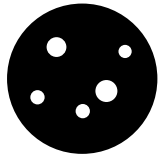
Single-cell RNA-seq workshop: tutorial

Raymond Louie, PhD
Sara Ballouz, PhD

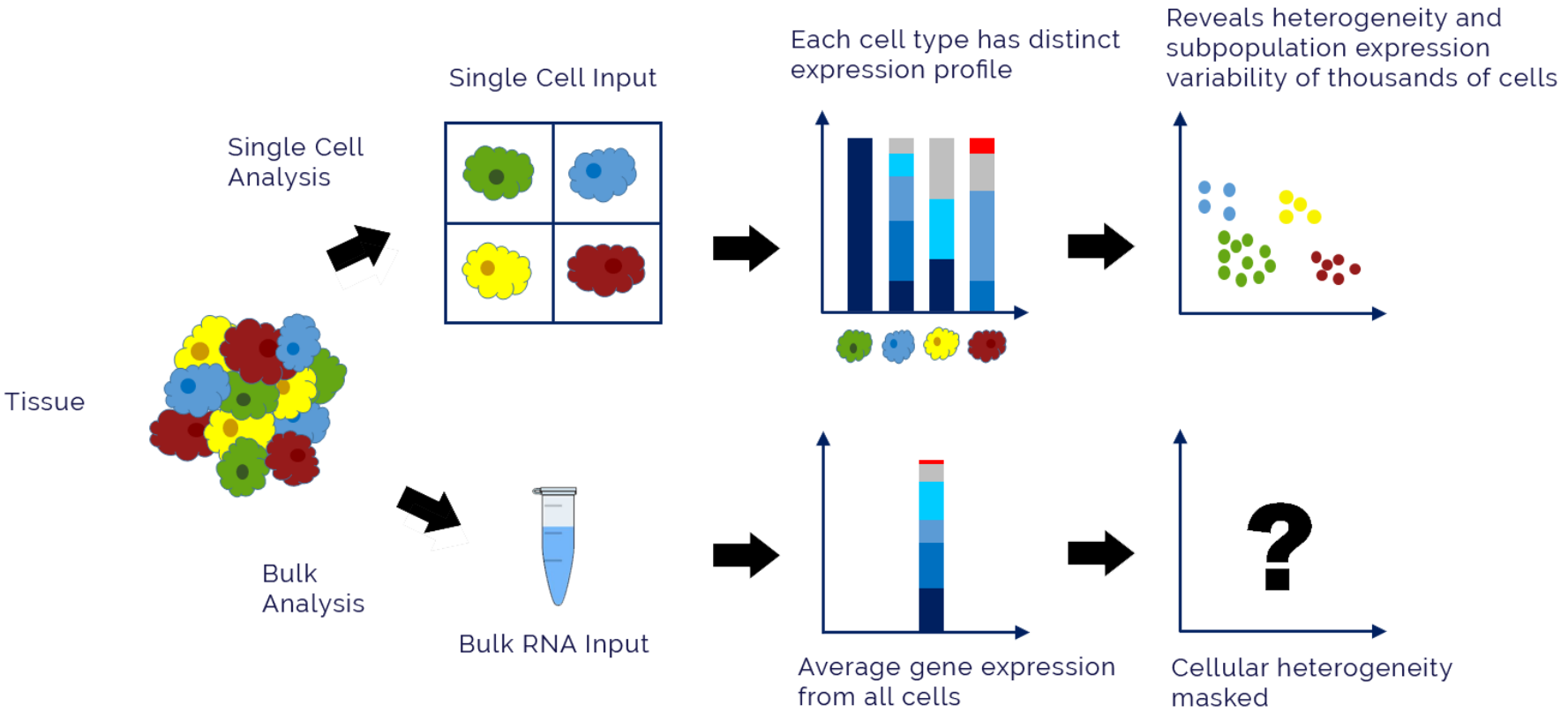
Overview of the Single Cell tutorial

1. Introduction to single cell
2. Setting up Rstudio, data and count matrix
3. Pre-processing
4. Application 1: Cell annotation
5. Application 2: Case vs Control

