Lab 8

Single-cell RNA seq

Maarten de Vries

slides adapted from Qing Zhang, Aviv Regev (Broad@15), Mary Piper (HBC)

Seurat walkthrough from https://satijalab.org/seurat/articles/pbmc3k_tutorial.html

Outline

- HW5 on scRNA-seq released, due April 11th
  - scRNA analysis with Seurat
- Today:
  - Brief review of scRNA-seq (20-30 mins)
  - Hands-on Seurat tutorial (20 mins)
  - Homework pointers
- Office hours
  - Maarten: Saturday 3/27 & 4/3 @ 10:30 am + Thursday @ 8 pm
  - Philip: Friday 3/26 & 4/2 @ 4:30 pm

Why scRNA-seq

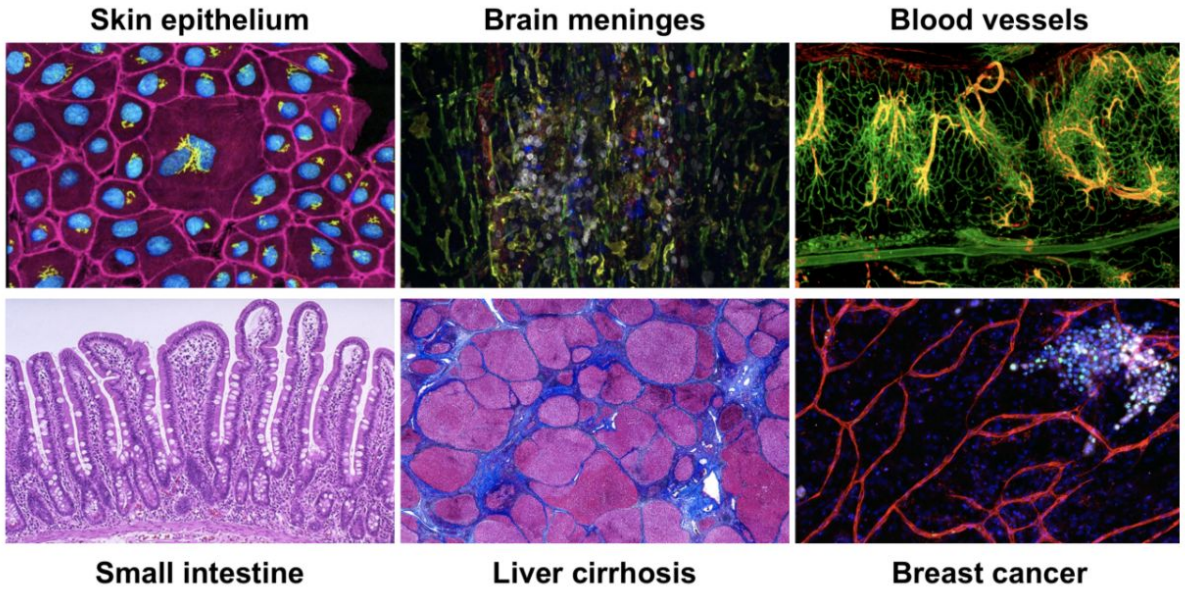
Bulk genomics

Single cell genomics

Spatial genomics

Aviv Regev, Broad@15 presentation

The body has many cell types



http://www.cell.com/pictureshow/skin | https://library.med.utah.edu/WebPath/webpath.html

Image credit: *courtesy of Dr. Ayshwarya Subramanian*

Capturing differences at the cell level

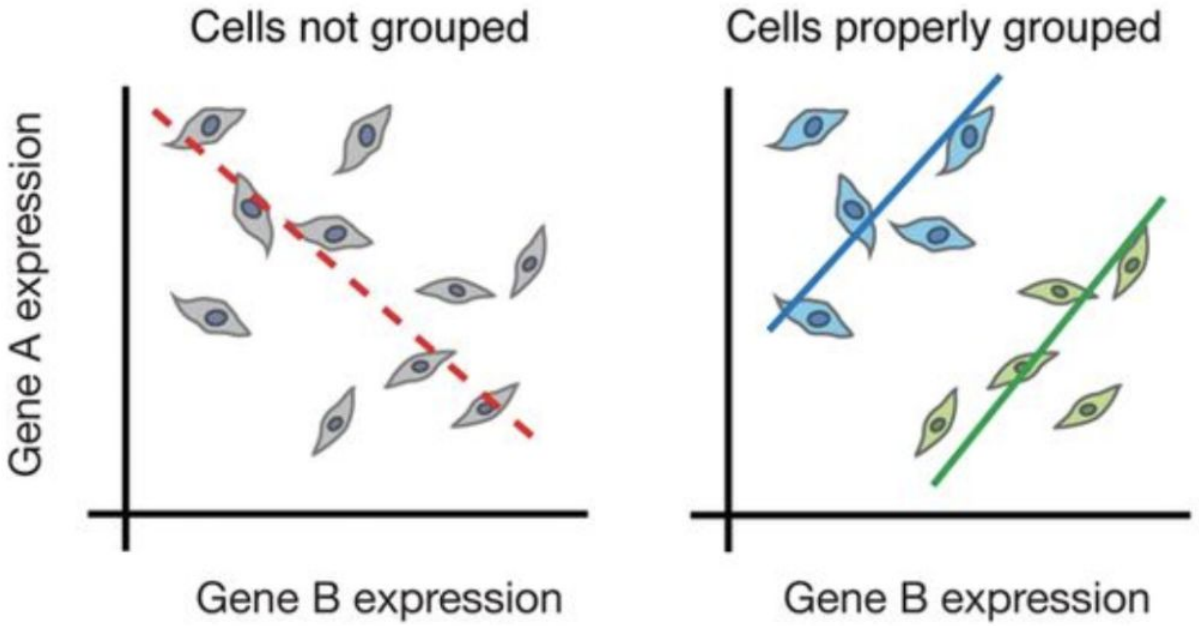
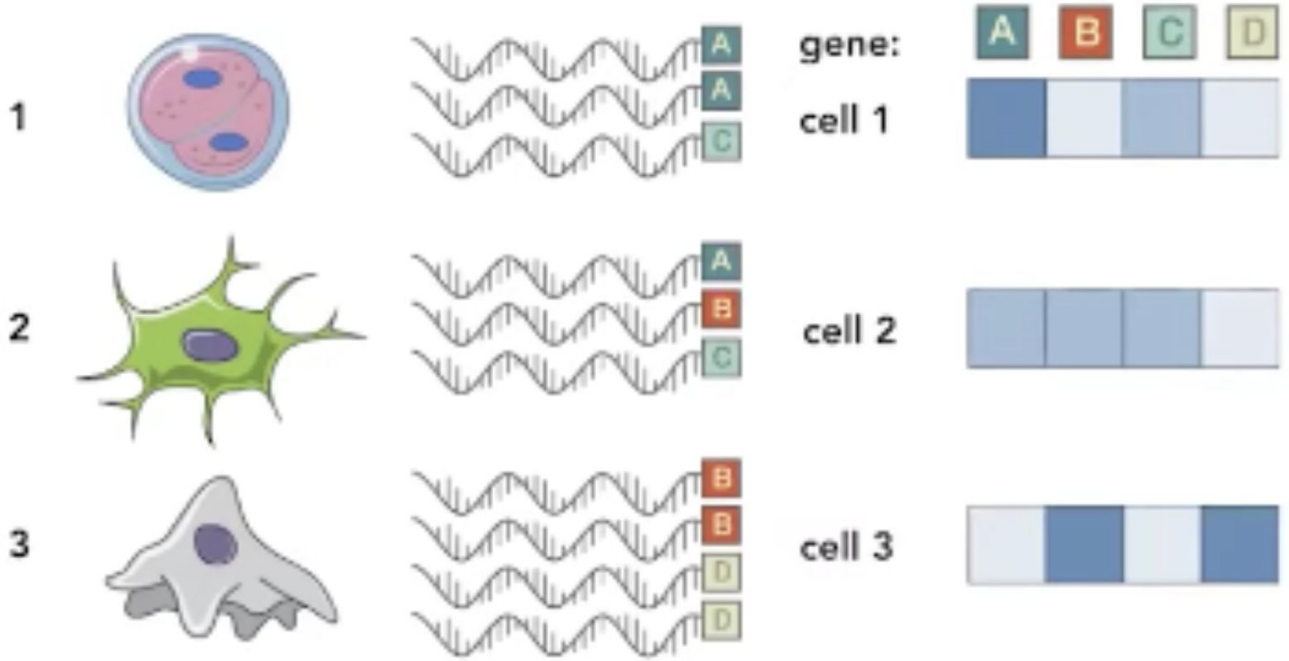
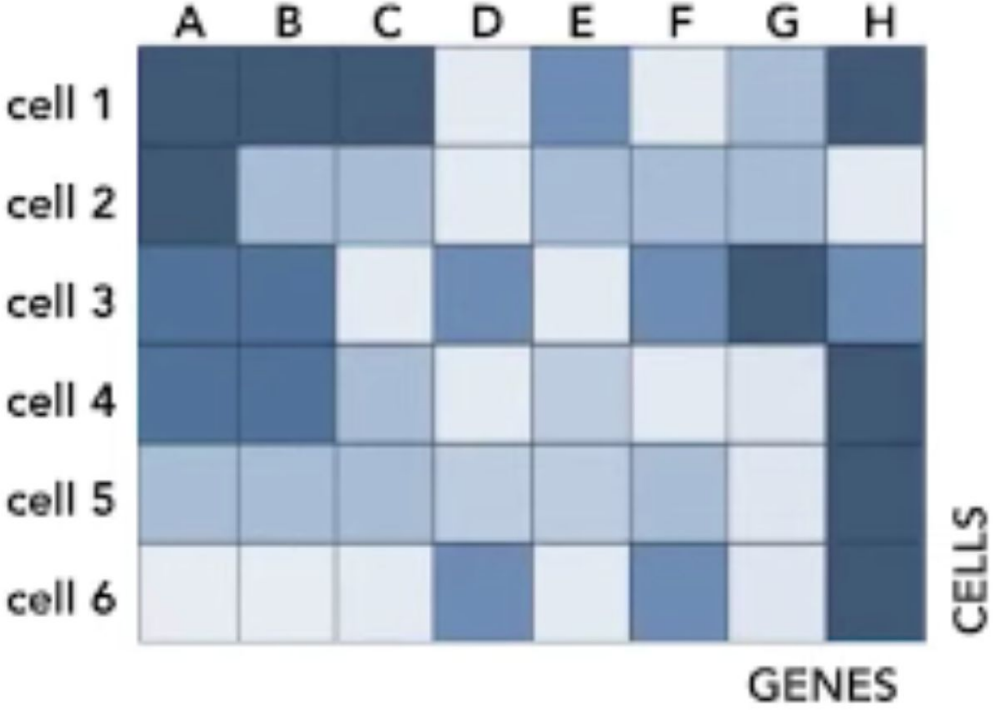


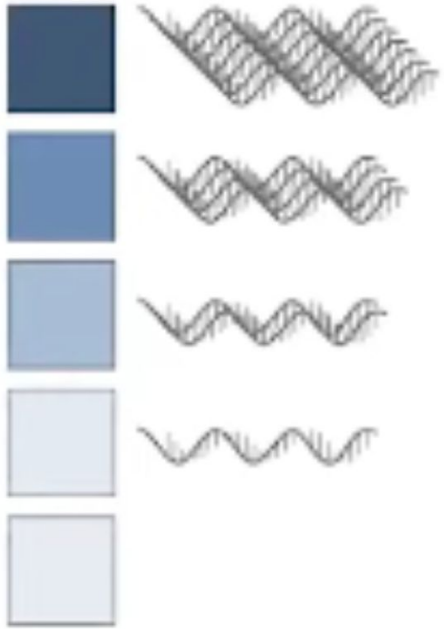
Image credit: *Trapnell, C. Defining cell types and states with single-cell genomics, Genome Research 2015 (doi: https://dx.doi.org/10.1101/gr.190595.115)*

