scRNA-seq Analysis Ecosystems

Tools, Resources, & Getting Started with Seurat

Alex Emmons, PhD

2026-05-13

Single-Cell and Spatial Omics

- April 29, 2026 - ***Introduction to Single-Cell & Spatial Omics: Technology, Core Resources & Workflow Overview*** (Mike Kelly, SCSC)
- May 6, 2026 - ***Single-Cell and Spatial Transcriptomics: SCSC (CCR SCAF) Support Workflows and Quality Assessment*** (Ian Taukulis, Kimia Dadkhah, SCSC)
- May 13, 2026 - ***scRNA-seq Analysis Ecosystems: Tools, Resources, & Getting Started with Seurat*** (Alex Emmons, BTEP)
- May 20, 2026 - ***Core Steps in scRNA-seq Analysis: QC to Clustering*** (Alex Emmons, BTEP).
- May 27, 2024 - ***Multi-sample analysis in single cell RNASeq: Batch correction, annotation, and differential expression*** (Nathan Wong, CCBR)
- June 3, 2024 - ***Multimodal Spatial Transcriptomic Analysis*** (Jing Bian, ABCS)
- June 10, 2024 - ***From One Cell to Multiple Molecular Views: A Practical Introduction to Single-Cell Multi-omics*** (Stefan Cordes, NHLBI)

Learning Objectives

- Understand the major ecosystems for analyzing scRNA-seq data
- Know where to find resources for learning R and Python
- Be aware of AI tools relevant to single-cell analysis
- Import count data into R using Seurat
- Create, navigate, and manipulate a Seurat object

Where We Are in the Workflow

`Raw FASTQ → Cell Ranger / STARsolo → Gene × Cell Count Matrix → [YOU ARE HERE]`

> **Note**
>
> This tutorial assumes preprocessing is **complete**: demultiplexing, FASTQ QC, alignment, and error correction have all been run. We are starting from a gene-by-cell count matrix.

If you want a more in-depth understanding of Cell Ranger and upstream steps, check out this earlier talk from 2024: *SCAF: Overview of Cell Ranger output files and single cell data analysis quality control*

Analysis Ecosystems

Three Major Ecosystems

	Seurat	Bioconductor	scverse
Language	R	R	Python
Philosophy	All-in-one	Modular	Modular + ML
Best for	Beginners, multimodal	Statistical control	Large-scale, ML
Core object	`SeuratObject`	`SingleCellExperiment`	`AnnData`

> Your choice depends on your background, your lab's preferences, and your scientific goals — and you don't have to commit to just one.

Ecosystem 1: Seurat (R) ✓

- Developed by the **Satija Lab**
- Unified framework: scRNA-seq, multimodal (RNA + ATAC), spatial
- Large community, extensive documentation
- **v5 additions**: layered assays, sketch-based workflows, disk-backed matrices, bridge integration
- Reference mapping via **Azimuth**

Pros: end-to-end workflows, great vignettes, scalable
Cons: updates can break things, some functions poorly documented, maintained by one group

Resources

- Getting started (v5)
- Seurat vignettes
- GitHub Issues
- v5 announcements

Ecosystem 2: Bioconductor (R)

- Built around the `SingleCellExperiment` object
- Modular — 70+ interoperable packages
- Emphasizes **statistical rigor** and **composability**
- Strong for users who want fine-grained control

Resources

- OSCA — Orchestrating Single-Cell Analysis
- OSTA — Orchestrating Spatial Transcriptomics
- Bioconductor support forum

Ecosystem 3: scverse (Python)

- Built around the **`AnnData`** object
- Collection of interoperable Python packages:
  - `scanpy` — core analysis workflows
  - `scvi-tools` — deep learning, batch correction
  - `muon` / `mudata` — multimodal analysis
  - `scirpy` — immune repertoire
  - `squidpy` — spatial transcriptomics
- Dominant in atlas-scale and ML-based workflows

Resources

- scverse learning hub
- scanpy tutorials
- Single-cell best practices
- BTEP Python Introductory Series

