Introduction to Seurat

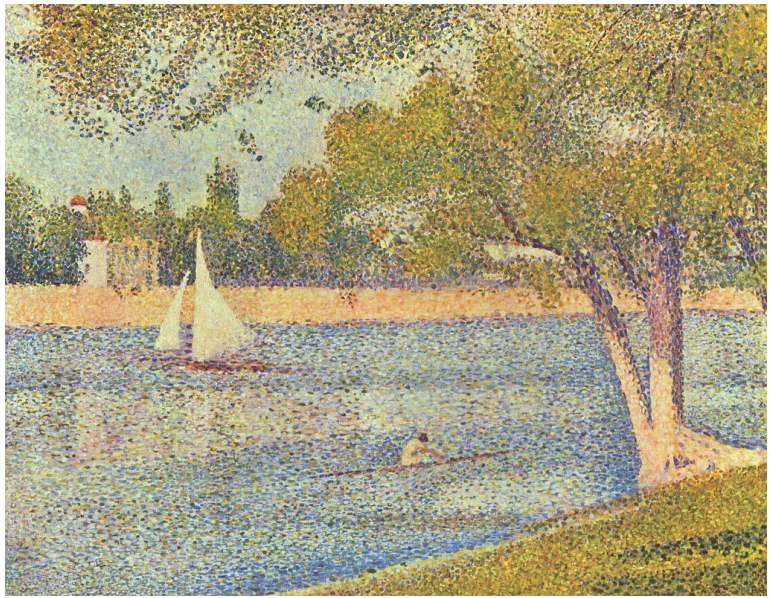
Emily Franz and Jack Hedberg

Outline: scRNA-Seq Analysis of one sample

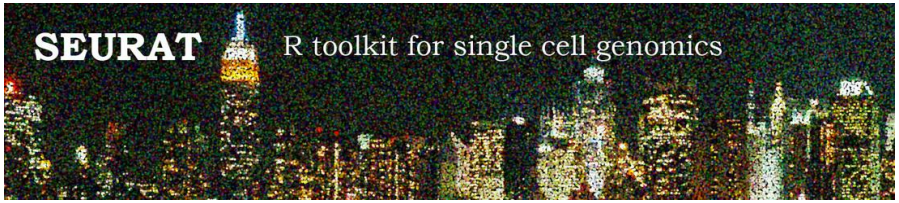
- Reading in CellRanger Outputs, Creating Seurat Objects in R
- Structure of a Seurat Object
- Major Object Processing Steps
  - Quality Control and Subsetting Low Quality Cells
  - Normalization
  - Identifying Highly Variable Features
  - Scaling
  - PCA
  - FindNeighbors(), FindClusters(), and Dimension Reduction
- High Yield Seurat Object Operations
  - ReadRDS and SaveRDS

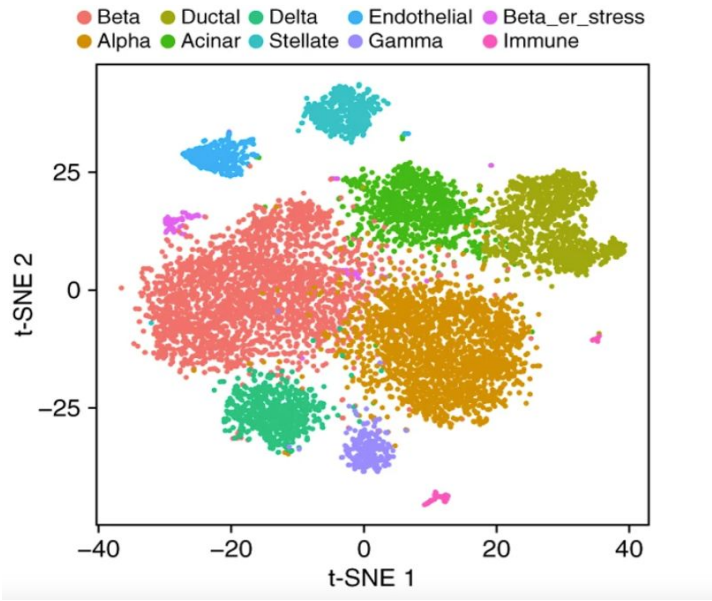
Georges Seurat

"Some say they see poetry in my paintings, I see only science." - Georges Seurat



satijalab.org/seurat





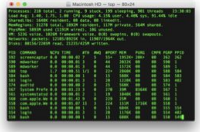
Butler, A., Hoffman, P., Smibert, P. *et al.* Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nat Biotechnol* 36, 411–420 (2018). https://doi.org/10.1038/nbt.4096

Where We Are in the Single Cell Sequencing Workflow

Sequencing

Raw Base Call (BCL) Files

Computer Terminal



CellRanger mkfastq

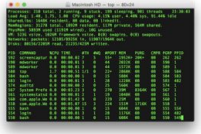
FASTQ File

```
@SEQ_ID GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAA
+
!''*((((***+))%%%++)(%%%%).1***-+*''))**55CCF
```

1. Sequence identifier (sequencing run + cluster)
2. Base calls (sequence itself)
3. Separator
4. Base call quality scores

CellRanger Count

Computer Terminal



Multiple Output Files

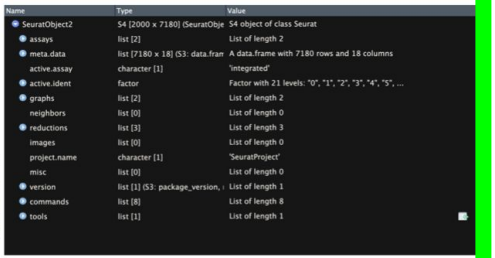
1. Web Summary.html
2. BAM
3. Matrices
4. Molecule Info (h5)
5. Secondary Analysis CSV
6. Loupe File (.cloupe)

Read10X()

RStudio

SEURAT R toolkit for single cell genomics

Seurat Object



How Gene Expression Values Are Stored in Seurat Objects

Counts matrix
covid@assays$RNA@counts

Each row is a gene

Each column is a cell

	(Cell 1) AGAGCATCG	(Cell 2) AAAGCAAAT	(Cell 3) TTATATGTA	(Cell 4) GCTCGTCAA	(Cell 5) AGTCATGAC	(Cell 6) GCGGCTCAC	(Cell 7) TTTCGCGTC	… etc
Gene 1								
Gene 2								
Gene 3								
Gene 4								

How a Final scRNA-seq Library Molecule Populates Information in a Seurat Object

scRNA-Seq Library Molecule

Seurat Object Counts Matrix

	AATGCTGA TCGATCTA GATGA	AAGTCTCC TGTTCTCAT CTGGTA	ATCGACTG CGTGTAGC TACACG	TTCGACAC GTAGCATG CTAGCC
TP53	.	.	2	.
ERCC4	.	.	.	.
ARF7	1	.	.	.
SMAD3	.	17	.	.
TGFB1	.	.	4	.
CD8A	.	.	.	.

