Core Steps in scRNA-seq Analysis: QC to Clustering

Single-Cell & Spatial Omics Series · NCI CCR BTEP · 2026

Alex Emmons, PhD

2026-05-20

Single-Cell & Spatial Omics 2026

Upcoming Events

- **Core Steps in scRNA-seq Analysis: QC to Clustering** ← *you are here*
- Multi-sample analysis in single cell RNASeq: Batch correction, annotation, and differential expression (5/27)
- Multimodal Spatial Transcriptomic Analysis (6/03)
- From One Cell to Multiple Molecular Views: A Practical Introduction to Single-Cell Multi-omics (6/10)

Resources

🔗 Series Home

📦 Download Tutorial Data

📧 ncibtep@nih.gov

Learning Objectives

By the end of this session you will be able to:

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

> ⚠ **Warning**
>
> Thresholds and clustering decisions shown here are **starting points** — revisit them in the context of your biology.

The Dataset

Lung regeneration after influenza infection Niethamer et al. 2025, *Cell Stem Cell*

Sample	Condition	Time Point
S89Inf42	Influenza	Day 42 post-infection
S91Hom42	Homeostasis	Day 42
S93Hom3	Homeostasis	Day 3
S99Inf42	Influenza	Day 42 post-infection

- **2 biological replicates** of the infected condition at day 42
- Comparison of injury-associated states vs. homeostatic controls
- Expect both **technical** and **biological** variation between samples

Load Packages & Object

```r
library(Seurat)
library(tidyverse)
library(hdf5r)
library(patchwork)
library(scuttle)
library(glmGamPoi)  # speeds up SCTransform
library(presto)     # speeds up FindAllMarkers
```

```r
lung <- readRDS("./outputs/merged_Seurat_lung.rds")
lung
Layers(lung)
```

> **Note**
>
> In **Seurat v5**, merged RNA assays retain per-sample matrices as separate layers (e.g., `counts.S89Inf42`). This enables per-sample QC even on a merged object.

Quality Control: The Goal

Remove **low-quality cells** before downstream analysis

`nCount_RNA`

Total UMI count per cell

- 🔺 High → possible doublet
- 🔻 Low → ambient RNA / empty droplet

`nFeature_RNA`

Genes detected per cell

- 🔺 High → possible doublet
- 🔻 Low → ambient RNA / low complexity

`percent.mt`

Mitochondrial read fraction

- 🔺 High → dying/damaged cell

> ⚠️ Always consider **biological explanations** — high mt% can reflect metabolic activity; low counts may indicate quiescent populations.

Add Mitochondrial Percentage

```r
# Mouse dataset — mitochondrial genes start with 'mt-'
# Human datasets use '^MT-'
lung[["percent.mt"]] <- PercentageFeatureSet(lung, pattern = "^mt-")
```

```r
# Extract metadata for plotting
metadata <- lung[[]]
head(metadata)
```

> **Note**
>
> **QC should be sample-aware.** Visualize metrics per sample — differences may reflect biology, not just technical issues.

QC Assessment Strategy

Metrics should be assessed jointly, not in isolation

Manual thresholds

- Subjective visual assessment
- Can be arbitrary
- ✅ What we do here

Adaptive thresholds

- Uses the data's own distribution
- Outliers defined as 3 or 5 MAD from the median
- `scuttle::isOutlier()`
- 📖 OSCA: QC outlier detection

Our Strategy:

- **Step 1: Visualize QC metrics**
- **Step 2: Decide on thresholds**
- **Step 3: Filter the Object**

Step 1: Visualize QC Metrics

```r
VlnPlot(lung, features = "nCount_RNA", layer = "counts",
        group.by = "orig.ident", alpha = 0.05)

metadata %>%
  ggplot(aes(x = nCount_RNA, fill = orig.ident, color = orig.ident)) +
  geom_density(alpha = 0.2) + theme_classic() + scale_x_log10() +
  geom_vline(xintercept = 600, color = "red", linetype = "dotted")
```

```r
metadata %>%
  ggplot(aes(x = nFeature_RNA, fill = orig.ident, color = orig.ident)) +
  geom_density(alpha = 0.2) + theme_classic() + scale_x_log10() +
  geom_vline(xintercept = c(375, 5500), color = "red", linetype = "dotted")

metadata %>%
  ggplot(aes(x = percent.mt, fill = orig.ident, color = orig.ident)) +
  geom_density(alpha = 0.2) + theme_classic() + scale_x_log10() +
  geom_vline(xintercept = 15)
```

Check out scCustomize for additional plotting options and flexible color palettes.

Step 2 / 3: Set QC Thresholds and Filter

Chosen thresholds (conservative starting point):

Metric	Filter
`nFeature_RNA`	> 375
`nCount_RNA`	> 600
`percent.mt`	< 15%

```r
keep_cells <- metadata %>%
  rownames_to_column("cell") %>%
  filter(nFeature_RNA > 375,
         nCount_RNA > 600,
         percent.mt < 15) %>%
  pull(cell)

lung_filt <- subset(lung, cells = keep_cells)
```

Check: Cells Before & After Filtering

```
bind_rows(
  before = metadata,
  after  = lung_filt[[]],
  .id = "stage"
) %>%
  count(stage, orig.ident) %>%
  ggplot(aes(x = orig.ident, y = n, fill = stage)) +
  geom_col(position = "dodge") + theme_classic() +
  geom_text(aes(label = n, group = stage),
            position = position_dodge(width = 0.9))
```

```
saveRDS(lung_filt, "./outputs/merged_Seurat_lung_qcfiltered.rds")
```

> **Tip**
>
> 💾 Save your filtered object — it's a useful checkpoint to return to.