Ecosystem Interoperability

The objects store the same things

Concept	Seurat	SingleCellExperiment	AnnData
Expression data	`assays` (e.g., RNA)	`assays` (counts, logcounts)	`X` and `layers`
Cell metadata	`meta.data`	`colData`	`obs`
Feature metadata	within assays	`rowData`	`var`
Dim. reductions	`reductions`	`reducedDims`	`obsm`
Graphs	`graphs`, `neighbors`	`colPairs`	`obsp`
Misc. results	`misc`, `commands`	`metadata`	`uns`

> ⚠ **Warning**
>
> **Matrix orientation differs.** Seurat and SCE store expression as **features × cells**. AnnData stores as **cells × genes**. Conversion tools handle this — but it's a common source of confusion.

Moving data between ecosystems

Recommended exchange format: `.h5ad`

- `anndataR` — preferred R package; reads/writes `.h5ad`, converts to/from both Seurat and SCE
- `zellkonverter` — reliable for SCE ↔ AnnData in Bioconductor workflows

> **Important**
>
> `SeuratDisk` is **not actively maintained** for Seurat v5. Avoid it for current workflows.

After any conversion, always verify:

- Cell and gene identifiers are intact
- Count layers are present and correct
- Metadata, embeddings, and graphs transferred correctly

Point-and-click alternatives

Not ready to code? GUI-based tools are available:

- **Partek Flow** — license available to NCI researchers
  - Intro to scRNA-seq with Partek Flow
  - scRNA-seq + scATAC-seq integration
  - Advanced scRNA-seq analysis
- **Qlucore Omics Explorer** — license available to NCI-CCR researchers

Learning Resources

R Resources

Getting started

- R Cheat Sheets (Posit)
- R Crash Course
- Glittr — 230+ bioinformatics R repos

BTEP courses

- R Introductory Series
- Data Wrangling with R
- Data Visualization with R
- Troubleshooting R Code

Python Resources

Getting started

- python.org beginners guide
- Glittr — Python bioinformatics repos

BTEP courses

- Python Introductory Education Series

How to find help

1. **Read the documentation first** — `?function_name`, `help()`, `vignette()`
2. **Check GitHub Issues** — e.g., Seurat Issues
3. **Search strategically**: *action + language + package*
   - ✅ `"how to rename a column in R with dplyr"`
   - ❌ `"R column rename"`
4. **Stack Overflow** — your best friend for error messages
5. **Use Generative AI** — BTEP GenAI Resource Guide
6. **Email BTEP** — ncibtep@nih.gov

GenAI and Single-Cell Analysis

Tool	What it does
SCassist	LLM-guided parameter choices and result interpretation in Seurat
CellWhisperer	Natural language interface for exploring scRNA-seq datasets
scGPT	Foundation model for cell annotation, representation learning
Geneformer	Foundation model trained on large transcriptomic datasets

> **Important**
>
> AI-generated suggestions are **decision support**, not ground truth. Always validate against QC metrics, marker genes, and biological context.

Accessing R / RStudio

Computing Options

Local

- Feasible for **≤ ~8 samples**
- Install R + RStudio
- NIH-managed machines: check with IT for admin privileges

Biowulf HPC (Recommended)

- > 90k processors, >600 pre-installed tools
- Seurat v5.4.0 pre-installed with R 4.5.2 ✅
- No package management needed
- Access via NIH HPC Open OnDemand

> **Note**
>
> This tutorial uses the **HPC's centrally managed R library**. If running locally, you may need to install packages manually and may encounter system dependency issues.

Connecting to RStudio on Biowulf

1. Connect to the NIH network
2. Go to hpcondemand.nih.gov and log in with your PIV card
3. Select **RStudio** from the interactive apps
4. Set your resources and launch:

Setting	Recommended
Hours	8
CPUs	8
Memory (GB)	100
Local Scratch (GB)	50

Note

Need a Biowulf account? Accounts are available to all NIH employees and contractors. Requires PI approval and $40/month. You'll receive `/home/USER (16 GB) and /data/$USER` (100 GB) storage.

