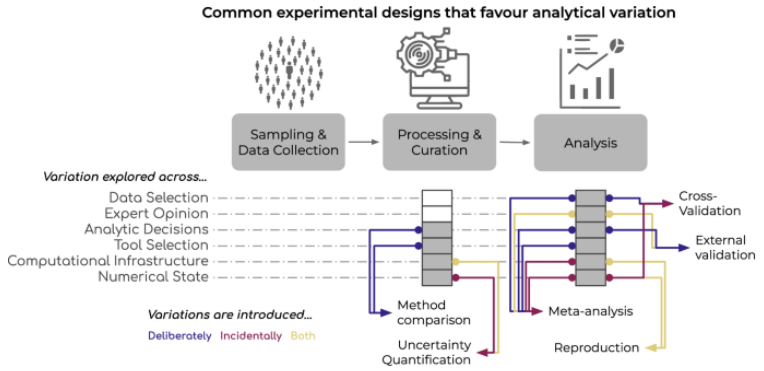


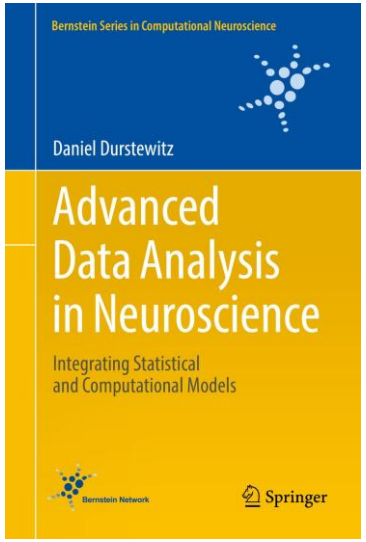
Processing and interpretation of neuroscience data:

Module 1 – Data overview and single cell sequencing

Dr. Kaja Moczulska & Dr. Łukasz Piszczek

Univ.-Prof. Dr. Wulf Haubensak
Department of Neuronal Cell Biology

Center For Brain Research (Zentrum für Hirnforschung)



Overview

Resources needed:

- Laptop
- Internet access
- Materials:
  - VM Machine
  - https://github.com/HugoMalagon/NeuroData_860.053-MUW

21.10
- Introduction to R, basics
- Visual analytics
- Grammar of graphics

28.10
- dimensional reduction: PCA, UMAP
- Normalization/scaling
- clustering: k-means, knn

4.11
- Intro to Seurat
- Single cell RNA seq
- Dataset merging and preprocessing

11.11
- clustering in Seurat
- DEG interpretation

„Exam“: 2-3 questions during classes, collected for the final

Day 3

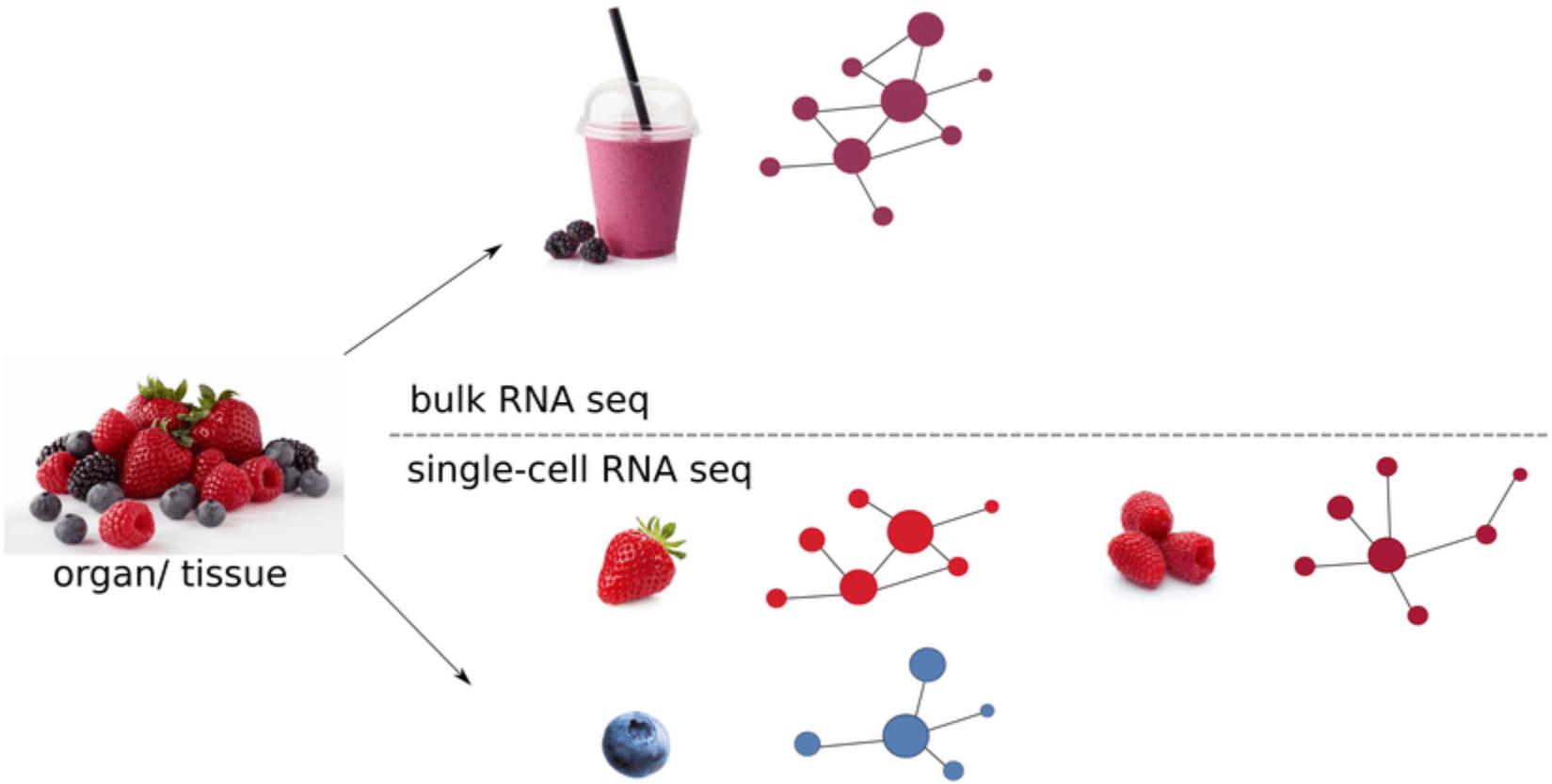
Intro to Seurat

Preprocessing of scRNAseq

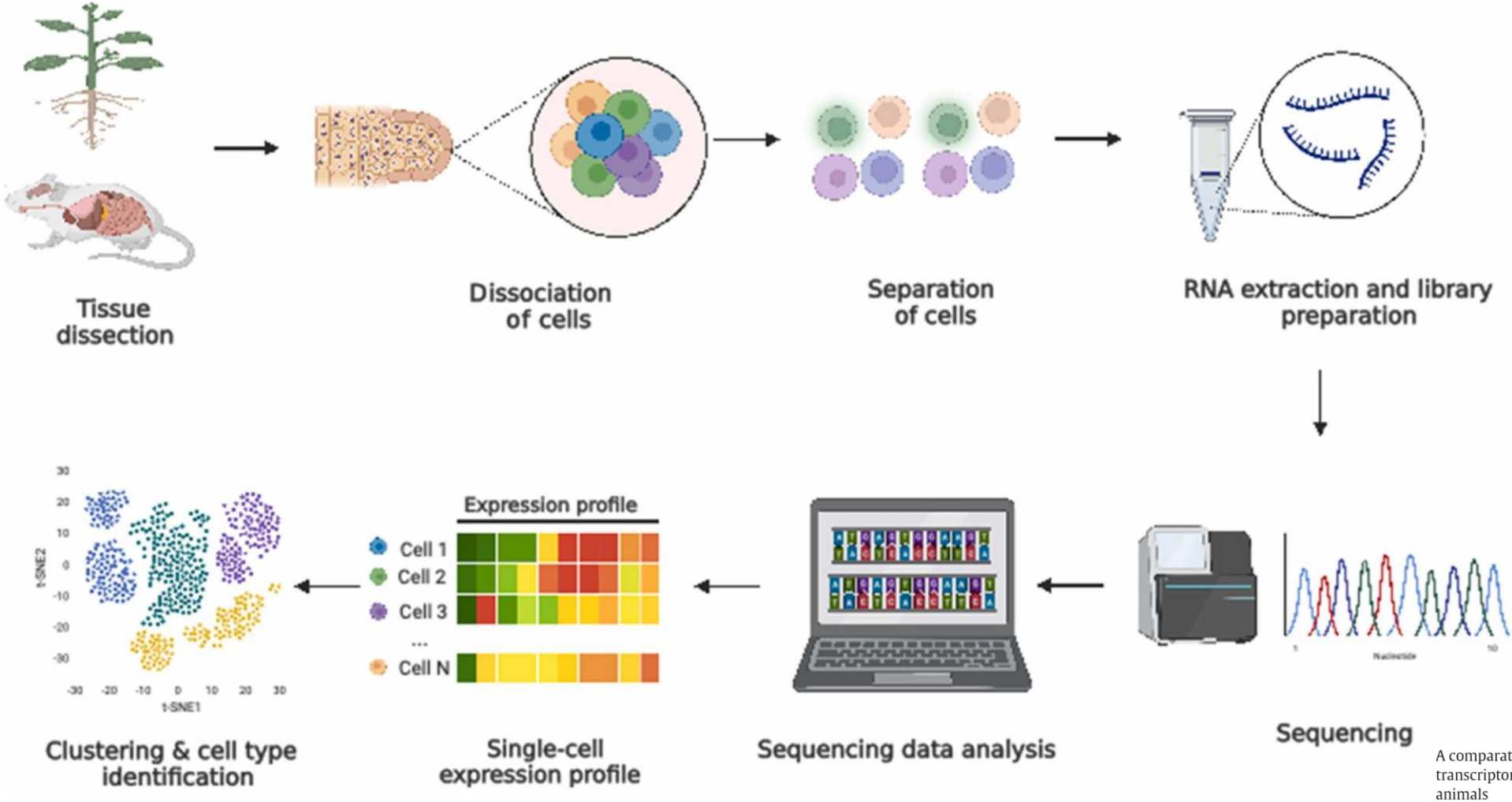
How to measure gene expression?

Method	Throughput	Quantitative	Sensitivity	Main Limitation
Northern Blotting	Low	Semi-quant	Moderate	Low throughput, laborious
qPCR	Low-Medium	Quantitative	High	Target-specific, limited multiplexing
Microarray	High	Quantitative	Moderate	Lower sensitivity than RNA-Seq
RNA-Seq	Very High	Quantitative	Very High	Expensive, computational load
In Situ Hybridization	Low	Qualitative	Moderate	Labor-intensive
SAGE	High	Quantitative	High	Largely replaced by RNA-Seq
Reporter Assays	Variable	Quantitative	Variable	Requires reporter construct
Digital PCR (dPCR)	Low	Quantitative	Very High	Specialized equipment

How to measure gene expression?



Single cell experiments workflow



A comparative analysis of single-cell transcriptomic technologies in plants and animals

Vamsidhar Reddy Netla [a 1], Harshraj Shinde [b 1], Gulshan Kumar [b], Ambika Dudhate [c], Jong Chan Hong [d], Ulhas Sopanrao Kadam [d]

Show more

Add to Mendeley Share Cite

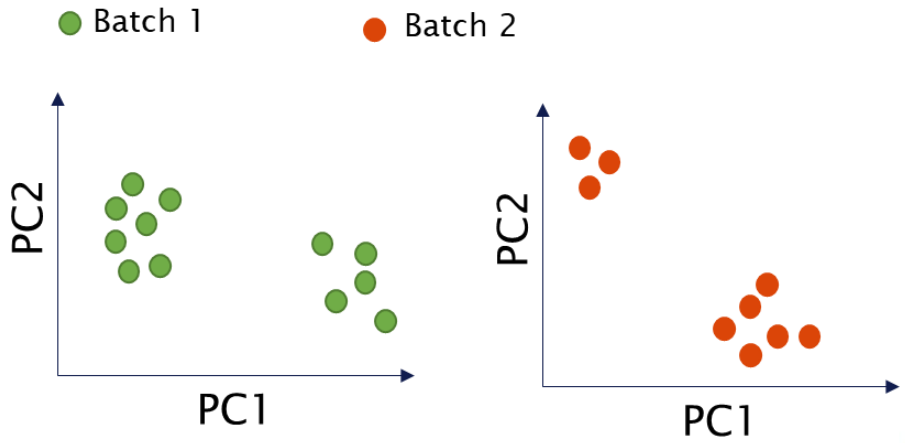
https://doi.org/10.1016/j.cpb.2023.100289

Get rights and content

Dataset integration

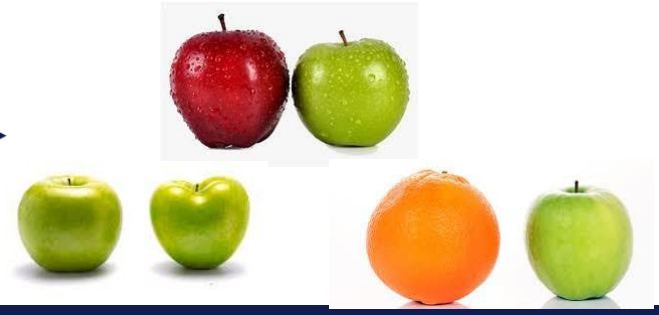
Why we should integrate the data?

- compare gene expression patterns between different conditions
- identify common and rare cell types
- build a large reference atlas
- …



What the source of data variation?

- technical factors
  - assays
  - platform
  - Protocol…
- biological variation
  - distinct tissue sampling regions
  - different genotype
  - different genetic background
  - different age
  - different species
  - different pre-treatment…



Data integration methods

- Promote cell type identity over background identity
  - in similar datasets
  - in different datasets
- The datasets should contain analogous cell types
- Popular algorithms applied
  - principal component analysis (PCA)
  - singular value decomposition (SVD)
  - canonical correlation analysis (CCA)

Table 6. Representative methods for single-cell and spatial transcriptomics integration.