+

Metadata Specifying Sample

Opening R, Reading in Covid Data, Create Seurat Object

- Install and load necessary packages

- Read in covid dataset

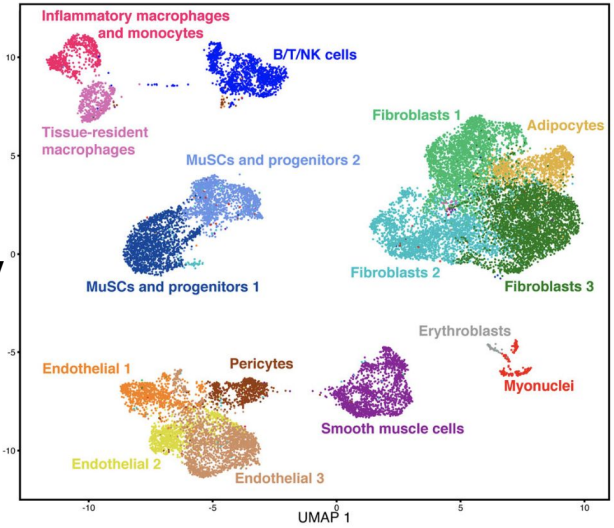
Problems Addressed by Major scRNA-Seq Analysis Steps

- Do not want poor quality cells decreasing the quality of the analysis-> QC, subsetting data
- Differential sequencing depth of different cells -> Normalization
- More highly abundant genes display larger values in their variation -> Scaling
- Measuring tens of thousands of different variables and need to distill this into a smaller number as the basis of assessing and visualizing biologically meaningful differences in the data -> Dimension Reduction, FindNeighbors, FindClusters

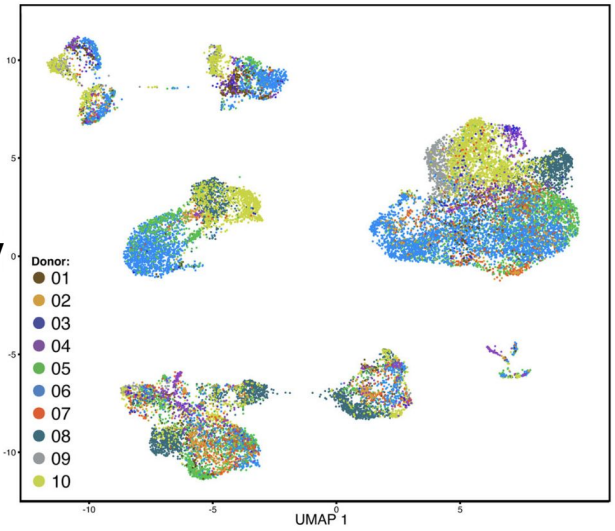
Metadata

- Additional data associated with specific cells
- Examples:
  - Sample of origin
  - Percent of each cell's total reads aligning to mitochondrial genes
  - Whether or not that cell expressed an interesting gene
  - Total number of features (genes) with at least 1 sequencing read in each cell (nFeatures)
  - Later: cell type annotations
- The same cell barcodes ('AATGTATCTAACTATA') used in the counts matrix are used in the metadata to identify the corresponding cells

Data grouped by 'Cell Type' Metadata



Data grouped by 'Donor' Metadata



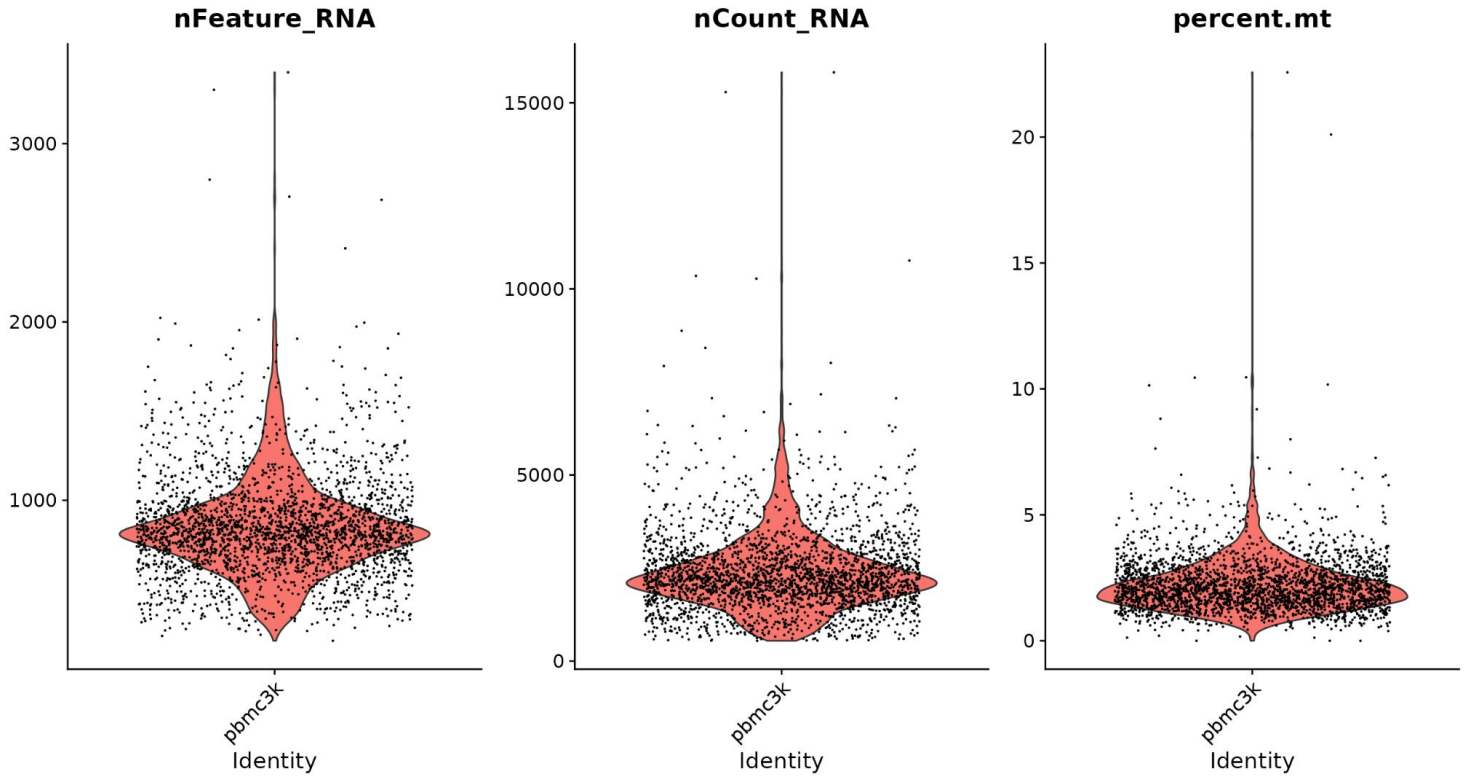
De Micheli, A.J., Spector, J.A., Elemento, O. *et al.* A reference single-cell transcriptomic atlas of human skeletal muscle tissue reveals bifurcated muscle stem cell populations. *Skeletal Muscle* 10, 19 (2020). https://doi.org/10.1186/s13395-020-00236-3