Overview of the Single Cell tutorial

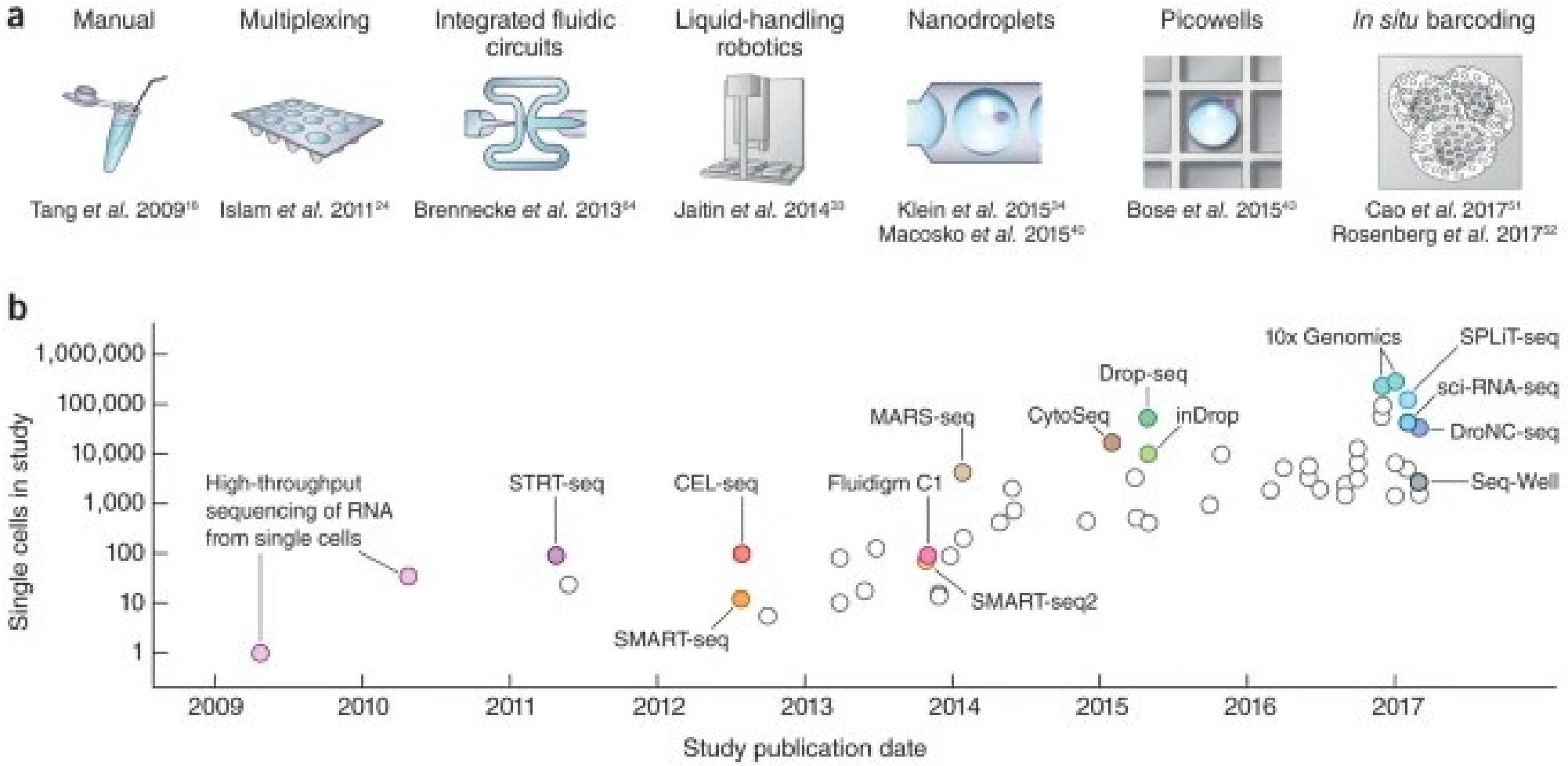
1. **Introduction to single cell** 
2. Setting up Rstudio, data and count matrix
3. Pre-processing
4. Application 1: Cell annotation
5. Application 2: Case vs Control

Overview of single cell



https://www.10xgenomics.com/blog/single-cell-rna-seq-an-introductory-overview-and-tools-for-getting-started