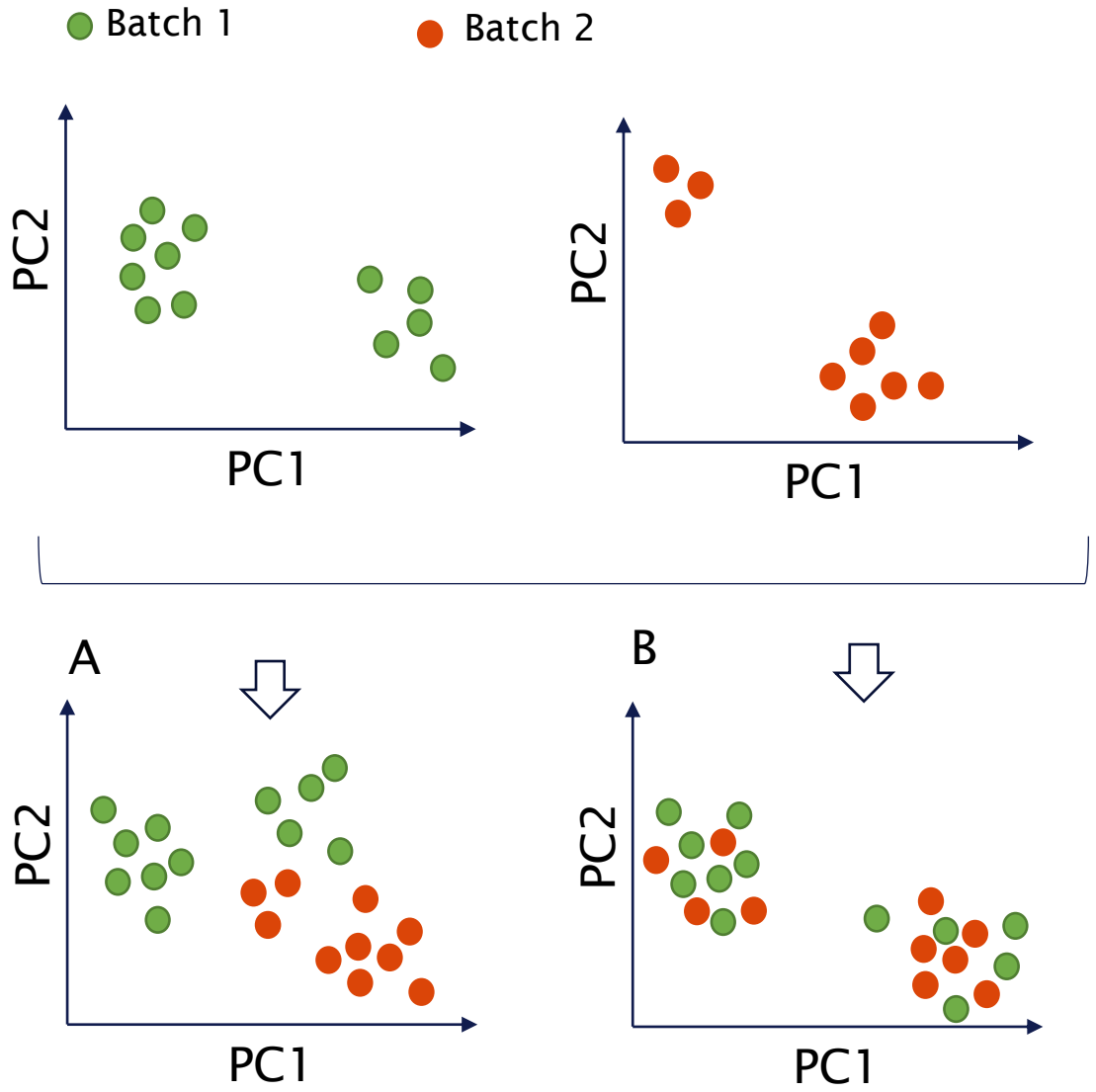
Tool	Input Data Demonstrated	scRNA-seq Data Preprocessing	Methods/Algorithms	Application/Output
Mapping				
scVI	scRNA-seq <-> scRNA-seq	Raw count matrix	Probabilistic modeling, neural networks, variational inference	scRNA-seq cell level and gene level batch correction scRNA-seq mapping
scANVI	scRNA-seq <-> scRNA-seq	Raw count matrix for UMI counts, gene length normalized count for read counts	Probabilistic modeling, neural networks, variational inference	scRNA-seq cell level and gene level batch correction scRNA-seq mapping annotation of single cells from annotated reference cells
MNN/fastMNN	scRNA-seq <-> scRNA-seq	Normalized with the library size, log transformed	Randomized SVD, MNN, weighted average of correction vectors	scRNA-seq cell level and gene level batch correction scRNA-seq mapping
Scanorama	scRNA-seq <-> scRNA-seq	L2-normalized for each cell	Randomized SVD, MNN, weighted average of correction vectors	scRNA-seq cell level and gene level batch correction scRNA-seq mapping
Seurat V3	scRNA-seq <-> scRNA-seq scRNA-seq <-> HPRI scRNA-seq <-> CITE-seq scRNA-seq <-> scATAC-seq	Normalized with the library size, log transformed, gene scaled	CCA, MNN, anchor scoring and weighting	scRNA-seq cell level and gene level batch correction scRNA-seq mapping Multimodal data mapping
Harmony	scRNA-seq <-> scRNA-seq scRNA-seq <-> HPRI	Normalized with the library size, log transformed, gene scaled, PCs from PCA	Maximum batch diversity soft k-means clustering, linear mixture model correction	scRNA-seq cell level batch correction scRNA-seq mapping Multimodal data mapping
LIGER	scRNA-seq <-> scRNA-seq scRNA-seq <-> HPRI scRNA-seq <-> single cell DNA methylation	Normalized with the library size, gene scaled but not centered	Integrative non-negative matrix factorization, shared factor neighborhood clustering	scRNA-seq cell level batch correction scRNA-seq mapping Multimodal data mapping
SpaGE	scRNA-seq <-> HPRI	Normalized with the library size, log transformed, gene scaled	SVD on the cosine similarity matrix of PCs from each modality	Multimodal data mapping
gimVI	scRNA-seq <-> HPRI	Raw count matrix	Probabilistic modeling, neural networks, variational inference	Multimodal data mapping (for gene imputation)
Tangram	scRNA-seq <-> spatial barcoding and HPRI	Normalized with the library size	Direct minimization of a Kullback–Leibler divergence and cosine distances	Multimodal data mapping
Cobolt	Unimodal data sets and multimodal data set	Raw count matrix	Probabilistic modeling, neural networks, variational inference	Multimodal data mapping
MultiVI	Unimodal data sets and multimodal data set	Raw count matrix	Probabilistic modeling, neural networks, variational inference	Multimodal data mapping
Seurat V5	Unimodal data sets and multimodal data sets	Depends on the mapping method in the first mapping step	Dictionary learning, Laplacian eigen-decomposition, sketching	Multimodal data mapping
Deconvolution				
Cell2location	scRNA-seq <-> spatial barcoding data	Raw count matrix	Bayesian negative binomial models, approximate variational inference	Estimate the absolute cell type abundance for each spot of spatial data
RCTD	scRNA-seq <-> spatial barcoding data	Raw count matrix	Poisson–lognormal models	Estimate the proportion of cell types for each spot of spatial data
stereoscope	scRNA-seq <-> spatial barcoding data	Raw count matrix	Negative binomial models	Estimate the proportion of cell types for each spot of spatial data
SpatialDWLS	scRNA-seq <-> spatial barcoding data	Normalized with the library size, log transformed	Enrichment analysis, dampened weighted least squares	Estimate the proportion of cell types for each spot of spatial data
SPOTlight	scRNA-seq <-> spatial barcoding data	Gene scaled	A seeded non-negative matrix factorization (NMF) regression and non-negative least squares	Estimate the proportion of cell types for each spot of spatial data
DestVI	scRNA-seq <-> spatial barcoding data	Raw count matrix	Probabilistic modeling, negative binomial models, neural networks, variational inference	Estimate both the proportion of cell types and the variations within each cell type for each spot of spatial

Open Access Review

A Review of Single-Cell RNA-Seq Annotation, Integration, and Cell–Cell Communication

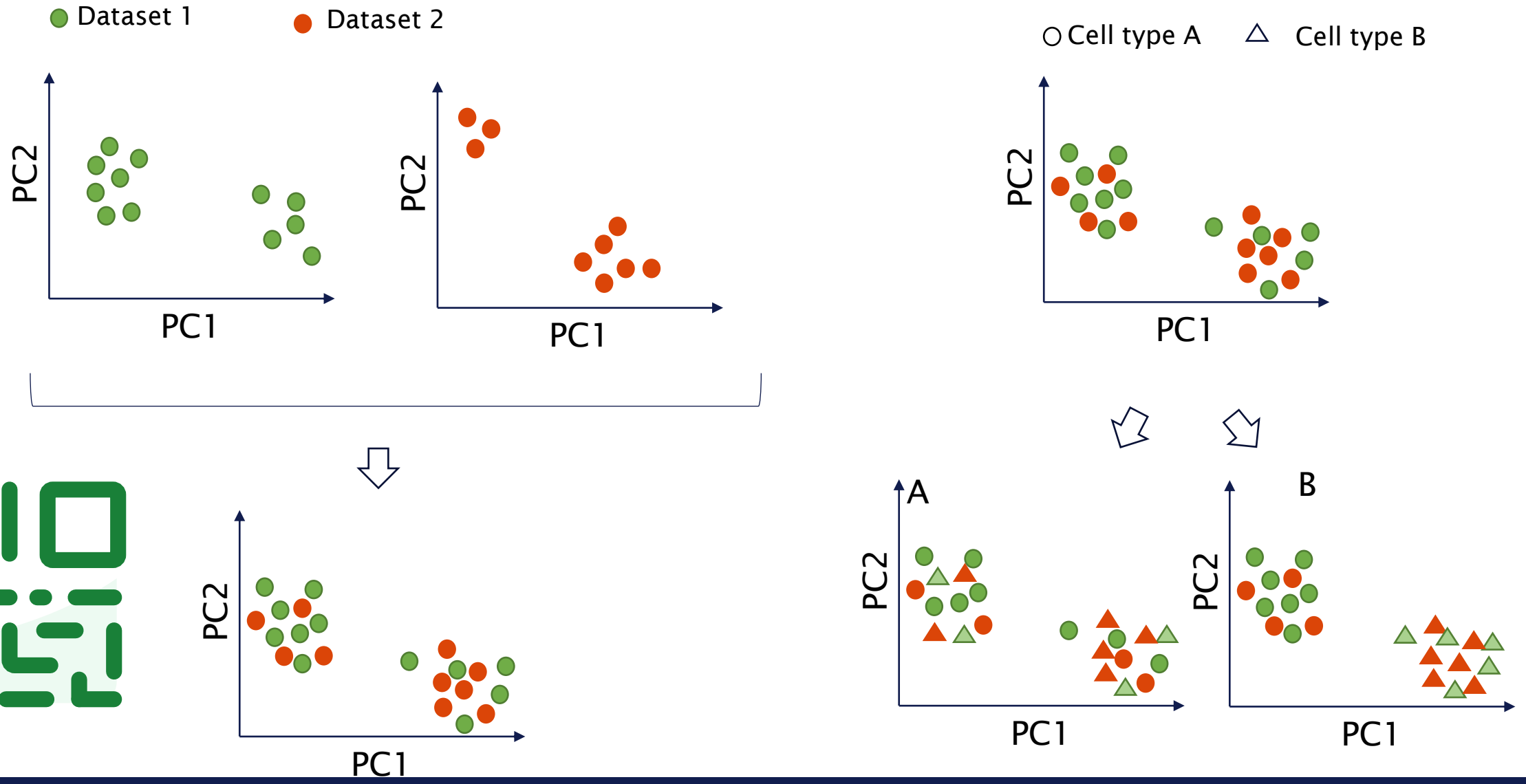
by Changde Cheng [1,†], Wenan Chen [2,†], Hongjian Jin [2,†] and Xiang Chen [1,*]

Comparing different data sets – data integration



Which outcome do you prefer?

Comparing different data sets – data integration



Which clusters represent better cell types?

Table 6. Representative methods for single-cell and spatial transcriptomics integration.

Tool	Input Data Demonstrated	scRNA-seq Data Preprocessing	Methods/Algorithms	Application/Output
Mapping				
scVI	scRNA-seq <-> scRNA-seq	Raw count matrix	Probabilistic modeling, neural networks, variational inference	scRNA-seq cell level and gene level batch correction scRNA-seq mapping
scANVI	scRNA-seq <-> scRNA-seq	Raw count matrix for UMI counts, gene length normalized count for read counts	Probabilistic modeling, neural networks, variational inference	scRNA-seq cell level and gene level batch correction scRNA-seq mapping annotation of single cells from annotated reference cells
MNN/fastMNN	scRNA-seq <-> scRNA-seq	Normalized with the library size, log transformed	Randomized SVD, MNN, weighted average of correction vectors	scRNA-seq cell level and gene level batch correction scRNA-seq mapping
Scanorama	scRNA-seq <-> scRNA-seq	L2-normalized for each cell	Randomized SVD, MNN, weighted average of correction vectors	scRNA-seq cell level and gene level batch correction scRNA-seq mapping
Seurat V3	scRNA-seq <-> scRNA-seq scRNA-seq <-> HPRI scRNA-seq <-> CITE-seq scRNA-seq <-> scATAC-seq	Normalized with the library size, log transformed, gene scaled	CCA, MNN, anchor scoring and weighting	scRNA-seq cell level and gene level batch correction scRNA-seq mapping Multimodal data mapping
Harmony	scRNA-seq <-> scRNA-seq scRNA-seq <-> HPRI	Normalized with the library size, log transformed, gene scaled, PCs from PCA	Maximum batch diversity soft k-means clustering, linear mixture model correction	scRNA-seq cell level batch correction scRNA-seq mapping Multimodal data mapping
LIGER	scRNA-seq <-> scRNA-seq scRNA-seq <-> HPRI scRNA-seq <-> single cell DNA methylation	Normalized with the library size, gene scaled but not centered	Integrative non-negative matrix factorization, shared factor neighborhood clustering	scRNA-seq cell level batch correction scRNA-seq mapping Multimodal data mapping
SpaGE	scRNA-seq <-> HPRI	Normalized with the library size, log transformed, gene scaled	SVD on the cosine similarity matrix of PCs from each modality	Multimodal data mapping
gimVI	scRNA-seq <-> HPRI	Raw count matrix	Probabilistic modeling, neural networks, variational inference	Multimodal data mapping (for gene imputation)
Tangram	scRNA-seq <-> spatial barcoding and HPRI	Normalized with the library size	Direct minimization of a Kullback–Leibler divergence and cosine distances	Multimodal data mapping
Cobolt	Unimodal data sets and multimodal data set	Raw count matrix	Probabilistic modeling, neural networks, variational inference	Multimodal data mapping
MultiVI	Unimodal data sets and multimodal data set	Raw count matrix	Probabilistic modeling, neural networks, variational inference	Multimodal data mapping
Seurat V5	Unimodal data sets and multimodal data sets	Depends on the mapping method in the first mapping step	Dictionary learning, Laplacian eigen-decomposition, sketching	Multimodal data mapping
Deconvolution				
Cell2location	scRNA-seq <-> spatial barcoding data	Raw count matrix	Bayesian negative binomial models, approximate variational inference	Estimate the absolute cell type abundance for each spot of spatial data
RCTD	scRNA-seq <-> spatial barcoding data	Raw count matrix	Poisson–lognormal models	Estimate the proportion of cell types for each spot of spatial data
stereoscope	scRNA-seq <-> spatial barcoding data	Raw count matrix	Negative binomial models	Estimate the proportion of cell types for each spot of spatial data
SpatialDWLS	scRNA-seq <-> spatial barcoding data	Normalized with the library size, log transformed	Enrichment analysis, dampened weighted least squares	Estimate the proportion of cell types for each spot of spatial data
SPOTlight	scRNA-seq <-> spatial barcoding data	Gene scaled	A seeded non-negative matrix factorization (NMF) regression and non-negative least squares	Estimate the proportion of cell types for each spot of spatial data
DestVI	scRNA-seq <-> spatial barcoding data	Raw count matrix	Probabilistic modeling, negative binomial models, neural networks, variational inference	Estimate both the proportion of cell types and the variations within each cell type for each spot of spatial

Open Access Review

A Review of Single-Cell RNA-Seq Annotation, Integration, and Cell–Cell Communication

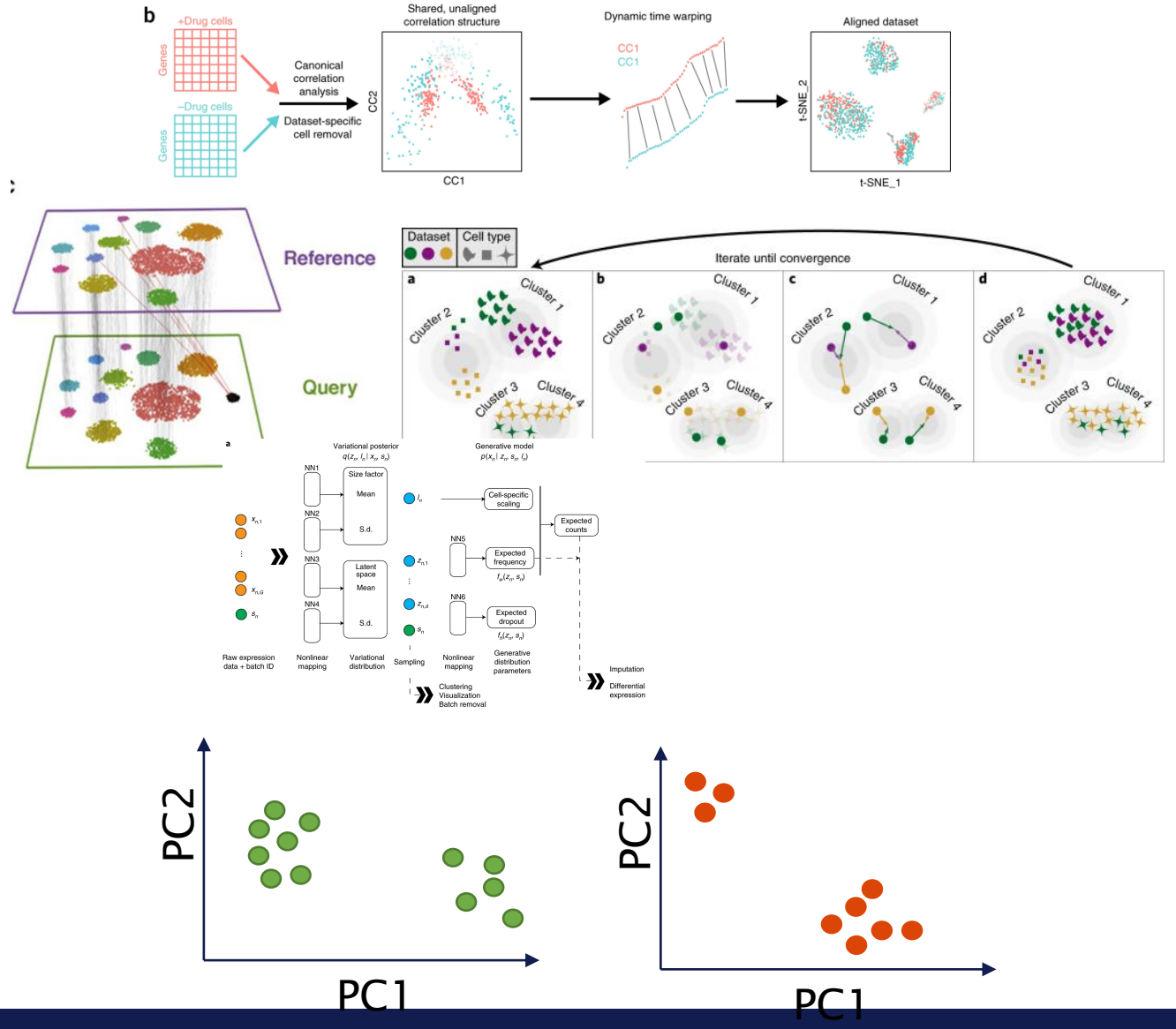
by Changde Cheng [1,†], Wenan Chen [2,†], Hongjian Jin [2,†] and Xiang Chen [1,*]

