Spatial Cell Typing

v0.1.2 spatial-transcriptomics transcriptomics genomics single-cell xenium merscope cosmx visium-hd clustering cell-typing

Clusters and cell types for single-cell spatial transcriptomics (10x Xenium, Vizgen MERSCOPE, NanoString CosMx; also Visium HD bins), the step after st-qc in the Single-cell Spatial Profiling workflow.

Repository	`https://github.com/pennsieve-apps/st-celltype`	License	Apache-2.0
Authors	Pennsieve (University of Pennsylvania)	Generated	2026-11-13 21:38 UTC
Sample run	`pipeline:feb17285-258b-4e42-844e-abc005e19917:43bab7e908c3`	Run time	42 s

This document shows what **Spatial Cell Typing** produces. Pennsieve ran version v0.1.2 once, on the test inputs the app declares for itself, and collected what it wrote. It is not the app's documentation and not its output: the outputs are the files listed under Results. Read it to decide whether the app does what you need before you run it on your own data.

Parts marked AI-WRITTEN were written by a language model (us.anthropic.claude-opus-5-5) from the files and records cited in this document; every number in them was checked against that material. Everything else, including every figure, table and value, is copied from the run's output files and its record. No figure was drawn or altered; images are downscaled copies.

IN THIS DOCUMENT

1 At a glance
2 About the app
3 Test inputs
4 Figures
5 Results
6 Run it yourself
7 How to cite
8 Files in this document

1 · At a glance

The Spatial Cell Typing app clusters and labels observations from single-cell spatial transcriptomics data. It clusters all samples jointly through PCA of log-normalized expression, Harmony integration by sample_id, a k-nearest-neighbour graph and Leiden clustering. It then scores each observation for marker gene sets and calls each cluster the cell type with the highest standardised score, or Unassigned. This run was a sample on test data written by the app's own test-data generator, with spatial and metadata files covering 2 samples. From these inputs it produced labels.csv with per-observation cluster and cell-type calls, cluster annotation, marker, score, summary and agreement tables, and figures: UMAPs, spatial cell-type maps per sample, a marker dotplot and composition bars.

AI-WRITTEN

RAN ON
2 inputs · 4 files · 2.3 MB

PRODUCED
2 outputs · 18 files · 1.0 MB

SHOWN HERE
6 figures

RUN TIME
42 s

2 · About the app

Clusters and cell types for single-cell spatial transcriptomics (10x Xenium, Vizgen MERSCOPE, NanoString CosMx; also Visium HD bins), the step after st-qc in the Single-cell Spatial Profiling workflow. All samples are clustered jointly: PCA of the log-normalized expression, optional Harmony integration by sample, a k-nearest-neighbour graph and Leiden clusters (C01, C02, …). Each observation is scored for marker gene sets (scanpy score_genes) from a user markers.csv or a built-in table of canonical markers drawn from PanglaoDB, matched to the panel case-insensitively (human and mouse), and each cluster (or each cell) is called the cell type with the highest standardised score, or Unassigned. Writes labels.csv (sample_id, obs_id, cluster, cell_type) for st-niches, st-ccc and st-report, cluster markers, scores and summaries, plus UMAPs, spatial cell-type maps, a marker dotplot and composition bars.

Takes

Input	What	Formats
spatial	st-qc's <sample_id>.h5ad, one per sample: X log-normalized (CSR), layers["counts"], obsm["spatial"], uns["pennsieve"]; obsm["X_pca"] is recomputed jointly over all samples.	`application/x-hdf5; profile=anndata`
metadata	st-prep's samples.csv (sample_id, path, platform, kind, n_obs, n_genes, n_obs_in_tissue and any extra sample-sheet columns); lists the samples to use and their order.	`text/csv, application/json`
markers	Optional markers.csv: long (cell_type, gene; one gene per row) or wide (cell_type, genes; genes separated by ';'). Replaces the built-in marker table.	`text/csv`