Quality Control: Defining and Removing Poor Quality Cells

- What Makes a ‘low quality’ cell?
  - Low number of detected genes
  - Low count depth
  - High fraction of mitochondrial counts
  - Can represent dying cells with broken membranes

- Things we do NOT address in this session
  - Doublets
  - Ambient RNA

Quality Control: Defining and Removing Poor Quality Cells



Normalization

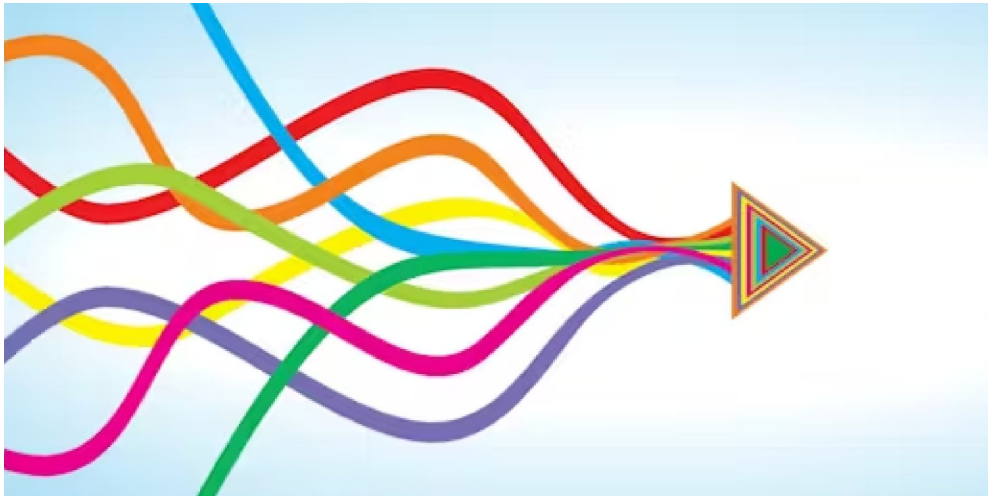
Normalization is the process of adjusting gene expression values to eliminate/reduce the influence of technical effects impacting all genes the same way.

Technical effects are inevitably introduced by the single cell sequencing workflow, they have nothing to do with biology, and we want them gone.

Sources can include:

- Sequencing depth variation
- Differences in cell lysis
- Reverse transcription efficiency
- Stochastic molecular sampling during sequencing

Identifying Highly Variable Features with FindVariableFeatures()



- Main purpose is to set the stage for scaling (described in next slides), in which performing scaling on only the most highly variable genes (features) can save computer power/time

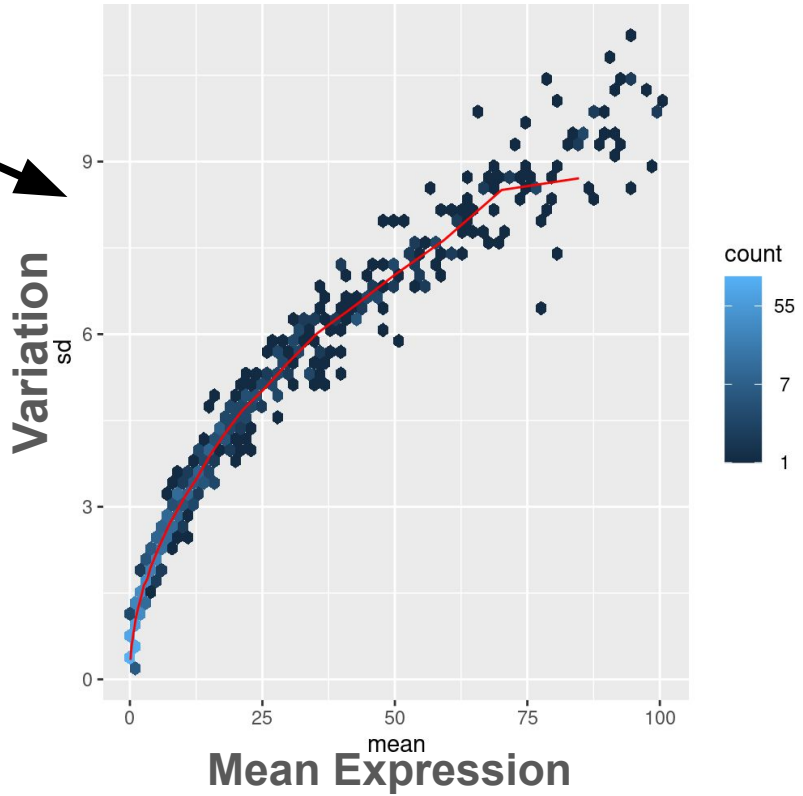
- However, in this vignette we will perform scaling on all genes

Scaling: Why It is Useful

Heteroscedasticity = as mean expression increases, so does observed variation in expression, even after normalization.

Scaling adjusts gene expression values to correct for this increasing variation in gene expression observed with increasing mean expression.

Puts gene expression values on the same ‘playing field’



https://bioconductor.org/packages/release/workflows/vignettes/rnaseqGene/inst/doc/rnaseqGene.html#the-variance-stabilizing-transformation-and-the-rlog

Major Takeaway from Normalization and Scaling

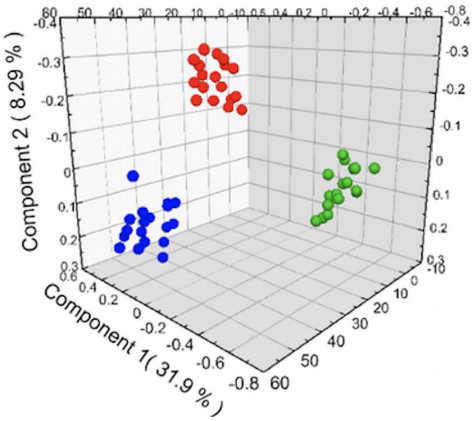
The goal of normalization and scaling is to make the quantitative gene expression values in our single cell sequencing dataset more reflective of biology, and less influenced by technical effects.

This better equips us to directly compare gene expression between different cells, different genes, and gain biological insights from those comparisons.