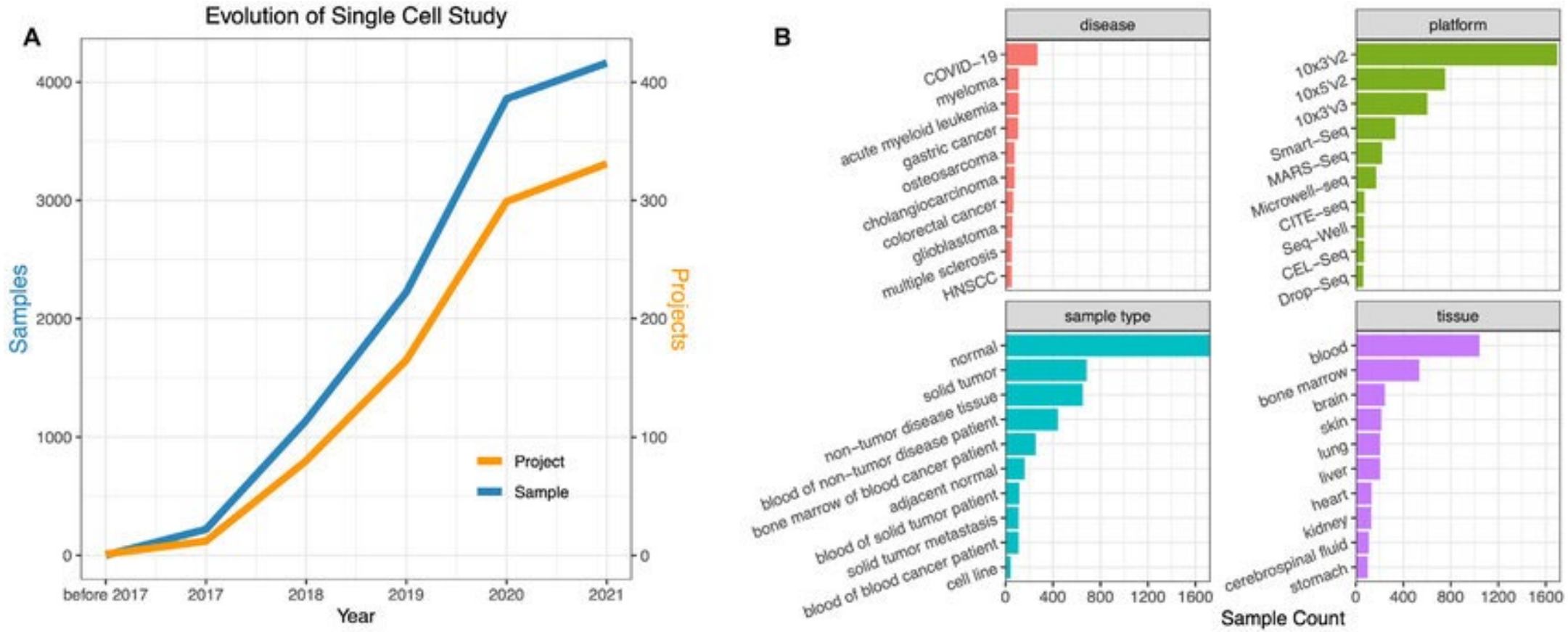
Exponential increase in the number of cells