Cells express the genes they need at the right quantities

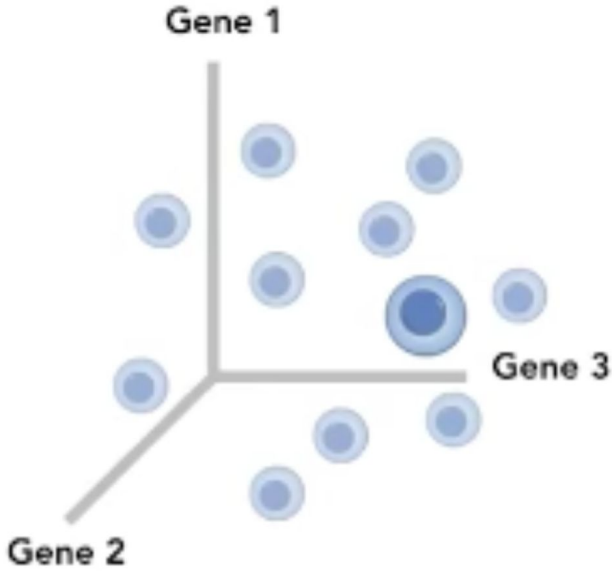


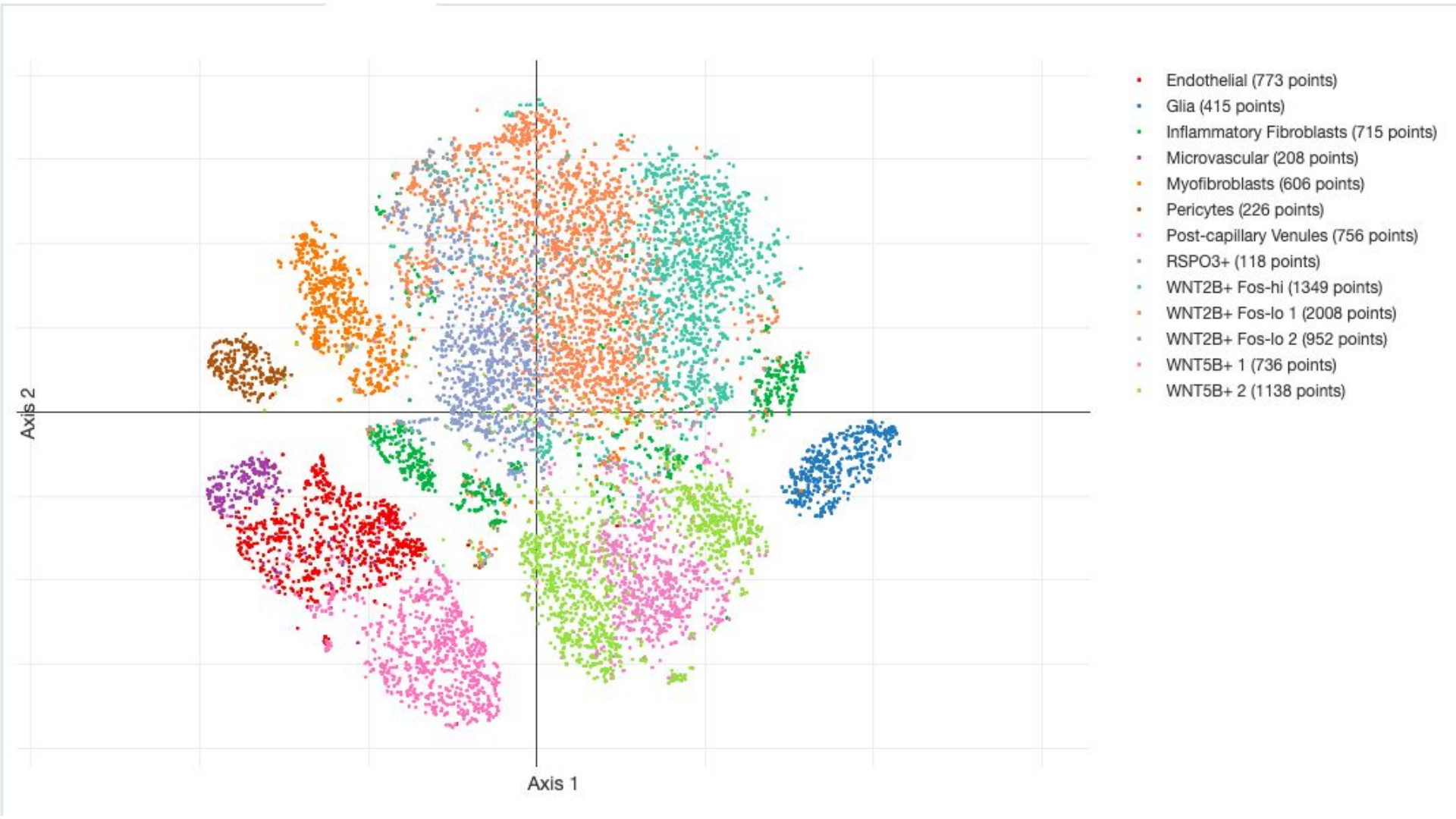
Gene expression profile is the calling card of a cell



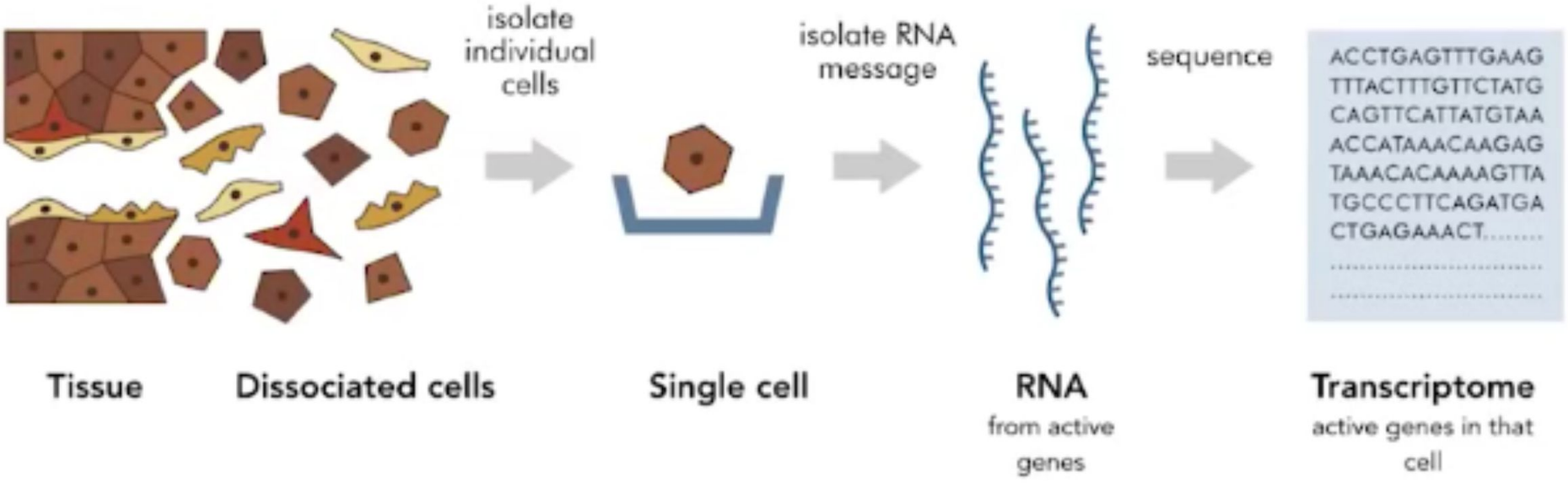


Cell as a point in 20,000+ dimensional gene expression space

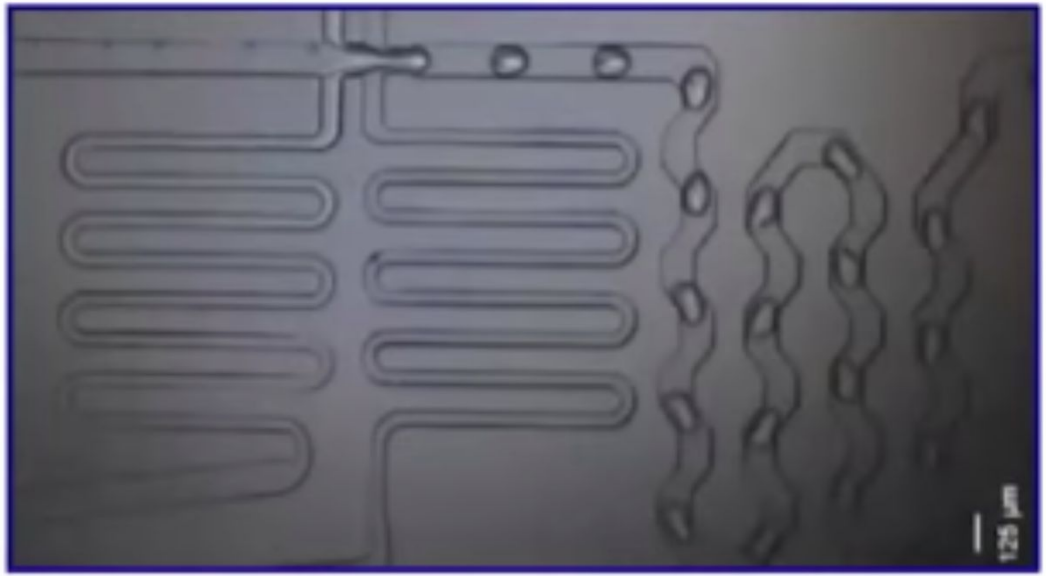



Endothelial (773 points)
Glia (415 points)
Inflammatory Fibroblasts (715 points)
Microvascular (208 points)
Myofibroblasts (606 points)
Pericytes (226 points)
Post-capillary Venules (756 points)
RSPO3+ (118 points)
WNT2B+ Fos-hi (1349 points)
WNT2B+ Fos-lo 1 (2008 points)
WNT2B+ Fos-lo 2 (952 points)
WNT5B+ 1 (736 points)
WNT5B+ 2 (1138 points)
Axis 2
Axis 1