Dimension Reduction: Principal Component Analysis (PCA)

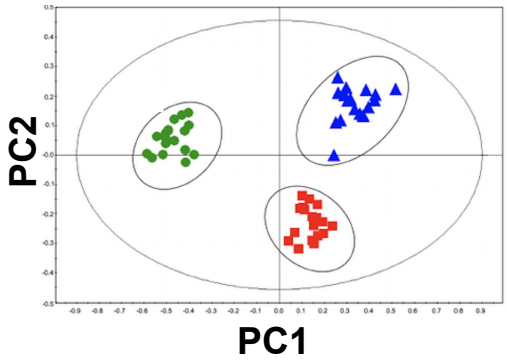
- In single cell sequencing, we have measured tens of thousands of variables (genes) in each cell
- Computationally and conceptually, this is not manageable
- Need to prioritize the sets of genes driving most of the variation in our data
- The process of identifying this is **dimension reduction**, and here we illustrate one form of it: PCA

3 Dimensions



PCA

2 Dimensions



20,000 Dimensions

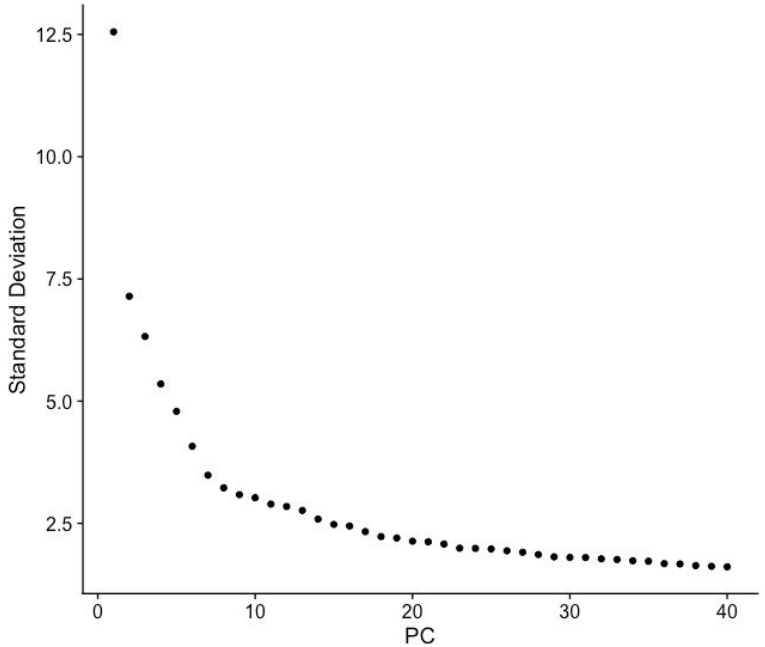
PCA

Small number of principal components

Dimension Reduction: Principal Component Analysis (PCA)

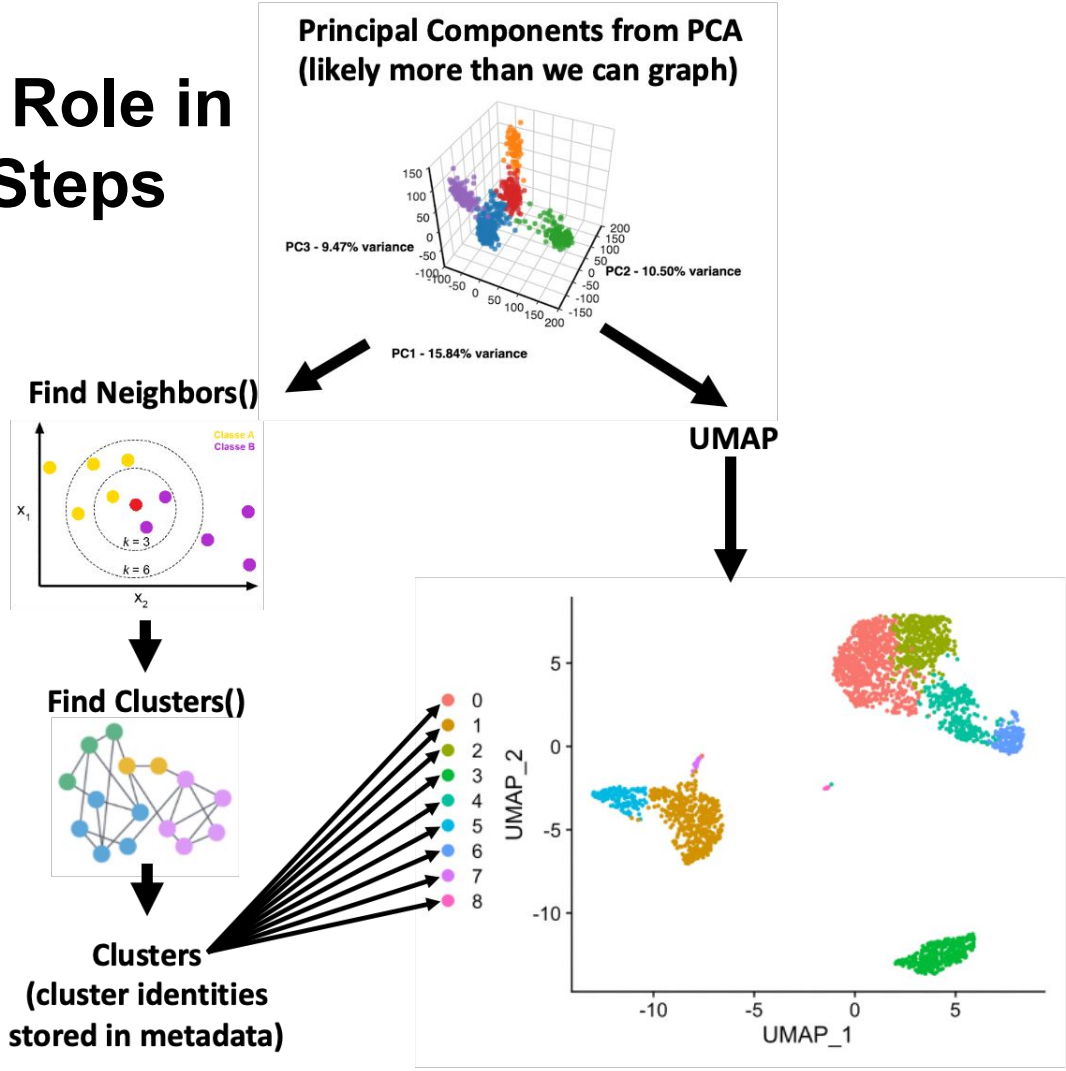
- After running PCA, want to know how many principal components needed to describe most of the variation in our data

- ElbowPlot plots the standard deviations of the principal components.
  - The left of the 'elbow' tends to represent the significant dimensions.
- Other visualization methods:
  - JackStraw (slow)
  - DimHeatmap



