The Single-cell Pediatric Cancer Atlas: Data portal and open-source tools for single-cell transcriptomics of pediatric tumors

This manuscript (permalink) was automatically generated from AlexsLemonade/ScPCA-manuscript@9a5148c on March 22, 2024.

Authors

- **Allegra G. Hawkins**
  0000-0001-6026-3660 · allyhawkins
  Childhood Cancer Data Lab, Alex's Lemonade Stand Foundation, Bala Cynwyd, PA, 19004, USA · Funded by Alex's Lemonade Stand Foundation Childhood Cancer Data Lab (CCDL)

- **Joshua A. Shapiro**
  0000-0002-6224-0347 · jashapiro
  Childhood Cancer Data Lab, Alex's Lemonade Stand Foundation, Bala Cynwyd, PA, 19004, USA · Funded by Alex's Lemonade Stand Foundation Childhood Cancer Data Lab (CCDL)

- **Stephanie J. Spielman**
  0000-0002-9090-4788 · sjspielman
  Childhood Cancer Data Lab, Alex's Lemonade Stand Foundation, Bala Cynwyd, PA, 19004, USA · Funded by Alex's Lemonade Stand Foundation Childhood Cancer Data Lab (CCDL)

- **David S. Mejia**
  TODO · davidsmejia
  Childhood Cancer Data Lab, Alex's Lemonade Stand Foundation, Bala Cynwyd, PA, 19004, USA · Funded by Alex's Lemonade Stand Foundation Childhood Cancer Data Lab (CCDL)

- **Deepashree Venkatesh Prasad**
  0000-0001-5756-4083 · dvenprasad
  Childhood Cancer Data Lab, Alex's Lemonade Stand Foundation, Bala Cynwyd, PA, 19004, USA · Funded by Alex's Lemonade Stand Foundation Childhood Cancer Data Lab (CCDL)

- **Nozomi Ichihara**
  · nozomione
  Childhood Cancer Data Lab, Alex's Lemonade Stand Foundation, Bala Cynwyd, PA, 19004, USA · Funded by Alex's Lemonade Stand Foundation Childhood Cancer Data Lab (CCDL)

- **Arkadii Yakovets**
  TODO · arkid15r
  Childhood Cancer Data Lab, Alex's Lemonade Stand Foundation, Bala Cynwyd, PA, 19004, USA · Funded by Alex's Lemonade Stand Foundation Childhood Cancer Data Lab (CCDL)

- **Kurt G. Wheeler**
  TODO · kurtwheeler
  Childhood Cancer Data Lab, Alex's Lemonade Stand Foundation, Bala Cynwyd, PA, 19004, USA · Funded by Alex's Lemonade Stand Foundation Childhood Cancer Data Lab (CCDL)

- **Chante J. Bethell**
  0000-0001-9653-8128 · cbethell
  Childhood Cancer Data Lab, Alex's Lemonade Stand Foundation, Bala Cynwyd, PA, 19004, USA; The University of Texas MD Anderson Cancer Center, UTHealth Houston Graduate School of Biomedical Sciences, Houston, TX, 77030, USA · Funded by Alex's Lemonade Stand Foundation Childhood Cancer Data Lab (CCDL)

- **Steven M. Foltz**
  0000-0002-9526-8194 · envest
  Childhood Cancer Data Lab, Alex's Lemonade Stand Foundation, Bala Cynwyd, PA, 19004, USA; Department of Pediatrics, Division of Oncology, Children's Hospital of Philadelphia, Philadelphia, PA, 19104, USA · Funded by Alex's Lemonade Stand Foundation Childhood Cancer Data Lab (CCDL)

- **Jennifer O'Malley**
  · Jen-OMalley
  Childhood Cancer Data Lab, Alex's Lemonade Stand Foundation, Bala Cynwyd, PA, 19004, USA · Funded by Alex's Lemonade Stand Foundation Childhood Cancer Data Lab (CCDL)

- **Casey S. Greene**
  0000-0001-8713-9213 · cgreene
  Childhood Cancer Data Lab, Alex's Lemonade Stand Foundation, Bala Cynwyd, PA, 19004, USA; Center for Health AI, University of Colorado School of Medicine, Aurora, CO, 80045, USA; Department of Biomedical Informatics, University of Colorado School of Medicine, Aurora, CO, 80045, USA · Funded by Alex's Lemonade Stand Foundation Childhood Cancer Data Lab (CCDL)

- **Jaclyn N. Taroni** ✉
  0000-0003-4734-4508 · jaclyn-taroni
  Childhood Cancer Data Lab, Alex's Lemonade Stand Foundation, Bala Cynwyd, PA, 19004, USA · Funded by Alex's Lemonade Stand Foundation Childhood Cancer Data Lab (CCDL)

✉ — Correspondence possible via GitHub Issues or email to Jaclyn N. Taroni <jaclyn.taroni@ccdatalab.org>.

Abstract

The Single-cell Pediatric Cancer Atlas (ScPCA) Portal (https://scpca.alexslemonade.org/) is a data resource for uniformly processed single-cell and single-nuclei RNA sequencing (RNA-seq) data and de-identified metadata from pediatric tumor samples. Originally comprised of data from 10 projects funded by Alex's Lemonade Stand Foundation, the Portal currently contains summarized gene expression data for over 500 samples from over 50 types of cancers from ALSF-funded and community-contributed datasets. In addition to gene expression data from single-cell and single-nuclei RNA-seq, the Portal holds data obtained from bulk RNA-seq, spatial transcriptomics, and feature barcoding methods, such as CITE-seq and cell hashing.

ScPCA data are available for download as `SingleCellExperiment` or `AnnData` objects and are ready for downstream analyses. Objects include raw counts and normalized gene expression data, PCA and UMAP coordinates, and automated cell type annotations. Additionally, all downloads include two summary reports for each library: a quality control report summarizing sample statistics and displaying visualizations of cell metrics and a cell type annotation report with comparisons among cell type annotation methods and diagnostic plots to assess annotation quality. Merged `SingleCellExperiment` and `AnnData` objects containing all gene expression data and metadata for all samples in an ScPCA project are also available for download. These objects are useful when performing analysis on multiple samples simultaneously. Comprehensive documentation about data processing and the contents of files on the Portal, including a guide to getting started working with an ScPCA dataset, can be found at http://scpca.readthedocs.io.

All data on the Portal were uniformly processed using `scpca-nf`, an open-source and efficient Nextflow workflow that uses `alevin-fry` to quantify all single-cell and single-nuclei RNA-seq data, any associated CITE-seq or cell hash data, spatial transcriptomics data, and bulk RNA-seq. Any pediatric cancer-relevant data sets processed with `scpca-nf` are eligible for inclusion on the ScPCA Portal, enabling continuous growth of the ScPCA Portal to help pediatric cancer researchers spend less time finding and processing data and more time answering their pressing research questions.

Introduction

Since the introduction of single-cell RNA-seq technology, the number of studies that employ single-cell RNA-seq has grown rapidly [1]. Unlike its predecessor, bulk RNA-seq, which averages the expression profiles of all cells within a sample, single-cell technology quantifies gene expression in individual cells. Tumors are known to be transcriptionally heterogeneous, so many studies have highlighted the importance of using single-cell RNA-seq in studying tumor samples [2]. Researchers can use single-cell RNA-seq of samples obtained from patient tumors to analyze and identify individual cell populations that may play important roles in tumor growth, resistance, and metastasis [3]. Additionally, single-cell RNA-seq data provides insight into how tumor cells interact with normal cells in the tumor microenvironment [4].

With the growing number of single-cell RNA-seq datasets, efforts have emerged to create central, harmonized sources for datasets. Harmonized data resources allow researchers to leverage more samples from various biological contexts to complete their analysis and elucidate previously unknown similarities across samples and disease types. The Human Cell Atlas (HCA) and Human Tumor Atlas Network (HTAN) are two of many such examples. The HCA, which aims to use single-cell genomics to provide a comprehensive map of all cell types in the human body [5], contains uniformly processed single-cell RNA-seq data obtained from normal tissue with few samples derived from diseased tissue.

The HTAN also hosts a collection of genomic data collected from tumors across multiple cancer types, including single-cell RNA-seq [6].

Existing resources have focused on making large quantities of harmonized data from normal tissue or adult tumor samples publicly available, but there are considerably fewer efforts to harmonize and distribute data from pediatric tumors. Pediatric cancer is much less common than adult cancer, so the number of available samples from pediatric tumors is smaller compared to the number of adult tumors [7] and access to data from pediatric tumors is often limited. Thus, it is imperative to provide harmonized data from pediatric tumors to all pediatric cancer researchers [8]. To address this unmet need, Alex's Lemonade Stand Foundation and the Childhood Cancer Data Lab developed and maintain the Single-cell Pediatric Cancer Atlas (ScPCA) Portal (https://scpca.alexslemonade.org/), an open-source data resource for single-cell and single-nuclei RNA-seq data of pediatric tumors.

The ScPCA Portal holds uniformly processed summarized gene expression from 10x Genomics droplet-based single-cell and single-nuclei RNA-seq for over 500 samples from a diverse set of over 50 types of pediatric cancers. Originally comprised of data from ten projects funded by Alex's Lemonade Stand Foundation, the Portal has since expanded to include data contributed by pediatric cancer research community members. In addition to gene expression data from single-cell and single-nuclei RNA-seq, the Portal includes data obtained from bulk RNA-seq, spatial transcriptomics, and feature barcoding methods, such as CITE-seq and cell hashing. All data provided on the portal are available in formats ready for downstream analysis with common workflow ecosystems such as `SingleCellExperiment` objects used by `R/Bioconductor` [9] or `AnnData` objects used by `Scanpy` and related Python modules [10]. Downloaded objects contain normalized gene expression counts, dimensionality reduction results, and cell type annotations.

To ensure that all current and future data on the Portal are uniformly processed, we created `scpca-nf`, an open-source Nextflow [11] pipeline (https://github.com/AlexsLemonade/scpca-nf). Using a consistent pipeline for all data increases transparency and allows users to perform analysis across multiple samples and projects without having to do any re-processing. The `scpca-nf` workflow uses `alevin-fry` [12] for fast and efficient quantification of single-cell gene expression for all samples on the Portal, including single-cell RNA-seq data and any associated CITE-seq or cell hash data. The `scpca-nf` pipeline also serves as a resource for the community, allowing others to process their own samples for comparison to samples available on the Portal and submit uniformly processed community contributions to the Portal.

Here, we present the Single-cell Pediatric Cancer Atlas as a resource for all pediatric cancer researchers. The ScPCA Portal provides downloads ready for immediate use, allowing researchers to skip time-consuming data re-processing and wrangling steps. We provide comprehensive documentation about data processing and the contents of files on the portal, including a guide to getting started working with an ScPCA dataset (https://scpca.readthedocs.io/). The ScPCA Portal advances pediatric cancer research by accelerating researchers' ability to answer important biological questions.

Results

The Single-cell Pediatric Cancer Atlas Portal

In March of 2022, the Childhood Cancer Data Lab launched the Single-cell Pediatric Cancer Atlas (ScPCA) Portal to make uniformly processed, summarized single-cell and single-nuclei RNA-seq data and de-identified metadata from pediatric tumor samples available for download. Data available on the Portal was obtained using two different mechanisms: raw data was accepted from ALSF-funded

investigators and processed using our open-source pipeline `scpca-nf`, or investigators processed their raw data using `scpca-nf` and submitted the output for inclusion on the Portal.

All samples on the Portal include a core set of metadata obtained from investigators, including age, sex, diagnosis, subdiagnosis (if applicable), tissue location, and disease stage. Some investigators submitted additional metadata, such as treatment and tumor stage, which can also be found on the Portal. All submitted metadata was standardized to maintain consistency across projects before adding to the Portal. In addition to providing a human-readable value for the submitted metadata, we also provide ontology term identifiers, if applicable. Submitted metadata was mapped to associated ontology term identifiers obtained from HsapDV (age) [13], PATO (sex) [14,15], NCBI taxonomy (organism) [16,17], MONDO (disease) [18,19], UBERON (tissue) [20,21,22], and Hancestro (ethnicity, if applicable) [23,24]. By providing these ontology term identifiers for each sample, users have access to standardized metadata terms that facilitate comparisons among datasets within the Portal as well as to data from other research projects.

The Portal contains data from 500 samples and over 50 tumor types. Figure 1A summarizes all samples from patient tumors and patient-derived xenografts currently available on the Portal. The total number of samples for each diagnosis is shown, along with the proportion of samples from each disease stage within a diagnosis group. The largest number of samples found on the Portal were obtained from patients with leukemia (n = 192). The Portal also includes samples from brain and central nervous system tumors (n = 154), sarcoma and soft tissue tumors (n = 68), and a variety of other solid tumors (n = 87). Most samples were collected at initial diagnosis (n = 424), with a smaller number of samples collected either at recurrence (n = 64), during progressive disease (n = 10), or post-mortem (n = 2). Along with the patient tumors, the Portal contains a small number of human tumor cell line samples (n = 4).

Each of the available samples contains summarized gene expression data from either single-cell or single-nuclei RNA sequencing. However, some samples also include additional data, such as quantified expression data from tagging cells with antibody-derived tags (ADT), such as CITE-seq antibodies [25], or multiplexing samples with hashtag oligonucleotides (HTO) [26] prior to sequencing. Out of the 500 samples, 96 have associated CITE-seq data, and 19 have associated multiplexing data. In some cases, multiple libraries from the same sample were collected for additional sequencing, either for bulk RNA-seq or spatial transcriptomics. Specifically, 118 samples on the Portal were sequenced using bulk RNA-seq and 94 samples were sequenced using spatial transcriptomics. A summary of the number of samples with each additional modality is shown in Figure 1B, and a detailed summary of the total samples with each sequencing method broken down by project is available in Table S1.

Samples on the Portal are organized by project, where each project is a collection of similar samples from an individual lab. Users can filter projects based on diagnosis, included modalities (e.g., CITE-seq, bulk RNA-seq), 10x Genomics kit version (e.g., 10Xv2, 10Xv3), and whether or not a project includes samples derived from patient-derived xenografts or cell lines. The project card displays an abstract, the total number of samples included, a list of diagnoses for all samples included in the Project, and links to any external information associated with the project, such as publications and links to external data, such as SRA or GEO (Figure 1C). The project card also indicates the type(s) of sequencing performed, including the 10x Genomics kit version, the suspension type (cell or nucleus), and if additional sequencing is present, like bulk RNA-seq or multiplexing.

Uniform processing of data available on the ScPCA Portal

All data available on the Portal was uniformly processed using `scpca-nf`, an open-source and efficient Nextflow [11] workflow for quantifying single-cell and single-nuclei RNA-seq data. Using

Nextflow as the backbone for the `scpca-nf` workflow ensures both reproducibility and portability. All dependencies for the workflow are handled automatically, as each process in the workflow is run in a Docker container. Nextflow is compatible with various computing environments, including high-performance computing clusters and cloud-based computing, allowing users to run the workflow in their preferred environment. Setup requires organizing input files and updating a single configuration file for your computing environment after installing Nextflow and either Docker or Singularity. Nextflow will also handle parallelizing sample processing as allowed by your environment, minimizing run time. The combination of being able to execute a Nextflow workflow in any environment and run individual processes in Docker containers makes this workflow easily portable for external use.

When building `scpca-nf`, we sought a fast and memory-efficient tool for gene expression quantification to minimize processing costs. We expected many users of the Portal to have their own single-cell or single-nuclei data processed with Cell Ranger [27,28], due to its popularity. Thus, selecting a tool with comparable results to Cell Ranger was also desirable. In comparing `alevin-fry` [12] to Cell Ranger, we found `alevin-fry` had a lower run time and memory usage (Figure S1A), while retaining comparable mean gene expression for all genes (Figure S1B), total UMIs per cell (Figure S1C), and total genes detected per cell (Figure S1D). (All analyses comparing gene expression quantification tools are available in a public analysis repository [29].) Based on these results, we elected to use `salmon alevin` and `alevin-fry` [12] in `scpca-nf` to quantify gene expression data.

`scpca-nf` takes FASTQ files as input (Figure 2A). Reads are aligned using the selective alignment option in `salmon alevin` to an index with transcripts corresponding to spliced cDNA and intronic regions, denoted by `alevin-fry` as a `splici` index. The output from `alevin-fry` includes a gene by cell count matrix for all barcodes identified, even those that may not contain true cells. This unfiltered counts matrix is stored in a `SingleCellExperiment` object [9] and output from the workflow as a file with the suffix `_unfiltered.rds`.

`scpca-nf` performs filtering of empty droplets, removal of low-quality cells, normalization, dimensionality reduction, and cell type annotation (Figure 2A). The unfiltered gene by cell counts matrices are filtered to remove any barcodes that are not likely to contain cells using `DropletUtils::emptyDropsCellRanger()` [30], and all cells that pass are saved in a `SingleCellExperiment` object and a file with the suffix `_filtered.rds`. Low-quality cells are identified and removed with `miQC` [31], which jointly models the proportion of mitochondrial reads and detected genes per cell and calculates a probability that each cell is compromised. The remaining cells' counts are normalized [32], and reduced-dimension representations are calculated using both principal component analysis (PCA) and uniform manifold approximation and projection (UMAP) [33]. Finally, cell types are classified using two automated methods, `SingleR` [34] and `CellAssign` [35]. The results from this analysis are stored in a processed `SingleCellExperiment` object saved to a file with the suffix `_processed.rds`.

To make downloading from the Portal convenient for R and Python users, downloads are available as either `SingleCellExperiment` or `AnnData` [36] objects. `scpca-nf` converts all `SingleCellExperiment` objects to `AnnData` objects, which are saved as `.hdf5` files (Figure 2A). Downloads contain the unfiltered, filtered, and processed objects from `scpca-nf` to allow users to choose to perform their own filtering and normalization or to start their analysis from a processed object.

All downloads from the Portal include a quality control (QC) report with a summary of processing information (e.g., `alevin-fry` version), library statistics (e.g., the total number of cells), and a collection of diagnostic plots for each library (Figure 2B-G). A knee plot displaying total UMI counts for all droplets (i.e., including empty droplets) indicates the effects of the empty drop filtering (Figure 2B).