Choice of methods to integrate datasets in Seurat

- RPCA – reciprocal PCA
- CCA - canonical correlation analysis
- Harmony – iterative clustering
- scVI – neural network model

What influences the choice of method

- How similar are 2 datasets
- How big is the data
- Scalability of method
- Size differences between the datasets
- Batch effect

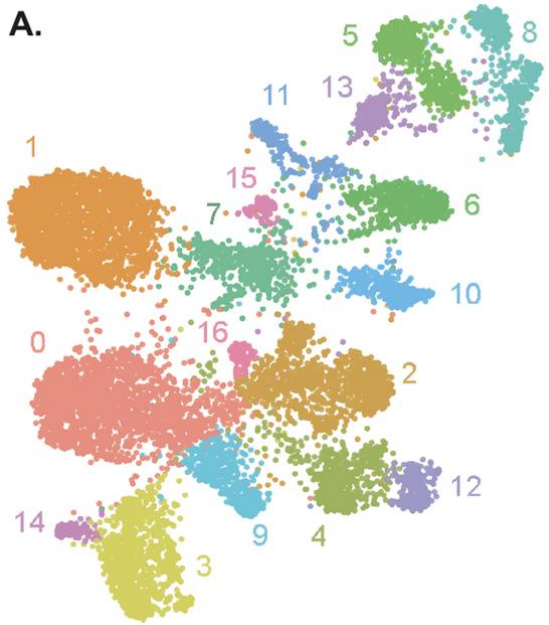


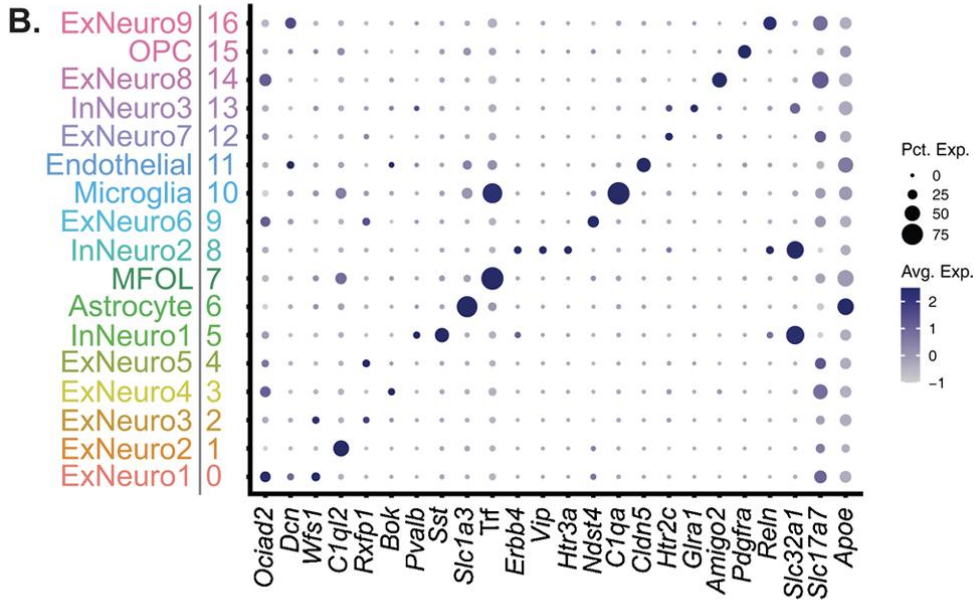
Examples of available data sets

Datasets (Showing 58 datasets)	Product	Species	Sample type	Cells or nuclei	Preservation
320k scFFPE From 8 Human Tissues 320k, 16-Plex	Flex Gene Expression v1.0	Human	Brain, Colon, Lung, Breast, Lymph node, Kidney, Skin, Cervix	N/A	Fixed
10k Mouse E18 Combined Cortex, Hippocampus and Subventricular Zone Cells, Chromium NextGEM Single Cell 3'	Universal 3' Gene Expression v3.1	Mouse	Brain	Cells	NA
10k Mouse E18 Combined Cortex, Hippocampus and Subventricular Zone Cells, Chromium GEM-X Single Cell 3'	Universal 3' Gene Expression v4	Mouse	Brain	Cells	NA
10k Adult Mouse Brain Nuclei Isolated with Chromium Nuclei Isolation Kit, Chromium NextGEM Single Cell 3'	Universal 3' Gene Expression v3.1	Mouse	Brain	Nuclei	Fresh Frozen
10k Adult Mouse Brain Nuclei Isolated with Chromium Nuclei Isolation Kit, Chromium GEM-X Single Cell 3'	Universal 3' Gene Expression v4	Mouse	Brain	Nuclei	Fresh Frozen
10k Mouse Forebrain FFPE Tissue Dissociated using gentleMACS Dissociator, Singleplex Sample (Next GEM)	Flex Gene Expression v1.0	Mouse	brain	Cells	FFPE
Mouse Brain Nuclei Isolated with Chromium Nuclei Isolation Kit, SaltyEZ Protocol, and 10x Complex Tissue DP (CT Sorted and CT Unsorted)	Epi Multiome ATAC + Gene Expression v1.0	Mouse	brain	Nuclei	Fresh Frozen
Mixture of Lung Cancer and Glioblastoma FFPE Tissues Dissociated Manually or using gentleMACS Dissociator, Multiplexed Samples, 4 Probe Barcodes (Next GEM)	Flex Gene Expression v1.0	Human	lung, brain	Cells	FFPE
5k Adult Mouse Brain Nuclei Isolated with Chromium Nuclei Isolation Kit	Universal 3' Gene Expression v3.1	Mouse	brain	Nuclei	Fresh Frozen
8k Adult Mouse Cortex Cells, ATAC v2, Chromium Controller	Epi ATAC v2	Mouse	brain	Nuclei	N/A
8k Adult Mouse Cortex Cells, ATAC v2, Chromium X	Epi ATAC v2	Mouse	brain	Nuclei	N/A
8k Adult Mouse Cortex Cells, ATAC v1.1, Chromium X	Epi ATAC v1.1	Mouse	brain	Nuclei	N/A
Multiomic Integration Neuroscience Application Note: Single Cell Multiome RNA + ATAC Alzheimer's Disease Mouse Model Brain Coronal Sections from One Hemisphere Over a Time Course	Epi Multiome ATAC + Gene Expression v1.0	Mouse	brain	Nuclei	Fresh Frozen, FFPE
Mouse E18 Combined Cortex, Hippocampus and Subventricular Zone Nuclei Multiplexed, 12 CMOs: 3'v3.1 Targeted, Custom Neuroscience Panel	Universal 3' Gene Expression v3.1	Mouse	brain, cortex, hippocampus, subventricular zone	Nuclei	N/A
Flash-Frozen Human Healthy Brain Tissue (3k)	Epi Multiome ATAC + Gene Expression v1.0	Human	brain, cerebellum	Nuclei	Frozen
Fresh Embryonic E18 Mouse Brain (5k)	Epi Multiome ATAC + Gene Expression v1.0	Mouse	cortex, hippocampus, ventricular zone, brain	Nuclei	Fresh

Dataset we will analyse today

- Alzheimer's disease model: ApoE knock out
- Single nucleus data
- 7 pooled hippocampi samples from 9-month-old C57BL/6 J wild type (WT) mice (ID GSM5067107)
- 7 pooled hippocampi samples from 9-month-old *Apoe* knockout (EKO) mice (ID GSM5067109)





differences in specific cell types.

All analyses were performed on GEO dataset GSE166261 [dataset] (Shi et al., 2021a, b) comprised of snRNAseq data generated by droplet-based Chromium Single Cell 3' Reagent Kit (10x Genomics) on 7 pooled hippocampi samples from 9-month-old C57BL/6 J wild type (WT) mice (Sample GSM5067107) and 7 pooled hippocampi samples from 9-month-old *Apoe* knockout (EKO) mice (Sample GSM5067109). The initial EKO mouse was generated using the 129P2/OlaHsd-derived E14Tg2a ES cell line (Piedrahita et al., 1992). The mice used to generate the samples for the snRNAseq data for this study were produced by backcrossing at least 10 generations to C57BL/6 J inbred mice.

Short Communication

Analysis of differential gene expression and transcript usage in hippocampus of *Apoe* null mutant mice: Implications for Alzheimer's disease

Andrew E. Weller*, Glenn A. Doyle, Benjamin C. Reiner, Richard C. Crist, Wade H. Berrettini

Center for Neurobiology and Behavior, Department of Psychiatry, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA, 19104, United States

General workflow

- Data import and Seurat object generation (already done)
- QC for mitochondrial genes, nr of genes and expression level
- Normalization: normalizing gene expression level for each cell
- Integration of data sets (AD and WT mice)
- Feature selection of variable genes
- Scaling: scaling expression for each gene to avoid overrepressentation of highly expressed genes
- Linear dimensional reduction: PCA
- Clustering: k-means for cluster selection
- Non-linear dimensional reduction: UMAP
- Feature selection and cluster annotation: looking at cluster specific markers
- (optional) Cell mapping, subclustering

How to find the data (example: GEO Gene Expression Omnibus)

ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM5067109

Scope: Self Format: HTML Amount: Quick GEO accession: GSE166261 GO

Series GSE166261 Query DataSets for GSE166261

Status	Public on Feb 06, 2021
Title	Overexpressing low-density lipoprotein receptor reduces tau-associated neurodegeneration via apoE-dependent and independent mechanisms
Organism	Mus musculus
Experiment type	Expression profiling by high throughput sequencing

differences in specific cell types.

All analyses were performed on GEO dataset GSE166261 [dataset] (Shi et al., 2021a, b) comprised of snRNAseq data generated by droplet-based Chromium Single Cell 3' Reagent Kit (10x Genomics) on 7 pooled hippocampi samples from 9-month-old C57BL/6 J wild type (WT) mice (Sample GSM5067107) and 7 pooled hippocampi samples from 9-month-old *Apoe* knockout (EKO) mice (Sample GSM5067109). The initial EKO mouse was generated using the 129P2/OlaHsd-derived E14Tg2a ES cell line (Piedrahita et al., 1992). The mice used to generate the samples for the snRNAseq data for this study were produced by backcrossing at least 10 generations to C57BL/6 J inbred mice.

Samples (6) Less...

GSM5067107	Isolated single nuclei from 7 pooled hippocampi of 9-month old WT mice
GSM5067108	Isolated single nuclei from 7 pooled hippocampi of 9-month old LDLR mice
GSM5067109	Isolated single nuclei from 7 pooled hippocampi of 9-month old EKO mice
GSM5067110	Isolated single nuclei from 8 pooled hippocampi of 9-month old P301S mice
GSM5067111	Isolated single nuclei from 7 pooled hippocampi of 9-month old P301S/LDLR mice
GSM5067112	Isolated single nuclei from 7 pooled hippocampi of 9-month old P301S/EKO mice

Series (1) GSE166261 Overexpressing low-density lipoprotein receptor reduces tau-associated neurodegeneration via apoE-dependent and independent mechanisms

Relations

BioSample	SAMN17812771
SRA	SRX10038115

Supplementary file	Size	Download	File type/resource
GSM5067109_EKO_barcodes.tsv.gz	23.9 Kb	(ftp)(http)	TSV
GSM5067109_EKO_genes.tsv.gz	212.7 Kb	(ftp)(http)	TSV
GSM5067109_EKO_matrix.mtx.gz	14.7 Mb	(ftp)(http)	MTX

SRA Run Selector

Raw data are available in SRA

Processed data provided as supplementary file

Download all 3 files

Introduction into Seurat

satijalab.org/seurat/

Seurat 5.2.0 | Install | Get started | Vignettes | Extensions | FAQ | News

Introductory Vignettes

- PBMC 3K guided tutorial
- Data visualization vignette
- SCTransform, v2 regularization
- Using Seurat with multi-modal data
- Seurat v5 Command Cheat Sheet