PCA Data Play a Crucial Role in Downstream Analytical Steps

- Results of PCA are input into multiple downstream steps
- Formal mathematical assignment of clusters is DISTINCT from UMAP (more in the coming slides)
- Most of the operations shown cannot be graphically represented as they are in high-dimensional space



Break

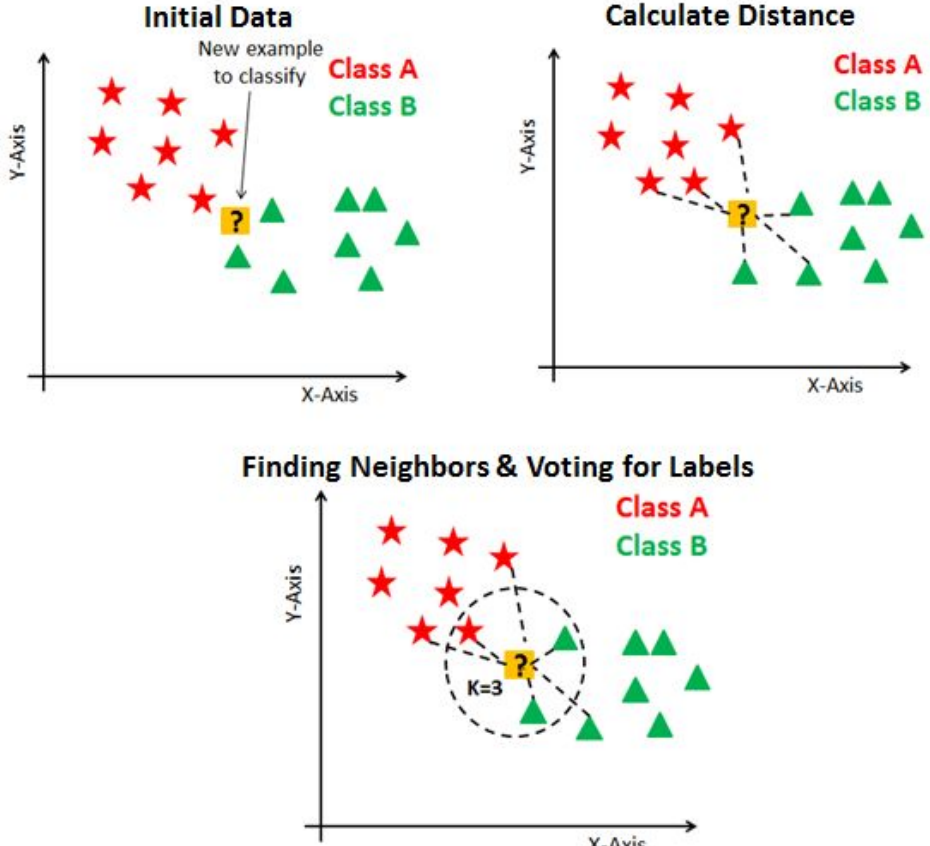
Clustering

This portion includes two steps: FindNeighbors and FindClusters.

The first, **FindNeighbors**:

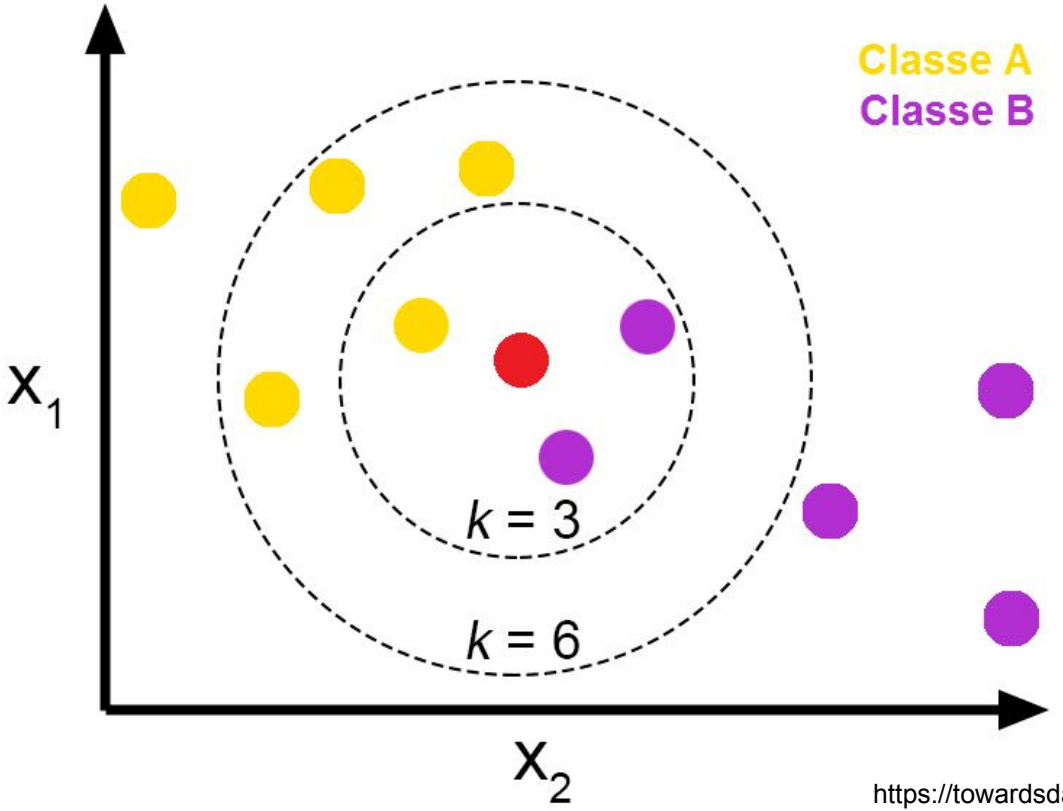
- Takes the principle components selected above and constructs a graph based on the euclidean distance from the PCA
- Finds the similarities between two cells based on overlap of their local neighborhoods.
- Does the above using the k-nearest neighbors (KNN) algorithm
    - k defines the number of neighbors to find
    - Default for Seurat is k=20

Clustering: Find Neighbors



https://towardsdatascience.com/knn-visualization-in-just-13-lines-of-code-32820d72c6b6

Clustering: Find Neighbors



https://towardsdatascience.com/knn-k-nearest-neighbors-1-a4707b24bd1d

Clustering: Find Clusters

The second step, **FindClusters**:

- Iteratively groups cells together with the ability to set the resolution, or granularity, of the clustering.
- The higher the resolution, the greater the number of clusters in the output.

Clustering: Find Clusters

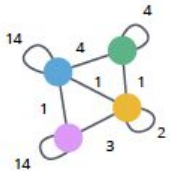
Pass 1

Step 0

Choose a start node and calculate the change in modularity that would occur if that node joins and forms a community with each of its immediate neighbors.