For each cell that remains after filtering empty droplets, the number of total UMIs, genes detected, and mitochondrial reads are calculated and summarized in a scatter plot (Figure 2C). We include plots showing the `miQC` model and which cells are kept and removed after filtering with `miQC` (Figure 2D-E). A UMAP plot with cells colored by the total number of genes detected and a faceted UMAP plot where cells are colored by the expression of a set of highly variable genes are also provided (Figure 2F-G).

Processing samples with additional modalities

`scpca-nf` includes modules for processing samples with sequencing modalities beyond single-cell or single-nuclei RNA-seq data: corresponding ADT or CITE-seq data [25], multiplexed data via cell hashing [26], spatial transcriptomics, or bulk RNA-seq.

Antibody-derived tags

To process ADT libraries, the ADT FASTQ files were provided as input into `scpca-nf` and quantified using `salmon alevin` and `alevin-fry` (Figure S2A). Along with the FASTQ files, `scpca-nf` takes a tab-separated values (TSV) file with one row for each ADT – containing the name used for the ADT and associated barcode – required to build an ADT-specific index for quantifying ADT expression with `alevin-fry`. The output from `alevin-fry` is the unfiltered ADT by cell counts matrix. The ADT by cell counts matrix is read into R alongside the gene by cell counts matrix and saved as an alternative experiment (`altExp`) within the main `SingleCellExperiment` object containing the unfiltered RNA counts. This `SingleCellExperiment` object containing both RNA and ADT counts is output from the workflow to a file with the suffix `_unfiltered.rds`.

`scpca-nf` does not filter any cells based on ADT expression or remove cells with low-quality ADT expression. Any cells removed after filtering empty droplets based on the unfiltered RNA counts matrix are also removed from the ADT counts matrix. The workflow calculates QC statistics for ADT counts using `DropletUtils::cleanTagCounts()` that are stored alongside the ADT by cell counts matrix in the filtered `SingleCellExperiment` object. The `SingleCellExperiment` object containing the filtered RNA and ADT counts matrix and associated ADT QC statistics is saved to a file with the suffix `_filtered.rds`.

The ADT by cell counts matrix is normalized by first determining the ambient profile and then using that profile to calculate median size factors with `scuttle::computeMedianFactors()` [37,38]. We skip normalization for cells with low-quality ADT expression, as indicated by `DropletUtils::cleanTagCounts()`. Although `scpca-nf` normalizes ADT counts, the workflow does not perform any dimensionality reduction of ADT data; only the RNA counts data are used as input for dimensionality reduction. The normalized ADT data are saved as an `altExp` within the processed `SingleCellExperiment` containing the normalized RNA data and is output to a file with the suffix `_processed.rds`. All files containing `SingleCellExperiment` objects and associated `altExp` objects are converted to `AnnData` objects and exported as separate RNA (`_rna.hdf5`) and ADT (`_adt.hdf5`) `AnnData` objects.

If a library contains associated ADT data, the QC report output by `scpca-nf` will include an additional section with a summary of ADT-related statistics, such as how many cells express each ADT, and ADT-specific diagnostic plots (Figure S2B-D). As mentioned above, `scpca-nf` uses `DropletUtils::cleanTagCounts()` to calculate QC statistics for each cell using ADT expression but does not filter any cells from the object. We include plots summarizing the potential effects of removing of low-quality cells based on RNA and ADT counts in the QC report (Figure S2B). The first quadrant indicates which cells would be kept if the object was filtered using both RNA and ADT quality

measures. The other facets highlight which cells would be removed if filtering was done using only RNA counts, only ADT counts, or both. The top four ADTs with the most variable expression are also identified and visualized using density plots to show the normalized ADT expression across all cells (Figure S2C) and UMAPs – calculated from RNA data – with cells colored by ADT expression (Figure S2D).

Multiplexed libraries

To process multiplexed libraries, the HTO FASTQ files are input to `scpca-nf` and quantified using `salmon alevin` and `alevin-fry` (Figure S2C). Along with the FASTQ files, `scpca-nf` requires two TSV files to process multiplexed data: one to build an HTO-specific index for quantifying HTO expression with `alevin-fry`, and a second to indicate which HTO was used for which sample when multiplexing the library. The unfiltered HTO by cell counts matrix output from `alevin-fry` is saved as an alternative experiment (`altExp`) within the main `SingleCellExperiment` containing the unfiltered RNA counts. This `SingleCellExperiment` object containing both RNA and HTO counts is output from the workflow to a file with the suffix `_unfiltered.rds`.

As with ADT data, `scpca-nf` does not filter any cells based on HTO expression, and any cells removed after filtering empty droplets based on the unfiltered RNA counts matrix are also removed from the HTO counts matrix with the remainder saved to a file with the `_filtered.rds` suffix. `scpca-nf` does not perform any additional filtering or processing of the HTO by cell counts matrix, so the same filtered matrix is saved to the file with the `_processed.rds` suffix.

Although `scpca-nf` quantifies the HTO data and includes an HTO by cell counts matrix in all objects, `scpca-nf` does not demultiplex the samples into one sample per library. Instead, `scpca-nf` applies multiple demultiplexing methods, including demultiplexing with `DropletUtils::hashedDrops()` [39], demultiplexing with `Seurat::HTODemux()` [26], and genetic demultiplexing when bulk RNA-seq data are available. `scpca-nf` uses the genetic demultiplexing method described in Weber *et al.* [40], which uses bulk RNA-seq as a reference for the expected genotypes found in each single-cell RNA-seq sample. The results from all available demultiplexing methods are saved in the filtered and processed `SingleCellExperiment` objects.

If a library has associated HTO data, an additional section is included in the `scpca-nf` QC report. This section summarizes HTO-specific library statistics, such as how many cells express each HTO. No additional plots are produced, but a table summarizing the results from all three demultiplexing methods is included.

Bulk and spatial transcriptomics

Some samples also included data from bulk RNA-seq and/or spatial transcriptomics libraries. Both of these additional sequencing methods are supported by `scpca-nf`. To quantify bulk RNA-seq data, `scpca-nf` takes bulk FASTQ files as input, trims reads using `fastp` [41], and then aligns and quantifies reads with `salmon` (Figure S3A) [42]. The output is a single TSV file with the gene by sample counts matrix for all samples in a given ScPCA project. This gene by sample matrix is only included with project downloads on the Portal.

To quantify spatial transcriptomics data, `scpca-nf` takes the RNA FASTQ and slide image as input (Figure S3B). As `alevin-fry` does not yet fully support spatial transcriptomics data, `scpca-nf` uses Space Ranger to quantify all spatial transcriptomics data [43]. The output includes the spot by gene matrix along with a summary report produced by Space Ranger.

Downloading projects from the ScPCA Portal

On the Portal, users can select to download data from individual samples or all data from an entire ScPCA project. When downloading data for an entire project, users can choose between receiving the individual files for each sample (default) or one file containing the gene expression data and metadata for all samples in the project as a merged object. Users also have the option to choose their desired format and receive the data as `SingleCellExperiment` (`.rds`) or `AnnData` (`.hdf5`) objects.

For downloads with samples as individual files, the download folder will include a sub-folder for each sample in the project (Figure 3A). Each sample folder contains all three object types (unfiltered, filtered, and processed) in the requested file format and the QC and cell type summary report for all libraries from the given sample. The objects house the summarized gene expression data and associated metadata for the library indicated in the filename.

All project downloads include a metadata file, `single_cell_metadata.tsv`, containing relevant metadata for all samples, and a `README.md` with information about the contents of each download, contact and citation information, and terms of use for data downloaded from the Portal (Figure 3A-B). If the ScPCA project includes samples with bulk RNA-seq, two additional files are included: a gene by sample counts matrix (`bulk_quant.tsv`) with the quantified gene expression data for all samples in the project, and a metadata file (`bulk_metadata.tsv`).

Merged objects

Providing data for all samples within a single file facilitates performing joint gene-level analyses, such as differential expression or gene set enrichment analyses, on multiple samples simultaneously. Therefore, we provide a single, merged object for each project containing all raw and normalized gene expression data and metadata for all single-cell and single-nuclei RNA-seq libraries within a given ScPCA project. We provide merged objects for all projects in the Portal except for those with multiplexing, due to potential ambiguity in identifying samples across multiplexed libraries. The data in the merged object has simply been combined without further processing; no batch-corrected or integrated data are included. If downloading data from an ScPCA project as a single, merged file, the download will include a single `.rds` or `.hdf5` file, a summary report for the merged object, and a folder with all individual QC and cell type reports for each library found in the merged object (Figure 3B).

To build the merged objects, we created an additional stand-alone workflow for merging the output from `scpca-nf`, `merge.nf` (Figure 3C). `merge.nf` takes as input the processed `SingleCellExperiment` objects output by `scpca-nf` for all single-cell and single-nuclei libraries included in a given ScPCA project. The gene expression data stored in all `SingleCellExperiment` objects are then merged to produce a single merged gene by cell counts matrix containing all cells from all libraries. The genes available in the merged object will be the same as those in each individual object, as all objects on the Portal were quantified using the same index. Where possible, library-, cell- and gene-specific metadata found in the individual processed `SingleCellExperiment` objects are also merged. The merged normalized counts matrix is then used to select high-variance genes in a library-aware manner before performing dimensionality reduction with both PCA and UMAP. `merge.nf` outputs the merged and processed object as a `SingleCellExperiment` object.

We also account for additional modalities in `merge.nf`. If at least one library in a project contains ADT data, the raw and normalized ADT data are also merged and saved as an `altExp` in the merged `SingleCellExperiment` object. If any libraries in a project are multiplexed, no merged object is created, as there is no guarantee that a unique HTO was used for each sample in a given project. All

merged `SingleCellExperiment` objects are converted to `AnnData` objects and exported as `.hdf5` files. If the merged object contains an `altExp` with merged ADT data, two `AnnData` objects are exported to create separate RNA (`_rna.hdf5`) and ADT (`_adt.hdf5`) objects.

`merge.nf` outputs a summary report for each merged object, which includes a set of tables summarizing the types of samples and libraries included in the project, such as types of diagnosis, and a faceted UMAP showing all cells from all libraries. In the UMAP, each panel represents a different library included in the merged object, with all cells from the specified library shown in color, while all other cells are gray. An example of this UMAP showing a subset of libraries from an ScPCA project is available in Figure 3D.

Annotating cell types

Assigning cell type labels to single-cell and single-nuclei RNA-seq data is often an essential step in analysis. Cell type annotation requires knowledge of the expected cell types in a dataset and the associated gene expression patterns for each cell type, which is available in publications or other public databases for some biological contexts. Automated cell type annotation methods leveraging public databases are an excellent initial step in the labeling process, as they can be applied consistently and transparently across all samples in a data set. As such, we include cell type annotations determined using two different automated methods, `SingleR` [34] and `CellAssign` [35], in all processed `SingleCellExperiment` and `AnnData` objects available for download on the Portal, saving users analysis time.

Annotating cell types with automated methods like `SingleR` and `CellAssign` requires the use of previously annotated reference data. For `SingleR`, this can be in the form of an annotated gene expression dataset from a microarray, bulk RNA-seq, or single-cell RNA-seq experiment. `CellAssign` requires a matrix of cell types and expected marker genes. Most public annotated reference datasets that can be used with these methods – including those we use for the Portal – are derived from normal tissue, making accurately annotating tumor datasets particularly difficult. Because there are limitations to the annotations provided on the Portal, comparing the two methods and observing consistent cell type annotations across methods can indicate higher confidence in the provided labels. For some ScPCA projects, submitters provided their own curated cell type annotations, including annotation of tumor cells and disease-specific cell states. These submitter-provided annotations can be found in all `SingleCellExperiment` and `AnnData` objects (unfiltered, filtered, and processed).

Choosing cell typing methods and references

`SingleR` is a reference-based annotation method that requires an existing bulk or single-cell RNA-seq dataset with annotations. To identify an appropriate reference to use with `SingleR`, we annotated a small number of samples across multiple disease types with all human-specific references available in the `celldex` package [34]. The output from `SingleR` includes a score matrix containing a score for each cell and all possible cell types found in the reference, where higher scores are associated with assigned cell types. We calculated the delta median statistic for each cell in the dataset by subtracting the median score from the score associated with the assigned cell type label. The delta median statistic helps evaluate how confident `SingleR` is in assigning each cell to a specific cell type, where low delta median values indicate ambiguous assignments and high delta median values indicate confident assignments [44]. Using this measure, we found that the `BlueprintEncodeData` reference [45,46], which includes a variety of normal cell types, tended to perform better than or at least similarly to other references across samples from different disease types (Figure S4). Based on these findings, we used the `BlueprintEncodeData` reference to

annotate cells from all libraries on the Portal, as using a single reference is potentially valuable for cross-project analyses.

In contrast, `CellAssign` is a marker-gene-based annotation method that requires a binary matrix with all cell types and all associated marker genes as the reference. We used the list of marker genes available as part of `PanglaoDB` [47] to construct organ-specific marker gene matrices with marker genes from all cell types listed for the specified organ. Since many cancers may have infiltrating immune cells, all immune cells were also included in each organ-specific reference. For each ScPCA project, we provided the organ-specific marker gene matrix relevant to the disease and tissue type from which the sample was obtained (e.g., for brain tumors, we used a brain-specific marker gene matrix with all brain and immune cell types). If `CellAssign` cannot find a likely cell type from the marker gene matrix, it does not assign a cell type. Because we annotate cells from tumor samples using references containing only normal cells, we anticipate that many cells, particularly the tumor cells, may not have an exact match; reporting this to the end user is valuable. Indeed, when applying `CellAssign` to tumor samples with our chosen reference, we observed that many of the cells were unassigned. We included an example in Figure S5A where unassigned cell types are labeled with `Unknown`. When comparing annotations obtained from `CellAssign` to submitter-provided annotations, we noticed the labels for non-tumor cells are similar between `CellAssign` and submitter annotations, while the tumor cells were not assigned using `CellAssign` (Figure S5B).

Adding cell type annotations to the ScPCA Portal

`scpca-nf` adds cell type annotations from `SingleR` and `CellAssign` to all processed `SingleCellExperiment` objects (Figure 4A). This requires two additional reference files as input to the workflow: a classification model built from a reference dataset for `SingleR` and a marker gene by cell type matrix for `CellAssign`. `SingleR::trainSingleR()` was used to build a classification model from the provided `BlueprintEncodeData` dataset and create the required `SingleR` input for `scpca-nf`. The classification model and processed `SingleCellExperiment` were used as input for `SingleR::classifySingleR()`, resulting in annotations for all cells and an associated score matrix. The score matrix containing a score for all cells and each possible cell type and the assigned cell types are added to the processed `SingleCellExperiment` object output by `scpca-nf`. Simultaneously, processed `SingleCellExperiment` objects are converted to `AnnData` objects for classification with `CellAssign`. `CellAssign` uses the converted `AnnData` object and the marker gene matrix to train a model and predict the most likely cell type from the possible cell types in the marker gene matrix. The prediction matrix, which contains a probability that each cell is one of each possible cell types, and the assigned cell types are added to the processed `SingleCellExperiment` object output by `scpca-nf`. The processed `SingleCellExperiment` object is then converted to an `AnnData` object to ensure cell type annotations are included in both data formats provided by `scpca-nf`.

An additional cell type report with information about reference sources, comparisons among cell type annotation methods, and diagnostic plots is also output by `scpca-nf`. Tables summarizing the number of cells assigned to each cell type for each method are shown alongside UMAPs coloring cells by the assigned cell type. The concordance of cell type annotations assigned between both methods can indicate higher confidence in the provided annotations. We therefore used the Jaccard similarity index to compare annotations between the two methods, as well as submitter-provided annotations, if available. This index is calculated between pairs of labels from each method and ranges from 0-1, with a value close to 1 indicating high agreement and a high proportion of overlapping cells and values close to 0 indicating a low proportion of overlapping cells. The Jaccard similarity index is displayed in a heatmap, an example of which is shown in Figure 4B.

The report also includes a diagnostic plot evaluating the confidence of cell type annotations determined by each method. To evaluate confidence in `SingleR` cell type annotations, the delta median statistic is calculated by subtracting the median score from the score associated with the assigned cell type label [44]. The distribution of delta median values for each cell type is shown in the cell type report, where a higher delta median statistic for a cell indicates higher confidence in the final cell type annotation (Figure S6A). `CellAssign` calculates the probability that each cell belongs to each possible cell type provided in the reference, and the cell type label with the highest probability is assigned as the cell type for that cell. These values range from 0 to 1, with larger values indicating greater confidence in a given cell type label, so we expect more confident labels to have most values close to 1. An example of the plot included in the report displaying the distribution of all probabilities for each cell type is shown in Figure S6B.

If the submitter provided cell types, the submitter annotations are compared to the annotations from both `SingleR` and `CellAssign`. A summary of this comparison is included in the cell type report along with a table summarizing the submitter cell type annotations and a UMAP plot where each cell is colored by the submitter annotation. The Jaccard similarity index is calculated for all pairs of cell type labels in submitter annotations and `SingleR` annotations and in submitter annotations and `CellAssign` annotations. The results from both comparisons are displayed in a stacked heatmap available in the report, an example of which is shown in Figure S7.

Materials and Methods

Data generation and processing

Raw data and metadata were generated and compiled by each lab and institution contributing to the Portal. Single-cell or single-nuclei libraries were generated using one of the commercially available kits from 10x Genomics. For bulk RNA-seq, RNA was collected and sequenced using either paired-end or single-end sequencing. For spatial transcriptomics, cDNA libraries were generated using the Visium kit from 10x Genomics. All libraries were processed using our open-source pipeline, `scpca-nf`, to produce summarized gene expression data.

Metadata

Submitters were required to submit the age, sex, organism, diagnosis, subdiagnosis (if applicable), and tissue of origin for each sample. The submitted metadata was standardized across projects, including converting all ages to years, removing abbreviations used in diagnosis, subdiagnosis, or tissue of origin, and using standard values across projects as much as possible for diagnosis, subdiagnosis, disease timing, and tissue of origin. For example, all samples obtained at diagnosis were assigned the value `Initial diagnosis` for disease timing.