Produces

Output	What	Formats
labels	labels/labels.csv (sample_id, obs_id, cluster, cell_type), cluster_markers.csv (cluster, gene, logfoldchange, pval_adj, pct_in, pct_out, rank, score), cell_type_summary.csv and cluster_summary.csv (sample_id, label, n_obs, fraction), cluster_annotation.csv (cluster, cell_type, score, n_obs, second_best, second_score), scores.csv (sample_id, obs_id, one z-scored column per marker set), agreement.csv and confusion.csv when obs["annotation"] exists, summary.json, provenance.json.	`text/csv, application/json`
figures	figures/umap_cluster.png, umap_cell_type.png, <sample_id>_cell_type.png (cell types in space), dotplot_marker_sets.png (marker-set scores by cluster), cell_type_by_sample.png (composition), summary.json, provenance.json.	`image/png, application/json`

3 · Test inputs

What the app ran on for this sample: the test data its app.yml declares, built for this run and discarded afterwards (app commit `2d4b2707049d`).

spatial 2 files · 2.3 MB

Written by the app's own test-data generator (python -m tests.make_fixture "$REVIEW_INPUT_DIR/fixture" --layout both).

In the app's input directory as `spatial/`.

metadata 2 files · 339 B

Written by the app's own test-data generator (python -m tests.make_fixture "$REVIEW_INPUT_DIR/fixture" --layout both).

In the app's input directory as `metadata/`.

4 · Figures

Figures marked (test) show the app's generated test data, not real measurements.

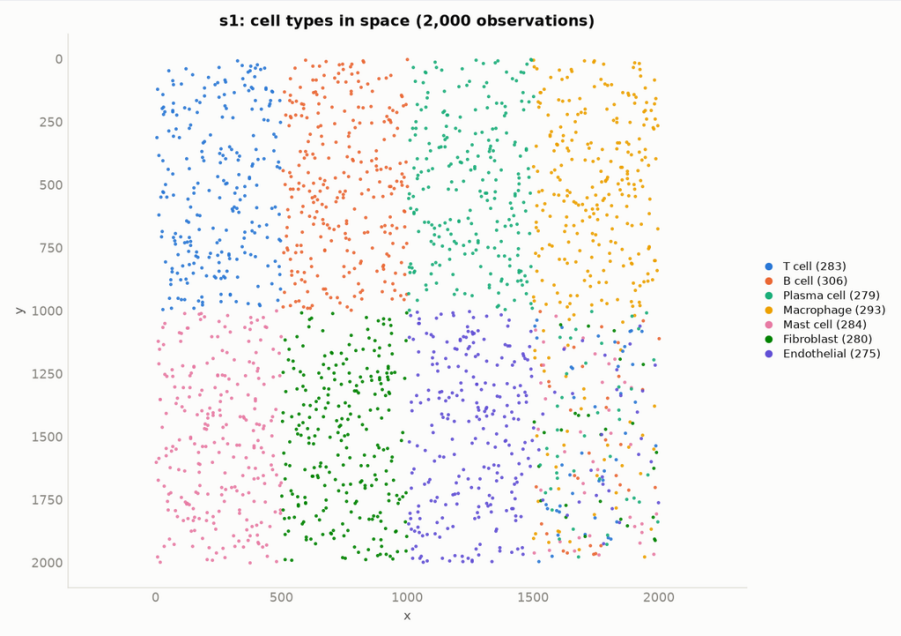


Figure 1 (test). Spatial plot of sample s1 with observations coloured by cell type, as annotated by st-celltype 0.1.1 using the builtin marker sets with cluster-level annotation. AI-WRITTEN