Data Integration

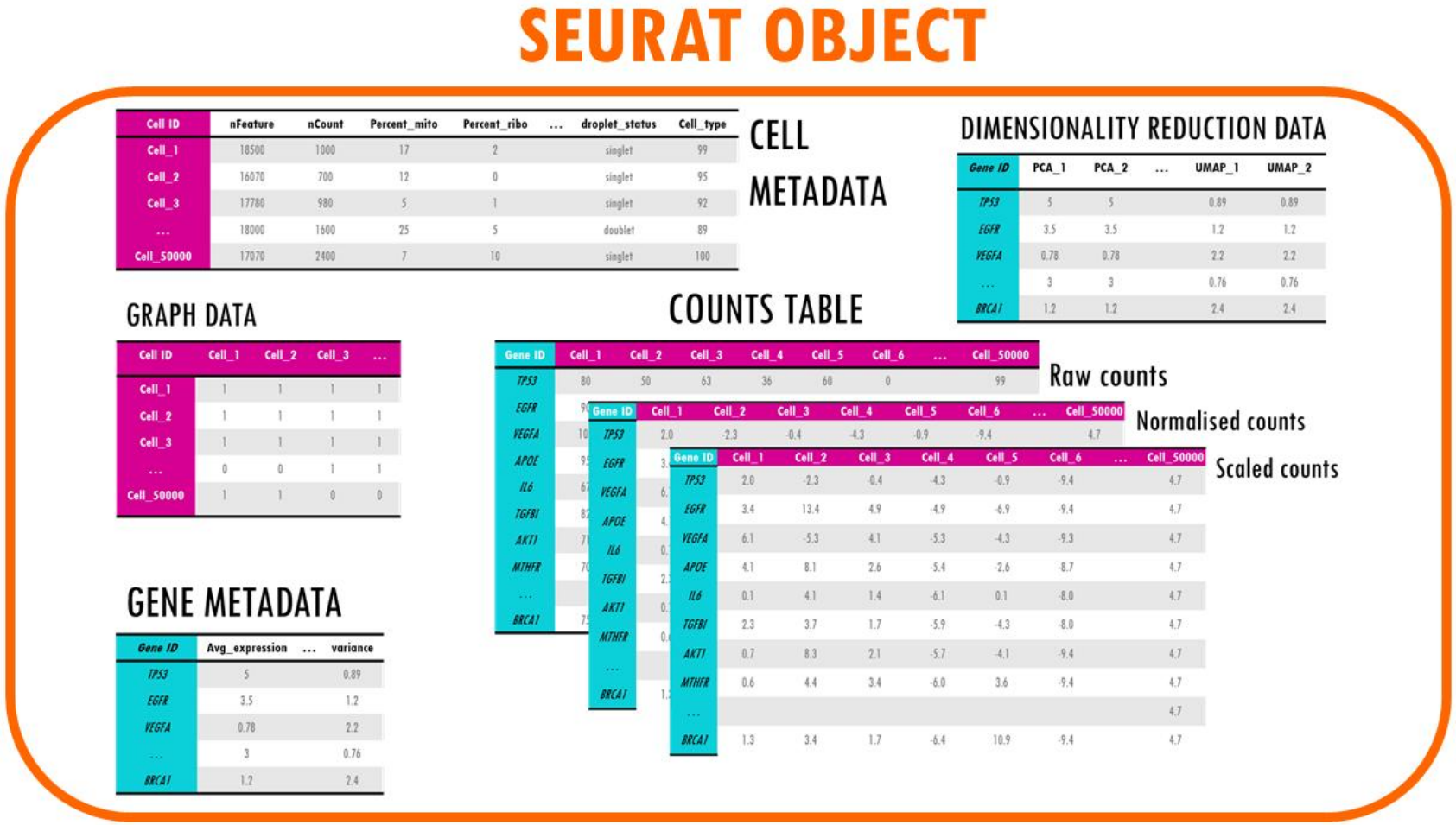
- Introduction to scRNA-seq integration

Seurat v5

https://satijalab.org/seurat/

Introduction into Seurat

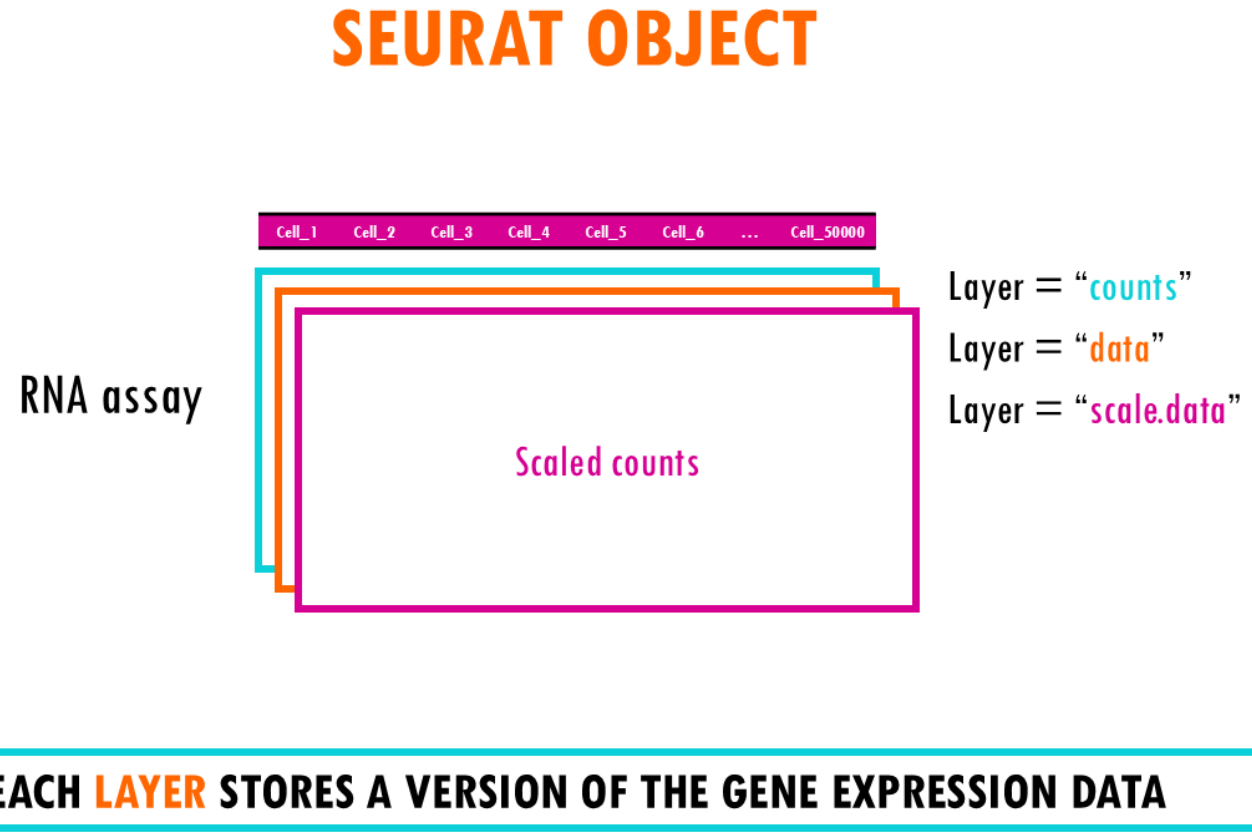
- What it is
- Why we use it



https://biostatsquid.com/seurat-objects-explained/

Introduction into Seurat

- What it is
- Why we use it



https://biostatsquid.com/seurat-objects-explained/

General workflow

- Data import and Seurat object generation (already done)
- QC for mitochondrial genes, nr of genes and expression level
- Normalization: normalizing gene expression level for each cell
- Integration of data sets (AD and WT mice)
- Feature selection of variable genes
- Scaling: scaling expression for each gene to avoid overrepressentation of highly expressed genes
- Linear dimensional reduction: PCA
- Clustering: k-means for cluster selection
- Non-linear dimensional reduction: UMAP
- Feature selection and cluster annotation: looking at cluster specific markers
- (optional) Cell mapping, subclustering

Part 1: Data import

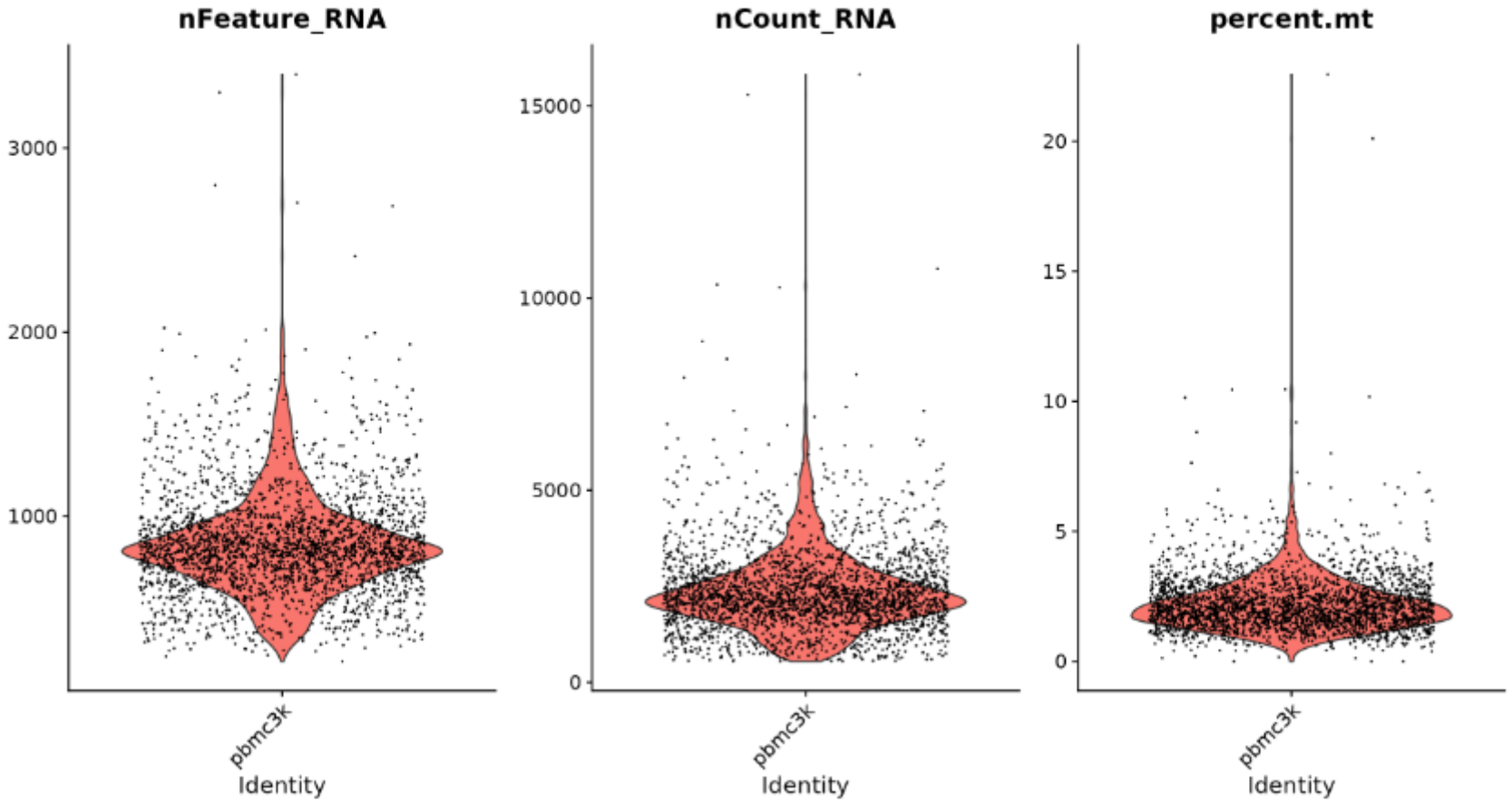
```
library(dplyr)
library(Seurat)
library(patchwork)

# Load the PBMC dataset
pbmc.data <- Read10X(data.dir = "/brahms/mollag/practice/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)
##  1 layer present: counts
```

https://satijalab.org/seurat/articles/pbmc3k_tutorial

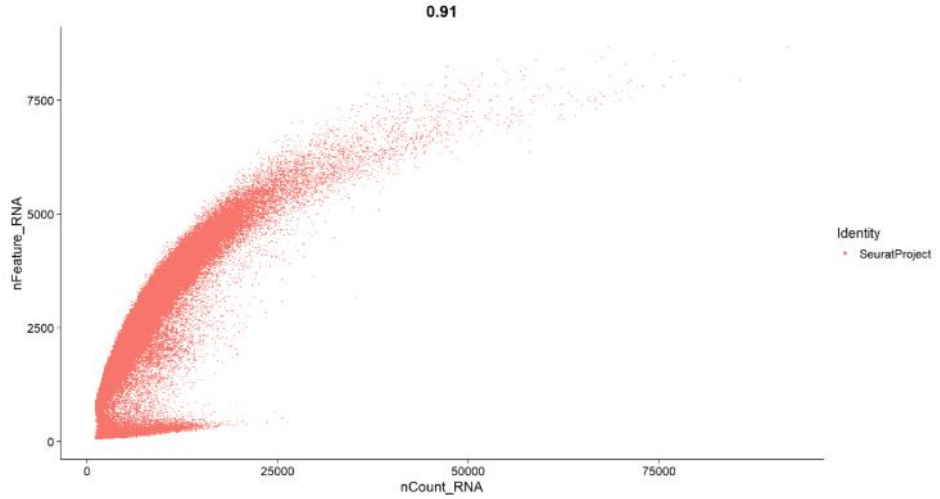
Part 2: Data filter

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

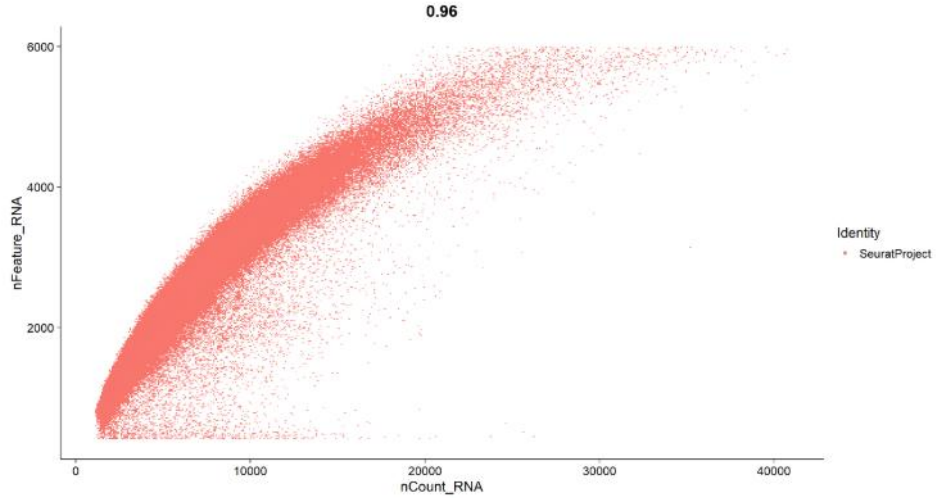


Part 2: Data filter

```
# FeatureScatter is typically used to visualize feature-feature relationships, but can be used for anything calculated by th
e object, i.e. columns in object metadata, PC scores etc.
FeatureScatter(obj, feature1 = "nCount_RNA", feature2 = "nFeature_RNA")
```

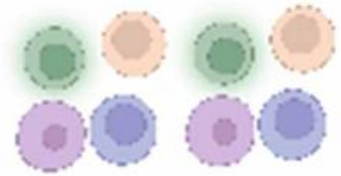


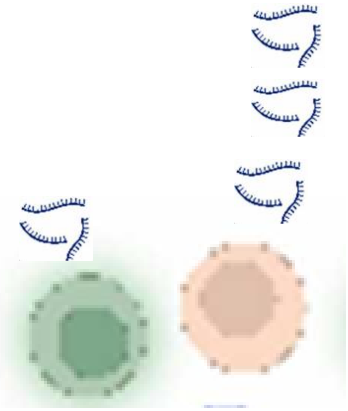

```
obj <- subset(obj, subset = nFeature_RNA > 400 & nFeature_RNA < 6000)
FeatureScatter(obj, feature1 = "nCount_RNA", feature2 = "nFeature_RNA")
```



https://biostatsquid.com/seurat-objects-explained/

Normalization





	Cell 1	Cell 2	Cell 3
Total reads	3	300	100
RAW			
Gene A	1	100	50
Gene B	2	200	50
NORMALISED			
Gene A	8.11	8.11	8.52
Gene B	8.81	8.81	8.52

LOG2 NORMALISATION

Gene A cell 1:
$\log_2(1/3*10000 + 1)$

Normalization: This step addresses differences between cells by adjusting for sequencing depth or other technical variations, ensuring that gene expression values are comparable across cells.

Part 2: Data normalization

Normalizing the data

After removing unwanted cells from the dataset, the next step is to normalize the data. By default, we employ a global-scaling normalization method "LogNormalize" that normalizes the feature expression measurements for each cell by the total expression, multiplies this by a scale factor (10,000 by default), and log-transforms the result. In Seurat v5, Normalized values are stored in `obj[["RNA"]]$data`.

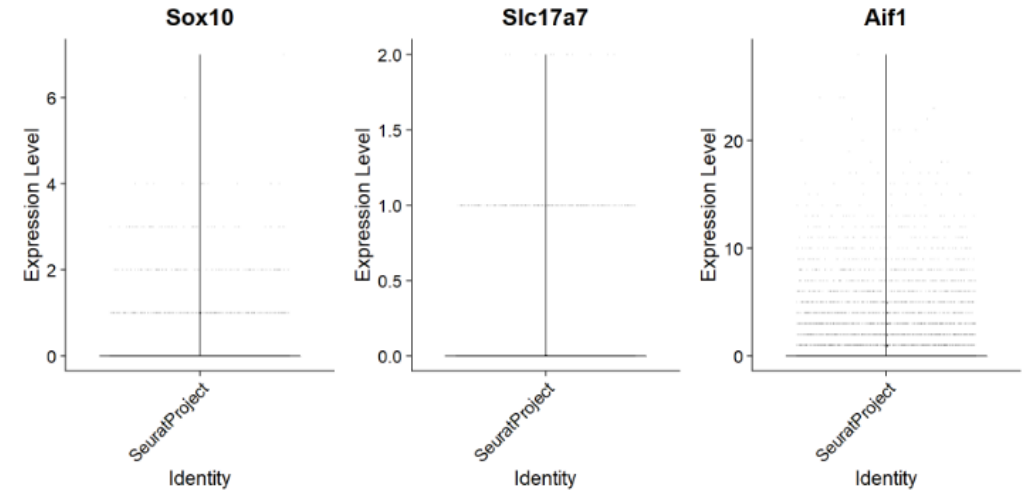
```
#obj <- NormalizeData(obj, normalization.method = "LogNormalize", scale.factor = 1e4)
```

For clarity, in this previous line of code (and in future commands), we provide the default values for certain parameters in the function call. However, this isn't required and the same behavior can be achieved with:

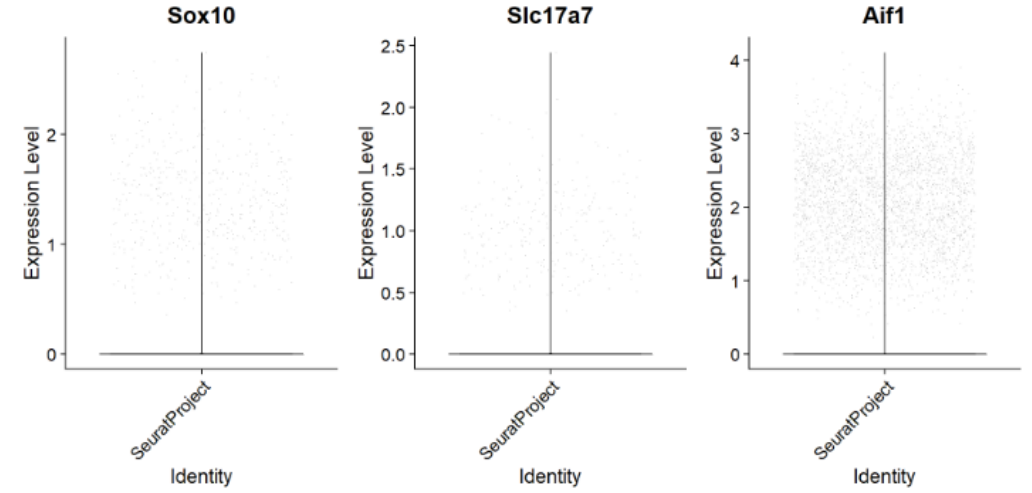
```
obj <- NormalizeData(obj)
```

Compare Normalization with Counts

```
VlnPlot(obj, features = c("Sox10", "Slc17a7", "Aif1"), ncol = 3, layer = "counts", alpha = 0.1)
```