In an effort to ensure sample metadata for ScPCA are compatible with CZI's CELLxGENE, ontology term identifiers were assigned to metadata categories for each sample following the guidelines present in the CELLxGENE schema [48,49], as shown in Table 1.

Table 1: Assignment of metadata fields to ontology terms.

Metadata field	Ontology term description
Age	Ontology term obtained from HsapDv [13]. For ages 0-11 months, the HsapDv for age in months was used. For ages 12 months and greater, the HsapDv for age in years was used.

Metadata field	Ontology term description
Sex	Ontology term obtained from PATO, either male (PATO:0000384), female (PATO:0000383), or unknown [14,15].
Organism	NCBI taxonomy term for organism. All current samples available on the Portal are from Homo sapiens or NCBITaxon:9606 [16,17].
Diagnosis	The most appropriate MONDO term based on the provided diagnosis [18,19]. An exact match was identified for most samples, but in a handful of cases, the most closely related term was used.
Tissue of origin	The most appropriate UBERON term based on the provided tissue of origin [20,21,22]. An exact match was identified for most samples, but in a handful of cases, the most closely related term was used.
Ethnicity (if applicable)	If the submitter provided ethnicity, the associated Hancestro term [23,24]. If ethnicity is unavailable, `unknown` is used.

Processing single-cell and single-nuclei RNA-seq data with alevin-fry

To quantify RNA-seq gene expression for each cell or nucleus in a library, `scpca-nf` uses `salmon alevin` [50] and `alevin-fry` [12] to generate a gene by cells counts matrix. Prior to mapping, we generated an index using transcripts from both spliced cDNA and unspliced cDNA sequences, denoted as the `splici` index [12]. The index was generated from the human genome, GRCh38, Ensembl version 104. `salmon alevin` was run using selective alignment to the `splici` index with the `--rad` option to generate a reduced alignment data (RAD) file required for input to `alevin-fry`.

The RAD file was used as input to the recommended `alevin-fry` workflow, with the following customizations. At the `generate-permit-list` step, we used the `--unfiltered-pl` option to provide a list of expected barcodes specific to the 10x kit used to generate each library. The `quant` step was run using the `cr-like-em` resolution strategy for feature quantification and UMI de-duplication.

Post alevin-fry processing of single-cell and single-nuclei RNA-seq data

The output from running `alevin-fry` includes a gene by cell counts matrix, with reads from both spliced and unspliced reads for all potential cell barcodes. This output is read into R to create a `SingleCellExperiment` using `fishpond::load_fry()`. The resulting `SingleCellExperiment` contains a `counts` assay with a gene by cell counts matrix where all spliced and unspliced reads for a given gene are totaled together. We also include a `spliced` assay that contains a gene by cell counts matrix with only spliced reads. These matrices include all potential cells, including empty droplets, and are provided for all Portal downloads in the unfiltered objects saved as `.rds` files with the `_unfiltered.rds` suffix.

Each droplet was tested for deviation from the ambient RNA profile using `DropletUtils::emptyDropsCellRanger()` and those with an FDR ≤ 0.01 were retained as likely cells. If a library did not have a sufficient number of droplets and `DropletUtils::emptyDropsCellRanger()` failed, cells with fewer than 100 UMIs were removed. Gene expression data for any cells that remain after filtering are provided in the filtered objects saved as `.rds` files with the `_filtered.rds` suffix.

In addition to removing empty droplets, `scpca-nf` also removes cells that are likely to be compromised by damage or low-quality sequencing. `miQC` was used to calculate the posterior probability that each cell is compromised [31]. Any cells with a probability of being compromised

greater than 0.75 and fewer than 200 genes detected were removed before further processing. The gene expression counts from the remaining cells were log-normalized using the deconvolution method from Lun, Bach, and Marioni [32]. `scran::modelGeneVar()` was used to model gene variance from the log-normalized counts and `scran::getTopHVGs()` was used to select the top 2000 high-variance genes. These were used as input to calculate the top 50 principal components using `scater::runPCA()`. Finally, UMAP embeddings were calculated from the principal components with `scater::runUMAP()`. The raw and log-normalized counts, list of 2000 high-variance genes, principal components, and UMAP embeddings are all stored in the processed objects saved as `.rds` files with the `_processed.rds` suffix.

Quantifying gene expression for libraries with CITE-seq or cell hashing

All libraries with antibody-derived tags (ADTs) or hashtag oligonucleotides (HTOs) were mapped to a reference index using `salmon alevin` and quantified using `alevin-fry`. The reference indices were constructed using the `salmon index` command with the `--feature` option. References were custom-built for each ScPCA project and constructed using the submitter-provided list of ADTs or HTOs and their barcode sequences.

The ADT by cell or HTO by cell counts matrix produced by `alevin-fry` were read into R as a `SingleCellExperiment` object and saved as an alternative experiment (`altExp`) in the same `SingleCellExperiment` object with the unfiltered gene expression counts data. The `altExp` within the unfiltered object contains all identified ADTs or HTOs and all barcodes identified in the RNA-seq gene expression data. Any barcodes that only appeared in either ADT or HTO data were discarded, and cell barcodes that were only found in the gene expression data (i.e., did not appear in the ADT or HTO data) were assigned zero counts for all ADTs and HTOs. Any cells removed after filtering empty droplets were also removed from the ADT and HTO counts matrices and before creating the filtered `SingleCellExperiment` object.

Processing ADT expression data from CITE-seq

The ADT count matrix stored in the unfiltered object was used to calculate an ambient profile with `DropletUtils::ambientProfileEmpty()`. This ambient profile was used to calculate quality-control statistics with `DropletUtils::cleanTagCounts()` for all cells remaining after removing empty droplets. Any negative or isotype controls were taken into account when calculating QC statistics. Cells with a high level of ambient contamination or negative/isotype controls were flagged as having low-quality ADT expression, but we did not remove any cells based on ADT quality from the object. The filtered and processed objects contain the results from running `DropletUtils::cleanTagCounts()`.

ADT count data were then normalized by calculating median size factors using the ambient profile with `scuttle::computeMedianFactors()`. If median-based normalization failed for any reason, ADT counts were log-transformed after adding a pseudocount of 1. Normalized counts are only available for any cells that would be retained after ADT filtering, and any cells that would be filtered out based on `DropletUtils::cleanTagCounts()` are assigned `NA`. The normalized ADT data are available in the `altExp` of the processed object.

Processing HTO data from multiplexed libraries

Although we did not perform any demultiplexing of samples within a multiplexed library, we did apply three different demultiplexing methods. Results from all three methods are included in the filtered and processed `SingleCellExperiment` objects along with the HTO counts data.

Genetic demultiplexing

If all samples in a multiplexed library were also sequenced using bulk RNA-seq, we performed genetic demultiplexing using genotype data from both bulk RNA-seq and single-cell or single-nuclei RNA-seq [40]. If bulk RNA-seq was not available, no genetic demultiplexing was performed.

Bulk RNA-seq reads for each sample were mapped to a reference genome using `STAR` [51], and multiplexed single-cell or single-nuclei RNA-seq reads were mapped to the same reference genome using `STARsolo` [52]. The mapped bulk reads were used to call variants and assign genotypes with `bcftools mpileup` [53]. `cellsnp-lite` was then used to genotype single-cell data at the identified sites found in the bulk RNA-seq data [54]. Finally, `vireo` was used to identify the sample of origin [54].

HTO demultiplexing

For all multiplexed libraries, we performed demultiplexing using `DropletUtils::hashedDrops()` and `Seurat::HTODemux()`. For both methods, we used the default parameters and only performed demultiplexing on the filtered cells present in the filtered object. The results from both these methods are available in the filtered and processed objects.

Quantification of spatial transcriptomics data

10x Genomics' Space Ranger [43] was used to quantify gene expression data from spatial transcriptomics libraries. `cellranger mkref` was used to create a reference index from the human genome, GRCh38, Ensembl version 104. The FASTQ files, microscopic slide image, and slide serial number were provided as input to `spaceranger count`. The raw and filtered counts matrix and the summary report output by `spaceranger count` are included in the folder output from `scpca-nf`.

Quantification of bulk RNA-seq data

`fastp` was used to trim adapters and perform quality and length filtering on all FASTQ files from bulk RNA-seq. We used a decoy-aware reference created from spliced cDNA sequences with the entire human genome sequence (GRCh38, Ensembl version 104) as the decoy [42]. The trimmed reads were then provided as input to `salmon quant` for selective alignment. In addition to using the default parameters for `salmon quant`, we applied the `--seqBias` and `--gcBias` flags to correct for sequence-specific biases due to random hexamer priming and fragment-level GC biases, respectively.

Cell type annotation

Cell type labels determined by both `SingleR` [34] and `CellAssign` [35] were added to processed `SingleCellExperiment` objects. If cell types were obtained from the submitter of the dataset, the submitter-provided annotations were incorporated into all `SingleCellExperiment` objects (unfiltered, filtered, and processed).

To prepare the references used for assigning cell types, we developed a separate workflow `build-celltype-index.nf` within `scpca-nf`. For `SingleR`, we used the `BlueprintEncodeData` from the `celldex` package [45,46] to train the `SingleR` classification model with `SingleR::trainSingleR()`. In the main `scpca-nf` workflow, this model and the processed `SingleCellExperiment` object were input to `SingleR::classifySingleR()`. The `SingleR` output of cell type annotations and a score matrix for each cell and all possible cell types were added

to the processed `SingleCellExperiment` object output. To evaluate confidence in `SingleR` cell type assignments, we also calculated a delta median statistic for each cell by subtracting the median cell type score from the score associated with the assigned cell type [44].

For `CellAssign`, marker gene references were created using the marker gene lists available on `PanglaoDB` [47]. Organ-specific references were built using all cell types in a specified organ listed in `PanglaoDB` to accommodate all ScPCA projects encompassing a variety of disease and tissue types. If a set of disease types in a given project encompassed cells that may be present in multiple organ groups, multiple organs were combined. For example, we created a reference containing bone, connective tissue, smooth muscle, and immune cells for sarcomas that appear in bone or soft tissue.

Given the processed `SingleCellExperiment` object and organ-specific reference, `scvi.external.CellAssign` was used in the main `scpca-nf` workflow to train the model and predict the assigned cell type. For each cell, `CellAssign` calculates a probability of assignment to each cell type in the reference. The probability matrix and a prediction based on the most probable cell type were added as cell type annotations to the processed `SingleCellExperiment` object output.

Generating merged data

Merged objects are created with the `merge.nf` workflow within `scpca-nf`. This workflow takes as input the processed `SingleCellExperiment` objects in a given ScPCA project output by `scpca-nf` and creates a single merged `SingleCellExperiment` object containing gene expression data and metadata from all libraries in that project. The merged object includes both raw and normalized counts for all cells from all libraries. Because the same reference index was used to quantify all single-cell and single-nuclei RNA-seq data, the set of genes is the same in the merged object and the individual objects. Library-, cell- and gene-specific metadata from each of the processed `SingleCellExperiment` objects are also combined and stored in the merged object. The `merge.nf` workflow does not perform batch-correction or integration. The counts in the merged object are therefore not batch-corrected.

The top 2000 shared high-variance genes are identified from the merged counts matrix by modeling variance using `scran::modelGeneVar()` and specifying library IDs for the `block` argument. These genes are used to calculate library-aware principal components with `batchelor::multiBatchPCA()`. The top 50 principal components were selected and used to calculate UMAP embeddings for the merged object.

If any libraries included in the ScPCA project contain additional ADT data, the ADT data are also merged and stored in the `altExp` slot of the merged `SingleCellExperiment` object. By contrast, if any libraries included in the ScPCA project are multiplexed and contain HTO data, no merged object is created.

Converting SingleCellExperiment objects to AnnData objects

`zellkonverter::writeH5AD()` [55] was used to convert `SingleCellExperiment` objects to `AnnData` format and export the objects as `.hdf5` files. For any `SingleCellExperiment` objects containing an `altExp` (e.g., ADT data), the RNA and ADT data were exported and saved separately as RNA (`_rna.hdf5`) and ADT (`_adt.hdf5`) files. Multiplexed libraries were not converted to `AnnData` objects, due to the potential for ambiguity in sample origin assignments.

All merged `SingleCellExperiment` objects were converted to `AnnData` objects and saved as `.hdf5` files. If a merged `SingleCellExperiment` object contained any ADT data, the RNA and ADT data were exported and saved separately as RNA (`_rna.hdf5`) and ADT (`_adt.hdf5`) objects. In contrast, if a merged `SingleCellExperiment` object contained HTO data due to the presence of any multiplexed libraries in the merged object, the HTO data was removed from the `SingleCellExperiment` object and not included in the exported `AnnData` object.

Code and data availability

All summarized gene expression data and de-identified metadata are available for download on the ScPCA Portal, https://scpca.alexslemonade.org/.

Documentation for the Portal can be found at https://scpca.readthedocs.io.

All original code was developed within the following repositories and is publicly available as follows:

- The `scpca-nf` workflow used to process all samples available on the Portal can be found at https://github.com/AlexsLemonade/scpca-nf.
- The Single-cell Pediatric Cancer Atlas Portal code can be found at https://github.com/AlexsLemonade/scpca-portal.
- Benchmarking of tools used to build `scpca-nf` can be found at https://github.com/AlexsLemonade/alsf-scpca/tree/main/analysis and https://github.com/AlexsLemonade/sc-data-integration/tree/main/celltype_annotation.
- All code for the underlying figures can be found at https://github.com/AlexsLemonade/scpca-paper-figures.
- The manuscript can be found at https://github.com/AlexsLemonade/ScPCA-manuscript.

Discussion

Here, we introduced the ScPCA Portal, a downloadable collection of uniformly processed, summarized single-cell and single-nuclei RNA-seq data and de-identified metadata from pediatric tumor samples. The Portal includes 500 samples from over 50 tumor types, making this the most comprehensive collection of publicly available single-cell RNA-seq datasets from pediatric tumor samples to our knowledge. Summarized data are available at three different processing stages: unfiltered, filtered, or processed objects, permitting users to choose to start from a processed object or perform their own processing, such as filtering and normalization. Processed objects containing normalized gene expression data, reduced dimensionality results from PCA and UMAP, and cell type annotations are provided to save researchers time and allow users to skip straight to downstream analysis, such as identifying marker genes or exploring genes of interest. Standardized metadata, containing human-readable values for all fields and ontology term identifiers for a subset of metadata fields, is included in a separate metadata file and the data objects for all samples. Every library includes a quality control report, which lets users assess data quality and identify low-quality libraries that they may wish to exclude from further downstream analyses.

Data on the Portal is available as either `SingleCellExperiment` or `AnnData` objects, so users can work in R or Python with the downloaded data using common analysis systems such as `Bioconductor` or `Scanpy`, depending on their preference. Providing data as `AnnData` objects also means users can easily integrate ScPCA data with data and tools available on other platforms. In particular, the format of the provided `AnnData` objects was designed to be mostly compliant with the requirements of CZI CELLxGENE [56,57,58], but these objects can also be used with UCSC Cell Browser [59,60] or Kana [61,62]. Additionally, users can choose to download a merged

`SingleCellExperiment` or `AnnData` object containing all gene expression data and metadata from all samples in a project. This is helpful for analyzing multiple samples simultaneously and performing analyses such as differential gene expression or gene set enrichment.

To provide users with cell type annotations, we used two automated methods, `SingleR` and `CellAssign`, which use public references. As the publicly available references we used do not contain tumor cells but only normal cells, we recognize that the annotations we provide are limited. Despite these limitations, these methods can provide a good starting point for users, particularly in helping to annotate populations of normal cells that may be present, as normal cells are represented in the reference.

Many samples on the Portal have additional sequencing data, including corresponding ADT data from CITE-seq, cell hashing data, bulk RNA-seq, or spatial transcriptomics, enabling users to gather more information about a single sample than they could from single-cell or single-nuclei RNA-seq alone. Samples with CITE-seq have additional information about cell-surface protein expression in individual cells, which can help determine cell types and correlate RNA to protein expression [25]. Spatial transcriptomics data on the Portal are not single-cell resolution, making it hard to identify cell types and spatial patterns from the spatial data alone. By providing matching single-cell RNA-seq, users can implement analysis tools, like those that use single-cell RNA-seq to deconvolute spatial data, to gain more insights about the spatial data [63]. Similarly, users can gain more insight from bulk RNA-seq data available on the Portal by integrating with single-cell RNA-seq data from the same sample [64,65].

We also introduced our open-source and efficient workflow for uniformly processing datasets available on the Portal, `scpca-nf`, which is available to the entire research community. In one command, `scpca-nf` can process raw data from various sequencing types, turning FASTQ files into processed `SingleCellExperiment` or `AnnData` objects ready for downstream analyses. Using Nextflow as the framework for `scpca-nf` means the workflow is both modular and portable. This makes it easy to add support for more modalities in the future, such as single-cell ATAC-seq, and allows others to run the workflow on their samples in their computing environment, maintaining the security of protected raw data. Processed output from running `scpca-nf` on samples from pediatric tumors, cell lines, or other model organisms is eligible for submission to the ScPCA Portal, enabling us to continue increasing the number of available samples freely available to all researchers.