figures · figures/s1_cell_type.png R9

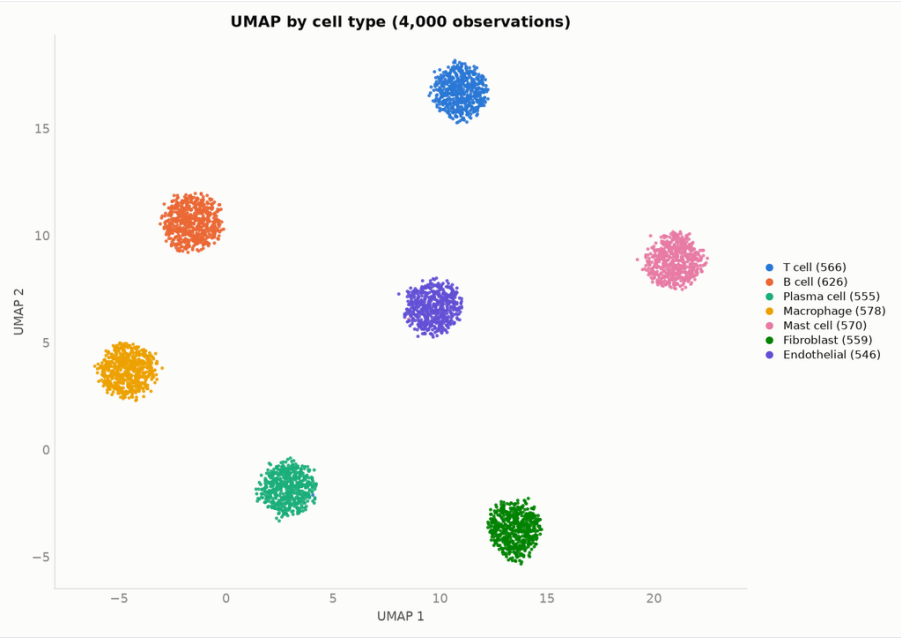


Figure 2 (test). UMAP embedding coloured by cell type, computed with scanpy.tl.umap on at most 50000 seeded observations after harmony integration by sample_id. AI-WRITTEN

figures · figures/umap_cell_type.png R10

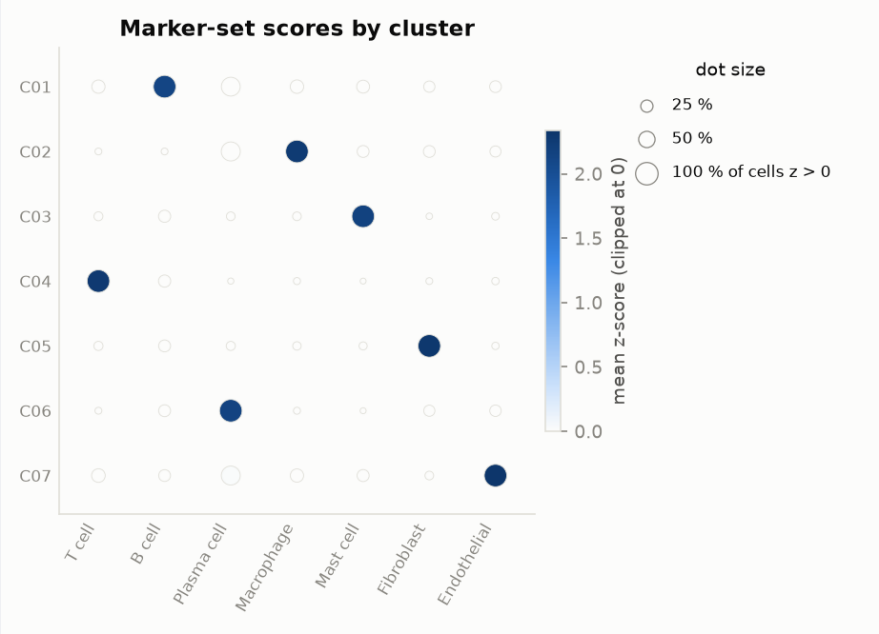


Figure 3 (test). Dot plot of builtin marker-set scores by Leiden cluster, showing mean z-score (clipped at 0) as colour and the fraction of cells with z > 0 as dot size. AI-WRITTEN