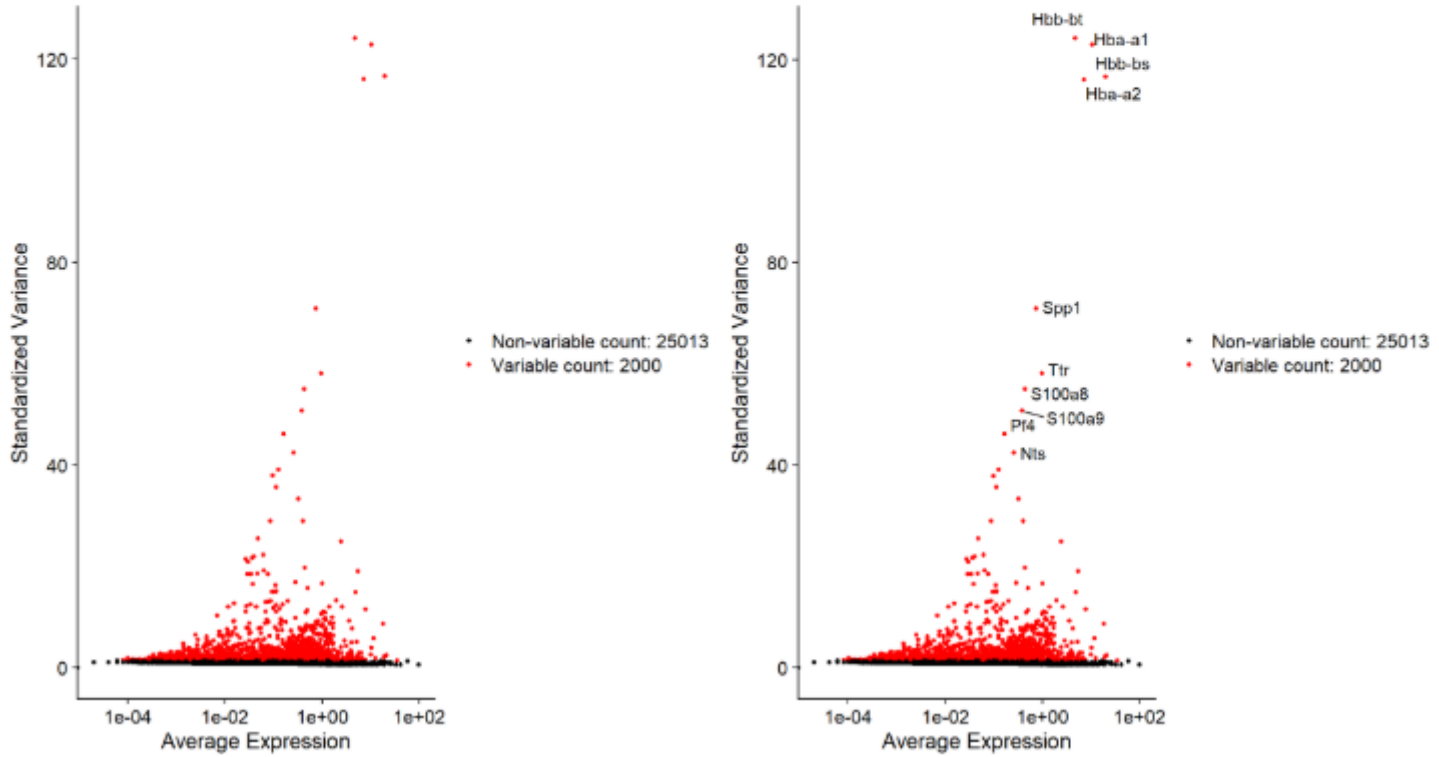



```
# We then visualize normalized data
VlnPlot(obj, features = c("Sox10", "Slc17a7", "Aif1"), ncol = 3, layer = "data", alpha = 0.1)
```

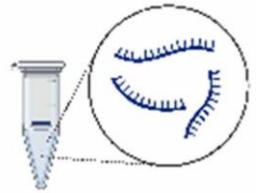


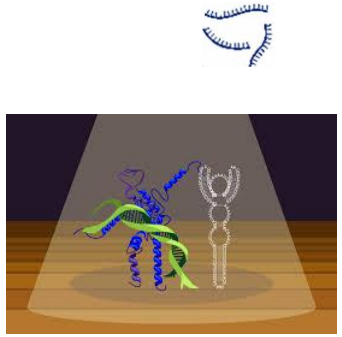
Part 3: Find variable features (feature selection)

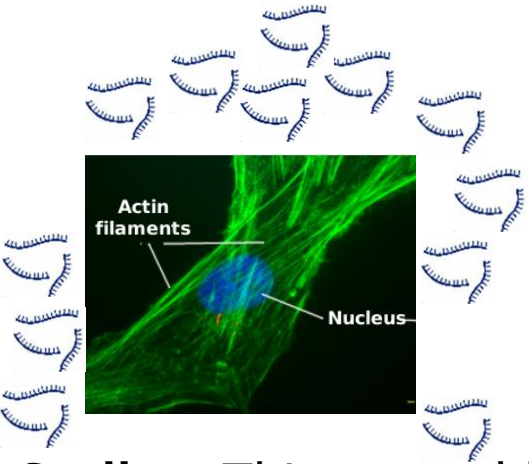
```
obj <- FindVariableFeatures(obj)
# Identify the 10 most highly variable genes
top10 <- head(VariableFeatures(obj), 10)

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



Scaling







RAW

	Cell 1	Cell 2	Cell 3
Gene A	1	100	50
Gene B	2	200	50

LOG2 NORMALISATION

NORMALISED

Gene A	8.11	8.11	8.52
Gene B	8.81	8.81	8.52

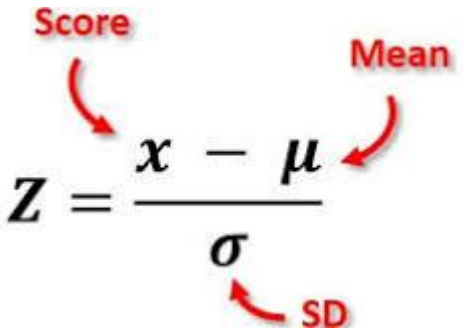
Z-SCORE

SCALED

Gene A	-0.57	-0.57	+1.19
Gene B	+0.58	+0.58	-1.16

Scaling: This step addresses differences between genes by adjusting for varying ranges in expression levels across genes, making sure that no single gene with very high expression dominates the analysis.

Part 4: scale the data



Scaling the data

Next, we apply a linear transformation ('scaling') that is a standard pre-processing step prior to dimensional reduction techniques like PCA. The `ScaleData()` function:

- 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 `obj[["sketch"]]$scale.data`
- By default, only variable features are scaled.
- You can specify the `features` argument to scale additional features

```
obj <- ScaleData(obj)
```

Part 5: PCA

For the first principal components, Seurat outputs a list of genes with the most positive and negative loadings, representing modules of genes that exhibit either correlation (or anti-correlation) across single-cells in the dataset.

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

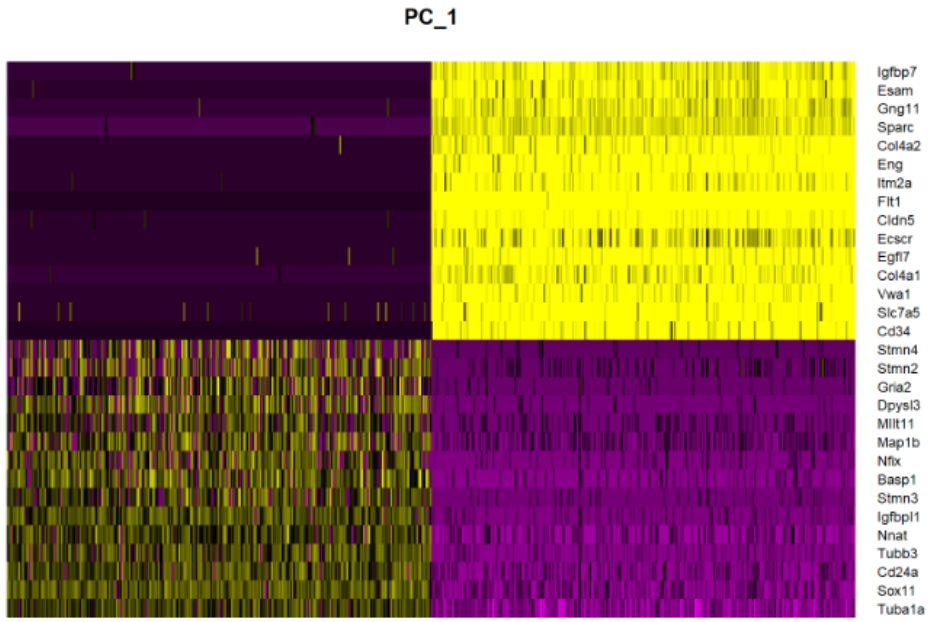
Seurat provides several useful ways of visualizing both cells and features that define the PCA, including `VizDimReduction()`, `DimPlot()`, and `DimHeatmap()`

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

```
## PC_ 1
## Positive:  Tuba1a, Sox11, Cd24a, Tubb3, Nnat
## Negative:  Igfbp7, Esam, Gng11, Sparc, Col4a2
## PC_ 2
## Positive:  C1qc, C1qb, Tyrobp, C1qa, Fcer1g
## Negative:  Tmsb10, Tuba1a, Tsc22d1, Sparcl1, Serpinh1
## PC_ 3
## Positive:  Tubb3, Tmsb10, Mllt11, Stmn2, Stmn3
## Negative:  Hmgb2, Dbi, Phgdh, Fabp7, Cks2
## PC_ 4
## Positive:  Vtn, Ndufa4l2, Kcnj8, Higd1b, Abcc9
## Negative:  Cldn5, Vwa1, Cd34, Pglyrp1, Ctla2a
## PC_ 5
## Positive:  Birc5, Ube2c, Ccna2, Nusap1, Spc25
## Negative:  Cimap3, Clu, Foxj1, Pierce1, Rsph1
```

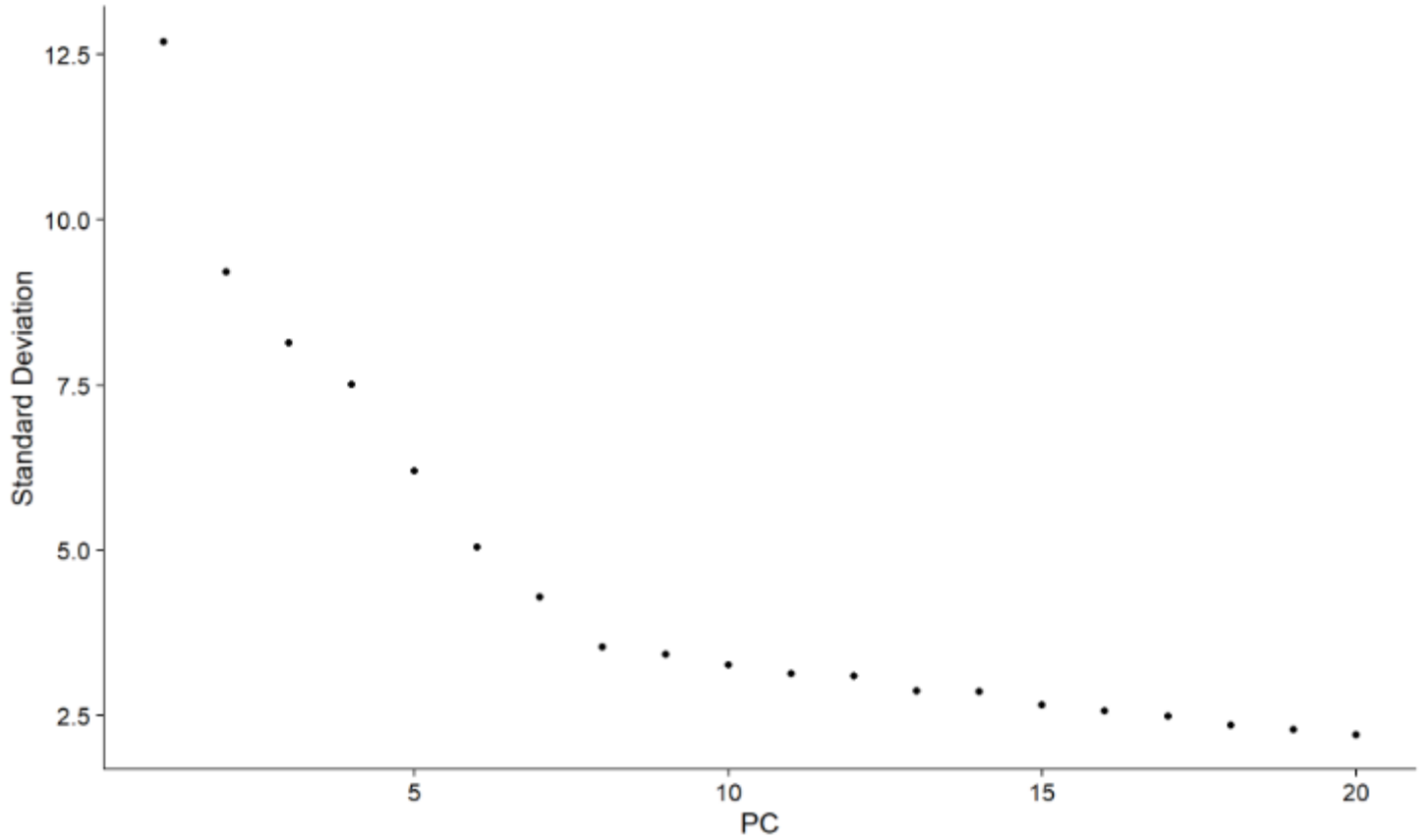
```
VizDimLoadings(obj, dims = 1:2, reduction = 'pca')
```

```
DimHeatmap(obj, dims = 1, cells = 500, balanced = TRUE)
```



Part 5: PCA

```
ElbowPlot(obj)
```



Part 6: Clustering

```
obj <- FindNeighbors(obj,reduction = "pca", dims = 1:50)
obj <- FindClusters(obj, resolution = 2) # the higher the number the higher the clusters
```

```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 49451
## Number of edges: 1969041
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.8779
## Number of communities: 54
## Elapsed time: 13 seconds
```

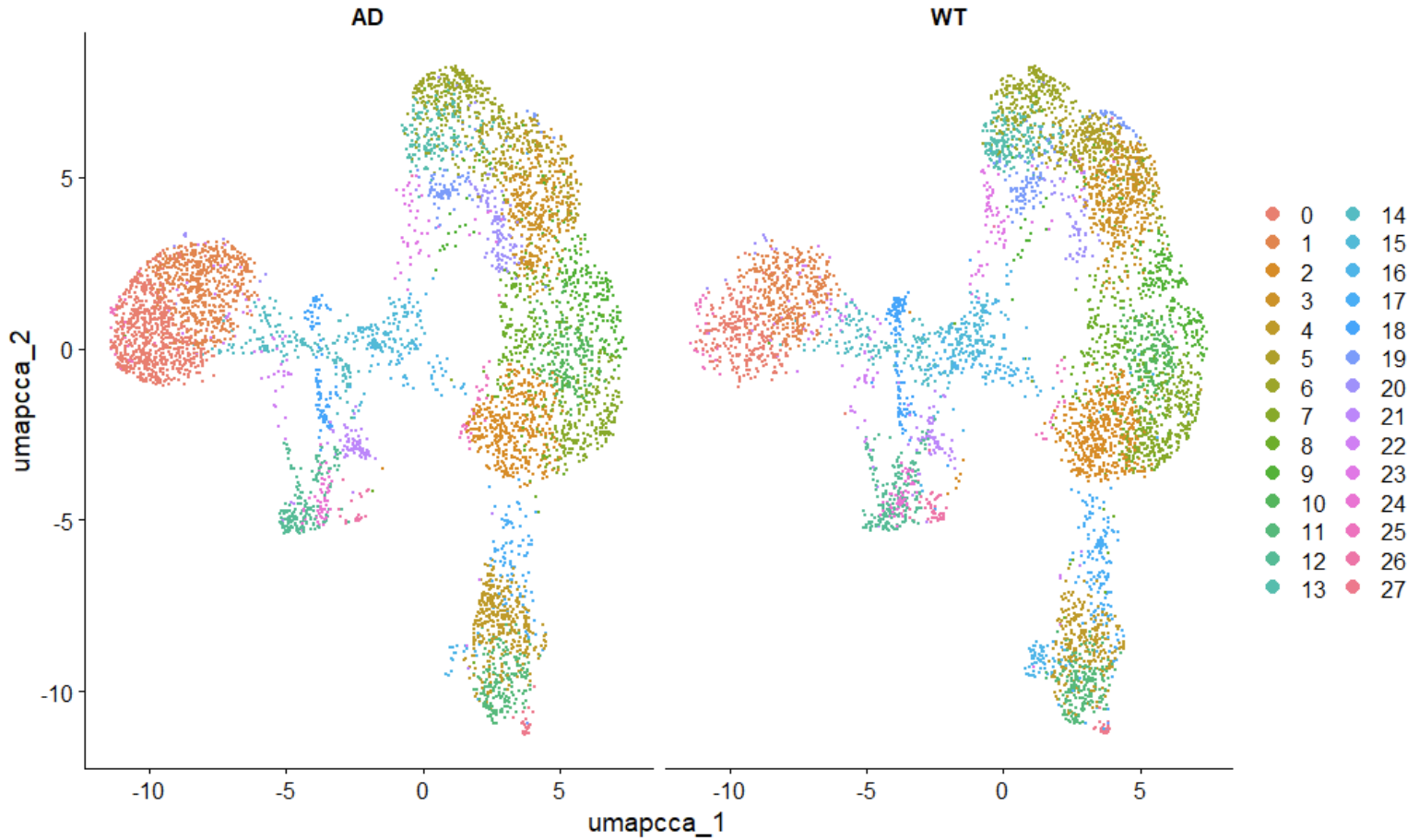
```
# Look at cluster IDs of the first 5 cells
head(Idents(obj), 5)
```