Figure Titles and Legends

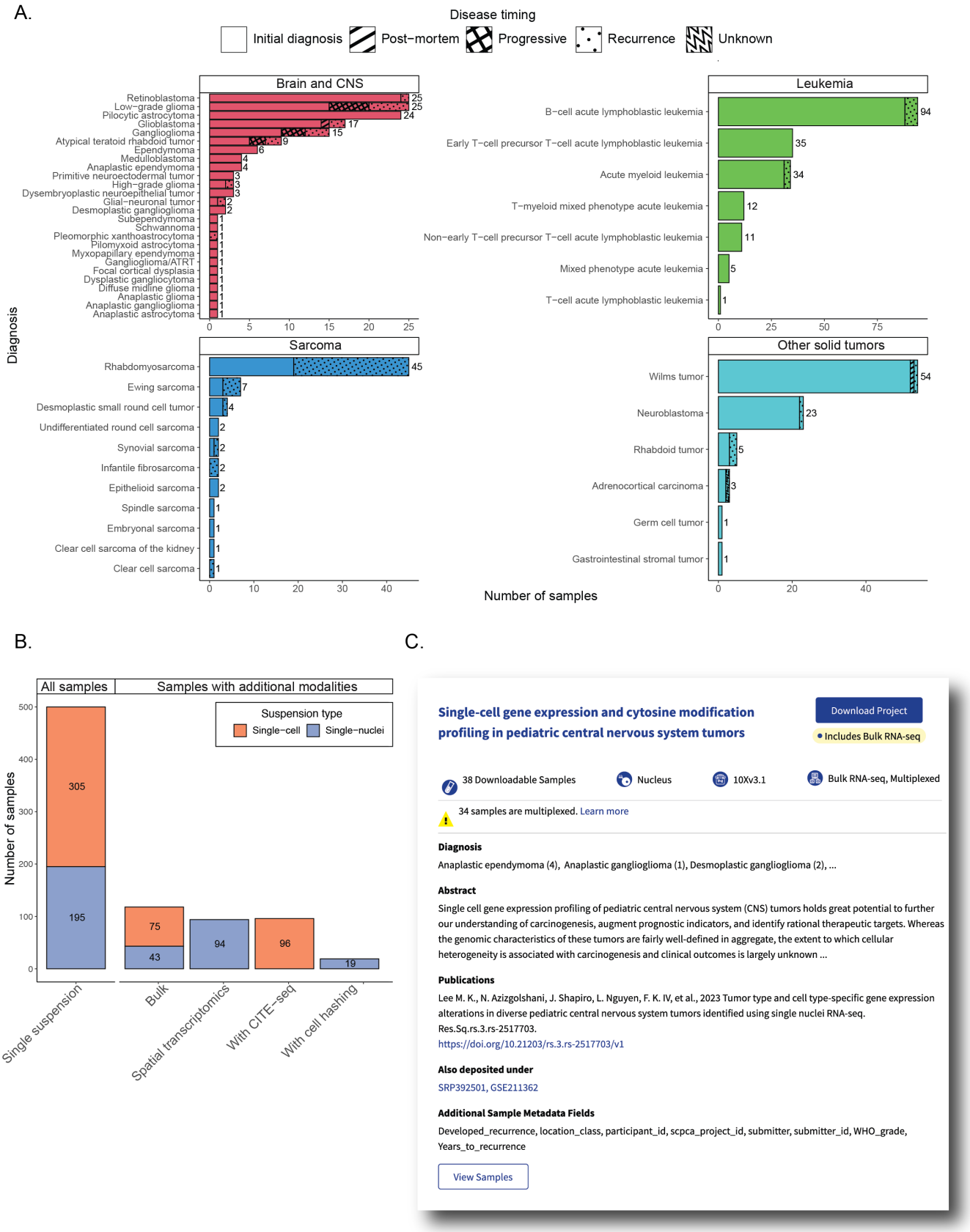


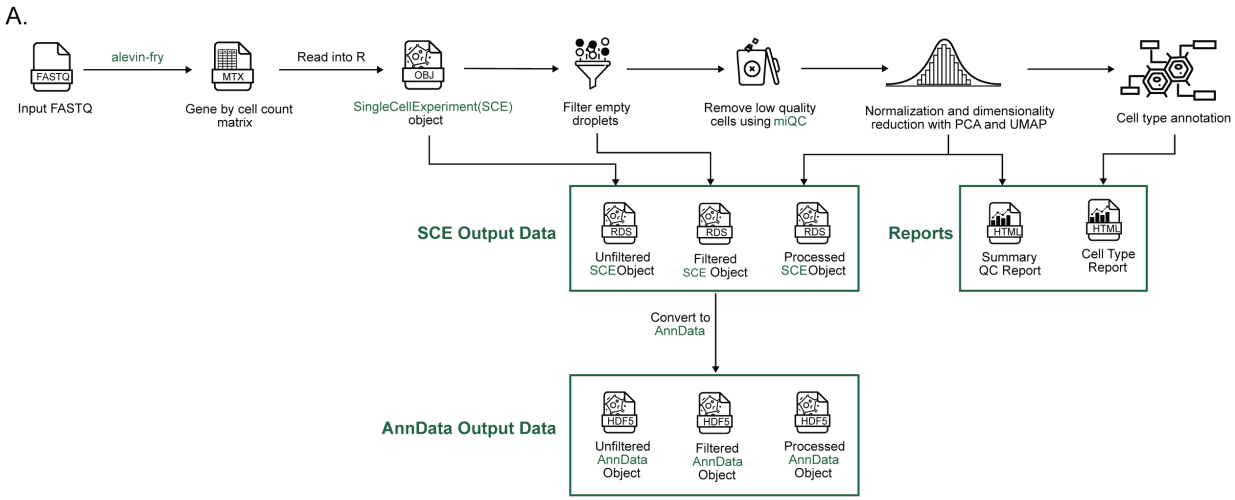
Figure 1: Overview of ScPCA Portal contents.

A. Barplots showing sample counts across four main cancer groupings in the ScPCA Portal, with each bar displaying the number of samples for each cancer type. Each bar is shaded based on the number

of samples with each disease timing, and total sample counts for each cancer type are shown to the right of each bar.

B. Barplot showing sample counts across types of modalities present in the ScPCA Portal. All samples in the portal are shown under the “All Samples” heading. Samples under the “Samples with additional modalities” heading represent a subset of the total samples with the given additional modality. Colors shown for each additional modality indicate the suspension type used, either single-cell or single-nuclei RNA-seq. For example, 75 single-cell samples and 43 single-nuclei samples have accompanying Bulk RNA-seq data.

C. Example of a project card as displayed on the “Browse” page of the ScPCA Portal. This project card is associated with project `SCPCP000009`. Project cards include information about the number of samples, technologies and modalities, additional sample metadata information, submitter-provided diagnoses, and a submitter-provided abstract. Where available, submitter-provided citation information, as well as other databases where this data has been deposited, are also provided.



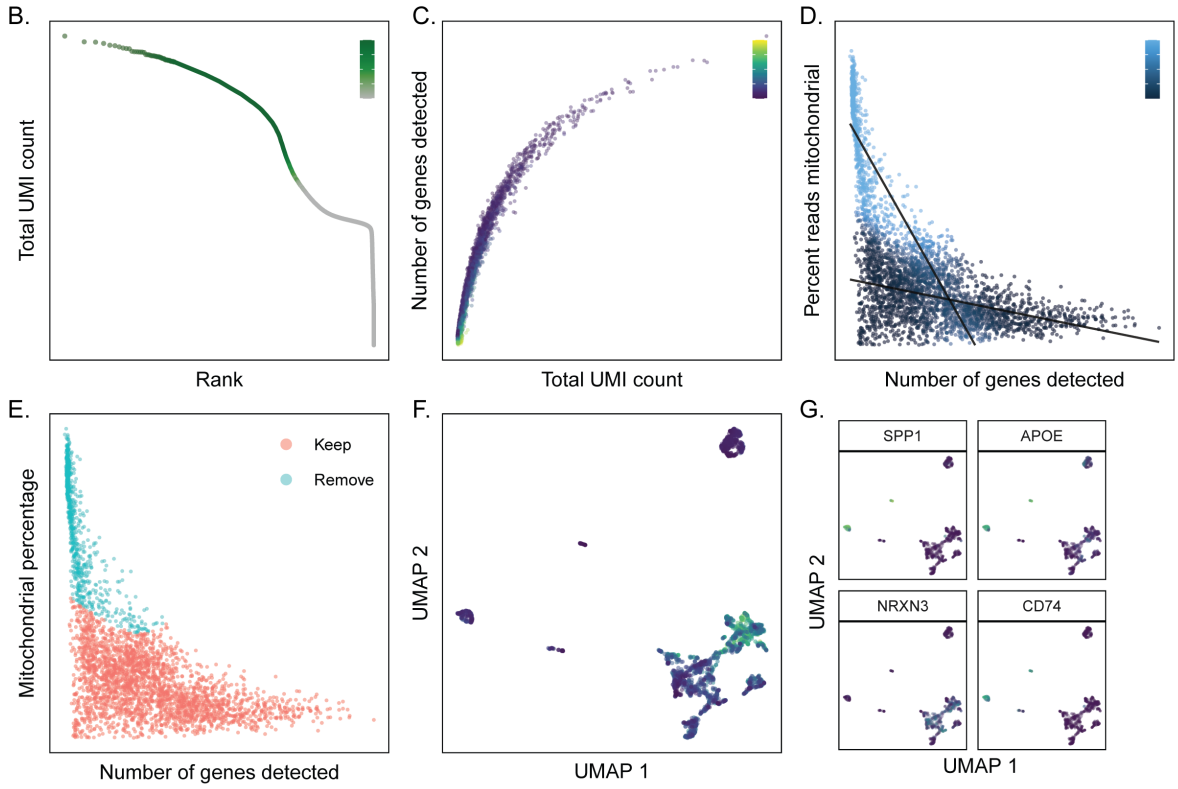


Figure 2: Overview of the `scpca-nf` workflow.

A. Overview of `scpca-nf`, the primary workflow for processing single-cell and single-nuclei RNA-seq data for the ScPCA Portal. Mapping is first performed with `alevin-fry` to generate a gene by cell count matrix, which is read into `R` and converted into a `SingleCellExperiment` (`SCE`) object. This `SCE` object is exported as the `Unfiltered SCE Object` before further post-processing. Next, empty droplets are filtered out, and the resulting `SCE` is exported as the `Filtered SCE Object`. The filtered object undergoes additional post-processing, including removing low-quality cells, normalizing counts, and performing dimension reduction including principal components analysis and UMAP calculation. The object undergoes cell type annotation and is exported as the `Processed SCE Object`. A summary QC report and a supplemental cell type report are prepared and exported. Finally, all `SCE` files are converted to `AnnData` format and exported. Panels B-G show example figures that appear in the summary QC report, shown here for `SCPCL000001`, as follows.

B. The total UMI count for each cell in the `Unfiltered SCE Object`, ordered by rank. Points are colored by the percentage of cells that pass the empty droplets filter.

C. The number of genes detected in each cell passing the empty droplets filter against the total UMI count. Points are colored by the percentage of mitochondrial reads in the cell.

D. `miQC` model diagnostic plot showing the percent of mitochondrial reads in each cell against the number of genes detected in the `Filtered SCE Object`. Points are colored by the probability that the cell is compromised as determined by `miQC`.

E. The percent of mitochondrial reads in each cell against the number of genes detected in each cell. Points are colored by whether the cell was kept or removed, as determined by both `miQC` and a minimum unique gene count cutoff, prior to normalization and dimensionality reduction.

F. UMAP embeddings of log-normalized RNA expression values where each cell is colored by the number of genes detected.

G. UMAP embeddings of log-normalized RNA expression values for the top four most variable genes, colored by the given gene's expression. In the actual summary QC report, the top 12 most highly variable genes are shown.

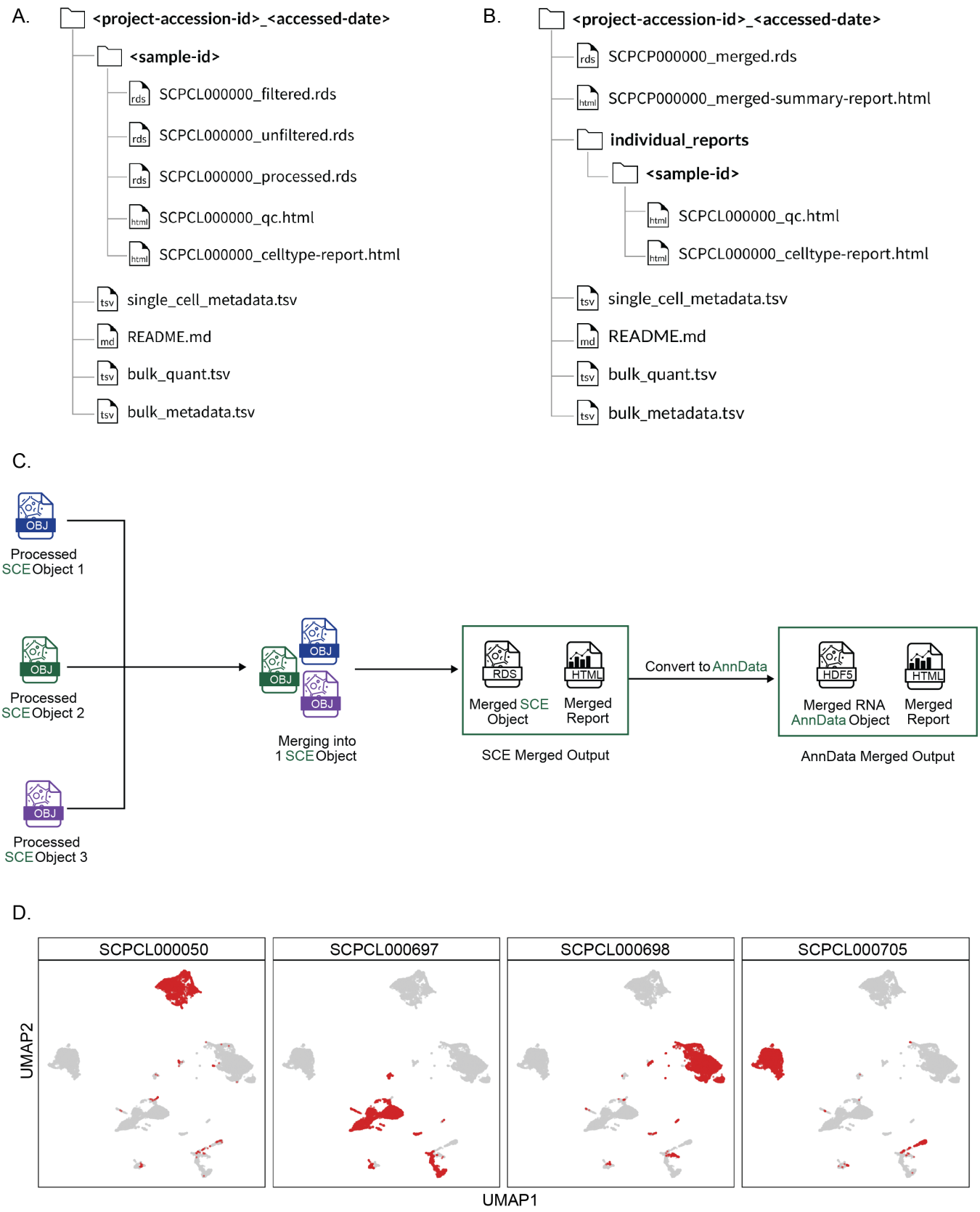


Figure 3: ScPCA Portal project download file structure and merged object workflow.

A. File download structure for an ScPCA Portal project download in `SingleCellExperiment` (`SCE`) format. The download folder is named according to both the project ID and the date it was downloaded. Download folders contain one folder for each sample ID, each containing the three versions (unfiltered, filtered, and processed) of the expression data as well as the summary QC report and cell type report all named according to the ScPCA library ID. The `single_cell_metadata.tsv` file contains sample metadata for all samples included in the download. The `README.md` file provides information about the contents of each download file, additional contact and citation information,

and terms of use for data downloaded from the ScPCA Portal. The files `bulk_quant.tsv` and `bulk_metadata.tsv` are only present for projects that also have bulk RNA-Seq data and contain, respectively, a gene by sample matrix of raw gene expression as quantified by `salmon`, and associated metadata for all samples with bulk RNA-Seq data.

B. File download structure for an ScPCA Portal merged project download in `SCE` format. The download folder is named according to both the project ID and the date it was downloaded. Download folders contain a single merged object containing all samples in the given project as well as a summary report briefly detailing the contents of the merged object. All summary QC and cell type reports for each individual library are also provided in the `individual_reports` folder arranged by their sample ID. As in panel (A), additional files `single_cell_metadata.tsv`, `bulk_quant.tsv`, `bulk_metadata.tsv`, and `README.md` are also included.

C. Overview of the merged workflow. Processed `SCE` objects associated with a given project are merged into a single object, including ADT counts from CITE-seq data if present, and a merged summary report is generated. Merged objects are available for download either in `SCE` or `AnnData` format.

D. Example of UMAPs as shown in the merged summary report. A grid of UMAPs is shown for each library in the merged object, with cells in the library of interest shown in red and all other cells belonging to other libraries shown in gray. The UMAP is constructed from the merged object such that all libraries contribute an equal weight, but no batch correction was performed. The libraries pictured are a subset of libraries in the ScPCA project `SCPCP000003`.