Step 1

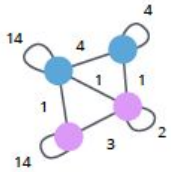
The start node joins the node with the highest modularity change. The process is repeated for each node with the above communities formed.



Step 2

Communities are aggregated to create super communities and the relationships between these super nodes are weighted as a sum of previous links. (Self-loops represent the previous relationships now hidden in the super node.)

Pass 2

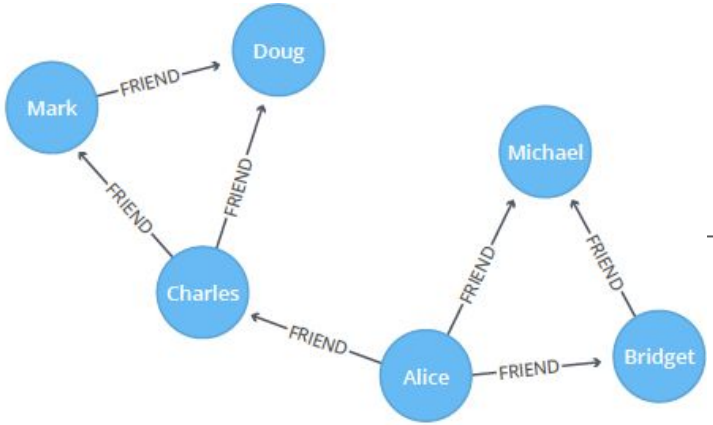


Step 1

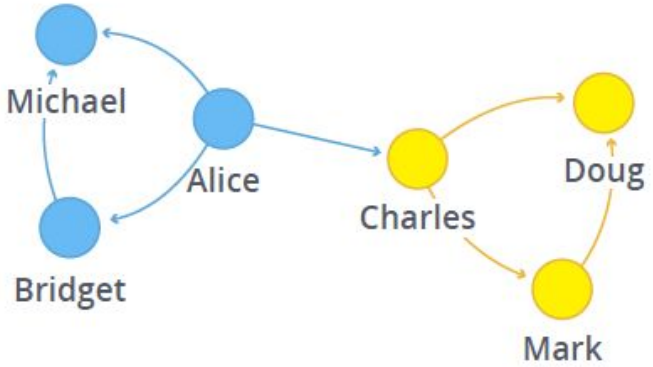
Step 2

Steps 1 and 2 repeat in passes until there is no further increase in modularity or a set number of iterations have occurred.

Clustering: Find Clusters



Graph Model



Visualization of Louvain

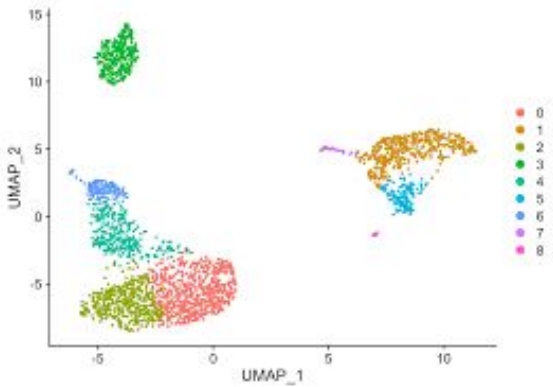
Run Non-linear Dimensional Reduction (UMAP/tSNE)

- Uses graph layout algorithms to arrange data in low-dimensional space
  - This step places similar cells together in a low-dimension (2D) graphical space

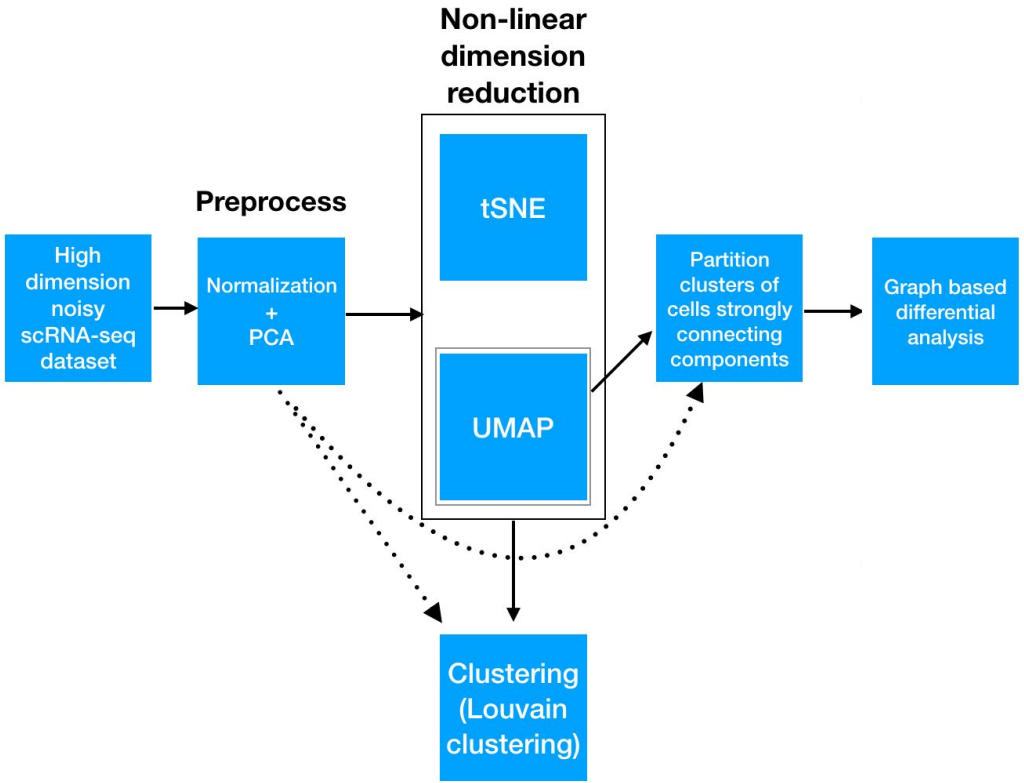
https://pair-code.github.io/understanding-umap/

Run Non-linear Dimensional Reduction (UMAP/tSNE)

- Those cells calculated as co-localized in the clusters generated above should co-localize in the UMAP or tSNE space, however, these are independent functions.
    - User should utilize the same PCs and resolution as defined during clustering
- Cells are colored by their cluster or identity class.