```
## AAACCTGAGATAGGAG-1 AAACCTGAGCGGCTTC-1 AAACCTGAGGAATCGC-1 AAACCTGAGGACACCA-1
##               <NA>               <NA>               <NA>               <NA>
## AAACCTGAGGCCCGTT-1
##                 17
## 54 Levels: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 ... 53
```

Part 7: Non-linear dimensional reduction (UMAP)

```r
```{r, UMAP}
Seurat_AD_WT <- RunUMAP(Seurat_AD_WT, reduction = "integrated.cca", dims = 1:13, reduction.name = "umap.cca")
DimPlot(Seurat_AD_WT, label = F, label.size = 3, reduction = 'umap.cca', split.by = 'condition')

```
```



Day 4

Clustering of cells in Seurat

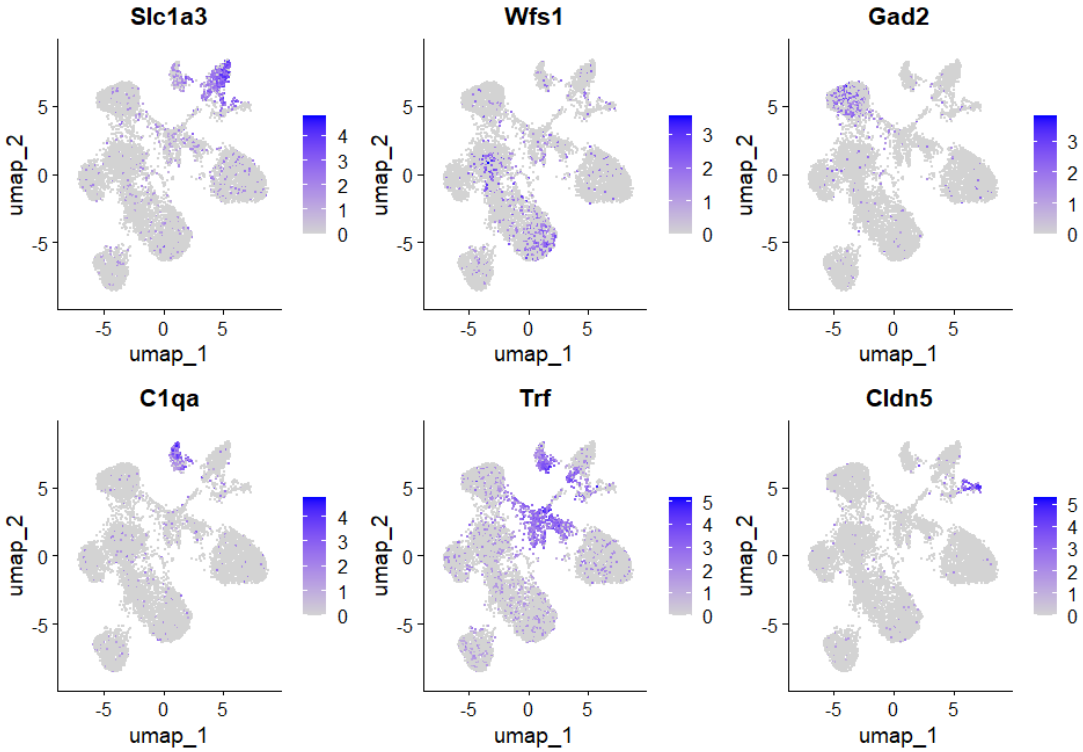
DEG and its interpretation

General workflow

- Data import and Seurat object generation (already done)
- QC for mitochondrial genes, nr of genes and expression level
- Normalization: normalizing gene expression level for each cell
- Integration of data sets (AD and WT mice)
- Feature selection of variable genes
- Scaling: scaling expression for each gene to avoid overrepressentation of highly expressed genes
- Linear dimensional reduction: PCA
- Clustering: k-means for cluster selection
- Non-linear dimensional reduction: UMAP
- Feature selection and cluster annotation: looking at cluster specific markers
- (optional) Cell mapping, subclustering

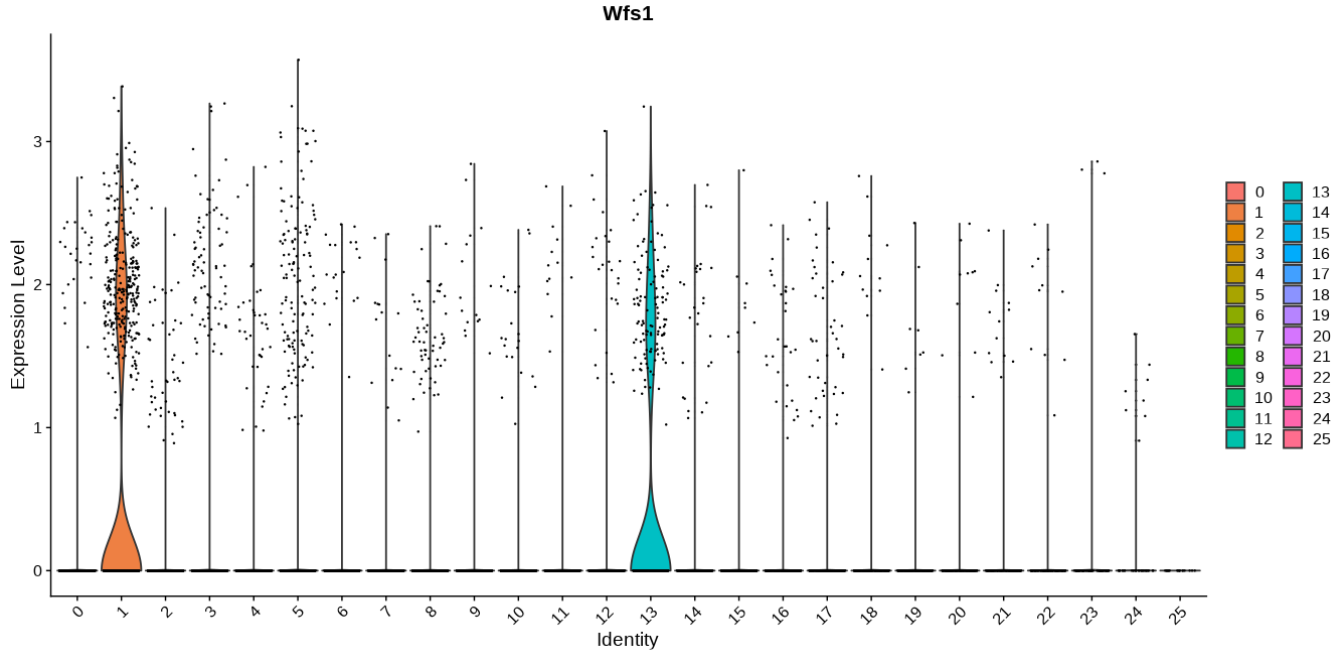
Part 8: Find markers

```r
```{r,fig.height = 7, fig.width = 10}
FeaturePlot(
  object = Seurat_AD_WT,
  features = c(
    "Slc1a3", #Astrocytes
    "Wfs1", # Glut Neurons or Slc17a7
    "Gad2", # GABA neurons
    "C1qa", # Microglia
    "Trf", # Myelin-forming mature oligodendrocytes
    "Cldn5"), # Endothelial cells
  ncol = 3)
```
```

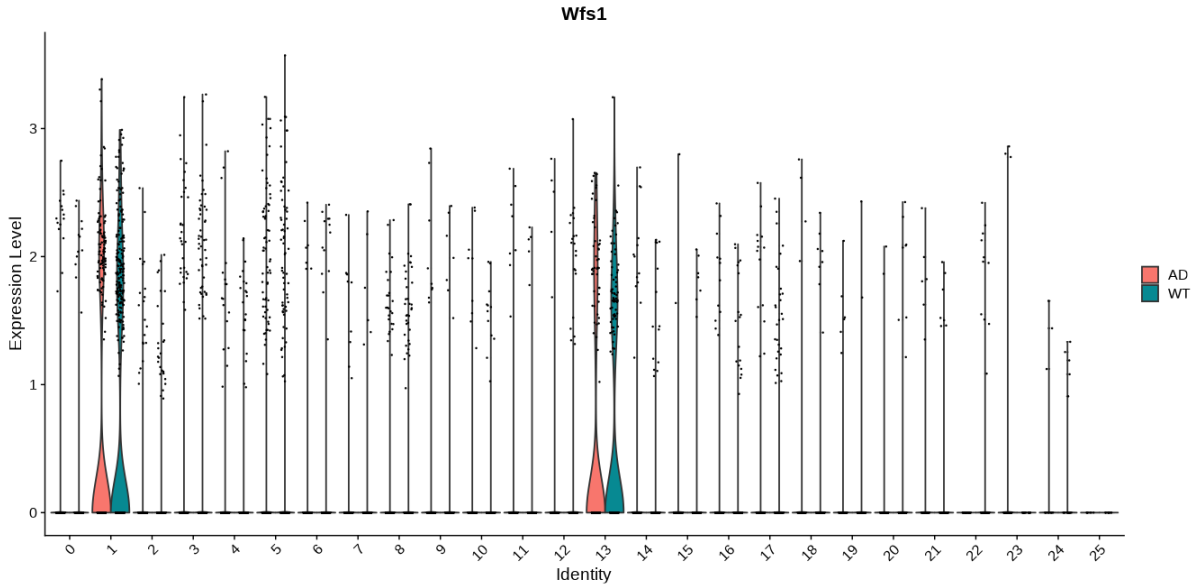
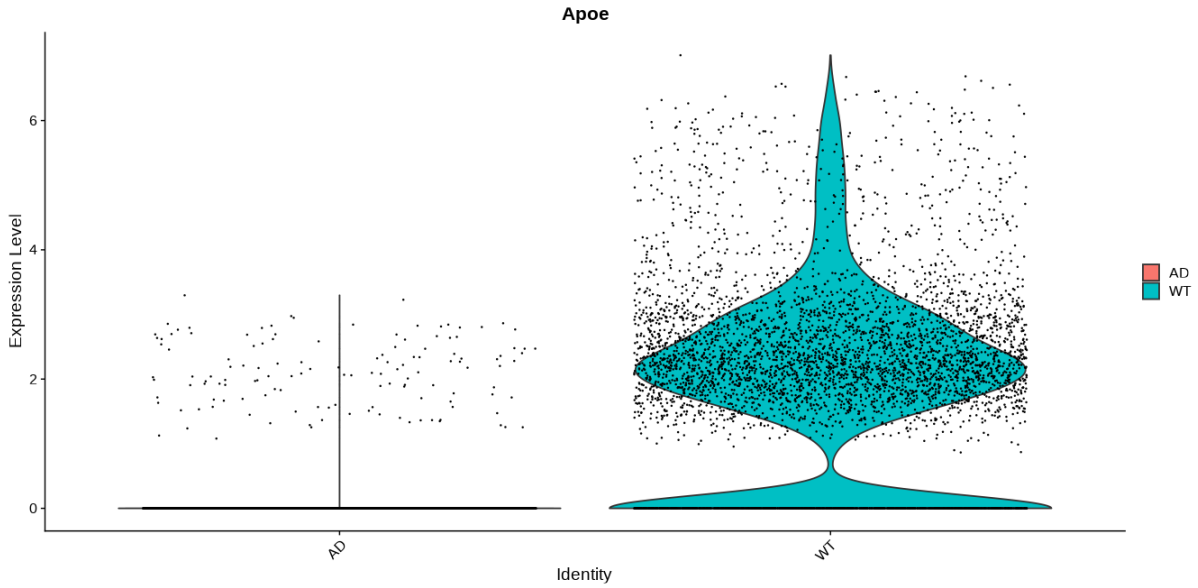


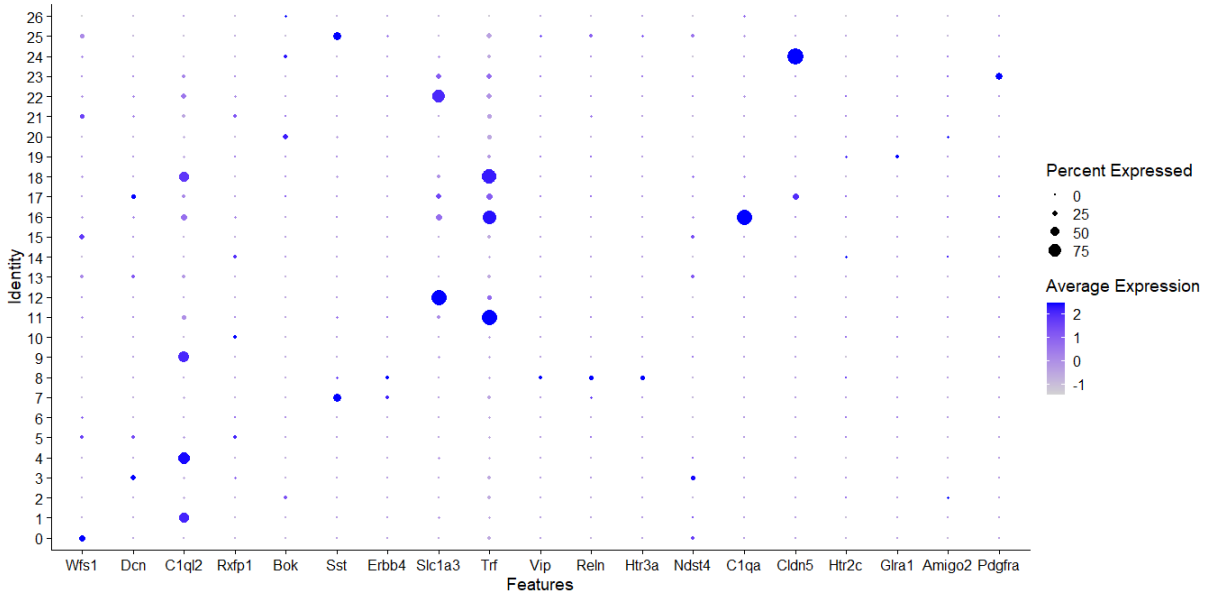
Differential gene expression

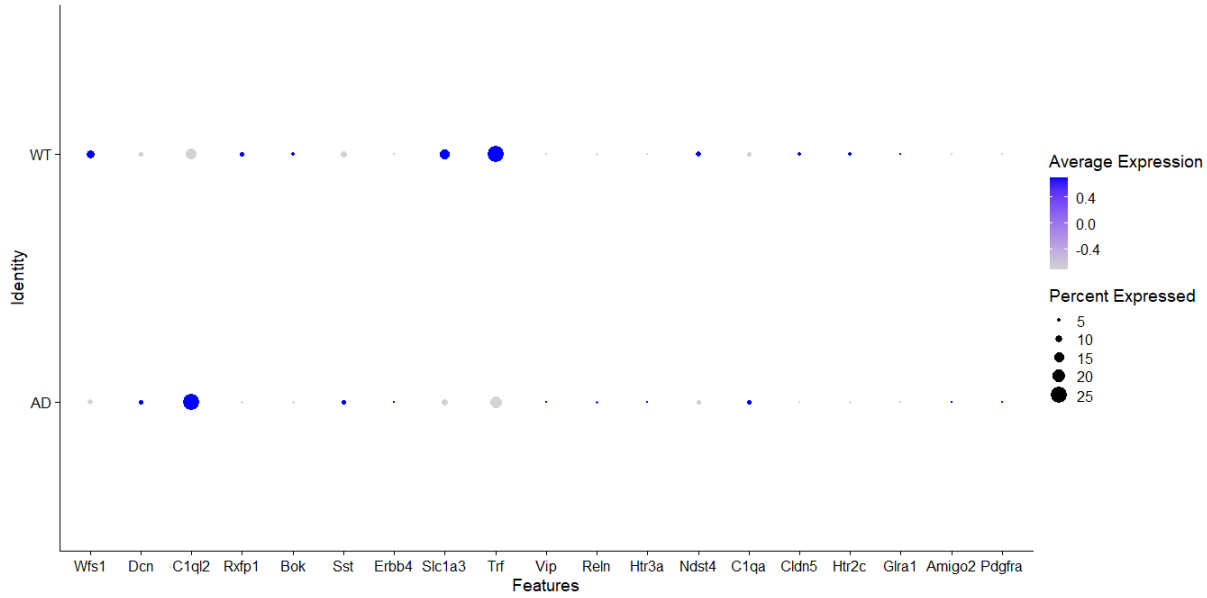
```r
VlnPlot(obj, 'Dlx2')
```



Part 8: Find markers

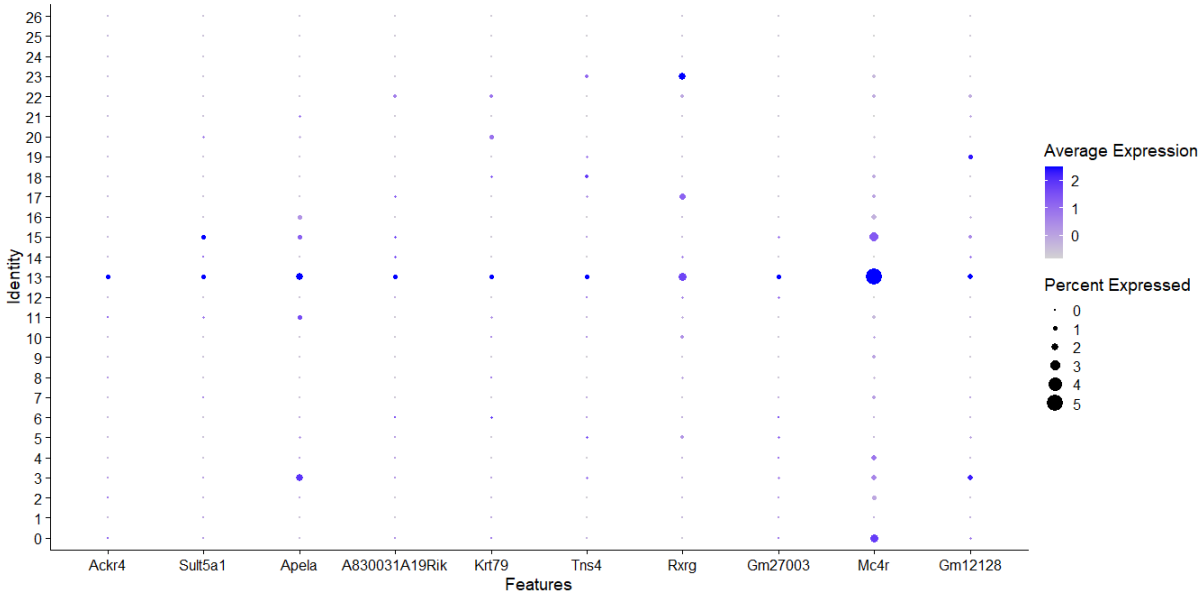
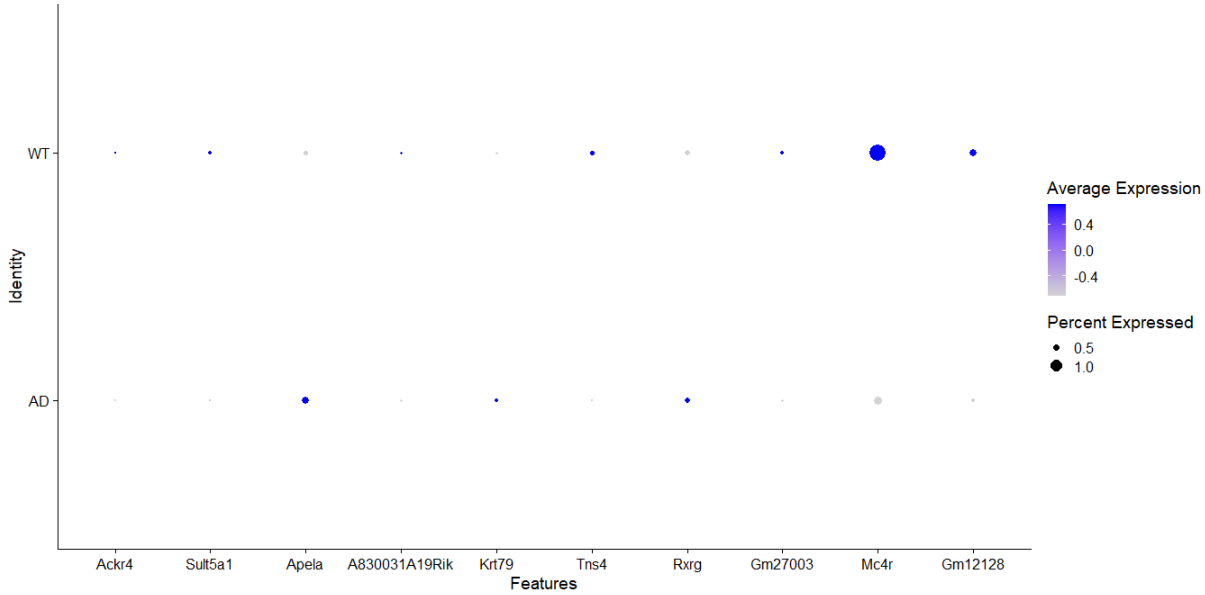
```r
```{r, fig.height = 7, fig.width = 14}
VlnPlot(Seurat_AD_WT, 'Wfs1')
VlnPlot(Seurat_AD_WT, 'Wfs1', split.by = 'condition')
VlnPlot(Seurat_AD_WT, 'Apoe', group.by = 'condition') # check knockout
```
```





Part 8: Find markers