Getting Started with Seurat

The Dataset

From Niethamer et al. 2025, *Cell Stem Cell* — lung regeneration after influenza infection

Sample	Shortname	Condition	Time Point
GSM8181589	`S89Inf42`	Tamoxifen + Influenza	Day 42 post-infection
GSM8181591	`S91Hom42`	Tamoxifen only	Homeostasis Day 42
GSM8181593	`S93Hom3`	Tamoxifen only	Homeostasis Day 3
GSM8181599	`S99Inf42`	Tamoxifen + Influenza	Day 42 post-infection

- **Lineage tracing**: Ki67-CreERT2; ROSA26-LSL-tdTomato reporter labels proliferating cells
- GSM8181589 and GSM8181599 are **biological replicates** of the infected condition
- Data available via GEO: GSE262927

The Preprocessing Workflow

```
Count Matrix
    ↓
1. Import data
    ↓
2. Quality control       (nFeature, nCount, % mitochondrial reads)
    ↓
3. Normalization         (NormalizeData or SCTransform)
    ↓
4. Feature selection     (highly variable genes)
    ↓
5. Scaling               (ScaleData — or handled by SCTransform)
    ↓
6. Dimensionality reduction  (PCA → UMAP / t-SNE)
    ↓
7. Clustering            (SNN graph → community detection)
    ↓
8. Downstream analysis   (markers, annotation, DE)
```

Analysis Strategy

Strategy A: Merge early (this tutorial)

- Merge into one object; samples kept in separate **layers**
- Visualize QC across all samples together
- Apply per-sample thresholds via `orig.ident`
- ✅ Best for interactive, exploratory analysis

Strategy B: Per-sample objects

- Create, QC, and normalize each sample separately
- Merge / integrate after individual processing
- ✅ Best for scripted, automated pipelines

Loading Packages

```
1 library(Seurat)
2 library(tidyverse)
```

> **Note**
>
> On Biowulf, Seurat v5.4.0 is pre-installed with R 4.5.2. No installation needed.
>
> To install locally: `install.packages("Seurat")`

Importing Data: Key Functions

Function	Use Case
`Read10X()`	Directory with `matrix.mtx`, `barcodes.tsv`, `features.tsv`
`Read10X_h5()`	Single HDF5 file from Cell Ranger
`ReadMtx()`	Flexible — local or remote, any platform
`ReadSTARsolo()`	STARsolo output

Tip

Functions for many other platforms exist too — Akoya CODEX, NanoString SMI, 10X Visium, 10X Xenium, Vizgen MERFISH. See the Seurat reference.

Loading Multiple H5 Files

```
# List all h5 files in the data directory
files <- list.files(path = "./data/", pattern = "*.h5")

# Read all into a named list of sparse matrices
h5_read <- lapply(paste0("./data/", files), Read10X_h5)

# Assign short sample names
names(h5_read) <- c("S89Inf42", "S91Hom42", "S93Hom3", "S99Inf42")
```

```
# Preview the sparse matrix — dots are zeros
h5_read$S89Inf42[1:5,1:5]
```

Creating Seurat Objects

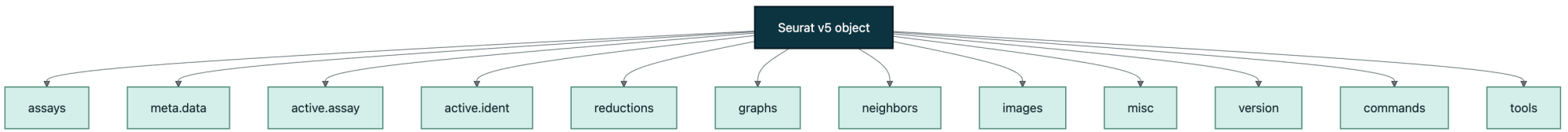
```r
# Create objects for all samples using mapply
lung <- mapply(CreateSeuratObject,
               counts  = h5_read,
               project = names(h5_read),
               MoreArgs = list(min.cells = 10, min.features = 200))

# Remove the original sparse matrices to free memory
rm(h5_read)

# Merge into a single object, prepending sample ID to cell barcodes
lung <- merge(lung[[1]], y = lung[2:length(lung)],
              add.cell.ids = names(lung), project = "Lung")
```