figures · figures/dotplot_marker_sets.png R11

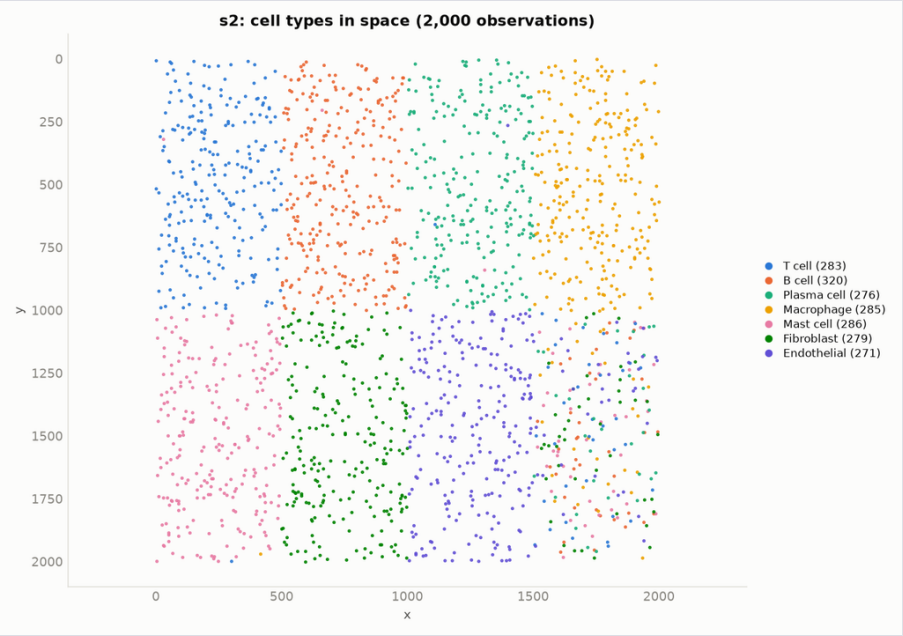


Figure 4 (test). Spatial plot of sample s2 with observations coloured by cell type, as annotated by st-celltype 0.1.1 using the builtin marker sets with cluster-level annotation. AI-WRITTEN

figures · figures/s2_cell_type.png R12

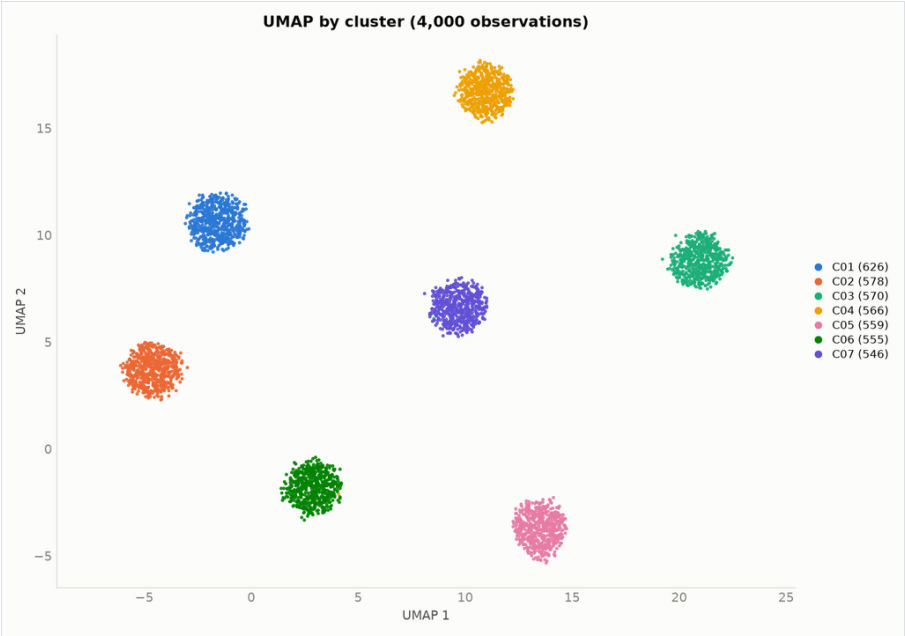


Figure 5 (test). UMAP embedding coloured by Leiden cluster (resolution 0.5), computed on the harmony-integrated neighbour graph with n_neighbors 15. AI-WRITTEN