Capturing a single cell

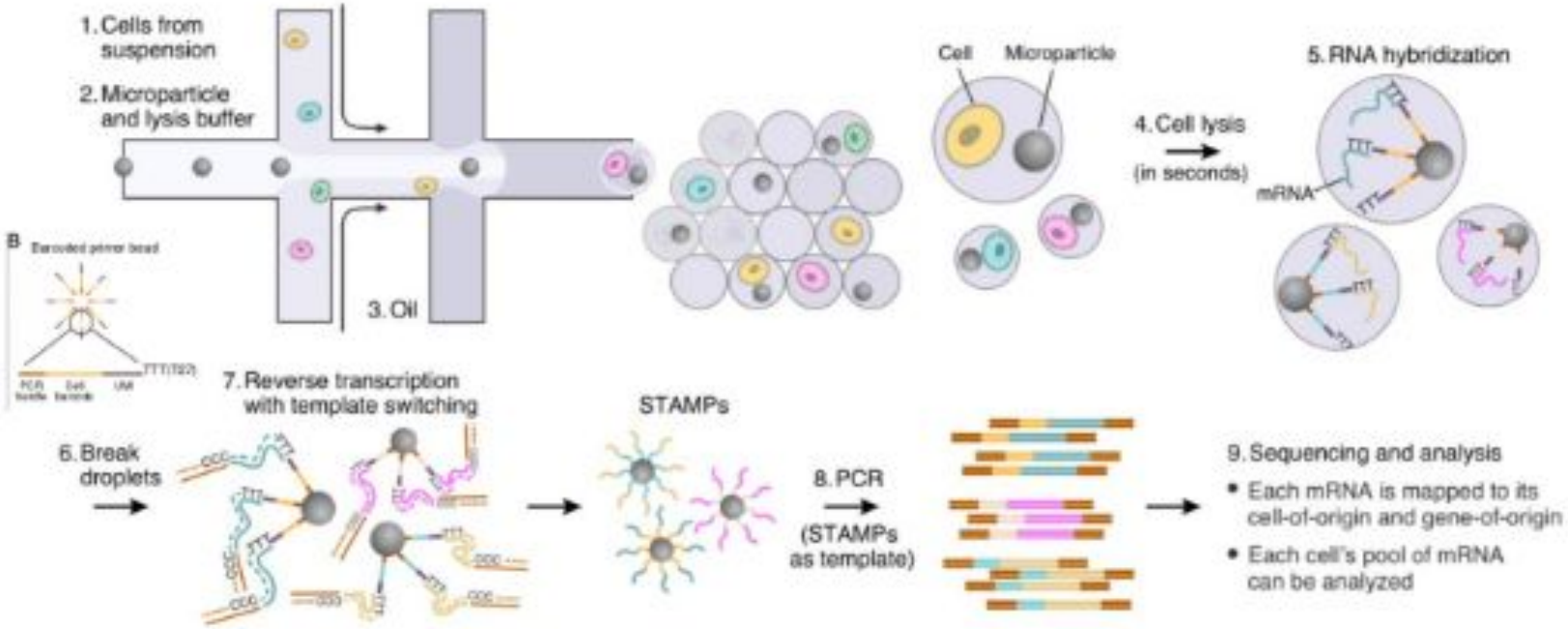


Droplet sequencing

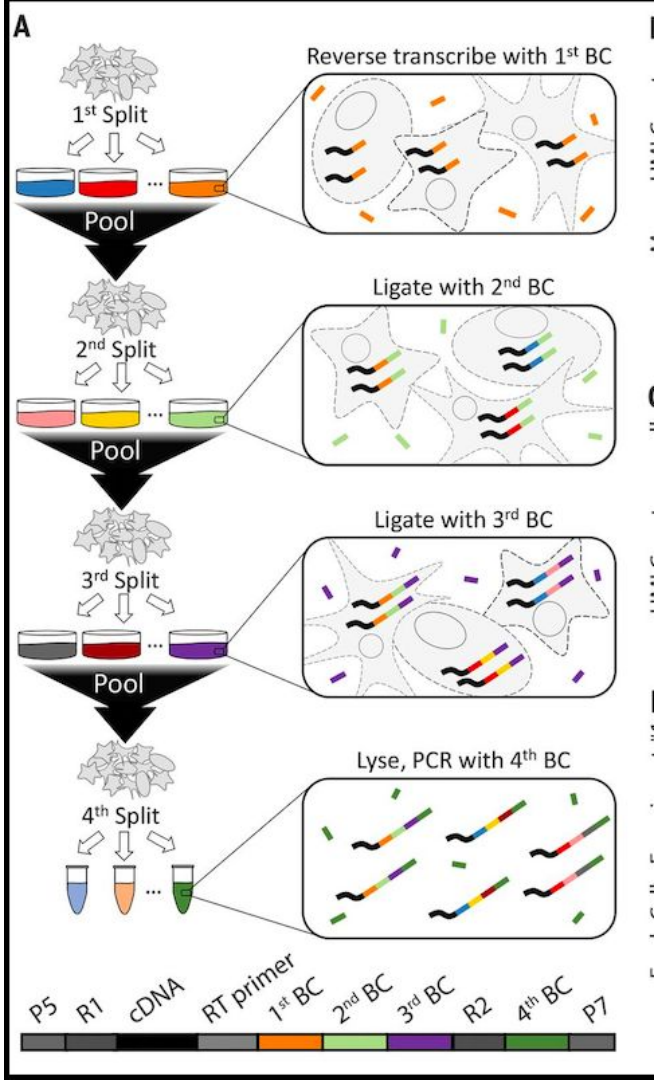


5,000 cells/second

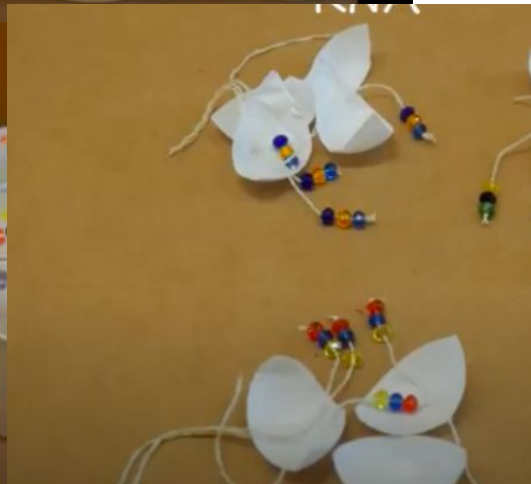
Droplet sequencing



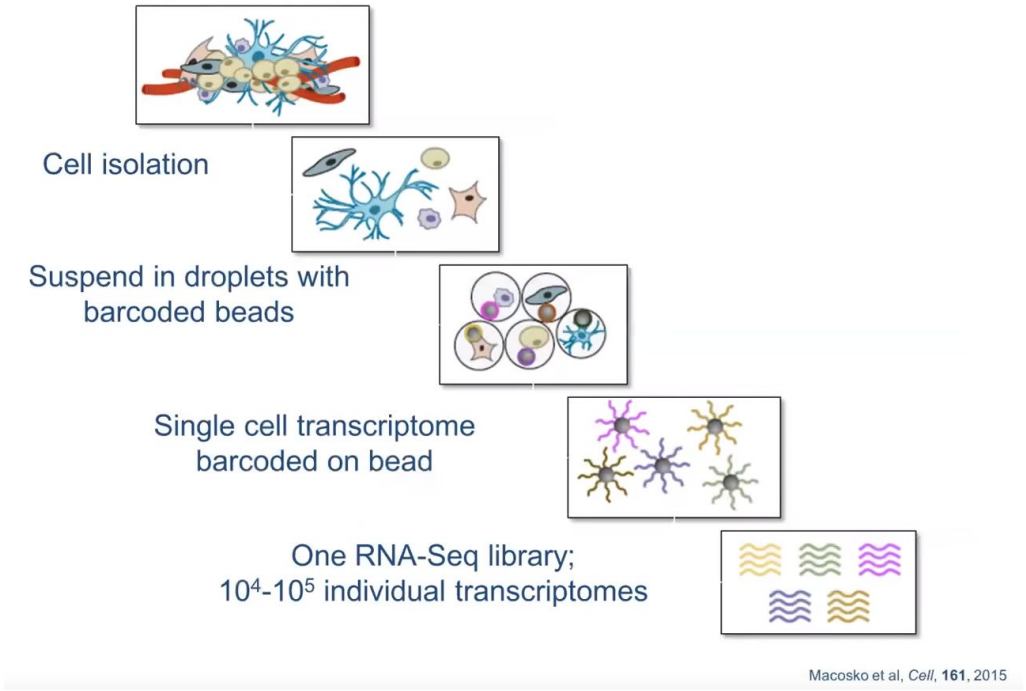
SPLIT-seq

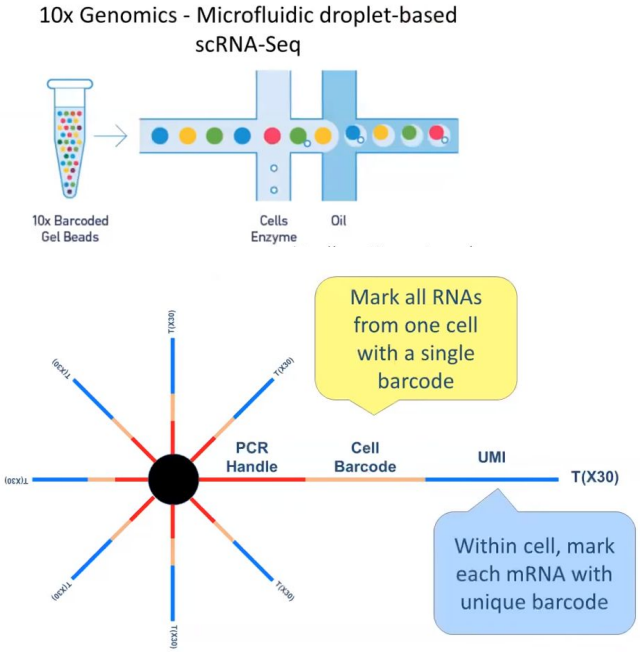

A
1st Split
Pool
Reverse transcribe with 1st BC
2nd Split
Pool
Ligate with 2nd BC
3rd Split
Pool
Ligate with 3rd BC
4th Split
Lyse, PCR with 4th BC
P5
R1
cDNA
RT primer
1st BC
2nd BC
3rd BC
R2
4th BC
P7

