367	1.817	<0.001	<0.001	<0.001	5732
R-HSA-1280218	Adaptive Immune System	714	0.357	1.767	<0.001	<0.001	<0.001	5521
R-HSA-168898	Toll-like Receptor Cascades	163	0.404	1.777	<0.001	<0.001	<0.001	6799
R-HSA-168179	Toll Like Receptor TLR1:TLR2 Cascade	110	0.434	1.809	<0.001	0.001	0.001	6735
R-HSA-181438	Toll Like Receptor 2 (TLR2) Cascade	110	0.434	1.809	<0.001	0.001	0.001	6735
R-HSA-1236975	Antigen processing-Cross presentation	94	0.457	1.863	<0.001	0.001	0.001	5640
R-HSA-168188	Toll Like Receptor TLR6:TLR2 Cascade	109	0.432	1.802	<0.001	0.002	0.001	6735
R-HSA-166016	Toll Like Receptor 4 (TLR4) Cascade	139	0.408	1.760	<0.001	0.002	0.002	5521
R-HSA-9662851	Anti-inflammatory response favouring Leishmania parasite infection	74	0.464	1.813	<0.001	0.003	0.002	4363
R-HSA-388841	Regulation of T cell activation by CD28 family	72	0.472	1.838	<0.001	0.003	0.002	5348
R-HSA-9660821	ADORA2B mediated anti-inflammatory cytokines production	39	0.521	1.803	0.001	0.013	0.010	5817
R-HSA-9664424	Cell recruitment (pro-inflammatory response)	26	0.584	1.846	0.002	0.016	0.011	2985
R-HSA-177243	Interactions of Rev with host cellular proteins	36	-0.487	-1.849	0.002	0.016	0.012	5258
R-HSA-5260271	Diseases of Immune System	29	0.560	1.810	0.003	0.023	0.017	4658
R-HSA-168181	Toll Like Receptor 7/8 (TLR7/8) Cascade	104	0.380	1.576	0.005	0.033	0.024	6735
R-HSA-176033	Interactions of Vpr with host cellular proteins	36	-0.441	-1.676	0.008	0.045	0.033	4970
R-HSA-1236974	ER-Phagosome pathway	79	0.395	1.565	0.008	0.045	0.033	5640
R-HSA-168138	Toll Like Receptor 9 (TLR9) Cascade	107	0.364	1.513	0.009	0.047	0.034	6735
R-HSA-168142	Toll Like Receptor 10 (TLR10) Cascade	96	0.372	1.522	0.009	0.050	0.037	6735
R-HSA-168176	Toll Like Receptor 5 (TLR5) Cascade	96	0.372	1.522	0.009	0.050	0.037	6735
R-HSA-68884	Mitotic Telophase/Cytokinesis	13	-0.614	-1.758	0.011	0.057	0.042	7041
R-HSA-2132295	MHC class II antigen presentation	118	0.349	1.468	0.012	0.059	0.043	5472
R-HSA-168164	Toll Like Receptor 3 (TLR3) Cascade	105	0.342	1.420	0.023	0.098	0.071	6916

Table C.29: Reactome pathways enriched for immune-related pathway (GSEA). Cluster 2 vs Cluster 4

ID	Description	setSize	enrichmentScore	NES	pvalue	p.adjust	qvalue	rank
R-HSA-9675136	Diseases of DNA Double-Strand Break Repair	41	-0.391	-1.534	0.015	0.070	0.051	5394
R-HSA-9701190	Defective homologous recombination repair (HRR) due to BRCA2 loss of function	41	-0.391	-1.534	0.015	0.070	0.051	5394
R-HSA-9701192	Defective homologous recombination repair (HRR) due to BRCA1 loss of function	25	-0.461	-1.600	0.020	0.088	0.064	5394
R-HSA-9701193	Defective homologous recombination repair (HRR) due to PALB2 loss of function	25	-0.461	-1.600	0.020	0.088	0.064	5394
R-HSA-9704331	Defective HDR through Homologous Recombination Repair (HRR) due to PALB2 loss of BRCA1 binding function	25	-0.461	-1.600	0.020	0.088	0.064	5394
R-HSA-9704646	Defective HDR through Homologous Recombination Repair (HRR) due to PALB2 loss of BRCA2/RAD51/RAD51C binding function	25	-0.461	-1.600	0.020	0.088	0.064	5394
R-HSA-5685939	HDR through MMEJ (alt-NHEJ)	12	-0.605	-1.678	0.021	0.091	0.066	5197
R-HSA-9670095	Inhibition of DNA recombination at telomere	58	0.382	1.433	0.037	0.131	0.095	7518