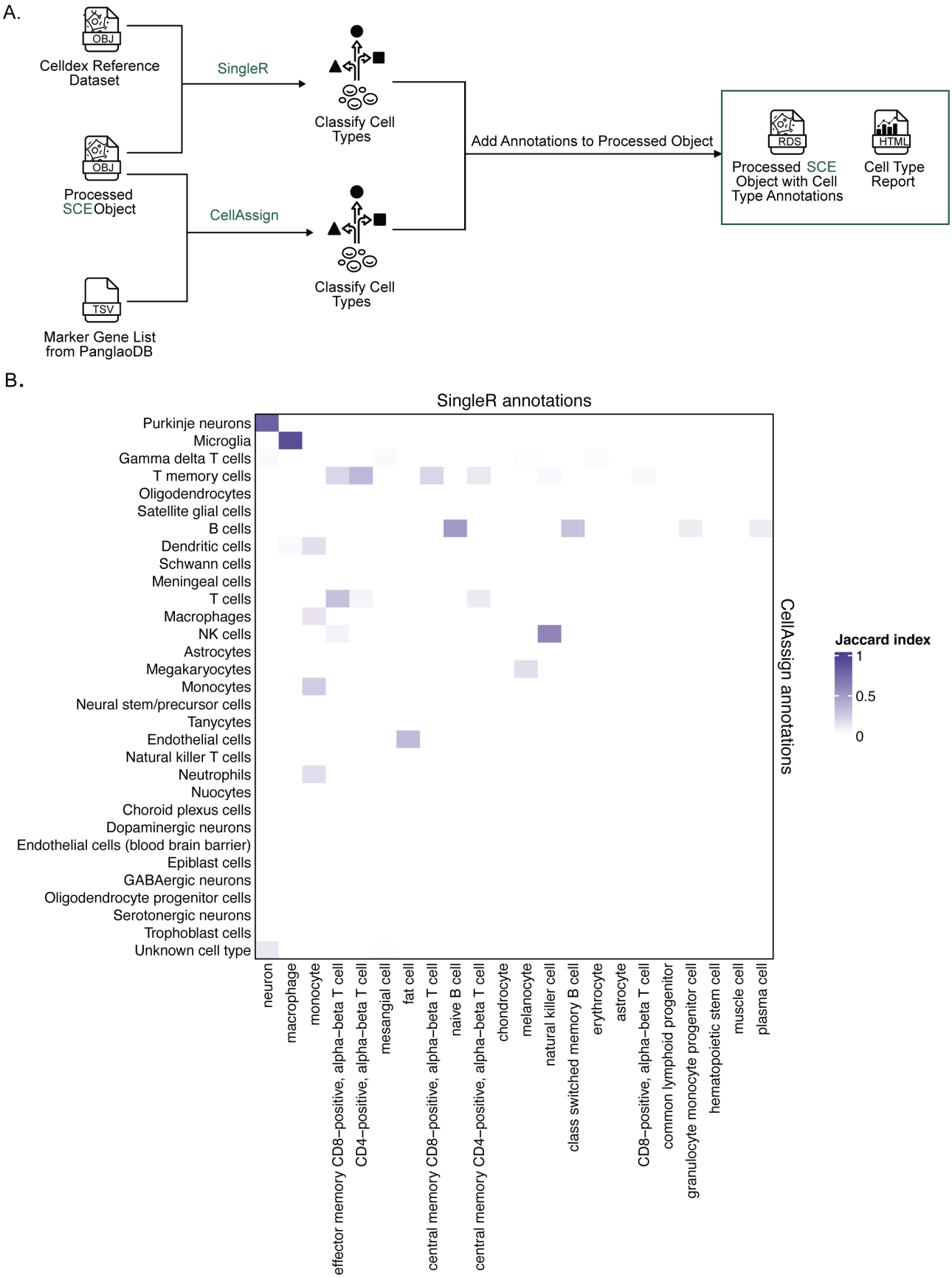


Figure 4: Cell type annotation in `scpca-nf`.

A. Expanded view of the process for adding cell type annotations within `scpca-nf`, as introduced in Figure 2A. Cell type annotation is performed on the `Processed SCE Object`. A `celldex` [34] reference dataset with ontology labels is used as input for annotation with `SingleR` [34], and a list of marker genes compiled from `PanglaoDB` [47] is used as input for annotation with `CellAssign` [35]. Results from cell type annotation are then added to the `Processed SCE Object`, and a cell

type summary report with information about reference sources, comparisons among cell type annotation methods, and diagnostic plots is created. Although not shown in this panel, cell type annotations are also included in the `Processed AnnData Object` created from the `Processed SCE Object` (Figure 2A).

B. Example heatmap as shown in the cell type summary report comparing annotations with `SingleR` and `CellAssign`. Heatmap cells are colored by the Jaccard similarity index. A value of 1 means that there is complete overlap between which cells are annotated with the two labels being compared, and a value of 0 means that there is no overlap between which cells are annotated with the two labels being compared. The heatmap shown is from library `SCPCL000498`.

Supplementary Figures and Tables

Table S1. Overview of ScPCA Portal Datasets. This table provides descriptions and sample and library counts for each project in the ScPCA Portal.

`scpca_project_id`: ScPCA project unique identifier. `Diagnosis group`: Diagnosis group as shown in Figure 1. `Diagnoses`: Full set of diagnoses for all samples associated with the project. `Total number of samples (S)`: Number of samples associated with the project. `Total number of libraries (L)`: Number of libraries associated with the project. Due to additional sequencing modalities and/or multiplexing, projects may have more libraries than samples. All remaining columns give the number of libraries (as designated with `(L)`) with the given suspension type or additional modality.

Table S2. Summary of references used for cell type annotation with `CellAssign`. This table provides a summary of the references used for assigning cell types for ScPCA projects using `CellAssign`. All references were built using all cell types from a specified set of organs present in `PanglaoDB`'s marker gene list.

`scpca_project_id`: ScPCA project unique identifier. `Diagnoses`: Full set of diagnoses for all samples associated with the project. `ScPCA reference name`: Name used to describe the custom reference. `PanglaoDB organs included in reference`: A list of all organs included in the reference with names of organs corresponding to organs listed in `PanglaoDB`. The reference includes marker genes for all cell types present in each organ.

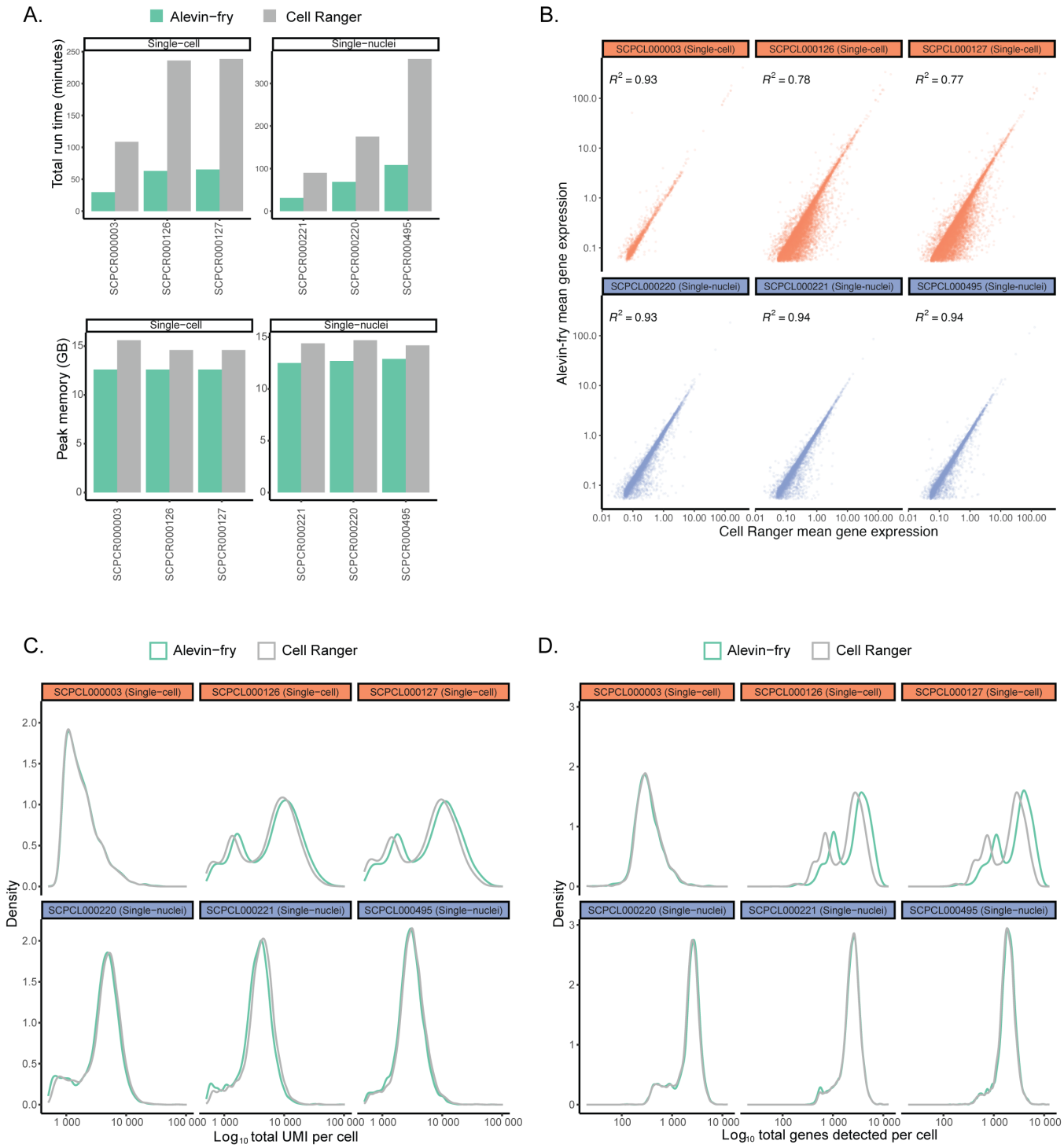


Figure S1: Results from benchmarking `alevin-fry` and `CellRanger` performance.

Each panel compares metrics for six representative ScPCA libraries, including three single-cell and three single-nuclei suspensions, obtained from processing libraries with both `alevin-fry` and `CellRanger`.

A. Runtime in minutes (top row) and peak memory in GB (bottom row) for six ScPCA libraries processed with `alevin-fry` and `CellRanger`. Processing with `alevin-fry` was consistently faster and more memory-efficient compared to processing with `CellRanger`.

Panels B-D show only cells present in both the `alevin-fry` and `CellRanger` output.

B. Comparison of mean gene expression values for six ScPCA libraries processed with `alevin-fry` and `CellRanger`, shown on a log-scale. Each point is a gene, and only genes detected in at least 5

cells are shown. R^2 values shown in the top left corner of each panel reflect broad agreement in mean gene expression values between platforms.

C. Comparison of log total UMI counts for six ScPCA libraries processed with `alevin-fry` and `CellRanger`. Distributions reflect broad agreement in the total UMI count per cell between platforms, although `alevin-fry` returned slightly higher values for certain single-cell libraries.

D. Comparison of log total genes detected per cell for six ScPCA libraries processed with `alevin-fry` and `CellRanger`. Distributions reflect broad agreement between platforms in the total number of genes detected per cell between platforms, although `alevin-fry` returned slightly higher values for certain single-cell libraries.

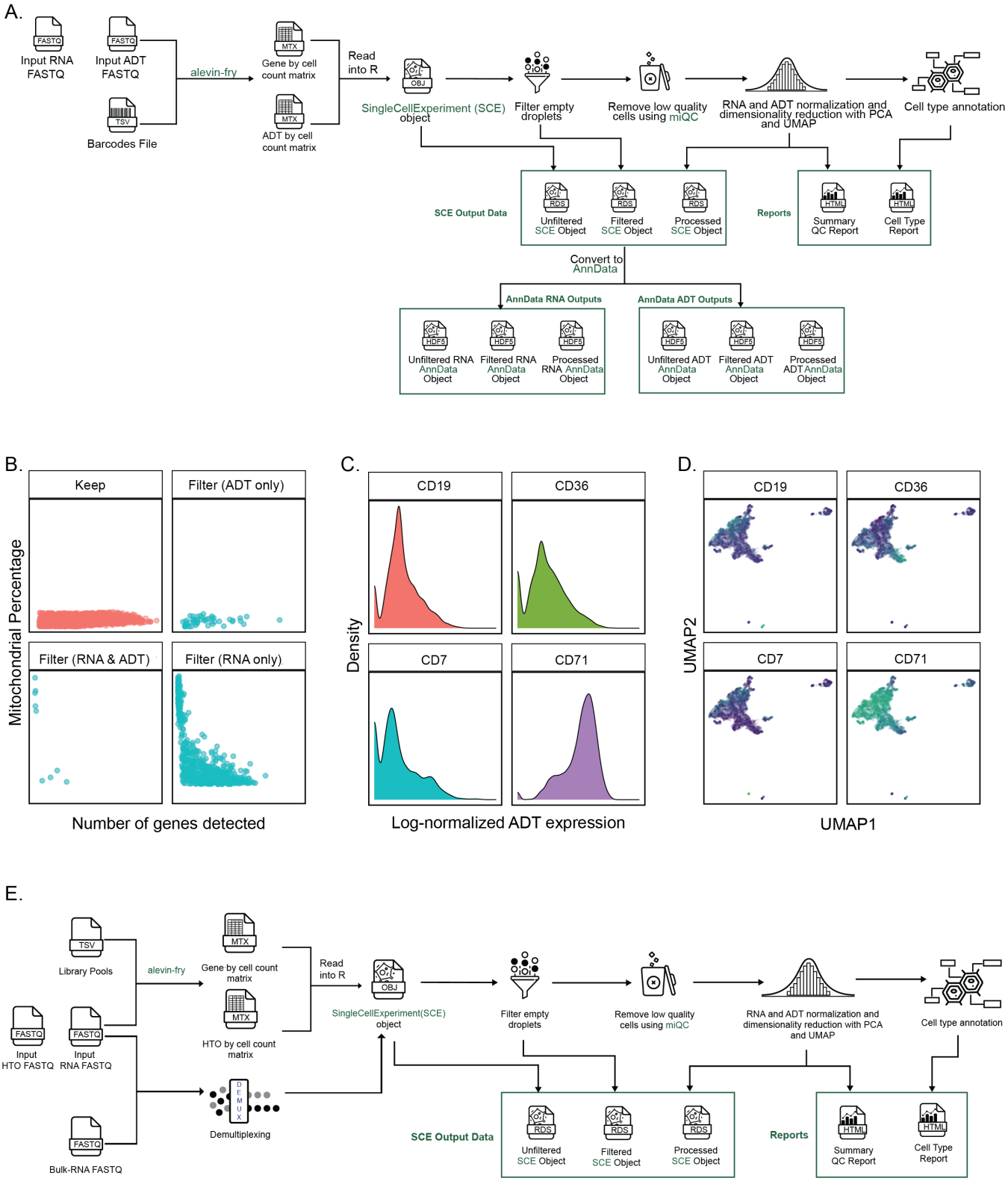


Figure S2: Processing additional single-cell modalities in `scpca-nf`.

A. Overview of the `scpca-nf` workflow for processing libraries with CITE-seq or antibody-derived tag (ADT) derived data. The workflow mirrors that shown in Figure 2A with several differences accounting for the presence of ADT data. First, both an RNA and ADT FASTQ file are required as input to `alevin-fry`, along with a TSV file containing information about ADT barcodes. The gene by cell and ADT by cell count matrices are produced and read into `R` to create a `SingleCellExperiment` (SCE) object. Second, during post-processing, statistics are calculated to filter cells based on ADT counts, but the filter is not applied. ADT counts are also normalized and included in the `Processed SCE Object`. Third, the summary QC report will include a `CITE-seq` section with additional information about

ADT-level processing. Fourth, the workflow exports `SCE` objects containing both RNA and ADT results, while separate `AnnData` objects for RNA and ADT are exported.

Panels B-D show example figures that appear in the CITE-seq section of the summary QC report, shown here for `SCPCL000290`.

B. The percent of mitochondrial reads in each cell against the number of genes detected in each cell. The panel labeled “Keep” displays cells that are retained based on both RNA and ADT counts. The panel labeled “Filter (ADT only)” displays cells that are filtered based on only ADT counts. The panel labeled “Filter (RNA only)” displays cells that are filtered based on only RNA counts. The panel labeled “Filter (RNA & ADT)” panel displays cells that are filtered based on both RNA and ADT counts.

C. Density plots of the log-normalized ADT counts shown for the four most variable ADTs in the library.

D. UMAP embeddings of log-normalized RNA expression values where each cell is colored by the expression of the given highly-variable ADT.

E. Overview of the `scpca-nf` workflow for multiplexed libraries. The workflow mirrors that shown in Figure 2A with several differences accounting for the presence of multiplexed data. First, both an RNA and HTO FASTQ file are required as input to `alevin-fry`, along with a TSV file providing information about library pools. The gene by cell and HTO by cell count matrices are produced and read into `R` to create a `SingleCellExperiment` (SCE) object. Second, in parallel, the RNA FASTQ file, the HTO FASTQ file, and, if available, a corresponding Bulk RNA FASTQ file for each sample present in the multiplexed library are provided to a demultiplexing subprocess. The workflow calculates demultiplexing results based on HTO counts, as well as genetic demultiplexing results if the library has corresponding bulk RNA FASTQ files. Demultiplexing results are stored in all exported `SCE` objects (`Unfiltered`, `Filtered`, and `Processed`), but libraries themselves are not demultiplexed. Third, only `SCE` files are provided for multiplexed libraries; no corresponding `AnnData` files are provided.

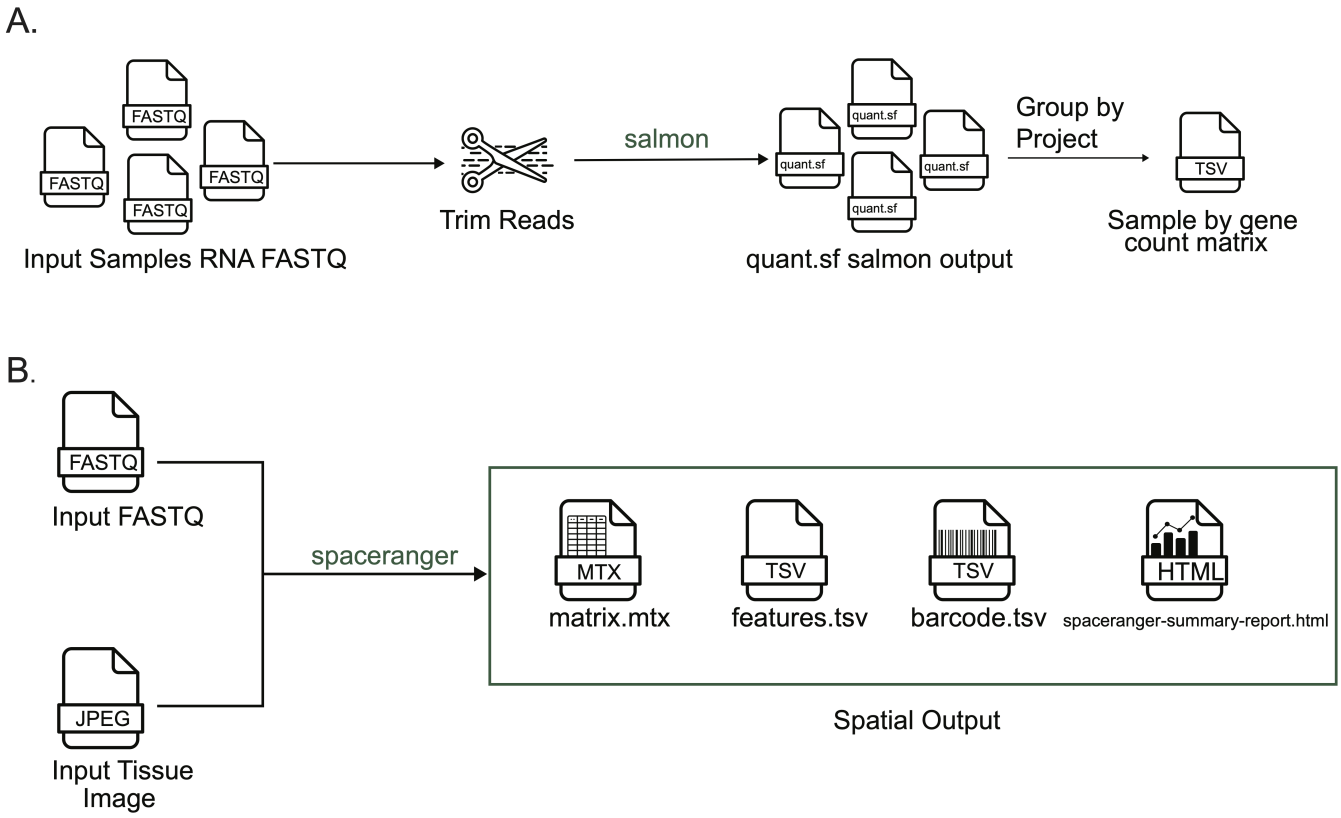


Figure S3: Processing other sequencing modalities with `scpca-nf`.