```
```{r, fig.height = 7, fig.width = 14}
DotPlot(Seurat_AD_WT, features= all_genes)
DotPlot(Seurat_AD_WT, features= all_genes, group.by = "condition")
```
```



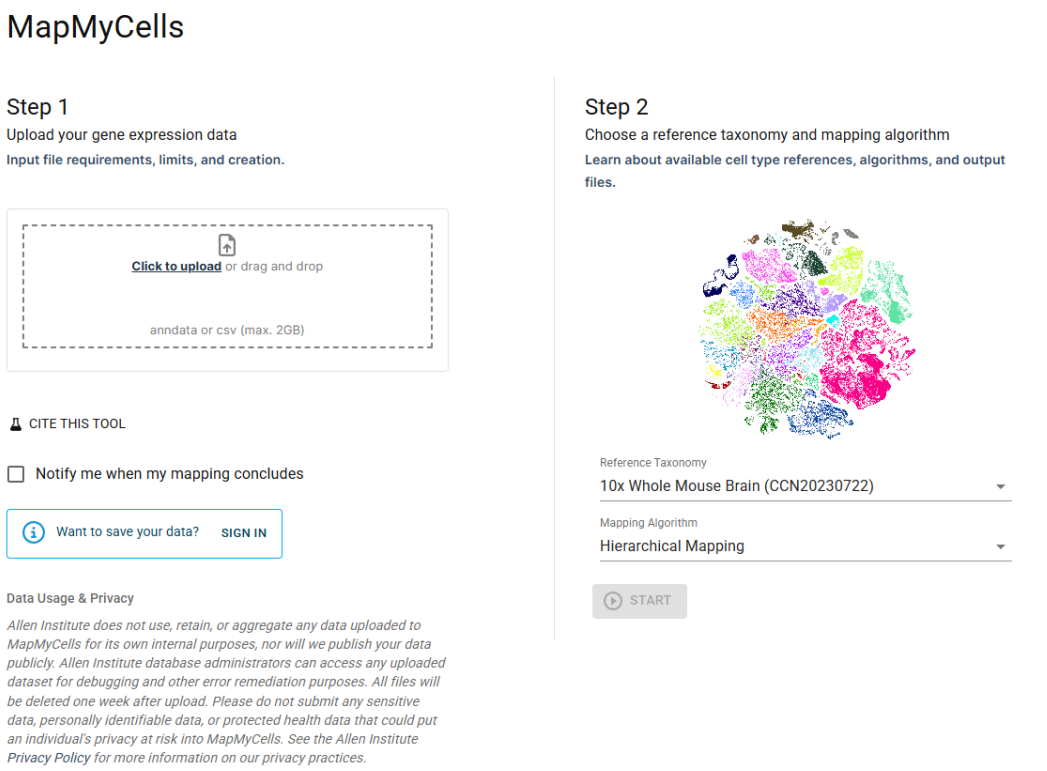


Part 8: Find markers

```r
```{r, fig.height = 7, fig.width = 14}
# find all markers of cluster 5
Idents(Seurat_AD_WT) <- "seurat_clusters"
cluster13.markers <- FindMarkers(Seurat_AD_WT, ident.1 = "13", only.pos = TRUE)
head(cluster13.markers, n = 10)

Idents(Seurat_AD_WT) <- "seurat_clusters"
cluster13.conserved <- FindConservedMarkers(Seurat_AD_WT, ident.1 = "13", grouping.var = "condition", verbose = FALSE)

# Find markers across conditions
Seurat_AD_WT$cluster_cond <- paste(Seurat_AD_WT$seurat_clusters,
                                   Seurat_AD_WT$condition, sep = "_")

Idents(Seurat_AD_WT) <- "cluster_cond"
cluster13.markers.cond <- FindMarkers(Seurat_AD_WT, ident.1 = "13_WT", ident.2 = "13_AD")

```

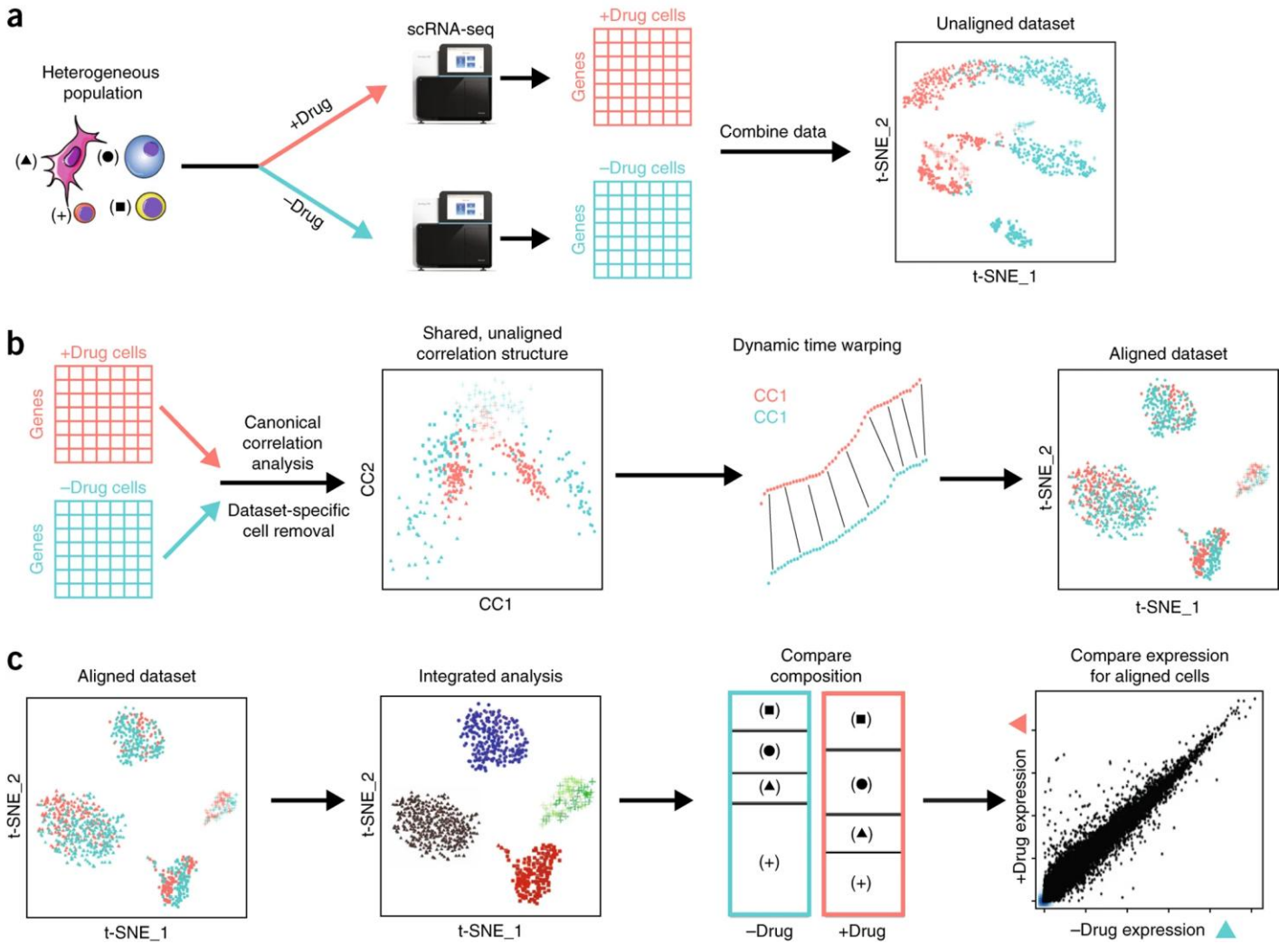
Plot the markers
```{r, fig.height = 7, fig.width = 14}
cluster13.markers %>%
    dplyr::filter(avg_log2FC > 2 & p_val < 0.05) %>%
    slice_max(avg_log2FC, n = 10) -> top10

Idents(Seurat_AD_WT) <- "seurat_clusters"
DotPlot(Seurat_AD_WT, features = rownames(top10))
DotPlot(Seurat_AD_WT, features = rownames(top10), group.by = "condition")
```
```

Markers

	p_val	avg_log2FC	pct.1	pct.2	p_val_adj
Cpne7	3.795059e-113	1.6148848	0.939	0.440	1.062541e-108
Pex5l	6.549430e-80	1.3841500	0.867	0.414	1.833709e-75
Man1a	1.168222e-55	1.7415038	0.408	0.123	3.270787e-51
Grin2b	6.088627e-55	0.6456170	1.000	0.913	1.704694e-50
Kcnn2	1.985708e-46	1.3211142	0.589	0.252	5.559586e-42
3110035E14Rik	1.610460e-42	1.1141428	0.614	0.279	4.508965e-38
Ntm	1.879055e-40	0.7167477	0.956	0.618	5.260977e-36
Chrd	8.107074e-40	1.0820214	0.683	0.354	2.269819e-35
Fam19a1	3.278474e-37	0.6712845	0.917	0.568	9.179072e-33
Fibcd1	4.322958e-37	1.6778961	0.267	0.076	1.210342e-32
Zeb2	6.237067e-33	0.7754559	0.853	0.579	1.746254e-28
Arhgef28	8.172101e-31	1.3445675	0.294	0.099	2.288025e-26
Mpped1	4.221407e-29	1.0897637	0.481	0.220	1.181910e-24
Brd9	7.888922e-28	0.7317513	0.836	0.619	2.208740e-23
Cadps2	6.051534e-27	1.3831535	0.256	0.086	1.694308e-22
Atp2b1	1.206030e-26	0.5222782	0.939	0.733	3.376643e-22
Grm5	2.330329e-26	0.5810710	0.939	0.721	6.524456e-22

Conserved Markers

	WT_p_val	WT_avg_log2FC	WT_pct.1	WT_pct.2	WT_p_val_adj	AD_p_val	AD_avg_log2FC	AD_pct.1	AD_pct.2	AD_p_val_adj	max_pval	minimump_p_val
Cpne7	1.896286e-53	1.5667888	0.921	0.475	5.309222e-49	3.859953e-62	1.6734406	0.959	0.406	1.080710e-57	1.896286e-53	7.719906e-62
Man1a	4.180960e-17	1.4020101	0.372	0.149	1.170585e-12	7.693521e-47	2.1012876	0.450	0.097	2.154032e-42	4.180960e-17	1.538704e-46
Pex5l	2.917237e-40	1.4236271	0.859	0.435	8.167680e-36	2.128116e-41	1.3607309	0.876	0.395	5.958298e-37	2.917237e-40	4.256231e-41
Grin2b	7.679557e-37	0.7202418	1.000	0.929	2.150122e-32	1.357831e-20	0.5502120	1.000	0.896	3.801656e-16	1.357831e-20	1.535911e-36
Kcnn2	1.643295e-19	1.3106818	0.571	0.288	4.600898e-15	6.643690e-30	1.3249140	0.609	0.216	1.860100e-25	1.643295e-19	1.328738e-29
Fam19a1	6.108650e-26	0.7810839	0.942	0.578	1.710300e-21	6.415784e-14	0.5597053	0.888	0.557	1.796291e-09	6.415784e-14	1.221730e-25
Arhgef28	8.345182e-10	1.1449230	0.262	0.114	2.336484e-05	1.180007e-25	1.5509784	0.331	0.085	3.303784e-21	8.345182e-10	2.360014e-25
Ntm	6.855173e-18	0.6564655	0.953	0.702	1.919311e-13	2.629994e-25	0.7718903	0.959	0.536	7.363456e-21	6.855173e-18	5.259987e-25
3110035E14Rik	7.409298e-20	0.9887040	0.660	0.348	2.074455e-15	3.045912e-25	1.2567730	0.562	0.212	8.527944e-21	7.409298e-20	6.091824e-25
Chrd	1.333034e-18	1.0821957	0.660	0.375	3.732227e-14	4.238994e-23	1.0950006	0.710	0.334	1.186834e-18	1.333034e-18	8.477988e-23
Zeb2	4.148900e-21	0.8652456	0.853	0.613	1.161609e-16	1.146133e-13	0.6508298	0.852	0.546	3.208944e-09	1.146133e-13	8.297800e-21
Fibcd1	1.303896e-18	1.7561711	0.277	0.088	3.650649e-14	4.053777e-20	1.5887512	0.254	0.065	1.134977e-15	1.303896e-18	8.107554e-20
Col5a2	4.191867e-10	2.1711345	0.110	0.029	1.173639e-05	1.079825e-19	2.5414465	0.107	0.014	3.023293e-15	4.191867e-10	2.159649e-19
Dcn	3.779572e-05	1.1004743	0.141	0.066	1.000000e+00	1.467456e-19	1.2494377	0.243	0.061	4.108582e-15	3.779572e-05	2.934911e-19
Brd9	1.091491e-10	0.5859223	0.806	0.663	3.055958e-06	1.828784e-19	0.8828290	0.870	0.575	5.120229e-15	1.091491e-10	3.657568e-19

Condition Markers

	p_val	avg_log2FC	pct.1	pct.2	p_val_adj
Apoe	5.241195e-46	8.1486988	0.780	0.006	1.467430e-41
Ttr	1.134299e-22	6.7309510	0.455	0.006	3.175811e-18
Rnf121	3.780637e-19	5.6537259	0.424	0.024	1.058503e-14
Gm26917	3.103626e-16	1.1466041	0.880	0.633	8.689531e-12
Gm38039	1.649691e-15	4.2406638	0.356	0.018	4.618805e-11
Eif2s3y	4.668433e-10	1.0712380	0.754	0.479	1.307068e-05
PISD	3.730705e-09	-0.7947013	0.796	0.917	1.044523e-04
Galnt1	1.822043e-06	-0.6015449	0.670	0.846	5.101357e-02
Slc24a5	3.015296e-06	-0.7756558	0.592	0.746	8.442226e-02
Timp4	8.124086e-06	-1.3849407	0.199	0.402	2.274582e-01
Sv2b	8.632651e-06	-0.9854270	0.351	0.550	2.416970e-01
mt-Nd4	8.915913e-06	0.7297622	0.728	0.580	2.496277e-01
AY036118	9.411056e-06	1.9050876	0.251	0.077	2.634907e-01
Ybx1	1.496202e-05	-1.1925602	0.220	0.414	4.189067e-01
Mbp	2.233738e-05	1.2708685	0.393	0.183	6.254018e-01
Ptcra	2.991650e-05	-1.4398573	0.110	0.278	8.376023e-01
Gm14966	4.146767e-05	3.9741931	0.110	0.006	1.000000e+00
Fam160b2	4.180642e-05	2.2633548	0.178	0.041	1.000000e+00
Bcas3	4.746139e-05	0.9550465	0.450	0.237	1.000000e+00

Part 8: Find markers





MapMyCells