Run Non-linear Dimensional Reduction (UMAP/tSNE)

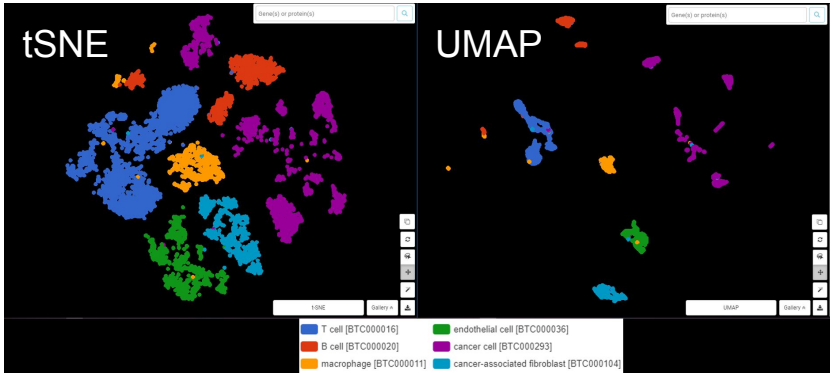


https://cole-trapnell-lab.github.io/monocle-release/monocle3/

Run Non-linear Dimensional Reduction (UMAP/tSNE)

tSNE vs.UMAP:

- UMAP lowers the dimensions of the high dimensional graph from above using compression
  - tends to provide better balance between local versus global structure
  - more time-effective.
- t-SNE moves the high dimensional graph to a lower dimensional space points by points.



Run Non-linear Dimensional Reduction (UMAP/tSNE)

For more information:

Comparing UMAP vs t-SNE in Single-cell RNA-Seq Data Visualization, Simply Explained (https://blog.bioturing.com/2022/01/14/umap-vs-t-sne-single-cell-rna-seq-data-visualization/)

Understanding UMAP (https://pair-code.github.io/understanding-umap/)

Structure of a Seurat Object Revisited

Seurat Object

@assays	List of assays 
@meta.data	Dataframe
@active.assay	Active, or default, assay
@active.ident	Active cluster identity
@graphs	List of Graph objects
@neighbors	k.param nearest neighbors
@reductions	List of dim. red. objects
@images	List of spatial image objects
@project.name	Name of the project
@misc	List of miscellaneous information
@version	Seurat Version
@commands	List of commands run on this object
@tools	List of miscellaneous data

Assay Object

@counts	Raw counts
@data	Normalized expression data
@scale.data	Scaled expression data
@key	Assay key
@assay.orig	Original assay
@var.features	Variable features
@meta.features	Feature-level metadata
@misc	Utility slot

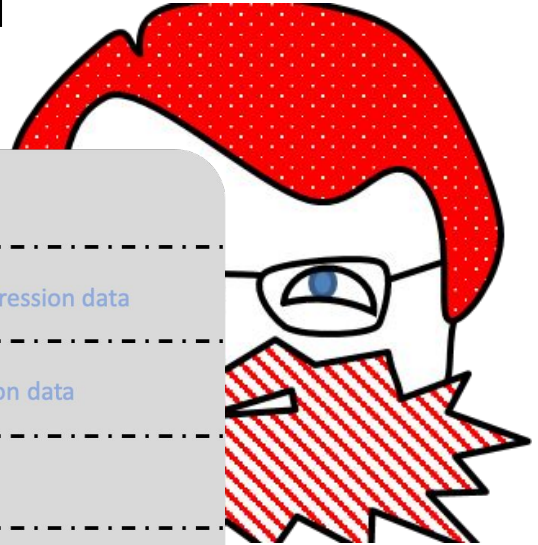


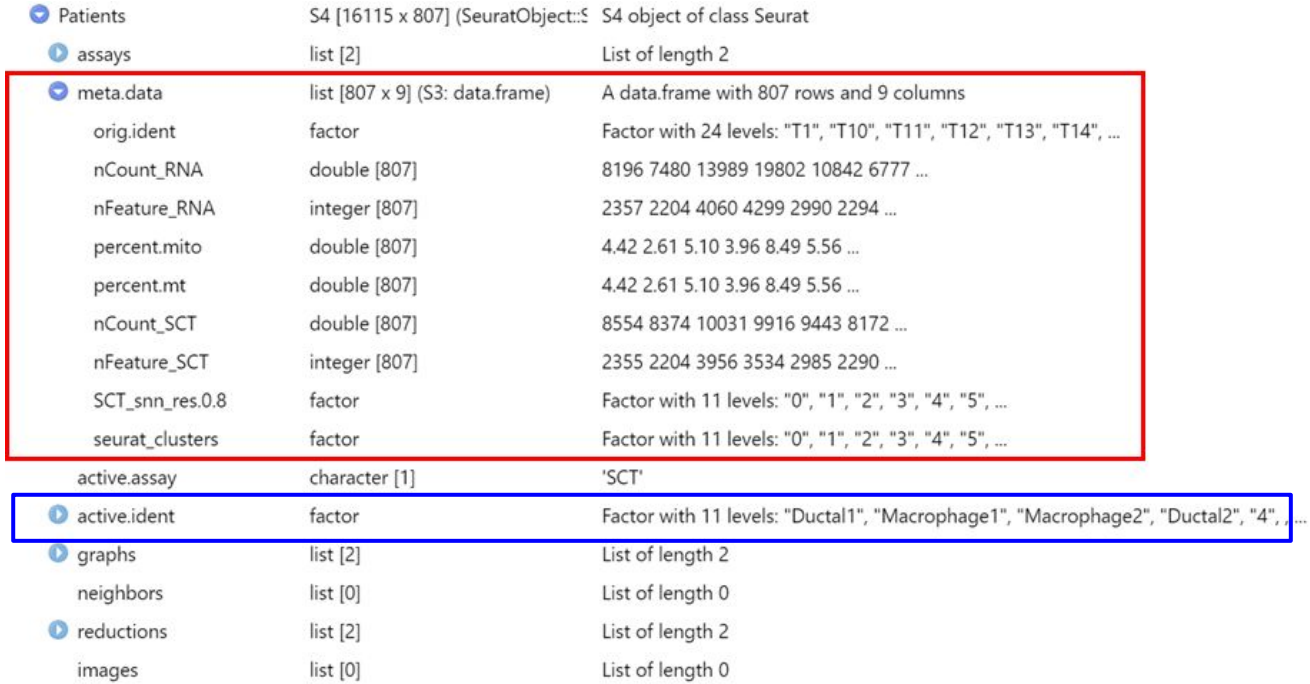
Figure by Matt Cannon

Metadata

Patients	S4 [16115 x 807] (SeuratObject::S	S4 object of class Seurat
assays	list [2]	List of length 2
meta.data	list [807 x 9] (S3: data.frame)	A data.frame with 807 rows and 9 columns
orig.ident	factor	Factor with 24 levels: "T1", "T10", "T11", "T12", "T13", "T14", ...
nCount_RNA	double [807]	8196 7480 13989 19802 10842 6777 ...
nFeature_RNA	integer [807]	2357 2204 4060 4299 2990 2294 ...
percent.mito	double [807]	4.42 2.61 5.10 3.96 8.49 5.56 ...
percent.mt	double [807]	4.42 2.61 5.10 3.96 8.49 5.56 ...
nCount_SCT	double [807]	8554 8374 10031 9916 9443 8172 ...
nFeature_SCT	integer [807]	2355 2204 3956 3534 2985 2290 ...
SCT_snn_res.0.8	factor	Factor with 11 levels: "0", "1", "2", "3", "4", "5", ...
seurat_clusters	factor	Factor with 11 levels: "0", "1", "2", "3", "4", "5", ...
active.assay	character [1]	'SCT'
active.ident	factor	Factor with 11 levels: "Ductal1", "Macrophage1", "Macrophage2", "Ductal2", "4", , ...
graphs	list [2]	List of length 2
neighbors	list [0]	List of length 0
reductions	list [2]	List of length 2
images	list [0]	List of length 0