SPLIT-seq: the movie



Recap: Single cell RNA seq



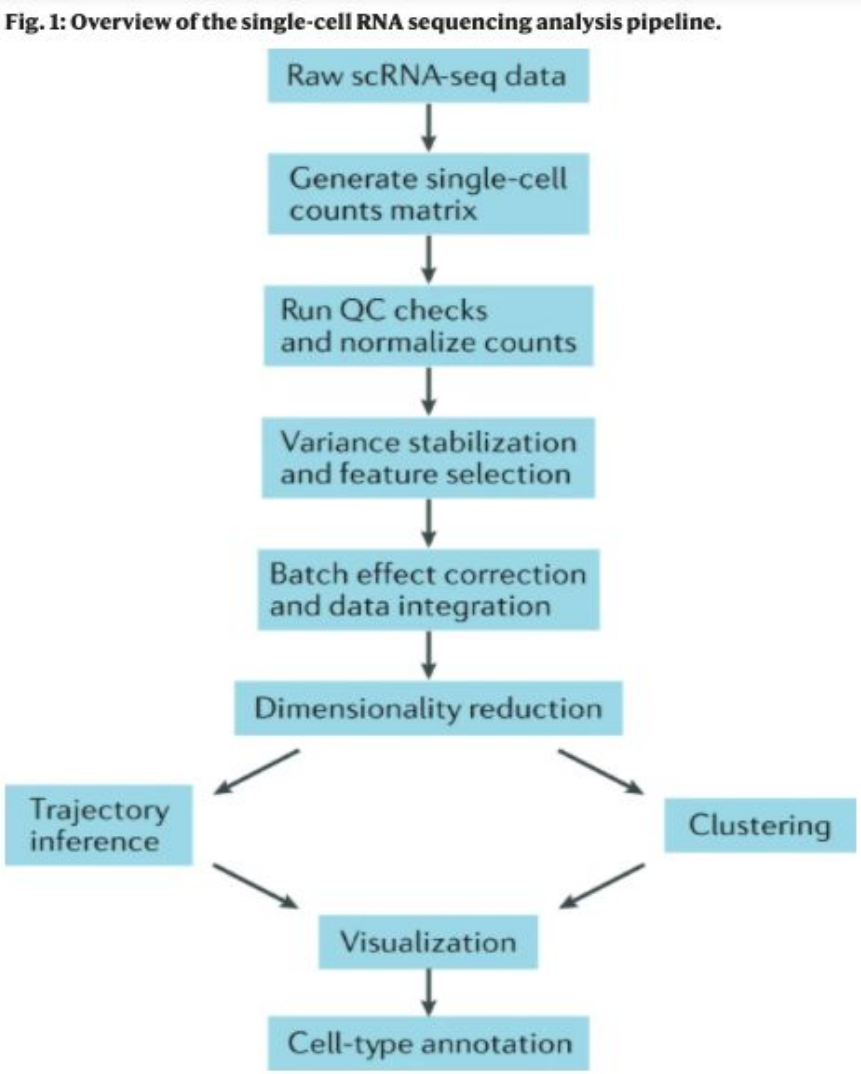


UMI: distinguish multiple copies of a transcript vs PCR artifacts

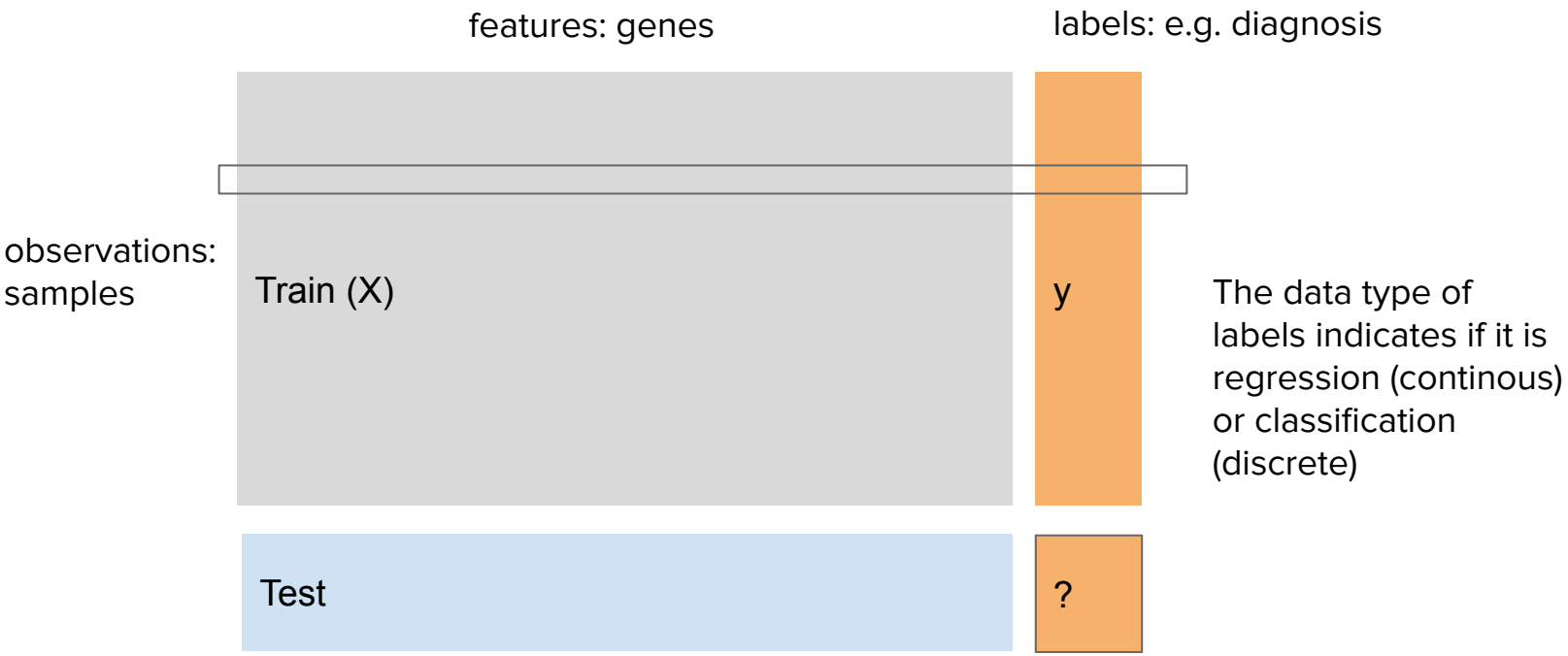
scRNA-seq analysis

Wu & Zhang

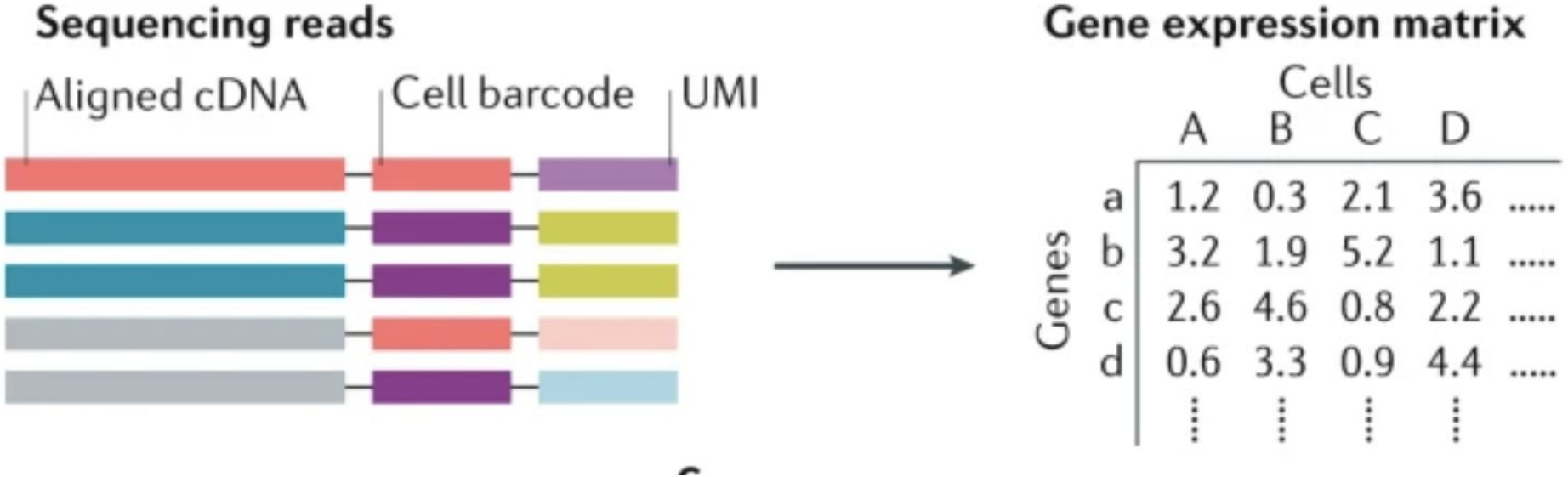
Nature Rev Neph 2020



Dataset

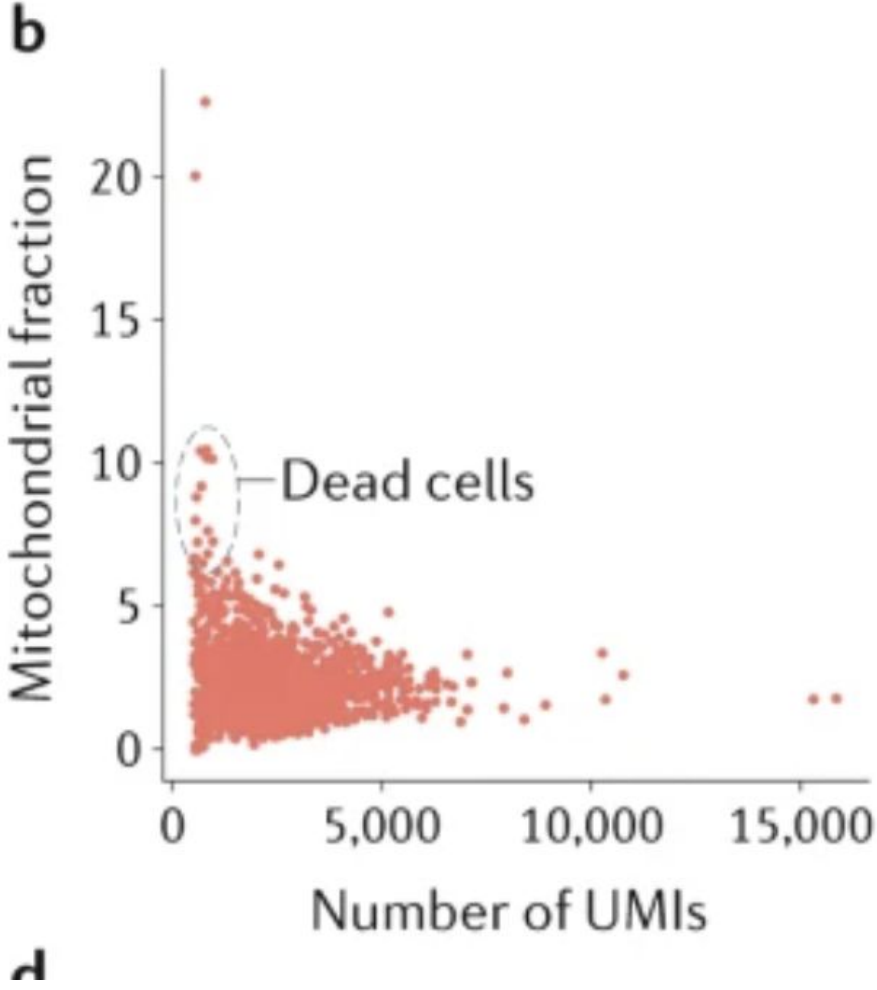

features: genes
labels: e.g. diagnosis
observations: samples
Train (X)
y
Test
?
The data type of labels indicates if it is regression (continous) or classification (discrete)

Generating the counts matrix

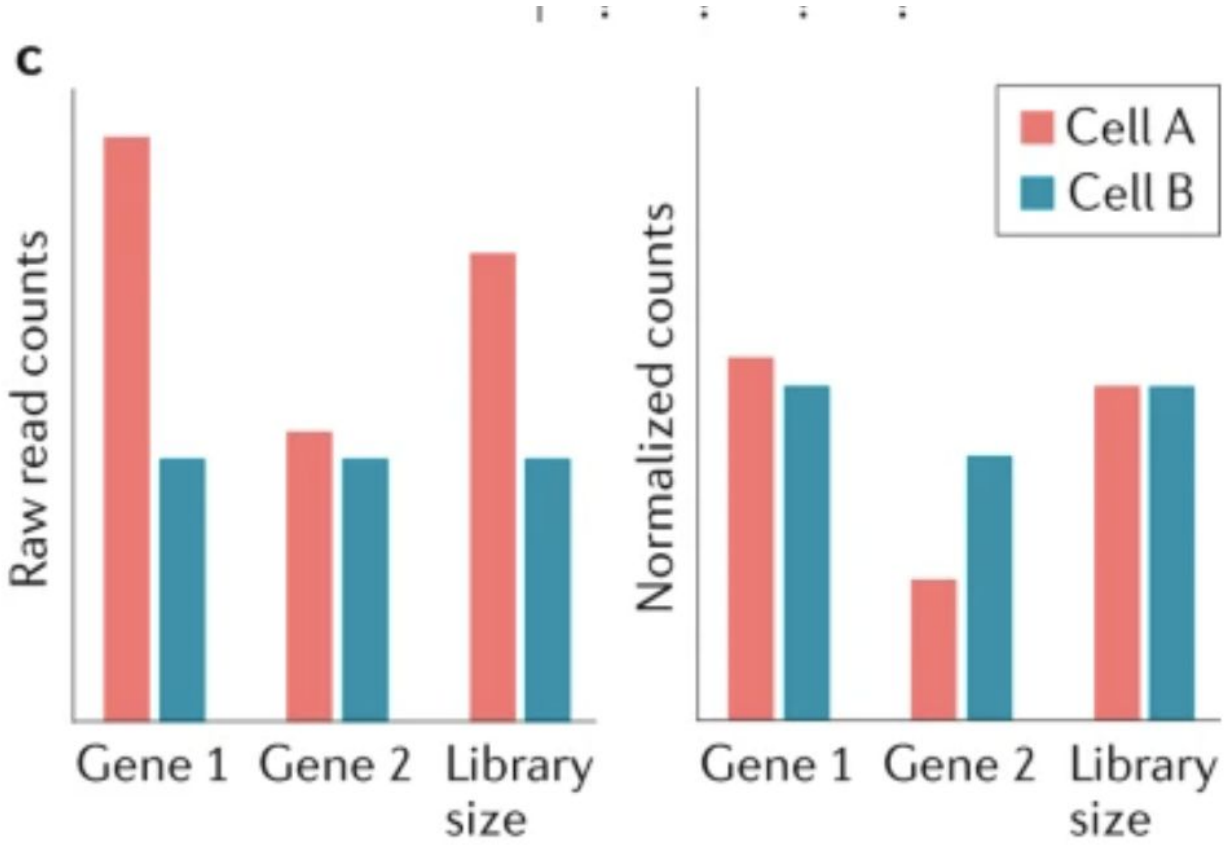


QC

- Detect cells with
  - Low read count
  - Low % of mapped reads
  - Too few/many genes
  - High mitochondrial fraction



Normalization



Batch effect

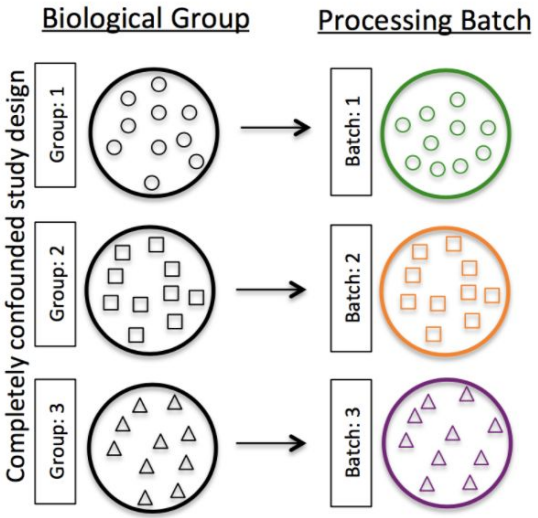
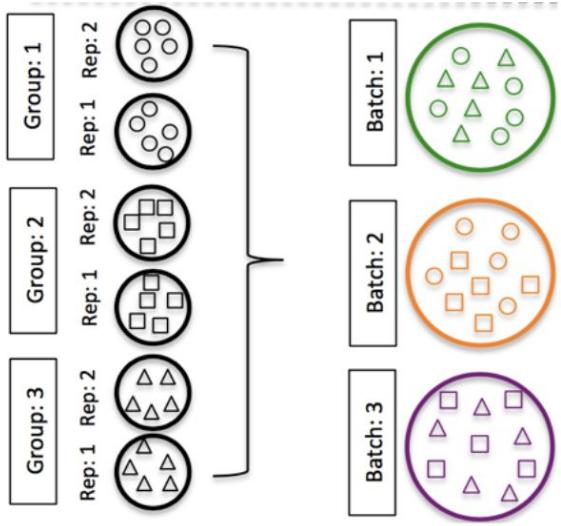


Image credit: *Hicks SC, et al., bioRxiv (2015)*



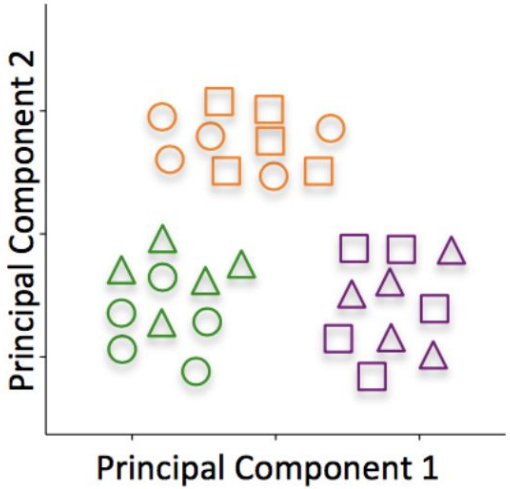


Image credit: *Hicks SC, et al., bioRxiv (2015)*

Dimensionality reduction

- Dimensionality reduction
    - Curse of dimensionality
    - Linear: Principal Component Analysis (PCA)
        - Finds linear combinations of genes that best capture the variance in data
        - Zero-inflated factor analysis (ZIFA) is a version of PCA designed to explicitly model the high expected count of zero values in scRNA-seq data.
    - Nonlinear: t-SNE, UMAP, Deep neural networks
- Imputation of zero-values
    - Technical limitations in RNA capture lead to zero-inflation
    - In part biological variance, in part noise
    - Imputation methods are available
        - MAGIC uses information from neighboring cells
        - Lots of deep learning methods recently developed

Clustering & Visualization

Fig. 4: Cell clustering in datasets with discrete cell types.