Table C.30: Reactome pathways enriched for HR-related pathway (GSEA). Cluster 2 vs Cluster 4

ID	Description	setSize	enrichmentScore	NES	pvalue	p.adjust	qvalue
GO:0097529	myeloid leukocyte migration	211	0.493	2.466	<0.001	<0.001	<0.001
GO:0002274	myeloid leukocyte activation	223	0.474	2.394	<0.001	<0.001	<0.001
GO:0002685	regulation of leukocyte migration	220	0.450	2.256	<0.001	<0.001	<0.001
GO:0050900	leukocyte migration	353	0.424	2.248	<0.001	<0.001	<0.001
GO:0030595	leukocyte chemotaxis	206	0.447	2.224	<0.001	<0.001	<0.001
GO:0071674	mononuclear cell migration	202	0.423	2.095	<0.001	<0.001	<0.001
GO:0097530	granulocyte migration	136	0.480	2.260	<0.001	<0.001	<0.001
GO:0002683	negative regulation of immune system process	495	0.329	1.817	<0.001	<0.001	<0.001
GO:0002687	positive regulation of leukocyte migration	149	0.461	2.186	<0.001	<0.001	<0.001
GO:0071675	regulation of mononuclear cell migration	134	0.450	2.107	<0.001	<0.001	<0.001
GO:0002758	innate immune response-activating signaling pathway	274	0.247	1.283	0.024	0.046	0.021

Table C.31: Immune-related GO Biological Process terms significantly enriched in Cluster 3 vs Cluster 4. GSEA Biological Process.

ID	Description	setSize	enrichmentScore	NES	pvalue	p.adjust	qvalue	rank
GO:0000724	double-strand break repair via homologous recombination	182	-0.311	-1.483	0.004	0.011	0.005	6132

Table C.32: HR-related GO Biological Process terms significantly enriched in Cluster 3 vs Cluster 4. GSEA Biological Process.

ID	Description	setSize	enrichmentScore	NES	pvalue	p.adjust	qvalue	rank
hsa04145	Phagosome	146	0.421	2.050	<0.001	<0.001	<0.001	4467
hsa04060	Cytokine-cytokine receptor interaction	228	0.370	1.888	<0.001	<0.001	<0.001	4944
hsa04620	Toll-like receptor signaling pathway	90	0.474	2.115	<0.001	<0.001	<0.001	4580
hsa04670	Leukocyte transendothelial migration	104	0.444	2.029	<0.001	<0.001	<0.001	4029
hsa04061	Viral protein interaction with cytokine and cytokine receptor	84	0.463	2.018	<0.001	<0.001	<0.001	5057
hsa04662	B cell receptor signaling pathway	80	0.467	2.020	<0.001	<0.001	<0.001	5200
hsa04610	Complement and coagulation cascades	68	0.440	1.859	<0.001	0.001	0.001	4221
hsa04660	T cell receptor signaling pathway	114	0.368	1.711	<0.001	0.003	0.002	5919
hsa05320	Autoimmune thyroid disease	33	0.500	1.771	0.002	0.007	0.004	7151
hsa05321	Inflammatory bowel disease	54	0.431	1.727	0.002	0.008	0.005	6201

Table C.33: KEGG pathways enriched for immune-related pathway (GSEA). Cluster 3 vs Cluster 4

ID	Description	setSize	enrichmentScore	NES	pvalue	p.adjust	qvalue	rank
hsa03460	Fanconi anemia pathway	50	-0.404	-1.570	0.008	0.025	0.014	6281

Table C.34: KEGG pathways enriched for hr-related pathway (GSEA). Cluster 3 vs Cluster 4