```
# Export data to MapMyCells
We will export the count matrix and map the cells with Allen Brain atlas
see: https://portal.brain-map.org/atlases-and-data/bkp/mapmycells
use: https://knowledge.brain-map.org/mapmycells/process/
## Extract the Count Matrix

```{r,fig.height = 7, fig.width = 10}
mat <- GetAssayData(object = subset(Seurat_AD_WT, downsample = 1000, subset = seurat_clusters %in% c(13)), assay = "RNA", slot = "counts")
# we need to transpose the matrix, as MapMyCells expects matrix in which rows are "cells" and columns are genes!
mat_t <- t(mat)
write.csv(mat_t, "./output/count_matrix_cluster13.csv")
```
```

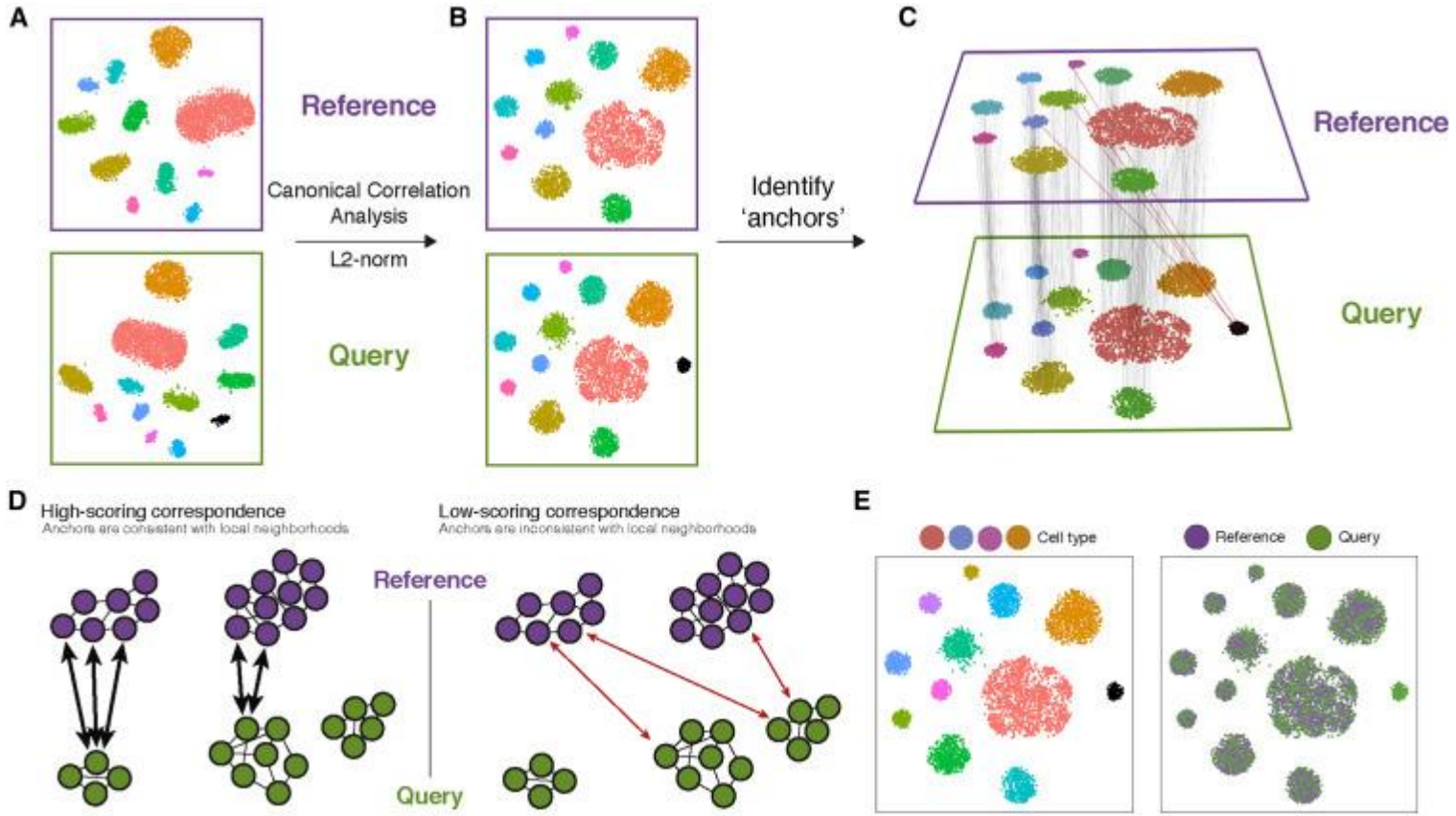


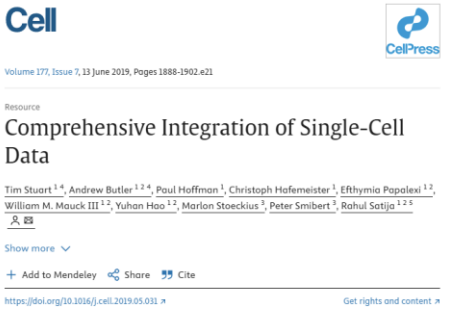
Thank you for the attention

CCA (canonical correlation analysis)



RPCA (reciprocal PCA)




Cell

CellPress

Volume 177, Issue 7, 13 June 2019, Pages 1888-1902.e21

Resource

Comprehensive Integration of Single-Cell Data

Tim Stuart [1,4], Andrew Butler [1,2,4], Paul Hoffman [1], Christoph Hafemeister [1], Efthymia Papalexi [1,2], William M. Mauck III [1,2], Yuhan Hao [1,2], Marlon Stoeckius [3], Peter Smibert [3], Rahul Satija [1,2,5]

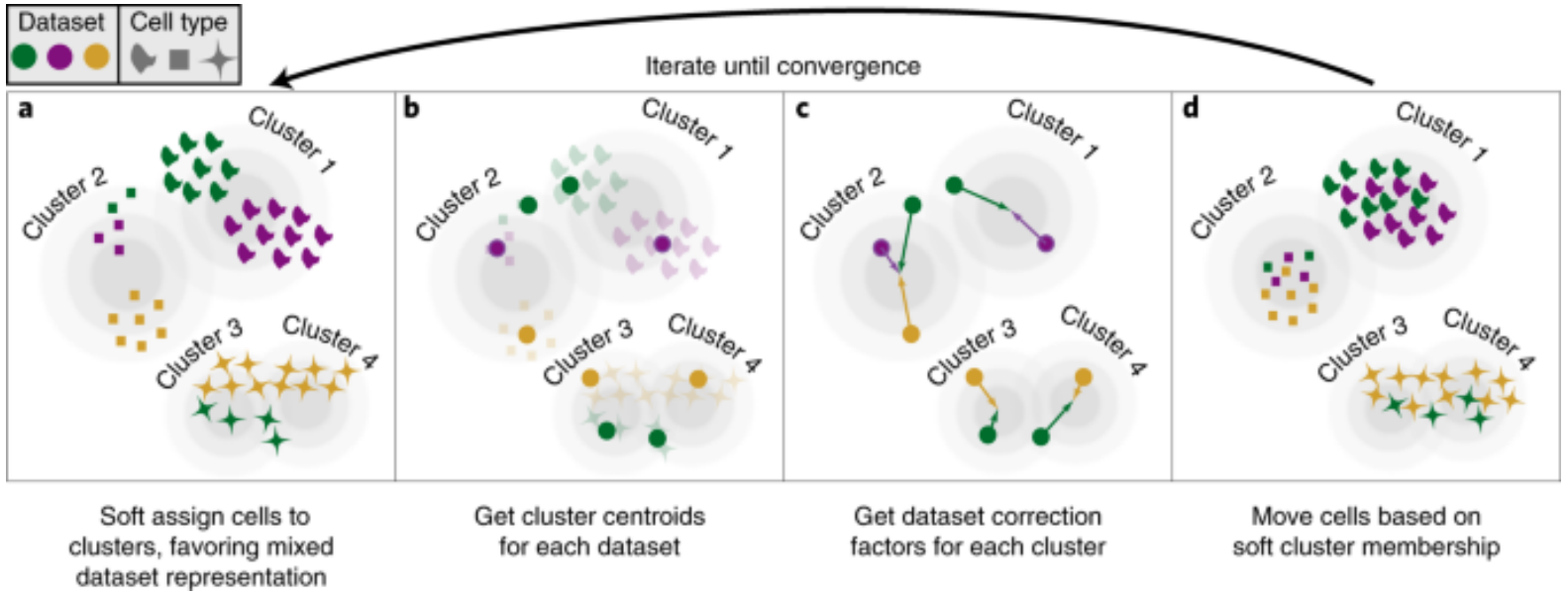
Show more

Add to Mendeley Share Cite

https://doi.org/10.1016/j.cell.2019.05.031

Get rights and content

Harmony



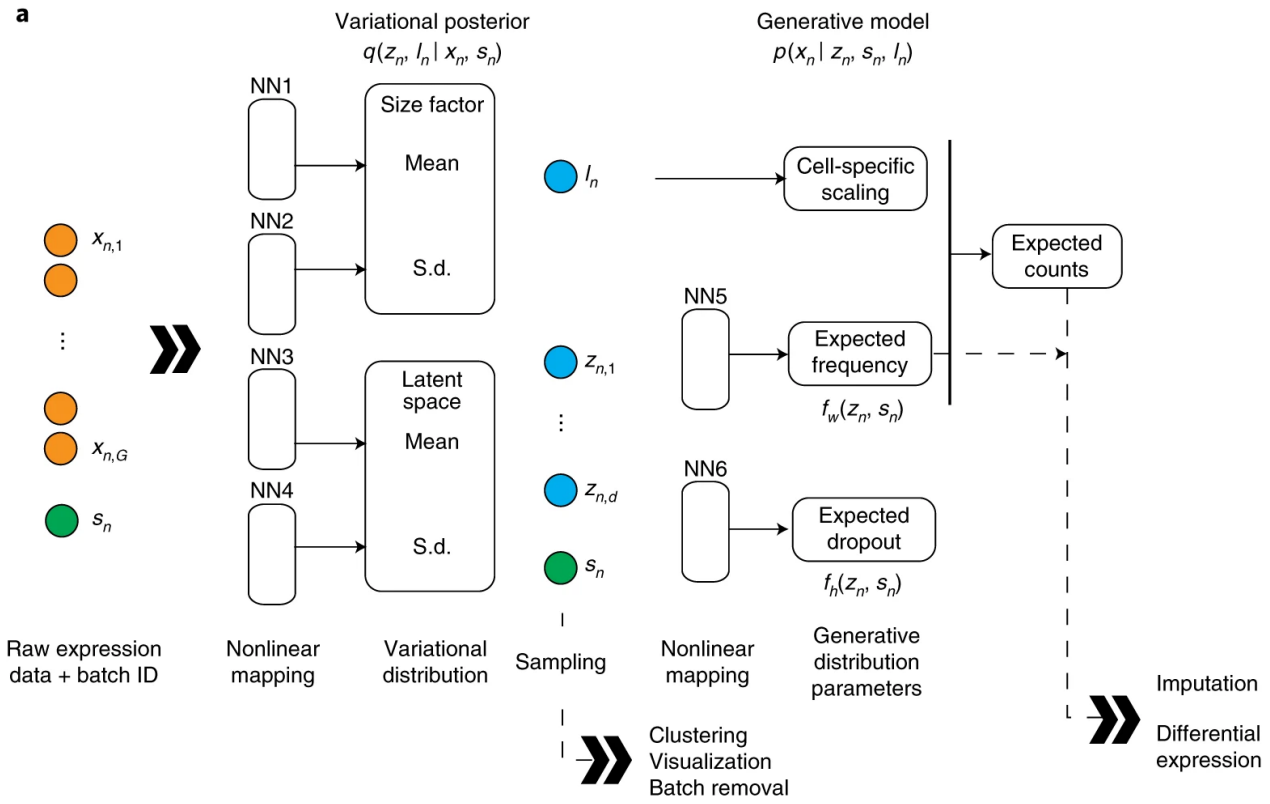
Article | Published: 18 November 2019

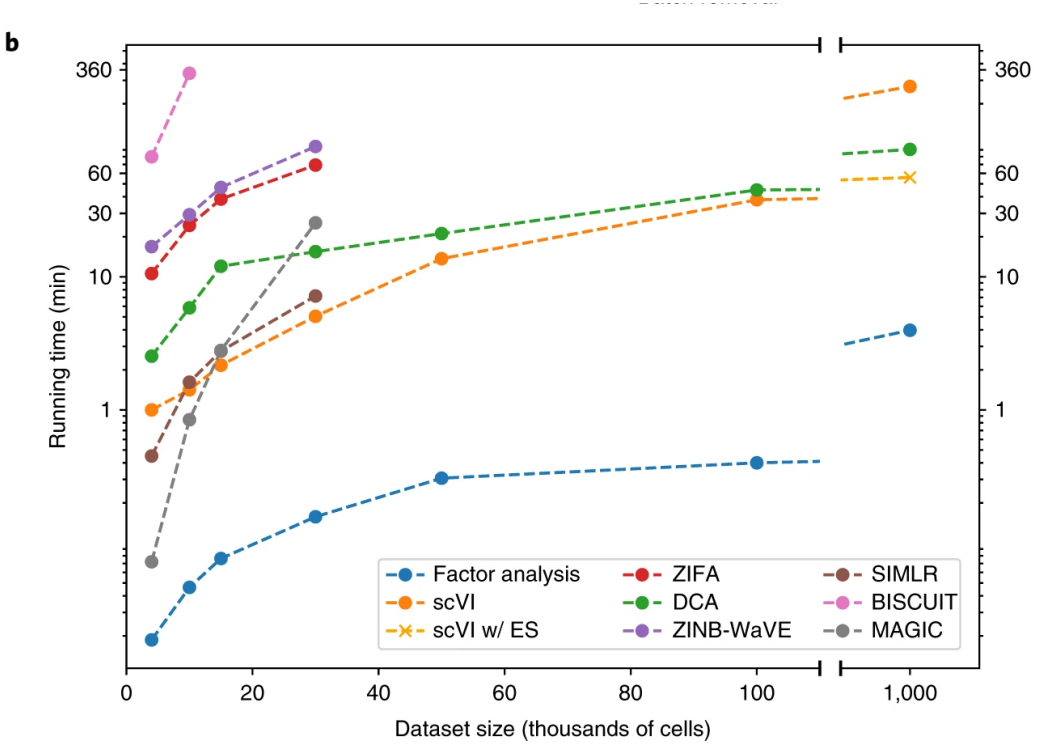
Fast, sensitive and accurate integration of single-cell data with Harmony

Ilya Korsunsky, Nghia Millard, Jean Fan, Kamil Slowikowski, Fan Zhang, Kevin Wei, Yuriy Baglaenko, Michael Brenner, Po-ru Loh & Soumya Raychaudhuri

Nature Methods **16**, 1289–1296 (2019) | Cite this article

scVI (single-cell variational inference) neural networks model





Article | Published: 30 November 2018

Deep generative modeling for single-cell transcriptomics

Romain Lopez, Jeffrey Regier, Michael B. Cole, Michael I. Jordan & Nir Yosef

Nature Methods **15**, 1053–1058 (2018) | Cite this article

117k Accesses | **2133** Citations | **186** Altmetric | Metrics