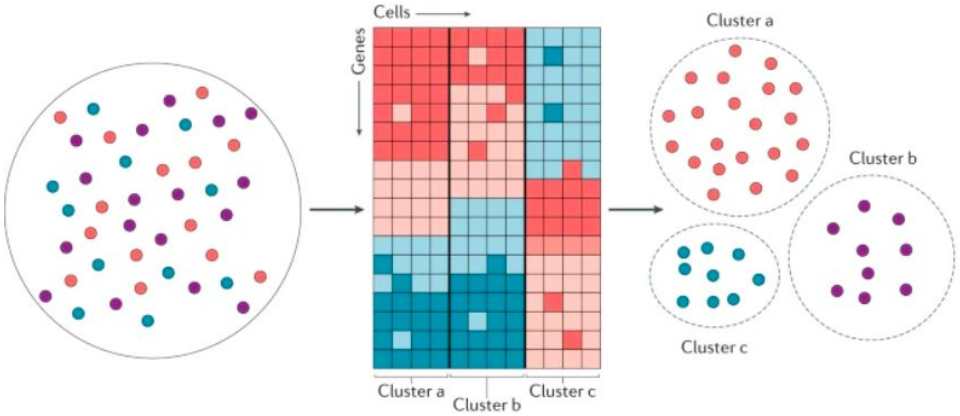
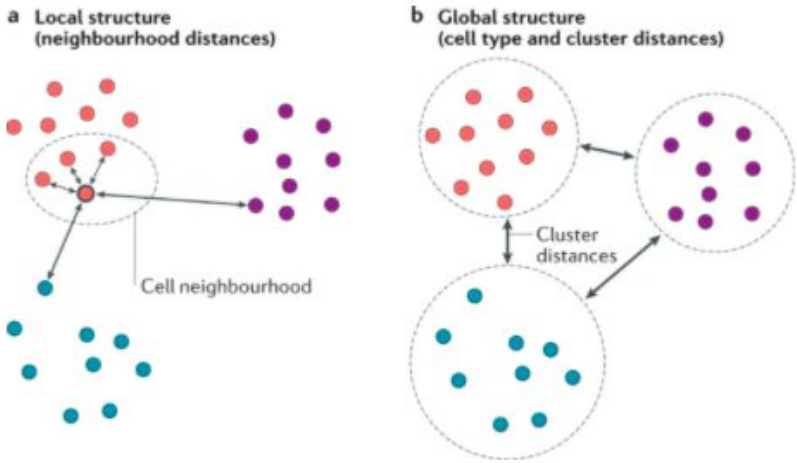


Fig. 6: Local and global structure in a dataset.



Cell-type annotation

- Time-consuming if done manually
- Find genes that are uniquely expressed in each cluster
- Match those genes to lists of canonical cell-type markers
- To accelerate this, use functional pathways and gene ontology terms
- New (semi-)automated cell-type classification methods are being developed.
- Novel cell types and states still need to be manually annotated, or do they?

Workflows

Conveniently, there exist toolkits that enable all of the aforementioned steps within a single workflow.

- Seurat (in R)
- SCANPY (in Python)
- MAESTRO (Philip’s Lab next week)

Seurat [v4.0]

https://satijalab.org/seurat/

R package for single cell analysis

Great vignettes!

Make sure you have Seurat 4.0 and
R version >4.0 installed

Setup Seurat object

```
library(dplyr)
library(Seurat)
library(patchwork)

# Load the PBMC dataset
pbmc.data <- Read10X(data.dir = "../data/pbmc3k/filtered_gene_bc_matrices/hg19/")
# Initialize the Seurat object with the raw (non-normalized data).
pbmc <- CreateSeuratObject(counts = pbmc.data, project = "pbmc3k", min.cells = 3, min.features = 200
pbmc
```

```
## An object of class Seurat
## 13714 features across 2700 samples within 1 assay
## Active assay: RNA (13714 features, 0 variable features)
```

Sparse matrix

```
# Lets examine a few genes in the first thirty cells
pbmc.data[c("CD3D", "TCL1A", "MS4A1"), 1:30]
```

```
## 3 x 30 sparse Matrix of class "dgCMatrix"
##
## CD3D  4 . 10 . . 1 2 3 1 . . 2 7 1 . . 1 3 . 2  3 . . . . . 3 4 1 5
## TCL1A . .  . . . . . . 1 . . . . . . . . . . .  . 1 . . . . . . . .
## MS4A1 . 6  . . . . . . 1 1 1 . . . . . . . . . 36 1 2 . . 2 . . . .
```

QC

- The number of unique genes detected in each cell.
  - Low-quality cells or empty droplets will often have very few genes
  - Cell doublets or multiplets may exhibit an aberrantly high gene count
- Similarly, the total number of molecules detected within a cell (correlates strongly with unique genes)
- The percentage of reads that map to the mitochondrial genome
  - Low-quality / dying cells often exhibit extensive mitochondrial contamination
  - We calculate mitochondrial QC metrics with the `PercentageFeatureSet()` function, which calculates the percentage of counts originating from a set of features
  - We use the set of all genes starting with `MT-` as a set of mitochondrial genes

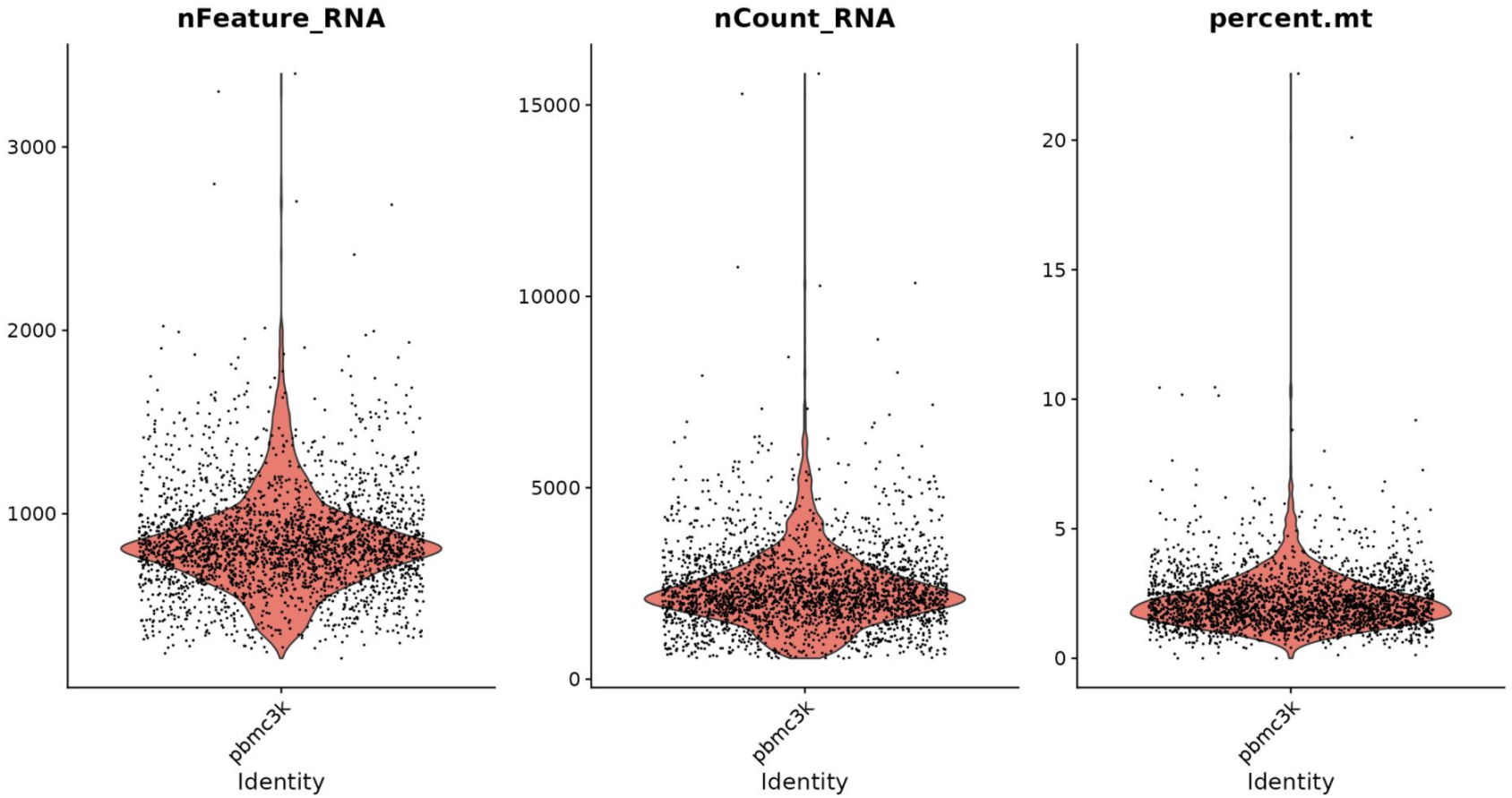
```
# The [[ operator can add columns to object metadata. This is a great place to stash QC stats
pbmc[["percent.mt"]] <- PercentageFeatureSet(pbmc, pattern = "^MT-")
```

QC metrics are stored in meta.data

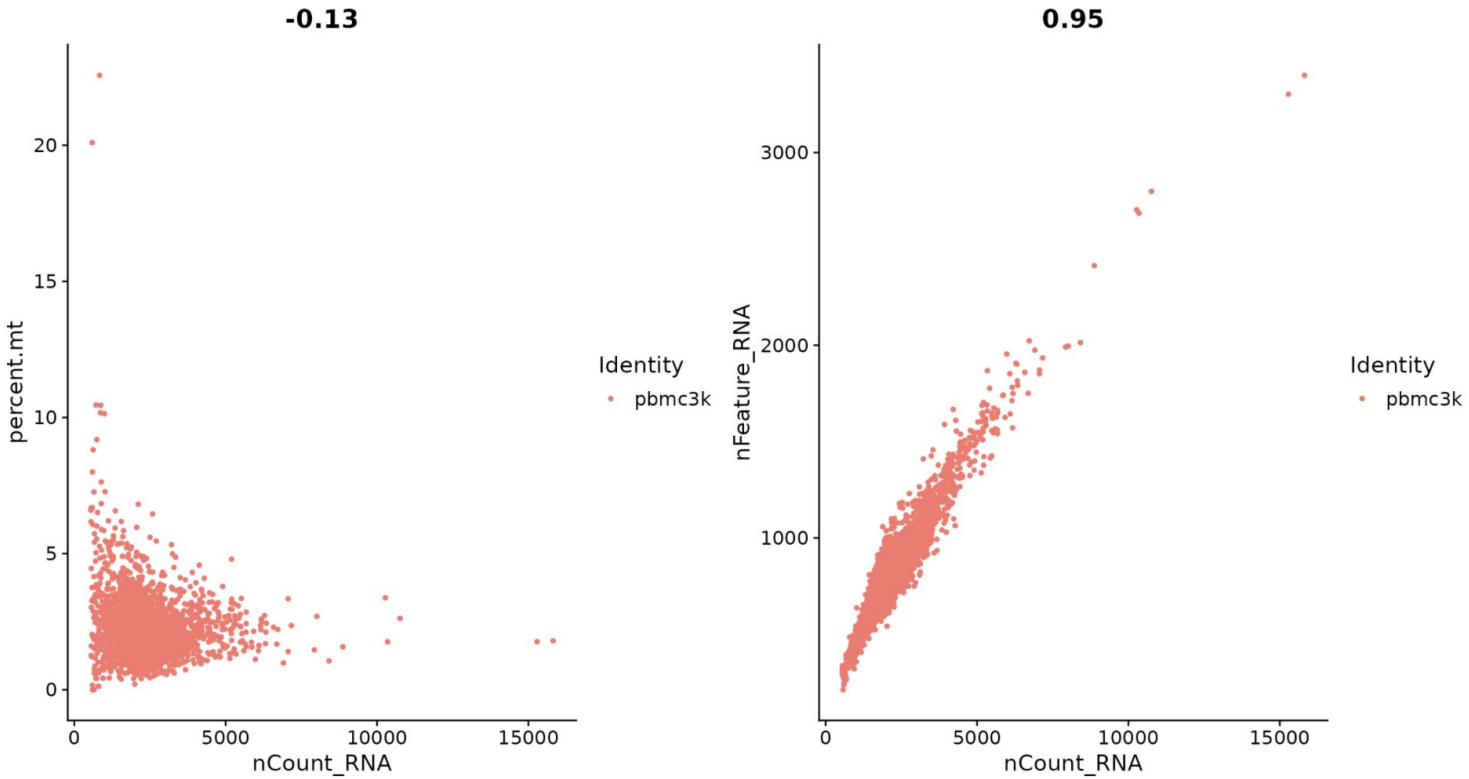
```
# Show QC metrics for the first 5 cells
head(pbmc@meta.data, 5)
```

```
##                 orig.ident nCount_RNA nFeature_RNA percent.mt
## AAACATACAACCAC-1      pbmc3k       2419          779  3.0177759
## AAACATTGAGCTAC-1      pbmc3k       4903         1352  3.7935958
## AAACATTGATCAGC-1      pbmc3k       3147         1129  0.8897363
## AAACCGTGCTTCCG-1      pbmc3k       2639          960  1.7430845
## AAACCGTGTATGCG-1      pbmc3k        980          521  1.2244898
```

```r
# Visualize QC metrics as a violin plot
VlnPlot(pbmc, features = c("nFeature_RNA", "nCount_RNA", "percent.mt"), ncol = 3)
```



```
# FeatureScatter is typically used to visualize feature-feature relationships, but can be used
# for anything calculated by the object, i.e. columns in object metadata, PC scores etc.

plot1 <- FeatureScatter(pbmc, feature1 = "nCount_RNA", feature2 = "percent.mt")
plot2 <- FeatureScatter(pbmc, feature1 = "nCount_RNA", feature2 = "nFeature_RNA")
plot1 + plot2
```