figures · figures/umap_cluster.png R13

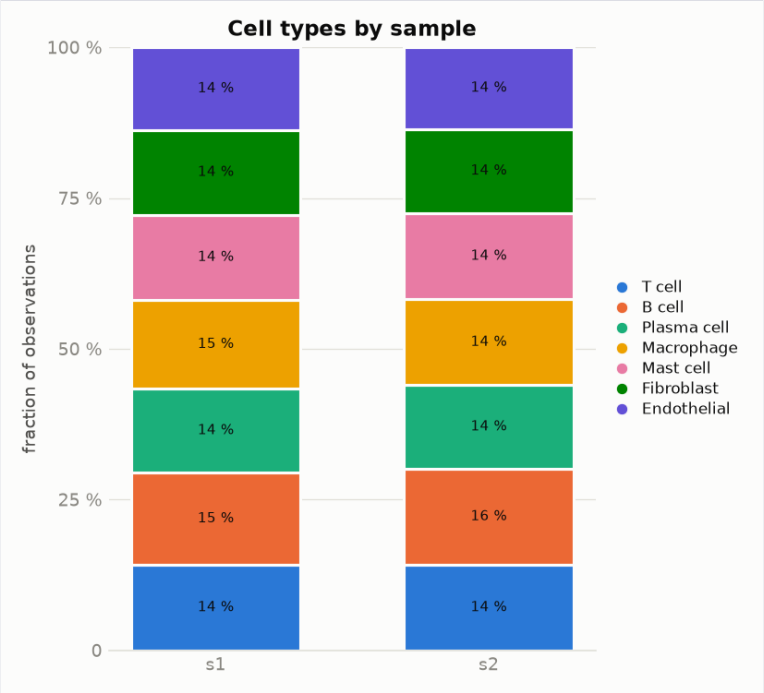


Figure 6 (test). Stacked bar chart of the fraction of observations assigned to each cell type in samples s1 and s2. AI-WRITTEN

figures · figures/cell_type_by_sample.png R14

5 · Results

What the app wrote, by output port, as delivered in `Spatial-Cell-Typing-v0.1.2-sample.zip`.

labels labels/labels.csv (sample_id, obs_id, cluster, cell_type), cluster_markers.csv (cluster, gene, logfoldchange, pval_adj, pct_in, pct_out, rank, score), cell_type_summary.csv and cluster_summary.csv (sample_id, label, n_obs, fraction), cluster_annotation.csv (cluster, cell_type, score, n_obs, second_best, second_score), scores.csv (sample_id, obs_id, one z-scored column per marker set), agreement.csv and confusion.csv when obs["annotation"] exists, summary.json, provenance.json.

The summary reports 4000 observations, 60 genes, 2 samples, 7 clusters and 7 cell types with fraction_unassigned 0.0, using builtin marker sets, Harmony integration and Leiden resolution 0.5.

AI-WRITTEN

Files

Total	10 files · 397.1 KB
.csv	8 · 383.2 KB
.json	2 · 14.0 KB

Recorded by the app R1

App	st-celltype 0.1.1
Ran	2026-11-12 05:42 UTC · 36.57 s in the app
Peak memory	after_reading 121 · total 623.4 · cgroup_peak — MB
Samples	2 ok · 0 failed
Tools	python 3.12.16 · scanpy 1.12.5 · anndata 0.13.4 · harmonypy 2.1.0 · igraph 1.0.0 · umap-learn 0.5.12 · scikit-learn 1.9.1 · numpy 2.5.4 · scipy 1.18.1 · pandas 3.1.0 · matplotlib 3.11.2 · h5py 3.16.0

`labels/summary.json` R3

n_samples	2
n_obs	4000
n_genes	60
n_clusters	7
n_cell_types	7
fraction_unassigned	0
figures	umap_cluster.png, umap_cell_type.png, s1_cell_type.png, s2_cell_type.png, dotplot_marker_sets.png, cell_type_by_sample.png

`labels/cluster_summary.csv` R5

10 of 14 rows.

sample_id	cluster	n_obs	fraction
s1	C01	306	0.153
s1	C02	293	0.1465
s1	C03	284	0.142
s1	C04	283	0.1415
s1	C05	280	0.14
s1	C06	279	0.1395
s1	C07	275	0.1375
s2	C01	320	0.16
s2	C02	285	0.1425
s2	C03	286	0.143

`labels/cell_type_summary.csv` R6

10 of 14 rows.

sample_id	cell_type	n_obs	fraction
s1	T cell	283	0.1415
s1	B cell	306	0.153
s1	Plasma cell	279	0.1395
s1	Macrophage	293	0.1465
s1	Mast cell	284	0.142
s1	Fibroblast	280	0.14
s1	Endothelial	275	0.1375
s2	T cell	283	0.1415
s2	B cell	320	0.16
s2	Plasma cell	276	0.138

`labels/agreement.csv` R7

6 of 6 rows.

sample_id	label	ari	nmi	n_obs
s1	cluster	1	1	2000
s2	cluster	0.991893	0.988034	2000
all	cluster	0.995943	0.993394	4000
s1	cell_type	1	1	2000
s2	cell_type	0.991893	0.988034	2000
all	cell_type	0.995943	0.993394	4000

1 more table(s) were read but are not shown: labels/cluster_annotation.csv R8.

figures figures/umap_cluster.png, umap_cell_type.png, <sample_id>_cell_type.png (cell types in space), dotplot_marker_sets.png (marker-set scores by cluster), cell_type_by_sample.png (composition), summary.json, provenance.json.