ID	Description	setSize	enrichmentScore	NES	pvalue	p.adjust	qvalue	rank
R-HSA-168249	Innate Immune System	914	0.379	2.205	<0.001	<0.001	<0.001	4688
R-HSA-1280215	Cytokine Signaling in Immune system	712	0.284	1.623	<0.001	<0.001	<0.001	6211
R-HSA-168898	Toll-like Receptor Cascades	163	0.371	1.806	<0.001	0.001	<0.001	5920
R-HSA-168179	Toll Like Receptor TLR1:TLR2 Cascade	110	0.412	1.889	<0.001	0.001	0.001	5920
R-HSA-181438	Toll Like Receptor 2 (TLR2) Cascade	110	0.412	1.889	<0.001	0.001	0.001	5920
R-HSA-168188	Toll Like Receptor TLR6:TLR2 Cascade	109	0.408	1.865	<0.001	0.001	0.001	5920
R-HSA-166016	Toll Like Receptor 4 (TLR4) Cascade	139	0.364	1.733	<0.001	0.004	0.003	5920
R-HSA-1280218	Adaptive Immune System	714	0.235	1.339	0.001	0.010	0.007	5815
R-HSA-9662851	Anti-inflammatory response favouring Leishmania parasite infection	74	0.402	1.718	0.001	0.010	0.008	2699
R-HSA-168181	Toll Like Receptor 7/8 (TLR7/8) Cascade	104	0.367	1.666	0.001	0.012	0.009	5920
R-HSA-168138	Toll Like Receptor 9 (TLR9) Cascade	107	0.361	1.646	0.001	0.015	0.011	5920
R-HSA-168142	Toll Like Receptor 10 (TLR10) Cascade	96	0.370	1.653	0.002	0.018	0.013	5920
R-HSA-168176	Toll Like Receptor 5 (TLR5) Cascade	96	0.370	1.653	0.002	0.018	0.013	5920
R-HSA-9660821	ADORA2B mediated anti-inflammatory cytokines production	39	0.475	1.773	0.004	0.030	0.023	2699
R-HSA-5260271	Diseases of Immune System	29	0.526	1.817	0.004	0.030	0.023	5787
R-HSA-68884	Mitotic Telophase/Cytokinesis	13	-0.666	-1.833	0.004	0.033	0.025	5263
R-HSA-177243	Interactions of Rev with host cellular proteins	36	-0.474	-1.710	0.007	0.044	0.033	5260
R-HSA-168164	Toll Like Receptor 3 (TLR3) Cascade	105	0.335	1.524	0.007	0.047	0.036	5920
R-HSA-1236975	Antigen processing-Cross presentation	94	0.341	1.522	0.010	0.061	0.046	5806
R-HSA-176033	Interactions of Vpr with host cellular proteins	36	-0.449	-1.620	0.012	0.067	0.050	5260
R-HSA-3134975	Regulation of innate immune responses to cytosolic DNA	14	0.599	1.707	0.013	0.070	0.053	4085
R-HSA-9664424	Cell recruitment (pro-inflammatory response)	26	0.475	1.602	0.020	0.093	0.070	5287

Table C.35: Reactome pathways enriched for immune-related pathway (GSEA). Cluster 3 vs Cluster 4

ID	Description	setSize	enrichmentScore	NES	pvalue	p.adjust	qvalue	rank
R-HSA-9670095	Inhibition of DNA recombination at telomere	58	0.476	1.942	<0.001	0.003	0.003	6419
R-HSA-912446	Meiotic recombination	71	0.405	1.716	0.001	0.014	0.010	4470
R-HSA-9701192	Defective homologous recombination repair (HRR) due to BRCA1 loss of function	25	-0.461	-1.521	0.035	0.136	0.103	6970
R-HSA-9701193	Defective homologous recombination repair (HRR) due to PALB2 loss of function	25	-0.461	-1.521	0.035	0.136	0.103	6970
R-HSA-9704331	Defective HDR through Homologous Recombination Repair (HRR) due to PALB2 loss of BRCA1 binding function	25	-0.461	-1.521	0.035	0.136	0.103	6970
R-HSA-9704646	Defective HDR through Homologous Recombination Repair (HRR) due to PALB2 loss of BRCA2/RAD51/RAD51C binding function	25	-0.461	-1.521	0.035	0.136	0.103	6970

Table C.36: Reactome pathways enriched for HR-related pathway (GSEA). Cluster 3 vs Cluster 4