Filtering out cells

- We filter cells that have unique feature counts over 2,500 or less than 200
- We filter cells that have >5% mitochondrial counts

```
pbmc <- subset(pbmc, subset = nFeature_RNA > 200 & nFeature_RNA < 2500 & percent.mt < 5)
```

Normalizing the data

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

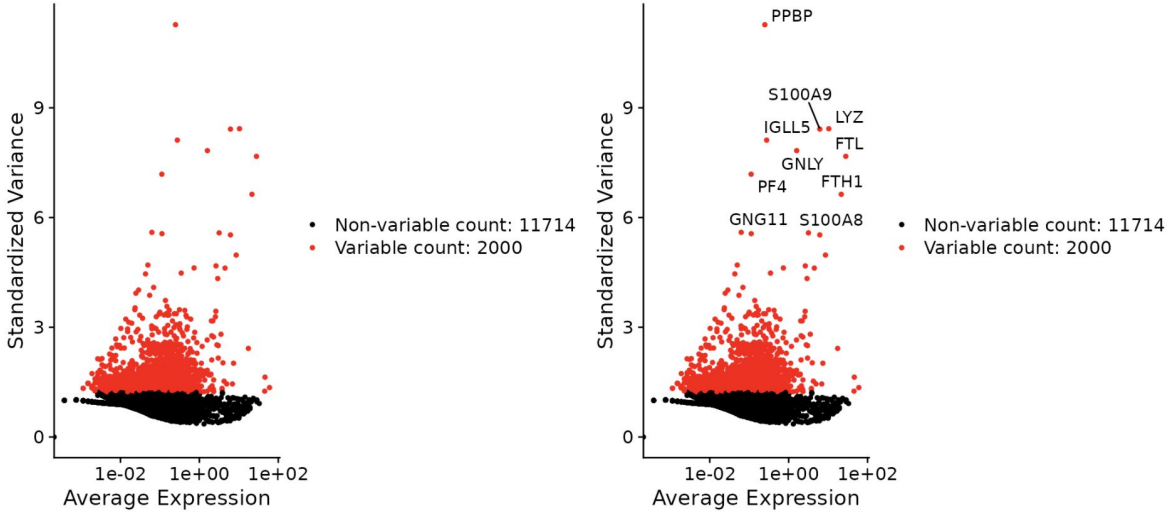
```
pbmc <- NormalizeData(pbmc)
```

Default:

- normalize feature expression measurements for each cell by total expression
- multiply by a scale factor (10,000 by default)
- log-transform the result
- `pbmc[["RNA"]]@data`

Feature selection

```r
pbmc <- FindVariableFeatures(pbmc, selection.method = "vst", nfeatures = 2000)

# Identify the 10 most highly variable genes
top10 <- head(VariableFeatures(pbmc), 10)

# plot variable features with and without labels
plot1 <- VariableFeaturePlot(pbmc)
plot2 <- LabelPoints(plot = plot1, points = top10, repel = TRUE)
plot1 + plot2
```



Scaling the data

- Shifts the expression of each gene, so that the mean expression across cells is 0
- Scales the expression of each gene, so that the variance across cells is 1
    - This step gives equal weight in downstream analyses, so that highly-expressed genes do not dominate
- The results of this are stored in `pbmc[["RNA"]]@scale.data`

```
all.genes <- rownames(pbmc)
pbmc <- ScaleData(pbmc, features = all.genes)
```

Linear dimensional reduction

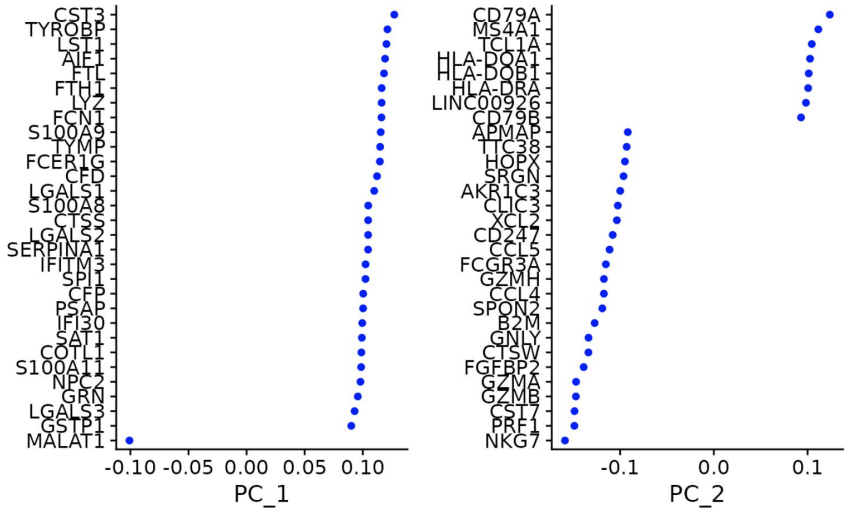
```
pbmc <- RunPCA(pbmc, features = VariableFeatures(object = pbmc))
```

```
# Examine and visualize PCA results a few different ways
print(pbmc[["pca"]], dims = 1:5, nfeatures = 5)
```

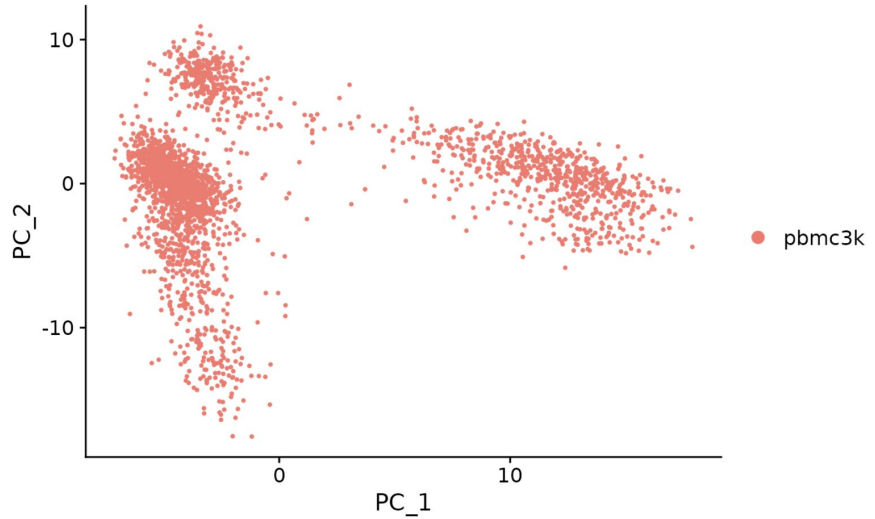
```
## PC_ 1
## Positive:  CST3, TYROBP, LST1, AIF1, FTL
## Negative:  MALAT1, LTB, IL32, IL7R, CD2
## PC_ 2
## Positive:  CD79A, MS4A1, TCL1A, HLA-DQA1, HLA-DQB1
## Negative:  NKG7, PRF1, CST7, GZMB, GZMA
## PC_ 3
## Positive:  HLA-DQA1, CD79A, CD79B, HLA-DQB1, HLA-DPB1
## Negative:  PPBP, PF4, SDPR, SPARC, GNG11
## PC_ 4
## Positive:  HLA-DQA1, CD79B, CD79A, MS4A1, HLA-DQB1
## Negative:  VIM, IL7R, S100A6, IL32, S100A8
## PC_ 5
```

More PCA visualizaitons

```
VizDimLoadings(pbmc, dims = 1:2, reduction = "pca")
```

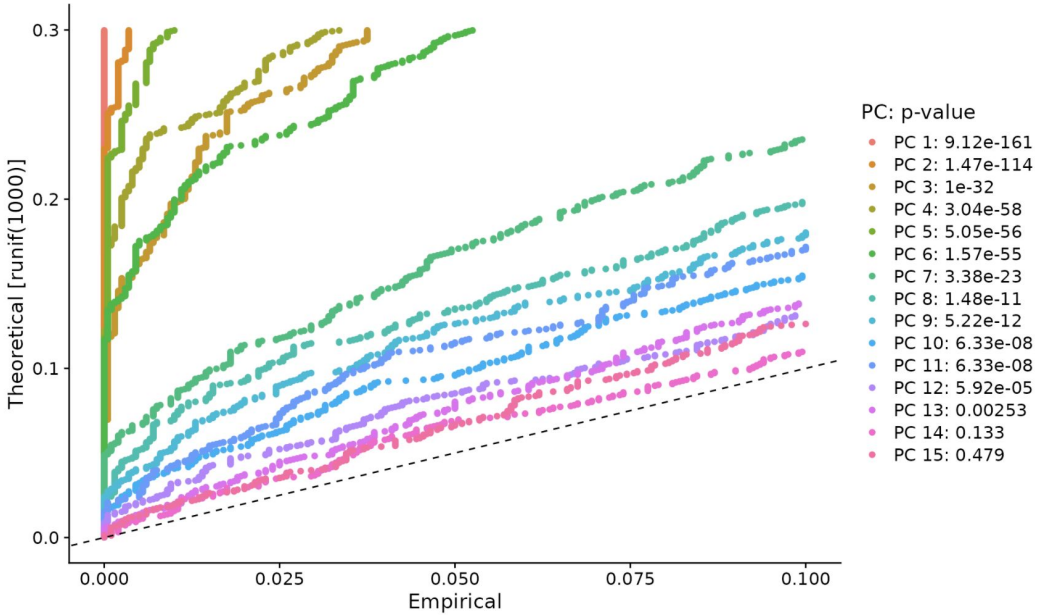



```
DimPlot(pbmc, reduction = "pca")
```