These are UMAPs coloured by Leiden cluster and by cell type, spatial cell-type plots for samples s1 and s2, a dot plot of builtin marker-set scores by cluster, and a stacked bar chart of cell-type fractions per sample.

AI-WRITTEN

Files

Total	8 files · 639.2 KB
.png	6 · 636.5 KB
.json	2 · 2.7 KB

Recorded by the app R2

App	st-celltype 0.1.1
Ran	2026-11-12 05:42 UTC · 36.57 s in the app
Peak memory	after_reading 121 · total 623.4 · cgroup_peak — MB
Samples	2 ok · 0 failed
Tools	python 3.12.16 · scanpy 1.12.5 · anndata 0.13.4 · harmonypy 2.1.0 · igraph 1.0.0 · umap-learn 0.5.12 · scikit-learn 1.9.1 · numpy 2.5.4 · scipy 1.18.1 · pandas 3.1.0 · matplotlib 3.11.2 · h5py 3.16.0

`figures/summary.json` R4

figures	umap_cluster.png, umap_cell_type.png, s1_cell_type.png, s2_cell_type.png, dotplot_marker_sets.png, cell_type_by_sample.png
n_samples	2
n_clusters	7
n_cell_types	7

6 · Run it yourself

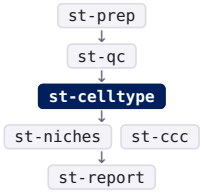
On Pennsieve

1. Open **Spatial Cell Typing** in the Pennsieve App Store and choose **Try this app**, or add it as a step in a workflow.
2. Give it **spatial** (application/x-hdf5; profile=anndata), **metadata** (text/csv, application/json), **markers** (text/csv) from one of your datasets. The sample's generated inputs are small synthetic files the app's repository can regenerate (`tests/` in the repository).
3. Pick a compute node and start the run.
4. Collect the results from the run's page: as a download, or into a dataset folder when the app is a workflow step.

Suggested workflows

Public workflows on Pennsieve that run Spatial Cell Typing as a step; open one and run it on your data.

Single-cell Spatial Profiling
OFFICIAL
Xenium/MERSCOPE/CosMx: QC, clustering and cell typing, niches, ligand-receptor



6 steps · **Open on Pennsieve ↗**

Locally, as a container

Every Pennsieve app is a container that reads `INPUT_DIR`, writes `OUTPUT_DIR` and takes its parameters from the environment, so it runs anywhere Docker does. Build it from the release's source and run it on a folder laid out as in section 3 (one subfolder per input port):

```
git clone https://github.com/pennsieve-apps/st-celltype st-celltype
cd st-celltype && git checkout v0.1.2
docker build -t st-celltype:v0.1.2 .

mkdir -p input output        # your inputs under ./input
docker run --rm \
  -v "$PWD/input:/input:ro" -v "$PWD/output:/output" \
```

```
  -e INPUT_DIR=/input -e OUTPUT_DIR=/output \
  st-celltype:v0.1.2
```

The app declares 8 GB of memory; give Docker at least that. The results land in `./output`, the same files as section 5.

7 · How to cite

Results made with this app rest on three things. Please cite all three.

1. **THE METHOD**
   F. Alexander Wolf, Philipp Angerer, Fabian J. Theis (2018). SCANPY: large-scale single-cell gene expression data analysis. Genome Biology. `doi:10.1186/s13059-017-1382-0`
2. **THIS RELEASE**
   Pennsieve (University of Pennsylvania) (2026). st-celltype (version v0.1.2). Pennsieve Discover. `doi:10.26275/8sgi-rlhl`
3. **PENNSIEVE**
   Name the platform as a resource, with its Research Resource Identifier `RRID:SCR_021677`.