Why Normalize?

Raw counts are **not directly comparable** across cells

- Cells vary in total mRNA content and technical capture efficiency
- Sources of technical variation include differences in cell lysis, reverse transcription efficiency, and stochastic molecular sampling
- Normalization minimizes technical effects so expression can be compared across cells

Standard workflow

1. `NormalizeData()`
2. `FindVariableFeatures()`
3. `ScaleData()`

Modern approach

✅ `SCTransform()` — one step, better statistical model

SCTransform

```
lung_filt <- SCTransform(
  lung_filt,
  vars.to.regress = "percent.mt",
  verbose = FALSE
)
```

SCTransform output — `SCT` assay layers:

Layer	Contents
`counts`	Regularized corrected counts
`data`	Log-normalized corrected counts
`scale.data`	Pearson residuals for variable features → used in PCA

```
DefaultAssay(lung_filt)  # should now be "SCT"
```

Dimensionality Reduction: PCA

Single-cell data is **highly dimensional** — thousands of genes per cell

PCA summarizes correlated gene expression patterns into principal components (PCs), which are used as the basis for clustering

```
lung_filt <- RunPCA(lung_filt, assay = "SCT", verbose = FALSE)
```

```
ElbowPlot(lung_filt, ndims = 40)
```

```
DimHeatmap(lung_filt, dims = 1:9, cells = 500, balanced = TRUE, ncol = 3)
DimHeatmap(lung_filt, dims = 20:30, cells = 500, balanced = TRUE, ncol = 3)
```

Graph-Based Clustering

Three steps:

1. `FindNeighbors()` — builds a KNN graph in PCA space; refines edge weights using Jaccard similarity
2. `FindClusters()` — partitions the graph into communities using modularity optimization
3. `RunUMAP()` — 2D visualization of the structure

Resolution parameter

Controls granularity

🔺 Higher → more clusters
🔻 Lower → fewer clusters

Try several and compare!

Run Clustering & UMAP

```
lung_filt <- FindNeighbors(lung_filt, dims = 1:30)
lung_filt <- FindClusters(lung_filt, resolution = 0.2)
lung_filt <- RunUMAP(lung_filt, dims = 1:30)
```

```
DimPlot(
  lung_filt, reduction = "umap",
  group.by = c("orig.ident", "seurat_clusters",
               "time_point", "treatment"),
  ncol = 2, alpha = 0.2
)

DimPlot(lung_filt, reduction = "umap",
        group.by = "orig.ident",
        split.by = "treatment", ncol = 2, alpha = 0.2)
```

Annotate with Canonical Markers

```
FeaturePlot(lung_filt, features = c("Epcam",   # epithelial
                                    "Ptprc"))  # immune (CD45)
```

Canonical markers are biologically validated genes known to define specific cell types

Combine with `VlnPlot()`, `DotPlot()` for detailed inspection

Annotation tools:

- Azimuth
- SingleR
- scType
- Garnett

Find Cluster Markers

```r
# Required before DE testing with SCT + multiple samples
lung_filt <- PrepSCTFindMarkers(lung_filt, verbose = TRUE)
```

```r
lung_filt_markers <- FindAllMarkers(lung_filt, only.pos = TRUE)

lung_filt_markers <- lung_filt_markers |>
  group_by(cluster) |>
  arrange(desc(avg_log2FC), pct.1, p_val_adj, .by_group = TRUE)
```

Column	Meaning
`pct.1`	% of cells in **this cluster** expressing the gene
`pct.2`	% in **all other clusters**
`avg_log2FC`	Fold change vs. other clusters

Save & Next Steps

```
saveRDS(lung_filt, "./outputs/merged_Seurat_lung_sct_clustered.rds")
```

Typical next steps:

- 🔗 **Integration** — if cells cluster primarily by sample or batch
- 🏷️ **Cluster annotation** — markers + reference tools + biology
- 🔬 **Subsetting & re-clustering** — zoom into populations of interest
- 📊 **Differential expression** — across clusters, conditions, or time points
- 🧮 **Cell composition analysis** — do proportions differ across treatments?
- 🌊 **Trajectory analysis** — model injury → regeneration state transitions

Resources & References

Tutorial & Data

- Series home
- Seurat v5 tutorial
- SCTransform vignette
- Single-Cell Best Practices
- HBC scRNA-seq training

Data

- Niethamer et al. 2025, *Cell Stem Cell*

Key Papers

- Luecken & Theis (2019) *Mol Syst Biol* — best practices review
- Heumos et al. (2023) *Nat Rev Genet* — best practices across modalities
- Hafemeister & Satija (2019) *Genome Biol* — SCTransform

Questions?

ncibtep@nih.gov

Office hours & video archive at bioinformatics.ccr.cancer.gov

Appendix: Adaptive QC with MAD

An alternative to manual thresholds — `scuttle::isOutlier()`

```
library(scuttle)

outlier_low_feat <- isOutlier(
  metadata$nFeature_RNA,
  nmads = 3,
  type = "lower",
  log = TRUE
)
summary(outlier_low_feat)
```

> Outliers are generally defined as 3 or 5 MADs from the median. More information on adaptive thresholds: OSCA: QC outlier detection

Appendix: Standard Normalization Workflow

For comparison — the classic three-step approach:

```
lung_filt <- NormalizeData(lung_filt,
               normalization.method = "LogNormalize",
               scale.factor = 10000)

lung_filt <- FindVariableFeatures(lung_filt,
               selection.method = "vst",
               nfeatures = 2000)

lung_filt <- ScaleData(lung_filt,
               vars.to.regress = "percent.mt")
```

We prefer `SCTransform()` for this dataset, but both are valid starting points.