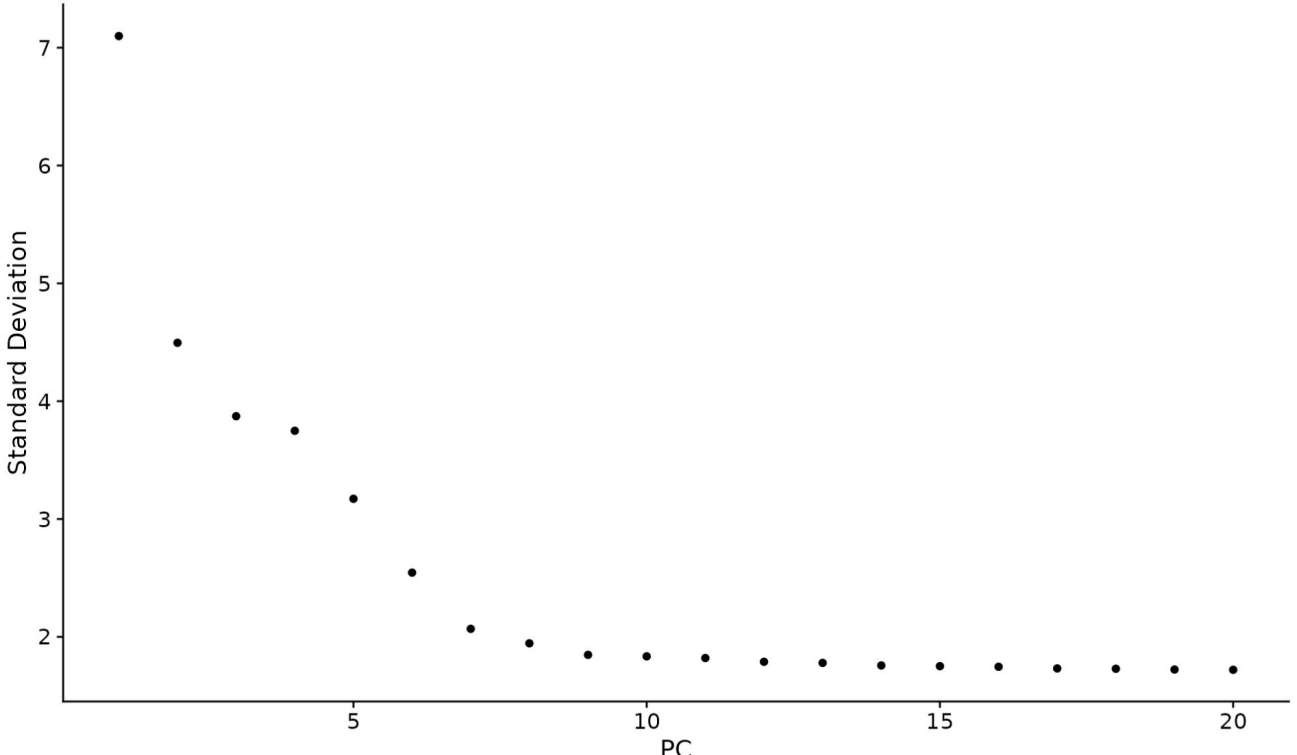
```
# NOTE: This process can take a long time for big datasets, comment out for expediency. More
# approximate techniques such as those implemented in ElbowPlot() can be used to reduce
# computation time
pbmc <- JackStraw(pbmc, num.replicate = 100)
pbmc <- ScoreJackStraw(pbmc, dims = 1:20)
```

```
JackStrawPlot(pbmc, dims = 1:15)
```



Elbow Plot

```
ElbowPlot(pbmc)
```



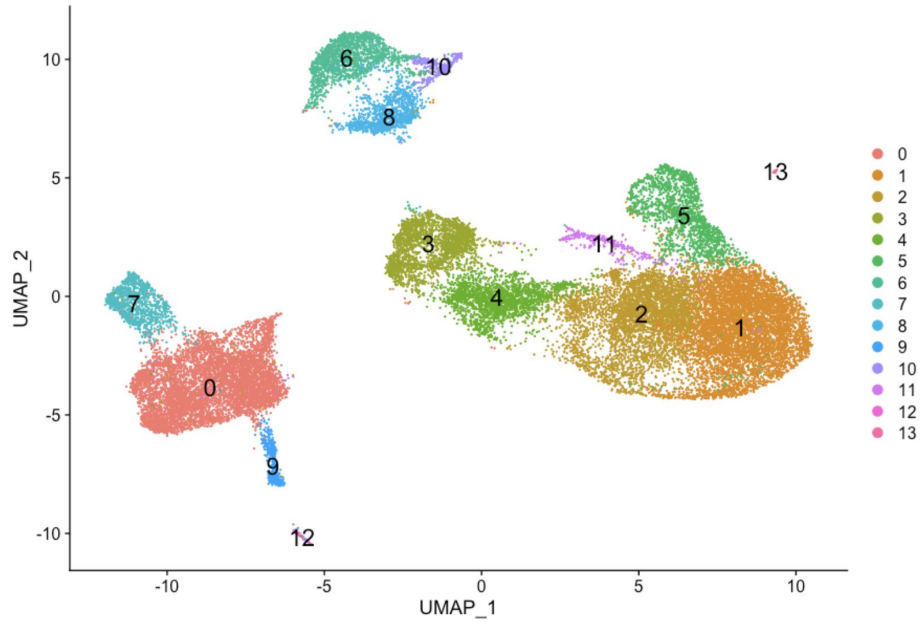
Clustering

```
pbmc <- FindNeighbors(pbmc, dims = 1:10)
pbmc <- FindClusters(pbmc, resolution = 0.5)
```

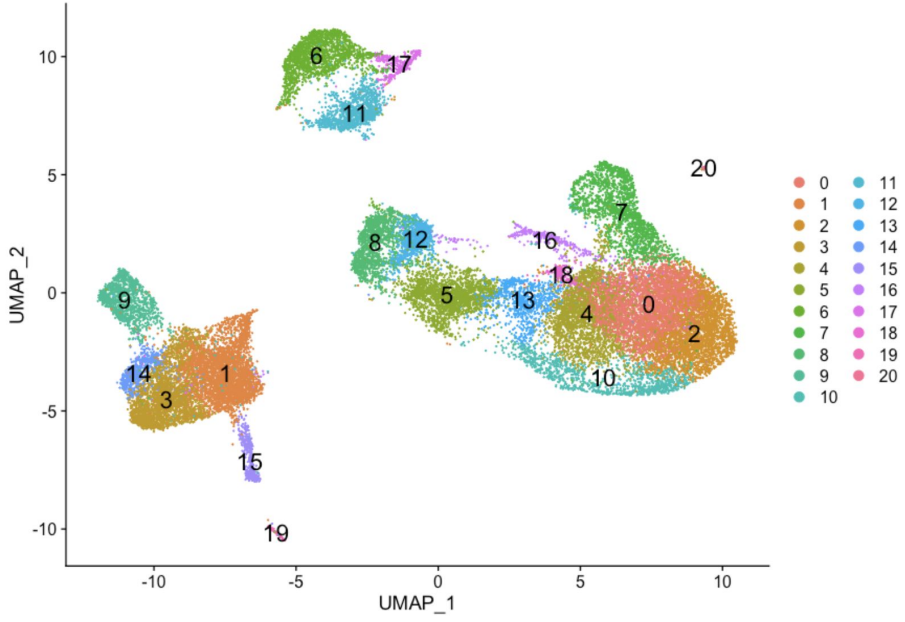
```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 2638
## Number of edges: 95965
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.8723
## Number of communities: 9
## Elapsed time: 0 seconds
```

Resolution

```r
# Assign identity of clusters
Idents(object = seurat_integrated) <- "integrated_snn_res.0.4"

# Plot the UMAP
DimPlot(seurat_integrated,
        reduction = "umap",
        label = TRUE,
        label.size = 6)
```



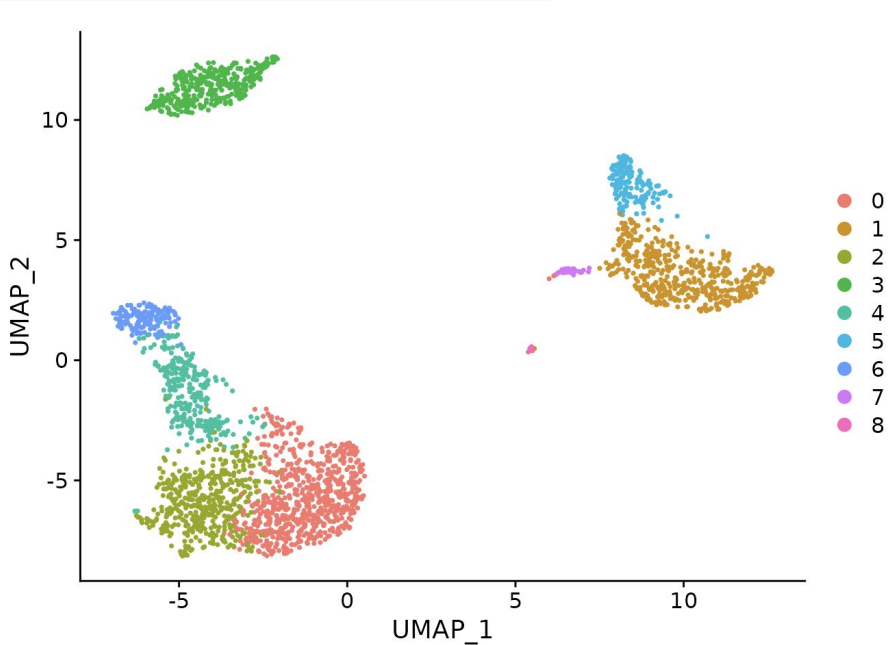

```r
# Assign identity of clusters
Idents(object = seurat_integrated) <- "integrated_snn_res.0.8"

# Plot the UMAP
DimPlot(seurat_integrated,
        reduction = "umap",
        label = TRUE,
        label.size = 6)
```



Non-linear dimension reduction

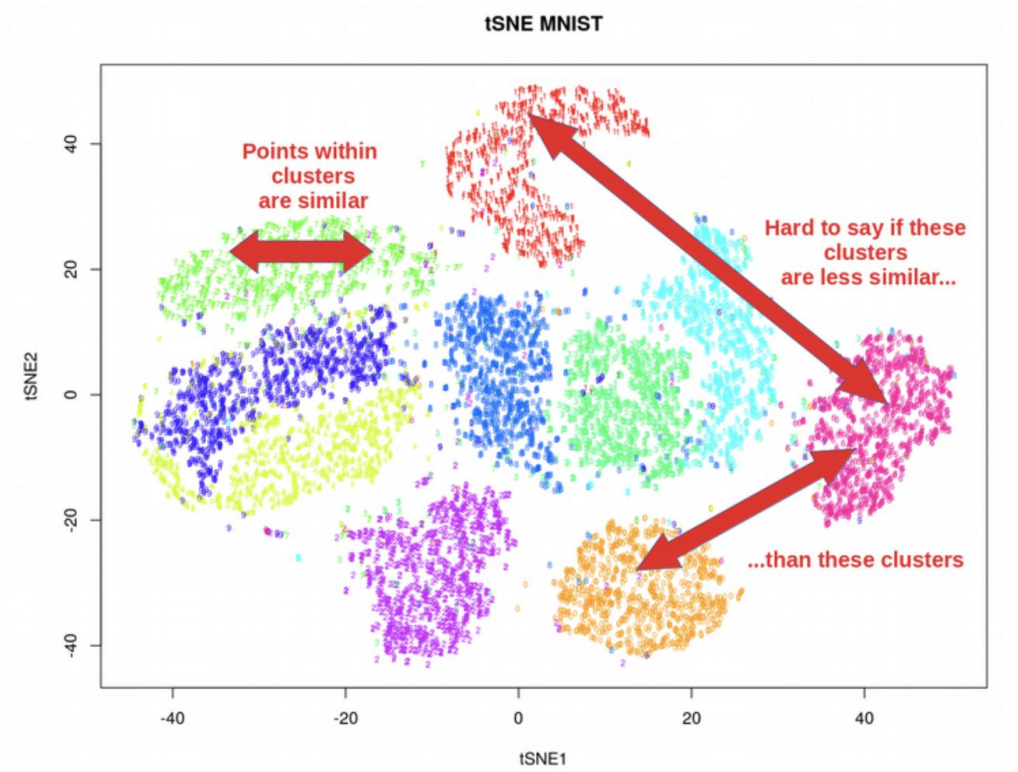
```
# If you haven't installed UMAP, you can do so via reticulate::py_install(packages =
# 'umap-learn')
pbmc <- RunUMAP(pbmc, dims = 1:10)
```

```
# note that you can set `label = TRUE` or use the LabelClusters function to help label
# individual clusters
DimPlot(pbmc, reduction = "umap")
```