Note

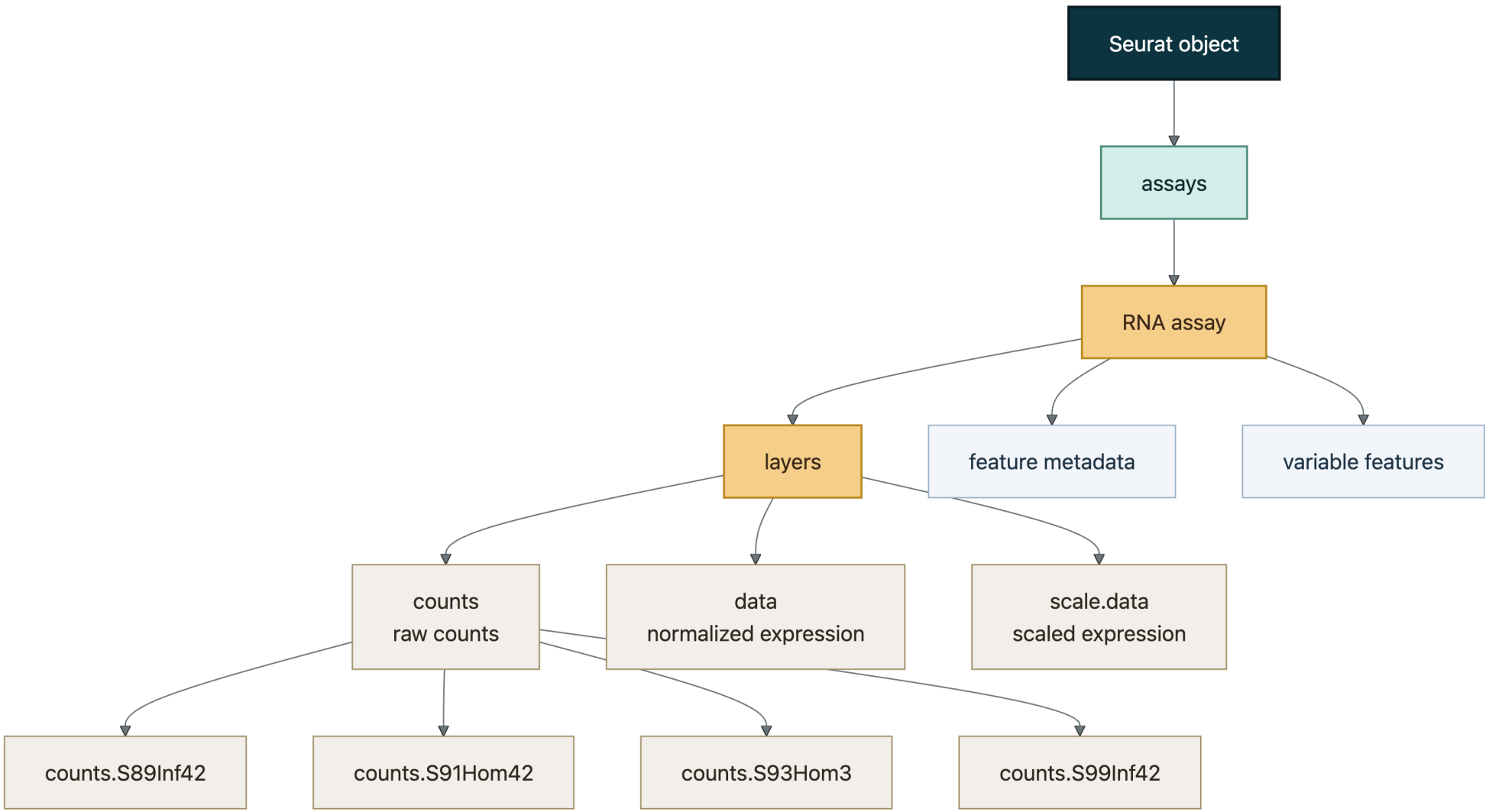
`min.cells = 10` removes genes detected in fewer than 10 cells.
`min.features = 200` removes cells with fewer than 200 detected genes.
`add.cell.ids` prevents barcode collisions across samples.