> **FOR YOUR METHODS SECTION**
> Analyses were run with Spatial Cell Typing v0.1.2 (doi:10.26275/8sgi-rlhl) on Pennsieve (RRID:SCR_021677), which implements the method of F (doi:10.1186/s13059-017-1382-0).

Also builds on

1. Isaac Virshup, Sergei Rybakov, Fabian J. Theis, Philipp Angerer, F. Alexander Wolf (2024). anndata: Access and store annotated data matrices. Journal of Open Source Software. `doi:10.21105/joss.04371`
2. Ilya Korsunsky, Nghia Millard, Jean Fan, Kamil Slowikowski, Fan Zhang, Kevin Wei, et al. (2019). Fast, sensitive and accurate integration of single-cell data with Harmony. Nature Methods. `doi:10.1038/s41592-019-0619-0`
3. V. A. Traag, L. Waltman, N. J. van Eck (2019). From Louvain to Leiden: guaranteeing well-connected communities. Scientific Reports. `doi:10.1038/s41598-019-41695-z`
4. Leland McInnes, John Healy, Nathaniel Saul, Lukas Großberger (2018). UMAP: Uniform Manifold Approximation and Projection. Journal of Open Source Software. `doi:10.21105/joss.00861`
5. Rahul Satija, Jeffrey A Farrell, David Gennert, Alexander F Schier, Aviv Regev (2015). Spatial reconstruction of single-cell gene expression data. Nature Biotechnology. `doi:10.1038/nbt.3192`
6. Oscar Franzén, Li-Ming Gan, Johan L M Björkegren (2019). PanglaoDB: a web server for exploration of mouse and human single-cell RNA sequencing data. Database. `doi:10.1093/database/baz046`
7. Dénes Türei, Alberto Valdeolivas, Lejla Gul, Nicolàs Palacio-Escat, Michal Klein, Olga Ivanova, et al. (2021). Integrated intra- and intercellular signaling knowledge for multicellular omics analysis. Molecular Systems Biology. `doi:10.15252/msb.20209923`
8. The igraph core team. igraph (Python interface): community_leiden, the Leiden implementation behind scanpy.tl.leiden(flavor='igraph')
9. Amanda Janesick, Robert Shelansky, Andrew D. Gottscho, Florian Wagner, Stephen R. Williams, Morgane Rouault, et al. (2023). High resolution mapping of the tumor microenvironment using integrated single-cell, spatial and in situ analysis. Nature Communications. `doi:10.1038/s41467-023-43458-x`

From the app's citation file (CITATION.cff) and the release's record on Pennsieve Discover.

8 · Files in this document

Every figure, table and value above is copied from one of these files in `Spatial-Cell-Typing-v0.1.2-sample.zip`, the sample run's download.

Ref	File	Size
R1	`labels/provenance.json`	2.5 KB
R2	`figures/provenance.json`	2.5 KB
R3	`labels/summary.json`	11.4 KB
R4	`figures/summary.json`	240 B
R5	`labels/cluster_summary.csv`	292 B
R6	`labels/cell_type_summary.csv`	378 B
R7	`labels/agreement.csv`	221 B
R8	`labels/cluster_annotation.csv`	382 B
R9	`figures/s1_cell_type.png`	135.9 KB
R10	`figures/umap_cell_type.png`	127.7 KB
R11	`figures/dotplot_marker_sets.png`	68.7 KB
R12	`figures/s2_cell_type.png`	134.7 KB
R13	`figures/umap_cluster.png`	123.9 KB
R14	`figures/cell_type_by_sample.png`	45.6 KB

About this document

- Generated 2026-11-13 21:38 UTC by Pennsieve's curation-service for Spatial Cell Typing v0.1.2, sample run pipeline:feb17285-258b-4e42-844e-abc005e19917:43bab7e908c3, app commit 2d4b2707049d.
- Material: the files the app wrote in the sample run (8 read, up to 256.0 KB each, headers only for large data files), the fixtures step's record of the test inputs, the app's app.yml and CITATION.cff, and the run's record.
- Model: us.anthropic.claude-opus-5-5. It wrote the captions and prose marked `AI-WRITTEN`; sentences with a number not found in the material were removed.
- A release report is generated for each release of the app; this one describes v0.1.2. It is a derived document, not part of the app's published release.
- 2 model-written sentence(s) were removed by the number check.