[Useful blog post]

t-SNE



Nikolay Oskolkov

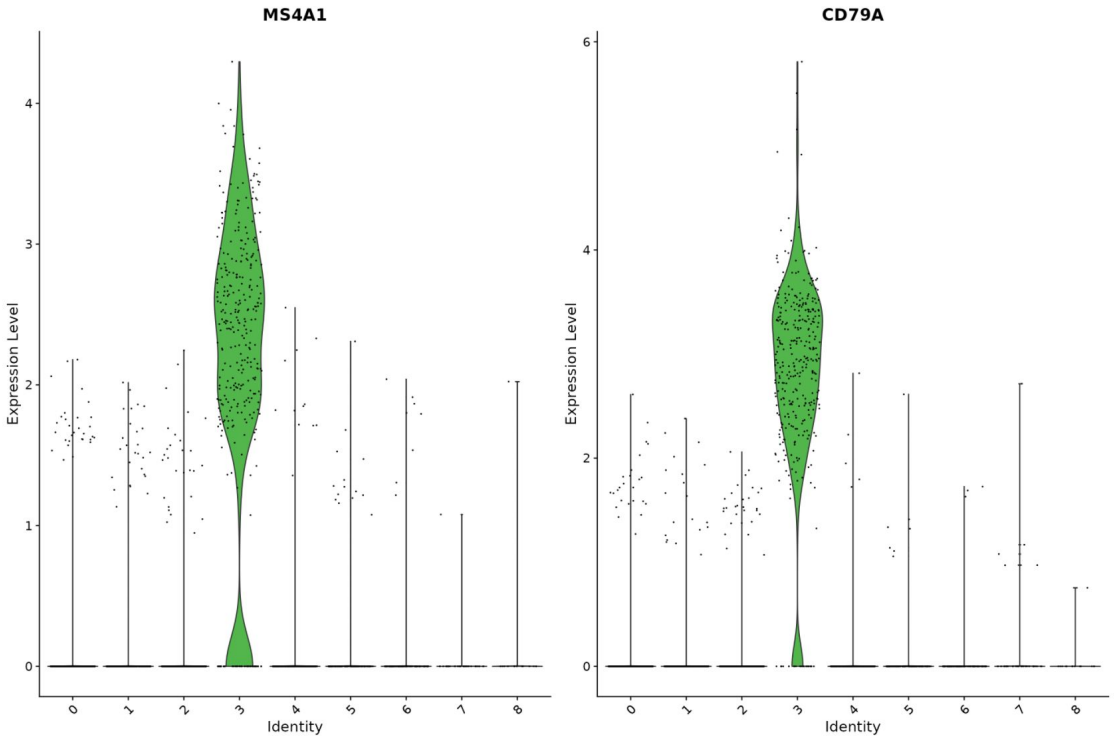
Finding differentially expressed features

```
# find markers for every cluster compared to all remaining cells, report only the positive ones
pbmc.markers <- FindAllMarkers(pbmc, only.pos = TRUE, min.pct = 0.25, logfc.threshold = 0.25)
pbmc.markers %>% group_by(cluster) %>% top_n(n = 2, wt = avg_log2FC)
```

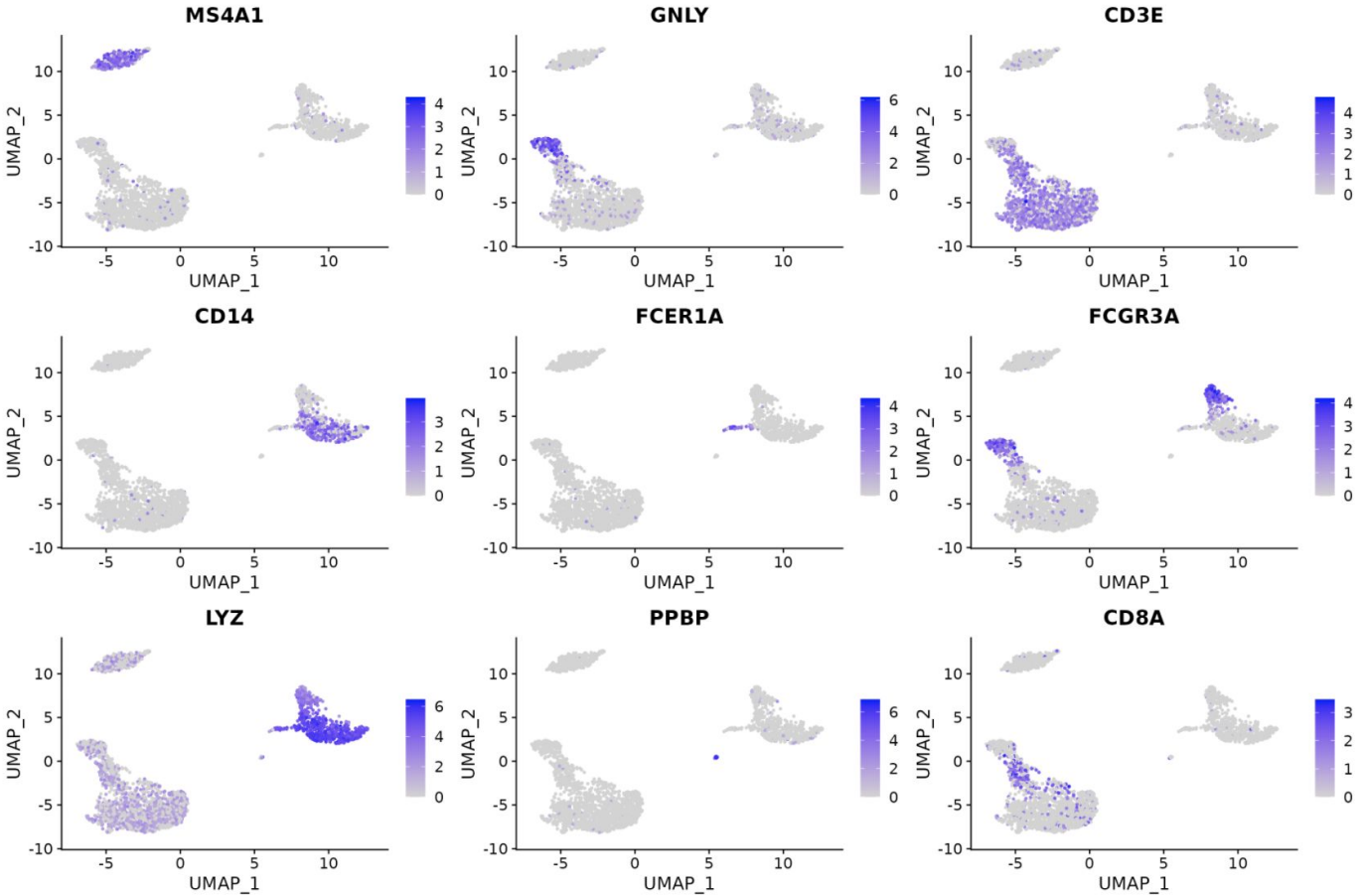
```
## # A tibble: 18 x 7
## # Groups:   cluster [9]
##        p_val avg_log2FC pct.1 pct.2 p_val_adj cluster gene
##        <dbl>      <dbl> <dbl> <dbl>     <dbl> <fct>   <chr>
##  1 1.74e-109       1.07 0.897 0.593 2.39e-105 0       LDHB
##  2 1.17e- 83       1.33 0.435 0.108 1.60e- 79 0       CCR7
##  3 0.              5.57 0.996 0.215 0.        1       S100A9
##  4 0.              5.48 0.975 0.121 0.        1       S100A8
##  5 7.99e- 87       1.28 0.981 0.644 1.10e- 82 2       LTB
##  6 2.61e- 59       1.24 0.424 0.111 3.58e- 55 2       AQP3
##  7 0.              4.31 0.936 0.041 0.        3       CD79A
##  8 9.48e-271       3.59 0.622 0.022 1.30e-266 3       TCL1A
##  9 1.17e-178       2.97 0.957 0.241 1.60e-174 4       CCL5
## 10 4.93e-169       3.01 0.595 0.056 6.76e-165 4       GZMK
## 11 3.51e-184       3.31 0.975 0.134 4.82e-180 5       FCGR3A
## 12 2.03e-125       3.09 1     0.315 2.78e-121 5       LST1
## 13 1.05e-265       4.89 0.986 0.071 1.44e-261 6       GZMB
## 14 6.82e-175       4.92 0.958 0.135 9.36e-171 6       GNLY
## 15 1.48e-220       3.87 0.812 0.011 2.03e-216 7       FCER1A
## 16 1.67e- 21       2.87 1     0.513 2.28e- 17 7       HLA-DPB1
## 17 7.73e-200       7.24 1     0.01  1.06e-195 8       PF4
## 18 3.68e-110       8.58 1     0.024 5.05e-106 8       PPBP
```

Visualize marker expression

```
VlnPlot(pbmc, features = c("MS4A1", "CD79A"))
```

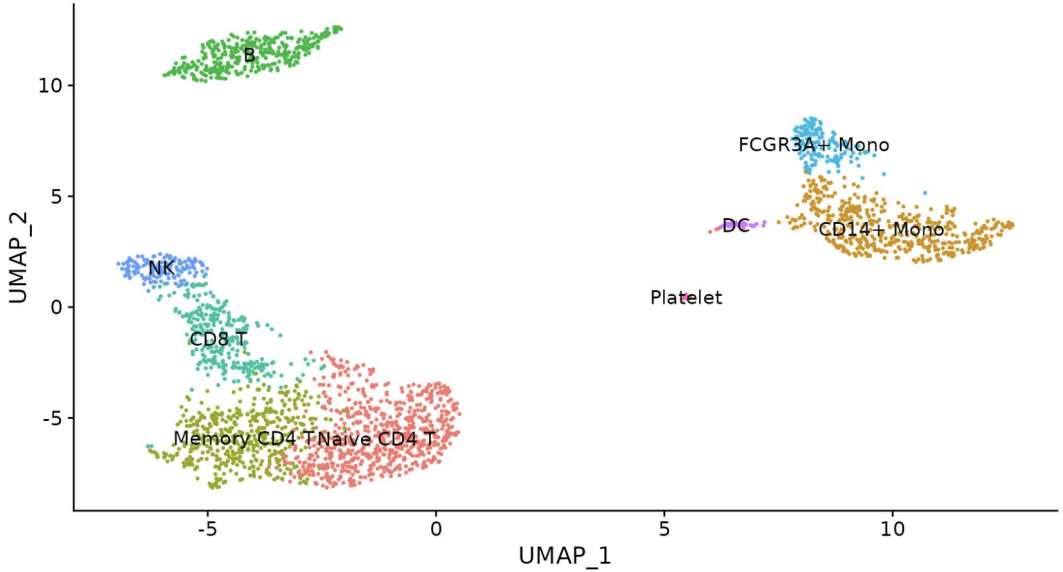


```
FeaturePlot(pbmc, features = c("MS4A1", "GNLY", "CD3E", "CD14", "FCER1A", "FCGR3A", "LYZ", "PPBP",
    "CD8A"))
```



Match clusters to canonical markers

```
new.cluster.ids <- c("Naive CD4 T", "CD14+ Mono", "Memory CD4 T", "B", "CD8 T", "FCGR3A+ Mono",
    "NK", "DC", "Platelet")
names(new.cluster.ids) <- levels(pbmc)
pbmc <- RenameIdents(pbmc, new.cluster.ids)
DimPlot(pbmc, reduction = "umap", label = TRUE, pt.size = 0.5) + NoLegend()
```



Further reading

https://hbctraining.github.io/scRNA-seq_online/schedule/

Harvard Chan
Bioinformatics Core
Bioinformatics for the Harvard Community

SATIJA LAB

Single Cell Genomics Day: A (Virtual) Practical Workshop

Details

WHEN:	**Friday March 26, 2021** 10:00 AM to 5:00 PM EDT
LIVESTREAM:	All talks will be openly livestreamed on this website. Registration is not required.

Homework - Part I (MAESTRO)

Initiating scRNA-seq

```
$ MAESTRO scrna-init --platform 10x-genomics --species GRCh38 \

--fastq-dir YOUR ANSWER HERE --fastq-prefix YOUR ANSWER HERE \

--cores 32 --rseqc --directory YOUR ANSWER HERE --outprefix YOUR ANSWER HERE \

--mapindex YOUR ANSWER HERE \

--whitelist YOUR ANSWER HERE \

  --umi-length 12 --lisadir YOUR ANSWER HERE --signature human.immune.CIBERSORT
```

Homework - Part I (MAESTRO)

Initializing scATAC-seq

```
$ MAESTRO scatac-init --platform 10x-genomics --format fastq --species GRCh38 \
--fastq-dir YOUR ANSWER HERE --fastq-prefix YOUR ANSWER HERE \
--cores 32 --directory YOUR ANSWER HERE --outprefix YOUR ANSWER HERE \
--peak-cutoff 100 --count-cutoff 1000 --frip-cutoff 0.2 --cell-cutoff 50 \
--giggleannotation YOUR ANSWER HERE \
--fasta YOUR ANSWER HERE \
--whitelist YOUR ANSWER HERE \
--rpmodel Enhanced \
--annotation --method RP-based --signature human.immune.CIBERSORT
```

Homework - Part II (scRNA-seq)

Useful Seurat functions

2) *Mitochondrial reads:* **PercentageFeatureSet()** and **VlnPlot()**

3) *# PCs statistically significant:* **JackStraw()**

Variability explained by each PC: https://github.com/satijalab/seurat/issues/982

4) *Visualization:* **DimPlot()** (and choose the appropriate 'reduction =')

5) *Clustering:* **FindClusters()**

6) *DE:* **FindAllMarkers()** and **FeaturePlot()**; **`top_n`** dplyr function may be useful

Homework - Part III (scATAC-seq)

8) **FindClusters()** & **DimPlot()**

9) **NormalizeData()**

Homework - Part IV (Integrating scRNA-seq & scATAC-seq data)

10) Find overlapped cells between RNA and ATAC and take this as subset

11) **FindTransferAnchors() & TransferData()**

https://satijalab.org/signac/articles/pbmc_vignette.html

Questions?