The Seurat Object

Object Structure: Top-level Slots



Object Structure: RNA Assay Layers



Accessing Data

```r
# What layers exist?
Layers(lung)

# Access the count layer — three equivalent approaches
lung[["RNA"]]$counts |> head()
lung@assays$RNA$counts |> head()
LayerData(lung, assay = "RNA", layer = "counts") |> head()

# Check Seurat version used to build the object
lung@version

# Check what commands have been run
lung@commands
```

Metadata

```
# View metadata
head(lung@meta.data)
head(lung[[]])          # shorthand

# Access specific columns
head(lung$orig.ident)
head(lung[["nCount_RNA"]])
head(lung[[c("orig.ident", "nCount_RNA")]])
```

Built-in columns after object creation:

Column	Meaning
`orig.ident`	Sample / project label
`nCount_RNA`	Total UMIs per cell
`nFeature_RNA`	Number of genes detected

Adding Custom Metadata

```r
# Treatment condition
lung$treatment <- ifelse(
  str_detect(lung$orig.ident, "S91Hom42|S93Hom3"),
  "Tamoxifen", "Tamoxifen_Influenza"
)

# Time point
lung$time_point <- case_when(
  str_detect(lung$orig.ident, "S93Hom3")  ~ "Homeostasis_Day3",
  str_detect(lung$orig.ident, "S91Hom42") ~ "Homeostasis_Day42",
  .default = "Infection_Day42"
)

# Combined label
lung$treatment_tp <- paste(lung$treatment, lung$time_point, sep = "_")
```

Useful Object Queries

```
# Cell names and feature names
head(Cells(lung, layer = "counts.S89Inf42"))
head(colnames(lung))
head(Features(lung))
head(rownames(lung))

# Dimensions
ncol(lung)       # number of cells
nrow(lung)       # number of features
dim(lung)        # rows x columns

# List assays (important for multimodal data)
Assays(lung)
```

Quick Cell Count Check

```
sample_colors <- c(
  S89Inf42 = "#0072B2", S91Hom42 = "#E69F00",
  S93Hom3  = "#009E73", S99Inf42 = "#CC79A7"
)

lung@meta.data %>%
  ggplot(aes(x = orig.ident, fill = orig.ident)) +
  geom_bar(color = "black") +
  stat_count(geom = "text",
             aes(label = after_stat(count)),
             position = position_stack(vjust = 0.5)) +
  scale_fill_manual(values = sample_colors) +
  theme_classic() +
  ggtitle("Number of Cells per Sample")
```

Save Your Object

```
1  saveRDS(lung, "./outputs/merged_Seurat_lung.rds")
```

> **Tip**
>
> Save at key checkpoints — pre-QC, post-QC, post-normalization, post-clustering. Use descriptive filenames so you know exactly what stage each file represents.

Summary

What We Covered

- ✅ Three major scRNA-seq ecosystems — Seurat, Bioconductor, scverse
- ✅ Object equivalences and interoperability across ecosystems
- ✅ Learning resources for R and Python
- ✅ AI tools emerging in single-cell analysis
- ✅ Computing options — local vs. Biowulf HPC
- ✅ Importing 10X HDF5 files with `Read10X_h5()`
- ✅ Creating and merging Seurat objects
- ✅ Navigating the Seurat object: slots, layers, metadata

Up Next: Standard Seurat Pre-Processing Workflow

1. Apply quality control filters to retain high-quality cells
2. Normalize the data with `SCTransform()`
3. Perform PCA and UMAP.
4. Cluster

Coming Up:

- Core Steps in scRNA-seq Analysis: QC to Clustering (5/20)
- Multi-sample analysis in single cell RNASeq: Batch correction, annotation, and differential expression (5/27)

Resources

- Seurat v5 Essential Commands
- OSCA Book
- Single-Cell Best Practices
- anndataR documentation
- BTEP GenAI Resource Guide
- NIH HPC Open OnDemand
- Seurat Objects Explained (biostatsquid)

Questions? → ncibtep@nih.gov