A. Overview of the bulk RNA-Seq workflow. A set of FASTQ files from libraries sequenced with bulk RNA-seq are provided as input. Reads are trimmed using `fastp`, and `salmon` is used to map reads and quantify counts. The quantified gene expression files output from `salmon` are then grouped by ScPCA Project ID, and a sample by gene count matrix is exported for each Project in TSV format.

B. Overview of the spatial transcriptomics workflow. The FASTQ file and tissue image for a given library are provided as input to `spaceranger`. The workflow directly returns the results from running `spaceranger` without any further processing.

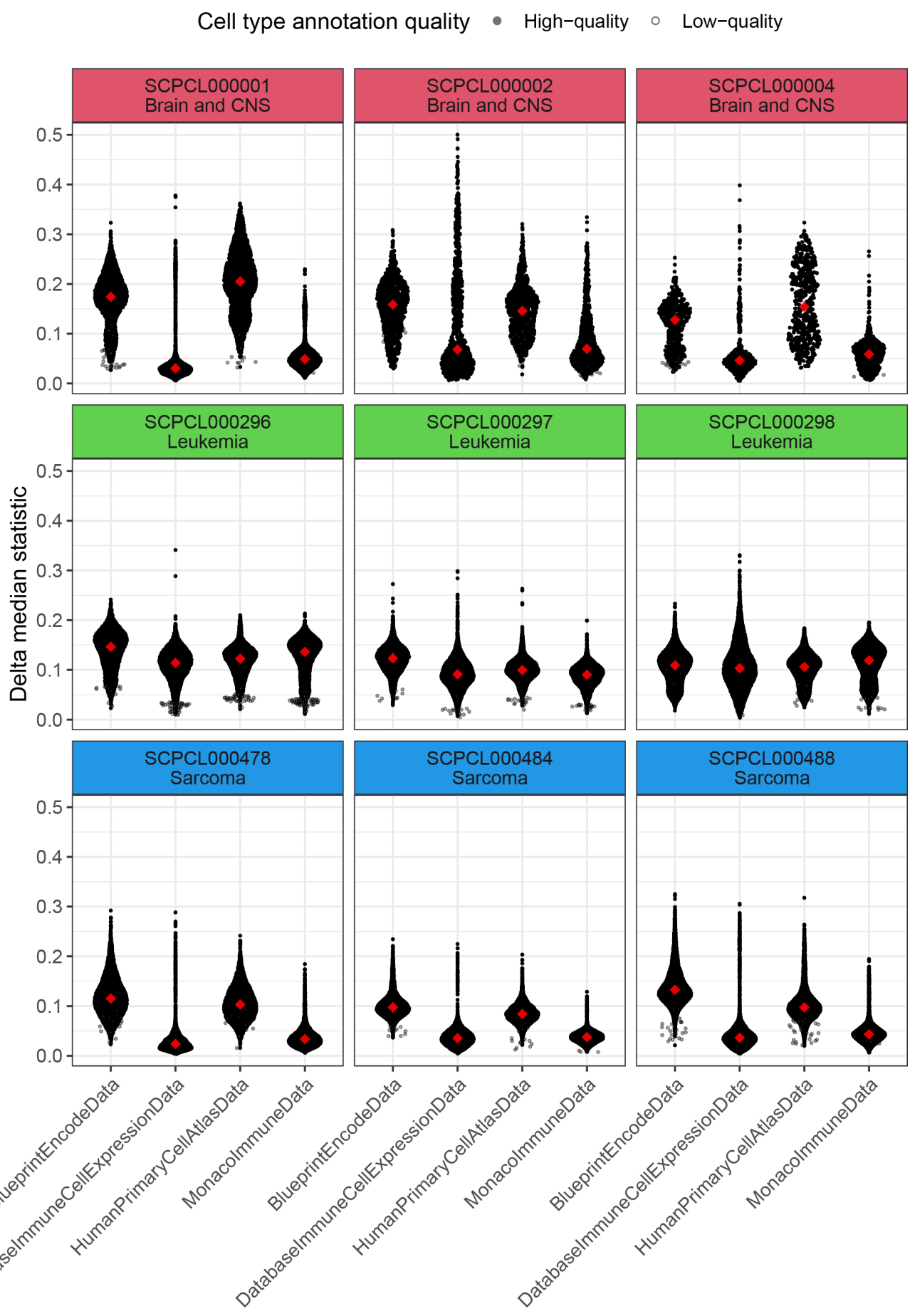
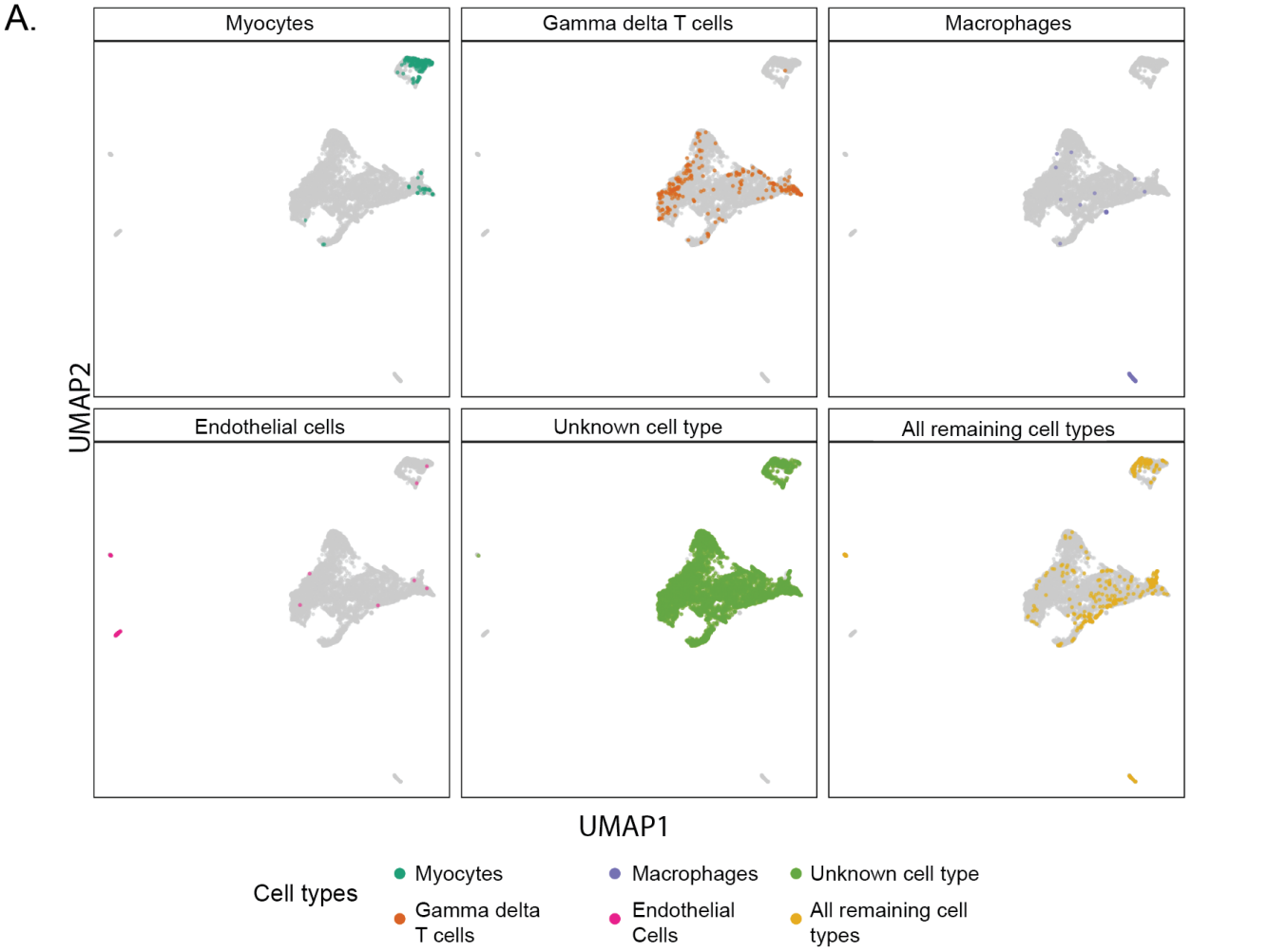
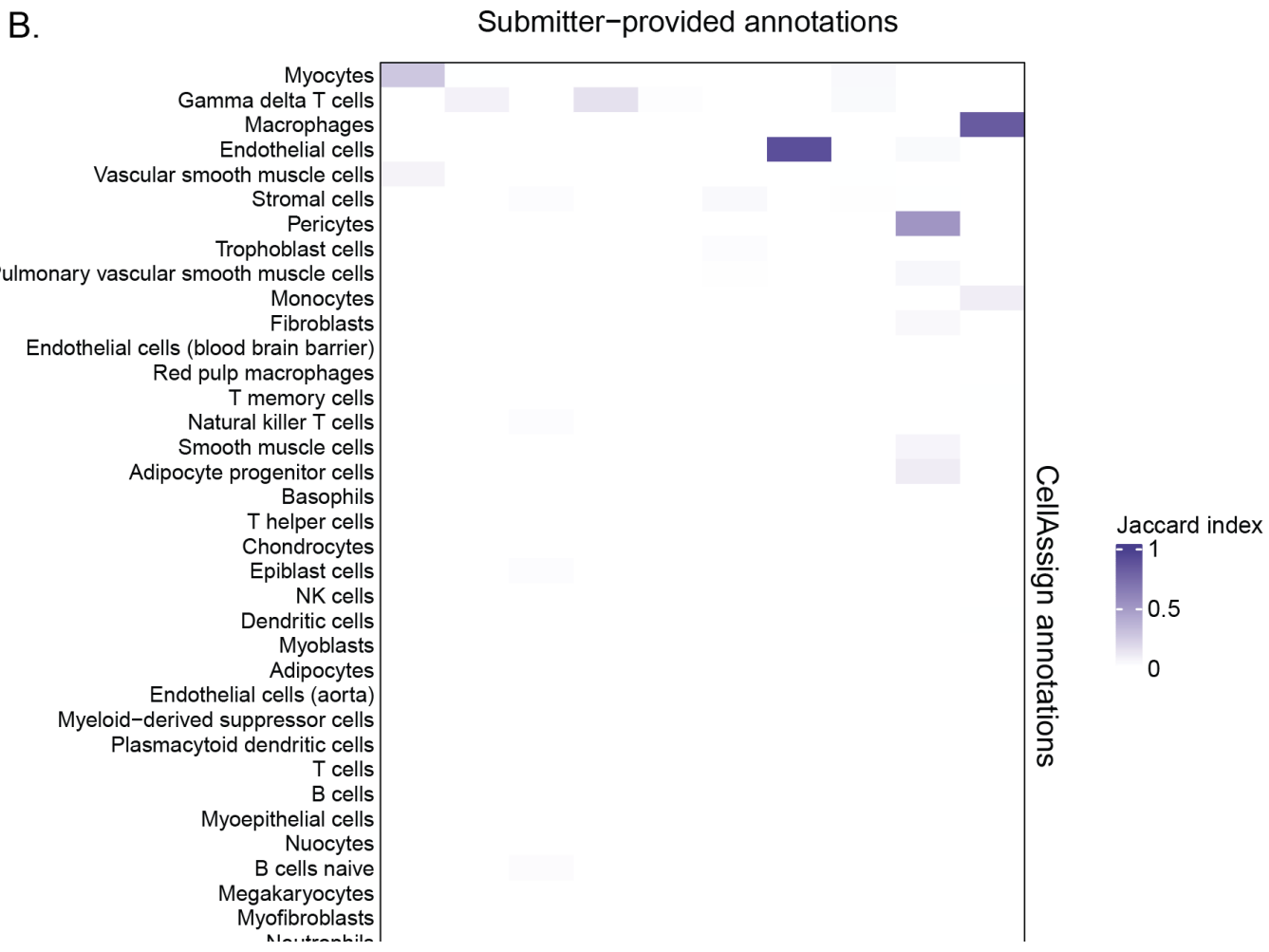


Figure S4: Evaluation of references available in the celldex package for use with SingleR.

`SingleR` was used to annotate ScPCA libraries using four different human-specific references from the `celldex` package. Libraries represent three different diagnosis groups in the ScPCA Portal -

Brain and CNS, Leukemia, and Sarcoma - as indicated in the labels for the individual panels. The distribution of the delta median statistic, calculated for each cell by subtracting the median delta score from the score of the annotated cell type label, is shown on the y-axis, while the `celldex` reference used is shown on the x-axis. Higher values indicate a higher quality cell type annotation, although there is no absolute scale for these values. Each black point represents a cell, where closed circles denote cells with high-quality annotations and open circles denote cells with low-quality annotations, as assessed by `SingleR`. Red diamonds represent the median delta median score for all cells with high-quality annotations in that library.


A.
Myocytes
Gamma delta T cells
Macrophages
Endothelial cells
Unknown cell type
All remaining cell types
UMAP2
UMAP1
Cell types
Myocytes
Gamma delta T cells
Macrophages
Endothelial Cells
Unknown cell type
All remaining cell types


B.
Submitter−provided annotations
Myocytes
Gamma delta T cells
Macrophages
Endothelial cells
Vascular smooth muscle cells
Stromal cells
Pericytes
Trophoblast cells
Pulmonary vascular smooth muscle cells
Monocytes
Fibroblasts
Endothelial cells (blood brain barrier)
Red pulp macrophages
T memory cells
Natural killer T cells
Smooth muscle cells
Adipocyte progenitor cells
Basophils
T helper cells
Chondrocytes
Epiblast cells
NK cells
Dendritic cells
Myoblasts
Adipocytes
Endothelial cells (aorta)
Myeloid−derived suppressor cells
Plasmacytoid dendritic cells
T cells
B cells
Myoepithelial cells
Nuocytes
B cells naive
Megakaryocytes
Myofibroblasts
CellAssign annotations
Jaccard index
1
0.5
0

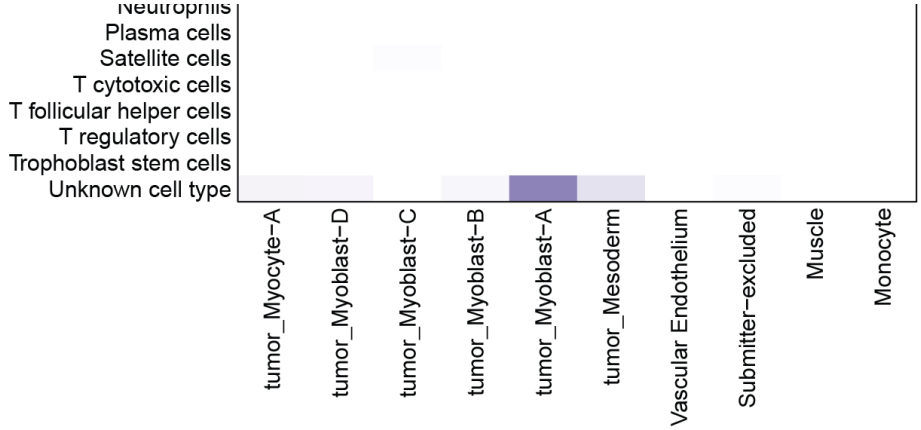
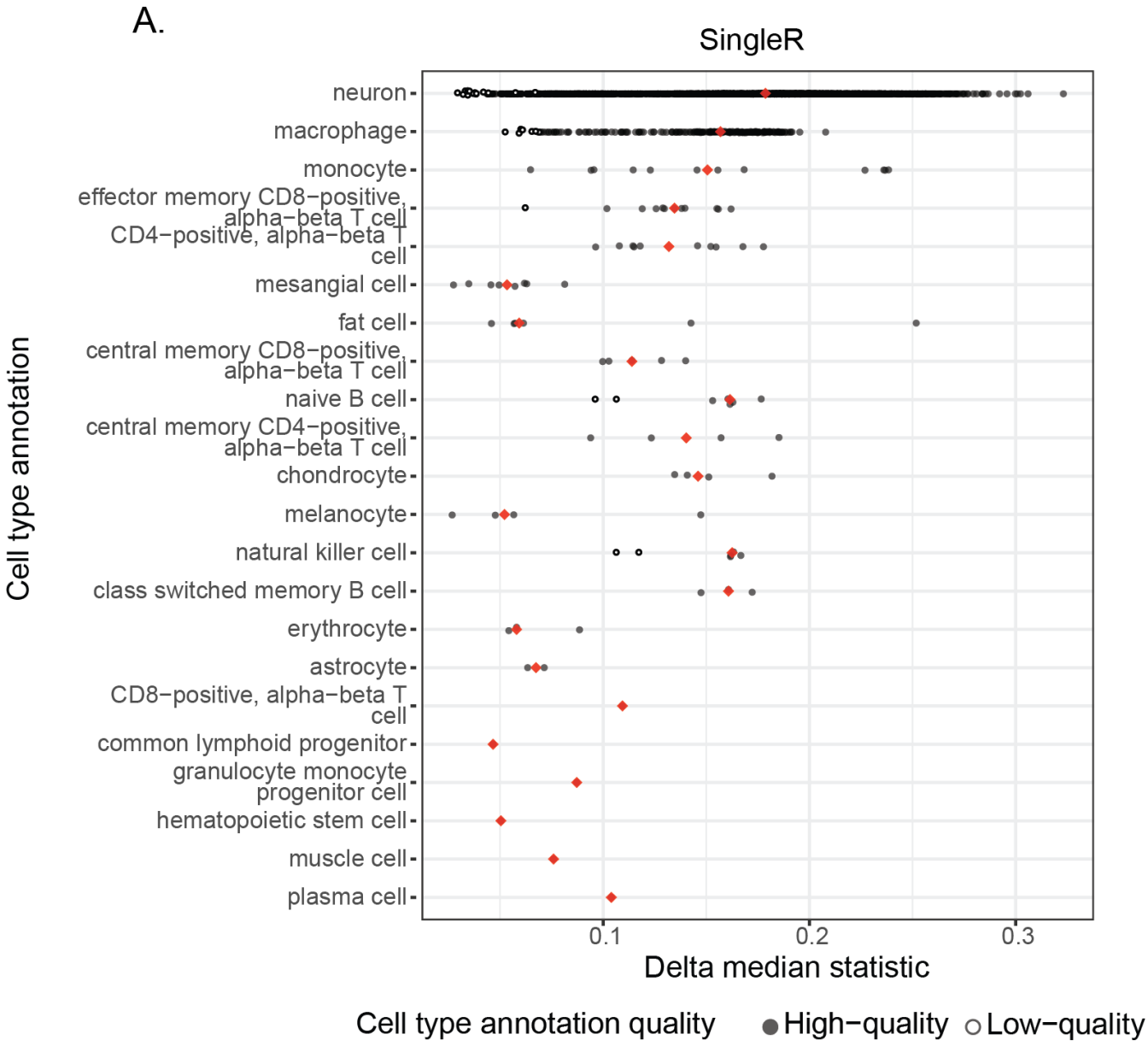