Active Idents

active.idents = the metadata that your cells are currently grouped by



Patients	S4 [16115 x 807] (SeuratObject::S	S4 object of class Seurat
assays	list [2]	List of length 2
meta.data	list [807 x 9] (S3: data.frame)	A data.frame with 807 rows and 9 columns
orig.ident	factor	Factor with 24 levels: "T1", "T10", "T11", "T12", "T13", "T14", ...
nCount_RNA	double [807]	8196 7480 13989 19802 10842 6777 ...
nFeature_RNA	integer [807]	2357 2204 4060 4299 2990 2294 ...
percent.mito	double [807]	4.42 2.61 5.10 3.96 8.49 5.56 ...
percent.mt	double [807]	4.42 2.61 5.10 3.96 8.49 5.56 ...
nCount_SCT	double [807]	8554 8374 10031 9916 9443 8172 ...
nFeature_SCT	integer [807]	2355 2204 3956 3534 2985 2290 ...
SCT_snn_res.0.8	factor	Factor with 11 levels: "0", "1", "2", "3", "4", "5", ...
seurat_clusters	factor	Factor with 11 levels: "0", "1", "2", "3", "4", "5", ...
active.assay	character [1]	'SCT'
active.ident	factor	Factor with 11 levels: "Ductal1", "Macrophage1", "Macrophage2", "Ductal2", "4", ...
graphs	list [2]	List of length 2
neighbors	list [0]	List of length 0
reductions	list [2]	List of length 2
images	list [0]	List of length 0

save and load

Purpose: load() and save() keep the object name and load the object into the current environment

- The object or data is stored as a binary version. This is a good way to transfer R objects
- Load and save can store one or more objects.

Ex. save(pbmc, file = “folder/pbmc.Rdata”)

load(“folder/name.Rdata”)

saveRDS and readRDS

Purpose: Means to save a single R object to a connection (typically a file) and to restore the object (serializes object)

- Serialization is the process of converting a data object into a series of bytes that saves the state of the object in an easily transmittable form.
- This differs from save and load, which save and restore one or more named objects into an environment
- Only saves one object. Saved in binary form.
- Does not retain metadata like object name, although infoRDS() can be used to retrieve some other metadata

Ex. saveRDS(object, file = "")

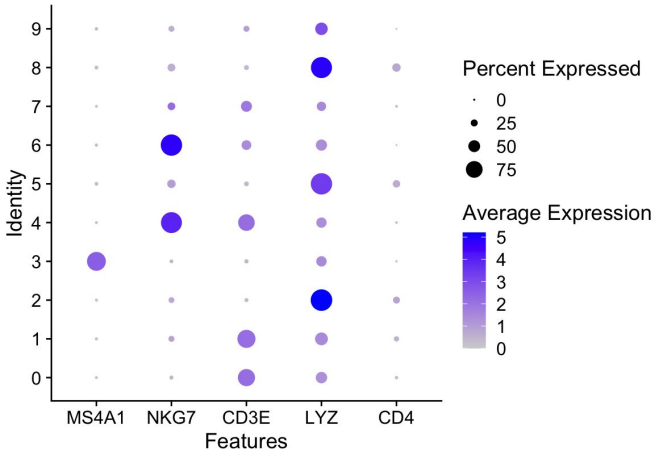
```
obj_name <- readRDS(file)
```

Break

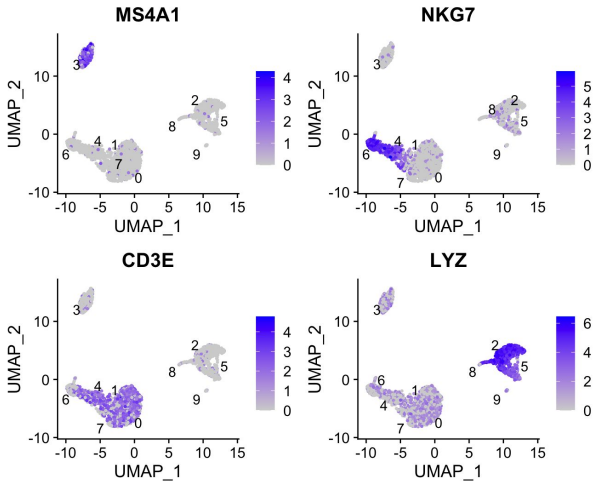
Challenge

Analyze the bone marrow single cell dataset to recreate the following plots:

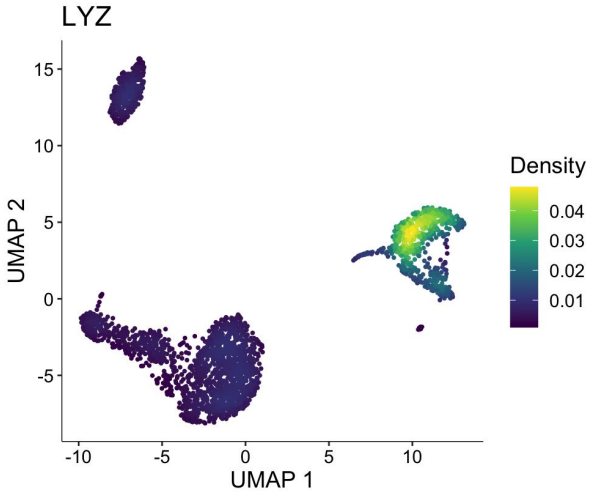
Dot plot showing cluster-wise expression of selected genes



Feature Plots showing expression of selected genes



Density Plot (Nebulosa Package) Showing Density of LYZ Expression



Single Cell Analysis Resources for Further Learning

- CrazyHotTommy GitHub and Blog: https://github.com/crazyhottommy/scRNAseq-analysis-notes
- Nature Reviews Genetics: Best practices for single-cell analysis across modalities: https://www.nature.com/articles/s41576-023-00586-w
- Statquest: Clearly explained + visualized statistical concepts relevant to single cell: https://www.youtube.com/channel/UCtYLUTtgS3k1Fg4y5tAhLbw
- Further understanding of UMAP with interactive animations: https://pair-code.github.io/understanding-umap/