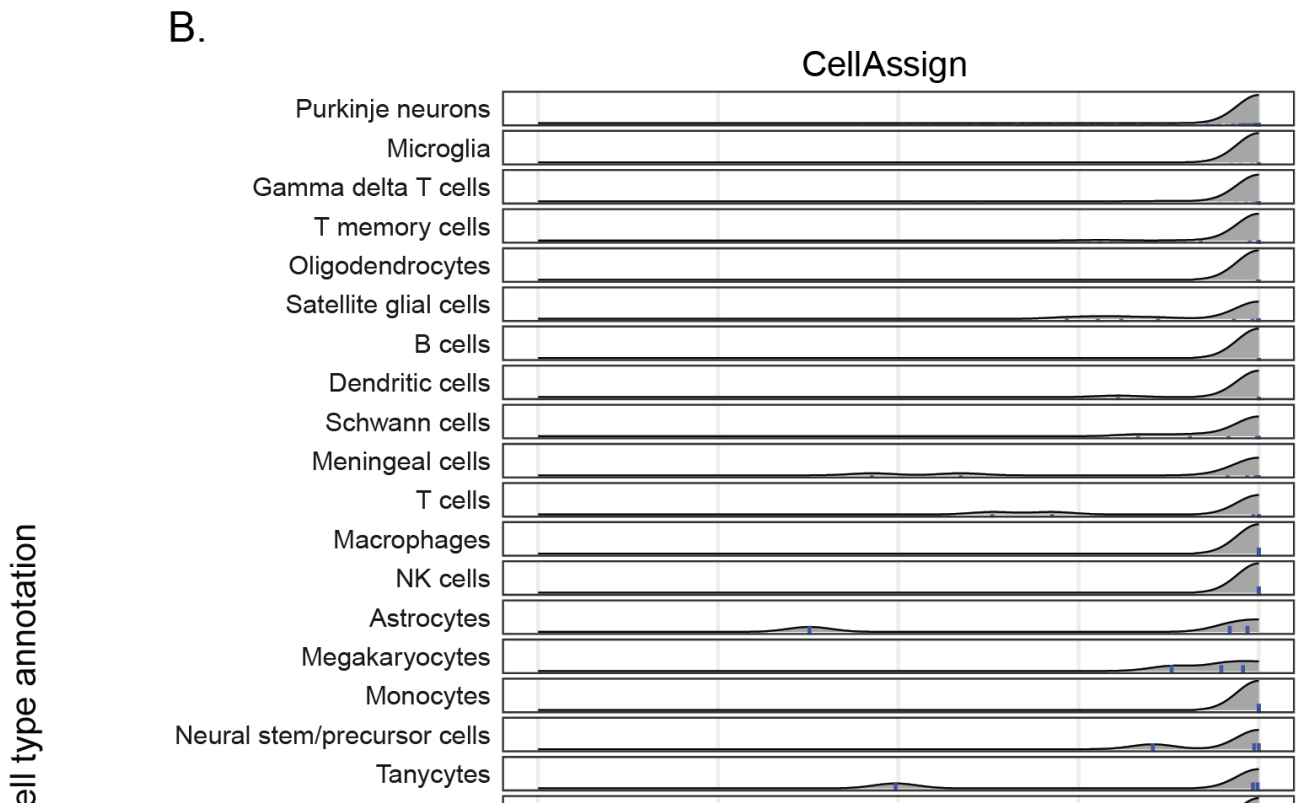
Figure S5: Cell type annotation with `CellAssign`.

Both plots in this figure are examples of plots that display results from annotating cells with `CellAssign` that can be found in the cell type summary report, shown here for library `SCPCL000490`.

A. A grid of UMAPs is shown for each cell type annotated using `CellAssign`, with the cell type of interest shown in color and all other cells belonging to other cell types shown in gray. The top four cell types with the greatest number of assigned cells are shown, while all other cells are grouped together and labeled with `All remaining cell types`. Any cells that are unable to be assigned by `CellAssign` are labeled with `Unknown cell type`.

B. This example heatmap displays a comparison between submitter-provided annotations and annotations with `CellAssign`, where heatmap cells are colored by the Jaccard similarity index. A value of 1 means that there is complete overlap between which cells are annotated with the two labels being compared, and a value of 0 means that there is no overlap between which cells are annotated with the two labels being compared.


A.
SingleR
neuron
macrophage
monocyte
effector memory CD8−positive, alpha−beta T cell
CD4−positive, alpha−beta T cell
mesangial cell
fat cell
central memory CD8−positive, alpha−beta T cell
naive B cell
central memory CD4−positive, alpha−beta T cell
chondrocyte
melanocyte
natural killer cell
class switched memory B cell
erythrocyte
astrocyte
CD8−positive, alpha−beta T cell
common lymphoid progenitor
granulocyte monocyte progenitor cell
hematopoietic stem cell
muscle cell
plasma cell
0.1
0.2
0.3
Delta median statistic
Cell type annotation
Cell type annotation quality
High−quality
Low−quality


B.
CellAssign
Purkinje neurons
Microglia
Gamma delta T cells
T memory cells
Oligodendrocytes
Satellite glial cells
B cells
Dendritic cells
Schwann cells
Meningeal cells
T cells
Macrophages
NK cells
Astrocytes
Megakaryocytes
Monocytes
Neural stem/precursor cells
Tanycytes
ell type annotation

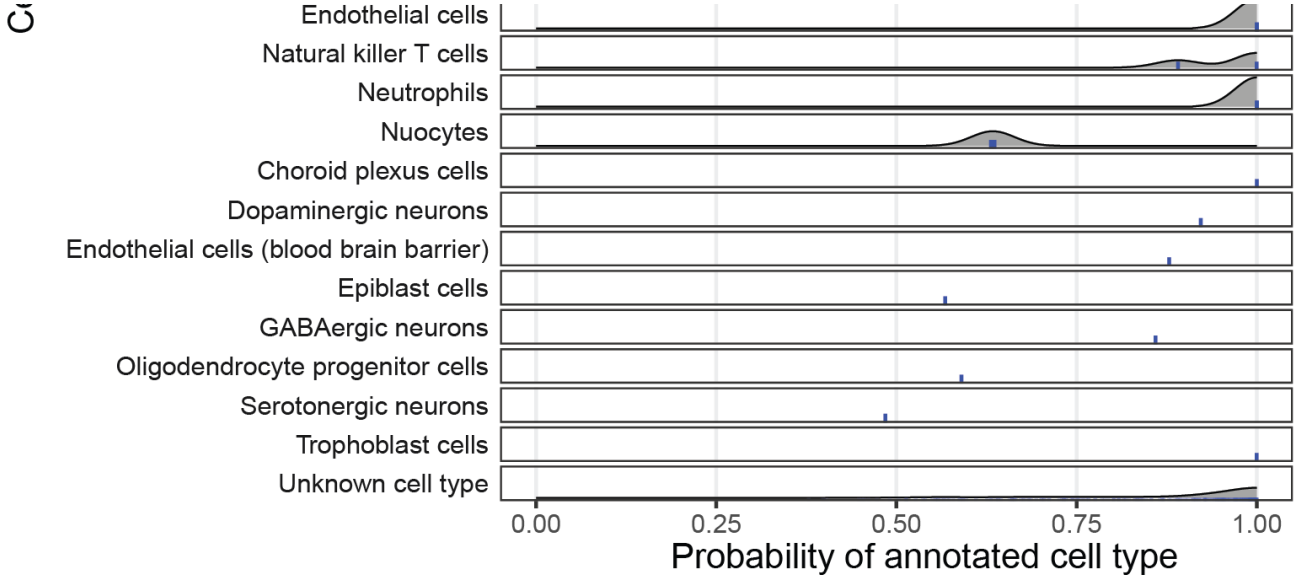


Figure S6: Assessment of cell type annotation quality.

Both plots in this figure are examples of diagnostic plots in the cell type summary report, shown for library `SCPCL000001`.

A. Diagnostic plot showing `SingleR` cell type annotation quality. Cell type annotations are shown on the y-axis, the delta median statistic is shown on the x-axis. The delta median statistic is calculated for each cell as the difference between the `SingleR` score of the annotated cell type label and the median score of all other cell type labels in the reference dataset. Higher values indicate a higher quality cell type annotation, although there is no absolute scale for these values. Each black point represents a cell, where closed circles denote cells with high-quality annotations and open circles denote cells with low-quality annotations, as assessed by `SingleR`. Red diamonds represent the median delta median scores for all cells with high-quality annotations associated with the given cell type label.

B. Diagnostic plot showing `CellAssign` cell type annotation quality. Cell type annotations are shown on the y-axis, and the probability of the annotated cell type as calculated by `CellAssign` is shown on the x-axis. Each row displays probabilities for only the cells associated with the given cell type annotation, and blue line segments show the probabilities for individual cells in each distribution. Taller line segments are shown for any distribution with five or fewer cells.

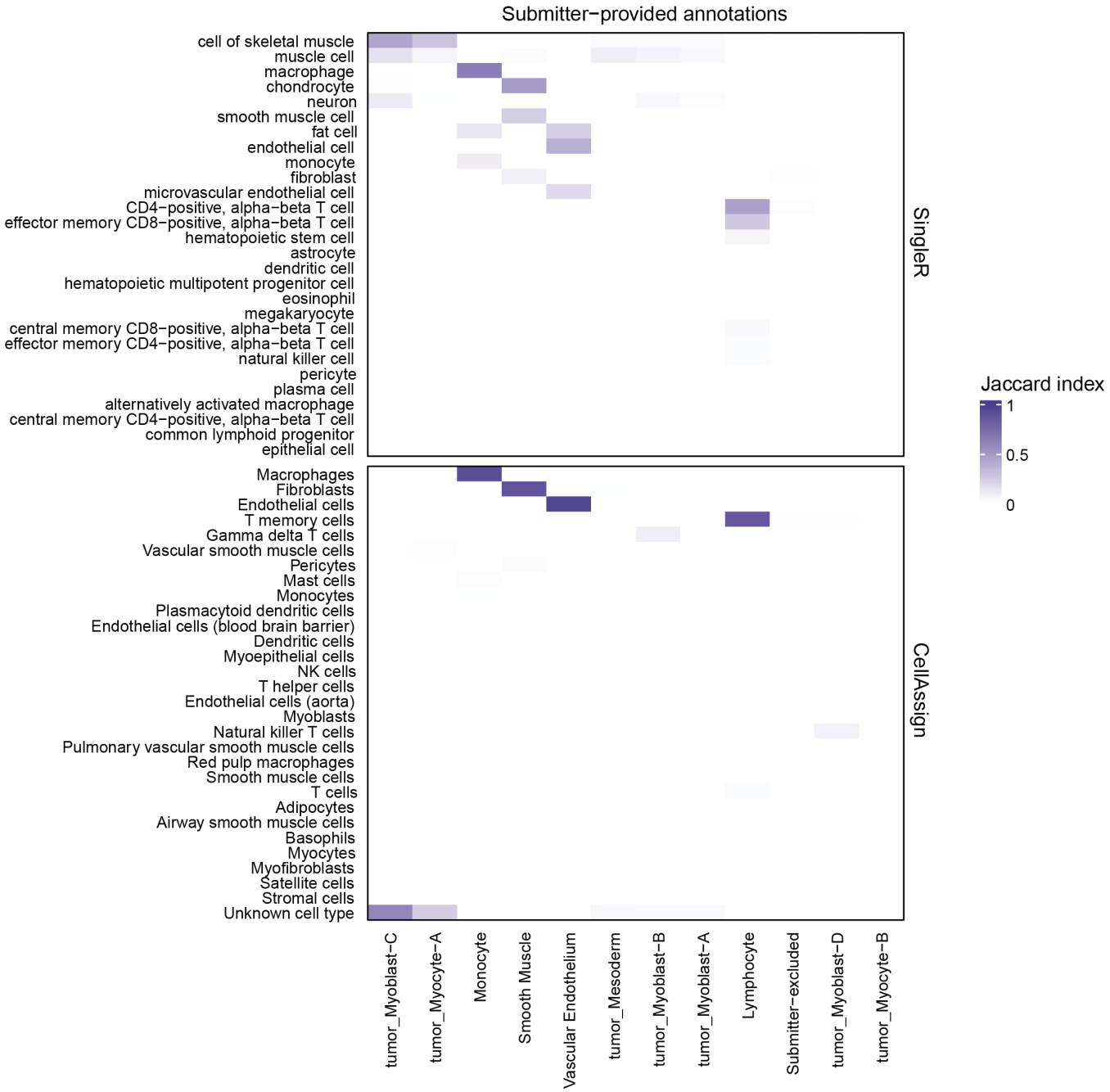


Figure S7: Comparison of cell type annotations across methods.

This example heatmap from the cell type summary report compares submitter-provided annotations to annotations with `SingleR` and `CellAssign`, shown for library `SCPCL000498`. This heatmap is only shown in the cell type summary report if submitters provided cell type annotations. Heatmap cells are colored by the Jaccard similarity index. A value of 1 means that there is complete overlap between which cells are annotated with the two labels being compared, and a value of 0 means that there is no overlap between which cells are annotated with the two labels being compared.

References

1. **Exponential scaling of single-cell RNA-seq in the past decade**
Valentine Svensson, Roser Vento-Tormo, Sarah A Teichmann
Nature Protocols (2018-03-01) https://doi.org/gc5ndt
DOI: 10.1038/nprot.2017.149 · PMID: 29494575

2. **Defining cell types and states with single-cell genomics**
Cole Trapnell
Genome Research (2015-10) https://doi.org/f7st9g
DOI: 10.1101/gr.190595.115 · PMID: 26430159 · PMCID: PMC4579334

3. **Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma**
Anoop P Patel, Itay Tirosh, John J Trombetta, Alex K Shalek, Shawn M Gillespie, Hiroaki Wakimoto, Daniel P Cahill, Brian V Nahed, William T Curry, Robert L Martuza, … Bradley E Bernstein
Science (2014-06-20) https://doi.org/gdm4dv
DOI: 10.1126/science.1254257 · PMID: 24925914 · PMCID: PMC4123637

4. **Cancer cell states recur across tumor types and form specific interactions with the tumor microenvironment**
Dalia Barkley, Reuben Moncada, Maayan Pour, Deborah A Liberman, Ian Dryg, Gregor Werba, Wei Wang, Maayan Baron, Anjali Rao, Bo Xia, … Itai Yanai
Nature Genetics (2022-08) https://doi.org/gqtn64
DOI: 10.1038/s41588-022-01141-9 · PMID: 35931863 · PMCID: PMC9886402

5. **The Human Cell Atlas**
Aviv Regev, Sarah A Teichmann, Eric S Lander, Ido Amit, Christophe Benoist, Ewan Birney, Bernd Bodenmiller, Peter Campbell, Piero Carninci, Menna Clatworthy, …
eLife (2017-12-05) https://doi.org/gcnzcv
DOI: 10.7554/elife.27041 · PMID: 29206104 · PMCID: PMC5762154

6. **The Human Tumor Atlas Network: Charting Tumor Transitions across Space and Time at Single-Cell Resolution**
Orit Rozenblatt-Rosen, Aviv Regev, Philipp Oberdoerffer, Tal Nawy, Anna Hupalowska, Jennifer E Rood, Orr Ashenberg, Ethan Cerami, Robert J Coffey, Emek Demir, … Xiaowei Zhuang
Cell (2020-04) https://doi.org/ggtkzd
DOI: 10.1016/j.cell.2020.03.053 · PMID: 32302568 · PMCID: PMC7376497

7. **Cancer in Children and Adolescents - NCI** (2023-09-29)
https://www.cancer.gov/types/childhood-cancers/child-adolescent-cancers-fact-sheet

8. **Use case driven evaluation of open databases for pediatric cancer research**
Fleur Jeanquartier, Claire Jean-Quartier, Andreas Holzinger
BioData Mining (2019-01-15) https://doi.org/ggjv7q
DOI: 10.1186/s13040-018-0190-8 · PMID: 30675185 · PMCID: PMC6334395

9. **Orchestrating single-cell analysis with Bioconductor**
Robert A Amezquita, Aaron TL Lun, Etienne Becht, Vince J Carey, Lindsay N Carpp, Ludwig Geistlinger, Federico Marini, Kevin Rue-Albrecht, Davide Risso, Charlotte Soneson, … Stephanie C Hicks
Nature Methods (2019-12-02) https://doi.org/ggdxgx
DOI: 10.1038/s41592-019-0654-x · PMID: 31792435 · PMCID: PMC7358058

10. **SCANPY: large-scale single-cell gene expression data analysis**
FAlexander Wolf, Philipp Angerer, Fabian J Theis
Genome Biology (2018-02-06) https://doi.org/gc22s9
DOI: 10.1186/s13059-017-1382-0 · PMID: 29409532 · PMCID: PMC5802054

11. **Nextflow enables reproducible computational workflows**
Paolo Di Tommaso, Maria Chatzou, Evan W Floden, Pablo Prieto Barja, Emilio Palumbo, Cedric Notredame
Nature Biotechnology (2017-04) https://doi.org/gfj52z
DOI: 10.1038/nbt.3820 · PMID: 28398311

12. **Alevin-fry unlocks rapid, accurate and memory-frugal quantification of single-cell RNA-seq data**
Dongze He, Mohsen Zakeri, Hirak Sarkar, Charlotte Soneson, Avi Srivastava, Rob Patro
Nature Methods (2022-03) https://doi.org/gptg86
DOI: 10.1038/s41592-022-01408-3 · PMID: 35277707 · PMCID: PMC8933848

13. **Ontology Lookup Service (OLS)** https://www.ebi.ac.uk/ols4/ontologies/hsapdv

14. **The anatomy of phenotype ontologies: principles, properties and applications**
Georgios V Gkoutos, Paul N Schofield, Robert Hoehndorf
Briefings in Bioinformatics (2017-04-06) https://doi.org/gk3928
DOI: 10.1093/bib/bbx035 · PMID: 28387809 · PMCID: PMC6169674

15. **Ontology Lookup Service (OLS)** https://www.ebi.ac.uk/ols4/ontologies/pato

16. **NCBI Taxonomy: a comprehensive update on curation, resources and tools**
Conrad L Schoch, Stacy Ciufo, Mikhail Domrachev, Carol L Hotton, Sivakumar Kannan, Rogneda Khovanskaya, Detlef Leipe, Richard Mcveigh, Kathleen O'Neill, Barbara Robbertse, … Ilene Karsch-Mizrachi
Database (2020-01-01) https://doi.org/gg7tjn
DOI: 10.1093/database/baaa062 · PMID: 32761142 · PMCID: PMC7408187

17. **Home - Taxonomy - NCBI** https://www.ncbi.nlm.nih.gov/taxonomy

18. **Mondo: Unifying diseases for the world, by the world**
Nicole A Vasilevsky, Nicolas A Matentzoglu, Sabrina Toro, Joseph E Flack IV, Harshad Hegde, Deepak R Unni, Gioconda F Alyea, Joanna S Amberger, Larry Babb, James P Balhoff, … Melissa A Haendel
Cold Spring Harbor Laboratory (2022-04-16) https://doi.org/gqx27c
DOI: 10.1101/2022.04.13.22273750

19. **Ontology Lookup Service (OLS)** https://www.ebi.ac.uk/ols4/ontologies/mondo

20. **Unification of multi-species vertebrate anatomy ontologies for comparative biology in Uberon**
Melissa A Haendel, James P Balhoff, Frederic B Bastian, David C Blackburn, Judith A Blake, Yvonne Bradford, Aurelie Comte, Wasila M Dahdul, Thomas A Dececchi, Robert E Druzinsky, … Christopher J Mungall
Journal of Biomedical Semantics (2014) https://doi.org/gtnrz4
DOI: 10.1186/2041-1480-5-21 · PMID: 25009735 · PMCID: PMC4089931

21. **Uberon, an integrative multi-species anatomy ontology**
Christopher J Mungall, Carlo Torniai, Georgios V Gkoutos, Suzanna E Lewis, Melissa A Haendel
Genome Biology (2012) https://doi.org/fxx6qr
DOI: 10.1186/gb-2012-13-1-r5 · PMID: 22293552 · PMCID: PMC3334586

22. **Ontology Lookup Service (OLS)** https://www.ebi.ac.uk/ols4/ontologies/uberon

23. **A standardized framework for representation of ancestry data in genomics studies, with application to the NHGRI-EBI GWAS Catalog**
Joannella Morales, Danielle Welter, Emily H Bowler, Maria Cerezo, Laura W Harris, Aoife C McMahon, Peggy Hall, Heather A Junkins, Annalisa Milano, Emma Hastings, ... Jacqueline AL MacArthur
Genome Biology (2018-02-15) https://doi.org/gf9pk3
DOI: 10.1186/s13059-018-1396-2 · PMID: 29448949 · PMCID: PMC5815218

24. **Ontology Lookup Service (OLS)** https://www.ebi.ac.uk/ols4/ontologies/hancestro

25. **Simultaneous epitope and transcriptome measurement in single cells**
Marlon Stoeckius, Christoph Hafemeister, William Stephenson, Brian Houck-Loomis, Pratip K Chattopadhyay, Harold Swerdlow, Rahul Satija, Peter Smibert
Nature Methods (2017-07-31) https://doi.org/gfkksd
DOI: 10.1038/nmeth.4380 · PMID: 28759029 · PMCID: PMC5669064

26. **Cell Hashing with barcoded antibodies enables multiplexing and doublet detection for single cell genomics**
Marlon Stoeckius, Shiwei Zheng, Brian Houck-Loomis, Stephanie Hao, Bertrand Z Yeung, William M Mauck III, Peter Smibert, Rahul Satija
Genome Biology (2018-12) https://doi.org/ggbm6p
DOI: 10.1186/s13059-018-1603-1 · PMID: 30567574 · PMCID: PMC6300015

27. **Massively parallel digital transcriptional profiling of single cells**
Grace XY Zheng, Jessica M Terry, Phillip Belgrader, Paul Ryvkin, Zachary W Bent, Ryan Wilson, Solongo B Ziraldo, Tobias D Wheeler, Geoff P McDermott, Junjie Zhu, ... Jason H Bielas
Nature Communications (2017-01-16) https://doi.org/f9mbtp
DOI: 10.1038/ncomms14049 · PMID: 28091601 · PMCID: PMC5241818

28. **Cell Ranger - Official 10x Genomics Support**
10x Genomics
https://www.10xgenomics.com/support/software/cell-ranger/latest

29. **AlexsLemonade/alsf-scpca**
Alex's Lemonade Stand Foundation
(2021-12-21) https://github.com/AlexsLemonade/alsf-scpca

30. **EmptyDrops: distinguishing cells from empty droplets in droplet-based single-cell RNA sequencing data**
Aaron TL Lun, Samantha Riesenfeld, Tallulah Andrews, The Phuong Dao, Tomas Gomes, John C Marioni
Genome Biology (2019-03-22) https://doi.org/gfxdhf
DOI: 10.1186/s13059-019-1662-y · PMID: 30902100 · PMCID: PMC6431044

31. **miQC: An adaptive probabilistic framework for quality control of single-cell RNA-sequencing data**
Ariel A Hippen, Matias M Falco, Lukas M Weber, Erdogan Pekcan Erkan, Kaiyang Zhang, Jennifer Anne Doherty, Anna Vähärautio, Casey S Greene, Stephanie C Hicks
PLOS Computational Biology (2021-08-24) https://doi.org/gng37g
DOI: 10.1371/journal.pcbi.1009290 · PMID: 34428202 · PMCID: PMC8415599

32. **Pooling across cells to normalize single-cell RNA sequencing data with many zero counts**
Aaron T L. Lun, Karsten Bach, John C Marioni
Genome Biology (2016-04-27) https://doi.org/gfgntn

DOI: 10.1186/s13059-016-0947-7 · PMID: 27122128 · PMCID: PMC4848819

33. **UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction**
Leland McInnes, John Healy, James Melville
arXiv (2020-09-21) https://arxiv.org/abs/1802.03426

34. **Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage**
Dvir Aran, Agnieszka P Looney, Leqian Liu, Esther Wu, Valerie Fong, Austin Hsu, Suzanna Chak, Ram P Naikawadi, Paul J Wolters, Adam R Abate, … Mallar Bhattacharya
Nature Immunology (2019-01-14) https://doi.org/gfv3p2
DOI: 10.1038/s41590-018-0276-y · PMID: 30643263 · PMCID: PMC6340744

35. **Probabilistic cell-type assignment of single-cell RNA-seq for tumor microenvironment profiling**
Allen W Zhang, Ciara O'Flanagan, Elizabeth A Chavez, Jamie LP Lim, Nicholas Ceglia, Andrew McPherson, Matt Wiens, Pascale Walters, Tim Chan, Brittany Hewitson, … Sohrab P Shah
Nature Methods (2019-09-09) https://doi.org/ggr7ps
DOI: 10.1038/s41592-019-0529-1 · PMID: 31501550 · PMCID: PMC7485597

36. **anndata: Annotated data**
Isaac Virshup, Sergei Rybakov, Fabian J Theis, Philipp Angerer, FAlexander Wolf
Cold Spring Harbor Laboratory (2021-12-19) https://doi.org/gst7w6
DOI: 10.1101/2021.12.16.473007

37. **scuttle**
Aaron Lun, Davis McCarthy
Bioconductor (2020) https://doi.org/gtkc7k
DOI: 10.18129/b9.bioc.scuttle

38. **Chapter 12 Integrating with protein abundance | Advanced Single-Cell Analysis with Bioconductor** https://bioconductor.org/books/3.16/OSCA.advanced/integrating-with-protein-abundance.html#cite-seq-median-norm

39. **DropletUtils**
Jonathan Griffiths Aaron Lun
Bioconductor (2018) https://doi.org/gtkc7d
DOI: 10.18129/b9.bioc.dropletutils

40. **Genetic demultiplexing of pooled single-cell RNA-sequencing samples in cancer facilitates effective experimental design**
Lukas M Weber, Ariel A Hippen, Peter F Hickey, Kristofer C Berrett, Jason Gertz, Jennifer Anne Doherty, Casey S Greene, Stephanie C Hicks
GigaScience (2021-09) https://doi.org/gmwhsc
DOI: 10.1093/gigascience/giab062 · PMID: 34553212 · PMCID: PMC8458035

41. **fastp: an ultra-fast all-in-one FASTQ preprocessor**
Shifu Chen, Yanqing Zhou, Yaru Chen, Jia Gu
Bioinformatics (2018-09-01) https://doi.org/gd9mrb
DOI: 10.1093/bioinformatics/bty560 · PMID: 30423086 · PMCID: PMC6129281

42. **Salmon provides fast and bias-aware quantification of transcript expression**
Rob Patro, Geet Duggal, Michael I Love, Rafael A Irizarry, Carl Kingsford
Nature Methods (2017-03-06) https://doi.org/gcw9f5
DOI: 10.1038/nmeth.4197 · PMID: 28263959 · PMCID: PMC5600148

43. **Space Ranger - Official 10x Genomics Support**
10x Genomics
https://www.10xgenomics.com/support/software/space-ranger/latest

44. **Chapter 4 Annotation diagnostics | Assigning cell types with SingleR**
https://bioconductor.org/books/release/SingleRBook/annotation-diagnostics.html#based-on-the-deltas-across-cells

45. **BLUEPRINT: mapping human blood cell epigenomes**
JHA Martens, HG Stunnenberg
Haematologica (2013-10-01) https://doi.org/gd2xz4
DOI: 10.3324/haematol.2013.094243 · PMID: 24091925 · PMCID: PMC3789449

46. **An integrated encyclopedia of DNA elements in the human genome** *Nature* (2012-09)
https://doi.org/bg9d
DOI: 10.1038/nature11247 · PMID: 22955616 · PMCID: PMC3439153

47. **PanglaoDB: a web server for exploration of mouse and human single-cell RNA sequencing data**
Oscar Franzén, Li-Ming Gan, Johan LM Björkegren
Database (2019-01-01) https://doi.org/ggkzxr
DOI: 10.1093/database/baz046 · PMID: 30951143 · PMCID: PMC6450036

48. **Cellxgene Data Portal**
Cellxgene Data Portal
https://cellxgene.cziscience.com/

49. https://github.com/chanzuckerberg/single-cell-curation/blob/main/schema/3.0.0/schema.md

50. **Alignment and mapping methodology influence transcript abundance estimation**
Avi Srivastava, Laraib Malik, Hirak Sarkar, Mohsen Zakeri, Fatemeh Almodaresi, Charlotte Soneson, Michael I Love, Carl Kingsford, Rob Patro
Genome Biology (2020-09-07) https://doi.org/gg98sd
DOI: 10.1186/s13059-020-02151-8 · PMID: 32894187 · PMCID: PMC7487471

51. **STAR: ultrafast universal RNA-seq aligner**
Alexander Dobin, Carrie A Davis, Felix Schlesinger, Jorg Drenkow, Chris Zaleski, Sonali Jha, Philippe Batut, Mark Chaisson, Thomas R Gingeras
Bioinformatics (2012-10-25) https://doi.org/f4h523
DOI: 10.1093/bioinformatics/bts635 · PMID: 23104886 · PMCID: PMC3530905

52. **STARsolo: accurate, fast and versatile mapping/quantification of single-cell and single-nucleus RNA-seq data**
Benjamin Kaminow, Dinar Yunusov, Alexander Dobin
Cold Spring Harbor Laboratory (2021-05-05) https://doi.org/gqj7ft
DOI: 10.1101/2021.05.05.442755

53. **Twelve years of SAMtools and BCFtools**
Petr Danecek, James K Bonfield, Jennifer Liddle, John Marshall, Valeriu Ohan, Martin O Pollard, Andrew Whitwham, Thomas Keane, Shane A McCarthy, Robert M Davies, Heng Li
GigaScience (2021-01-29) https://doi.org/gjxzc9
DOI: 10.1093/gigascience/giab008 · PMID: 33590861 · PMCID: PMC7931819

54. **Cellsnp-lite: an efficient tool for genotyping single cells**
Xianjie Huang, Yuanhua Huang
Bioinformatics (2021-05-08) https://doi.org/gqcggs

DOI: 10.1093/bioinformatics/btab358 · PMID: 33963851

55. **zellkonverter**
Luke Zappia, Aaron Lun
Bioconductor (2020) https://doi.org/gtnrz5
DOI: 10.18129/b9.bioc.zellkonverter

56. **cellxgene: a performant, scalable exploration platform for high dimensional sparse matrices**
Colin Megill, Bruce Martin, Charlotte Weaver, Sidney Bell, Lia Prins, Seve Badajoz, Brian McCandless, Angela Oliveira Pisco, Marcus Kinsella, Fiona Griffin, … Ambrose Carr
Cold Spring Harbor Laboratory (2021-04-06) https://doi.org/gst8vt
DOI: 10.1101/2021.04.05.438318

57. **CZ CELL×GENE Discover: A single-cell data platform for scalable exploration, analysis and modeling of aggregated data**
, Shibla Abdulla, Brian Aevermann, Pedro Assis, Seve Badajoz, Sidney M Bell, Emanuele Bezzi, Batuhan Cakir, Jim Chaffer, Signe Chambers, … Ambrose Carr
Cold Spring Harbor Laboratory (2023-11-02) https://doi.org/gtk3dd
DOI: 10.1101/2023.10.30.563174

58. **Cellxgene Data Portal**
Cellxgene Data Portal
https://cellxgene.cziscience.com/

59. **UCSC Cell Browser: visualize your single-cell data**
Matthew L Speir, Aparna Bhaduri, Nikolay S Markov, Pablo Moreno, Tomasz J Nowakowski, Irene Papatheodorou, Alex A Pollen, Brian J Raney, Lucas Seninge, WJames Kent, Maximilian Haeussler
Bioinformatics (2021-07-09) https://doi.org/gtk3db
DOI: 10.1093/bioinformatics/btab503 · PMID: 34244710 · PMCID: PMC8652023

60. https://cells.ucsc.edu/

61. **Powering single-cell analyses in the browser with WebAssembly**
Aaron Lun, Jayaram Kancherla
Cold Spring Harbor Laboratory (2022-03-04) https://doi.org/gtk3dc
DOI: 10.1101/2022.03.02.482701

62. **kana** https://www.kanaverse.org/kana/

63. **A comprehensive benchmarking with practical guidelines for cellular deconvolution of spatial transcriptomics**
Haoyang Li, Juexiao Zhou, Zhongxiao Li, Siyuan Chen, Xingyu Liao, Bin Zhang, Ruochi Zhang, Yu Wang, Shiwei Sun, Xin Gao
Nature Communications (2023-03-21) https://doi.org/gtk3c9
DOI: 10.1038/s41467-023-37168-7 · PMID: 36941264 · PMCID: PMC10027878

64. **Computational deconvolution of transcriptomics data from mixed cell populations**
Francisco Avila Cobos, Jo Vandesompele, Pieter Mestdagh, Katleen De Preter
Bioinformatics (2018-01-16) https://doi.org/gctpvd
DOI: 10.1093/bioinformatics/bty019 · PMID: 29351586

65. **Effective methods for bulk RNA-seq deconvolution using scnRNA-seq transcriptomes**
Francisco Avila Cobos, Mohammad Javad Najaf Panah, Jessica Epps, Xiaochen Long, Tsz-Kwong Man, Hua-Sheng Chiu, Elad Chomsky, Evgeny Kiner, Michael J Krueger, Diego di Bernardo, …

Pavel Sumazin
Genome Biology (2023-08-01) https://doi.org/gsmvqq
DOI: 10.1186/s13059-023-03016-6 · PMID: 37528411 · PMCID: PMC10394903