https://www.nature.com/articles/nprot.2017.149

UNSW SYDNEY

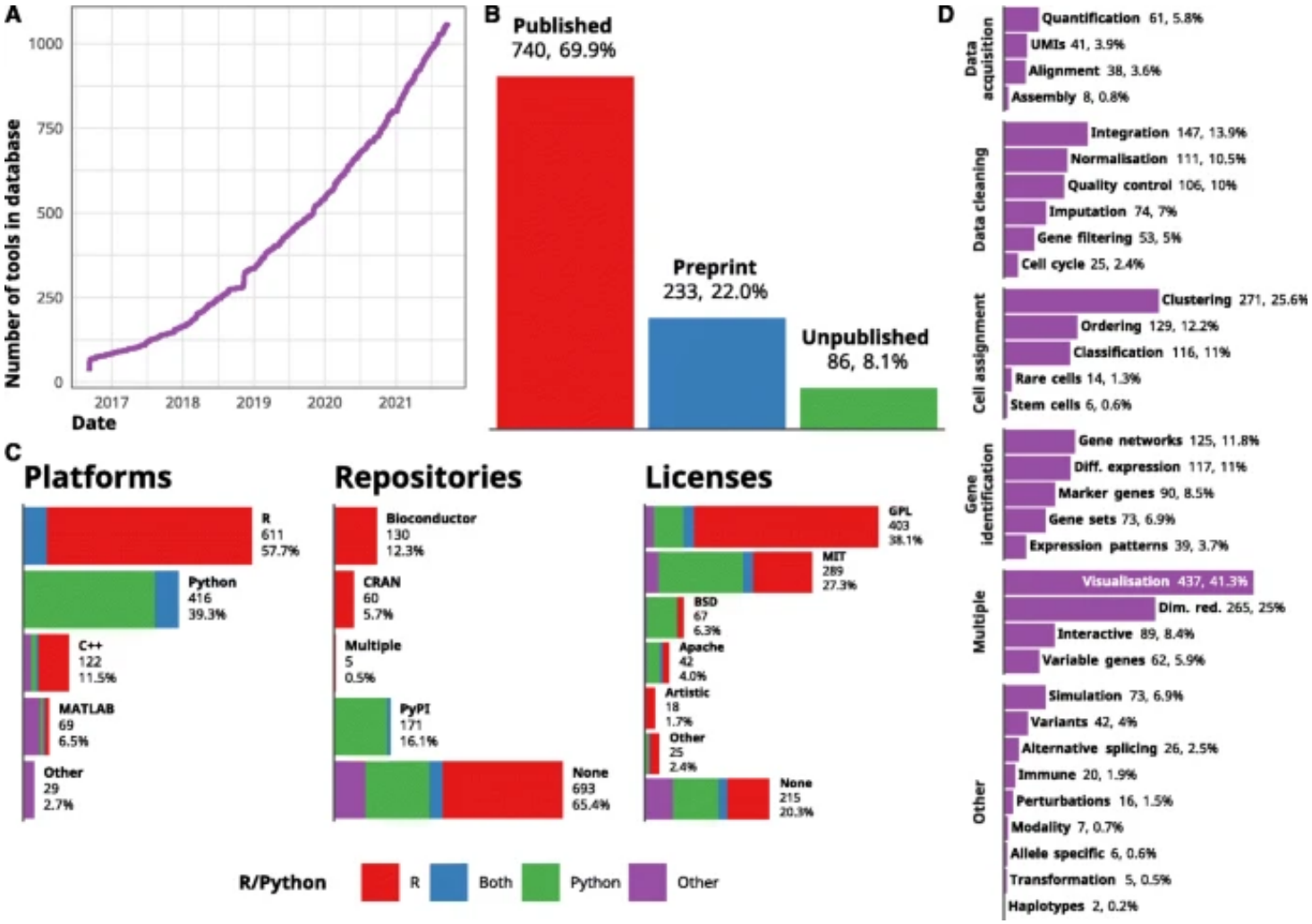
Increase in single-cell samples (2022)



https://academic.oup.com/nar/article/50/D1/D596/6430491

UNSW SYDNEY

Increase in single-cell tools (2021)



https://genomebiology.biomedcentral.com/articles/10.1186/s13059-021-02519-4

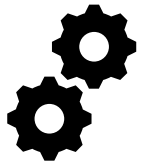
UNSW SYDNEY

Data access: some resources

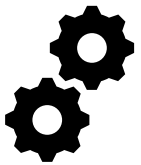
- 10X genomics:
  - https://www.10xgenomics.com/resources/datasets
- Human Cell Atlas:
  - https://data.humancellatlas.org/
- Cellxgene:
  - https://cellxgene.cziscience.com/datasets
- Single cell expression atlas
  - https://www.ebi.ac.uk/gxa/sc/home
- UCSC cell browser
  - https://cells.ucsc.edu/

Overview

1. Introduction to single cell
2. **Setting up Rstudio, data and count matrix**
3. Pre-processing
4. Application 1: Cell annotation
5. Application 2: Case vs Control



Accessing RStudio and setting up

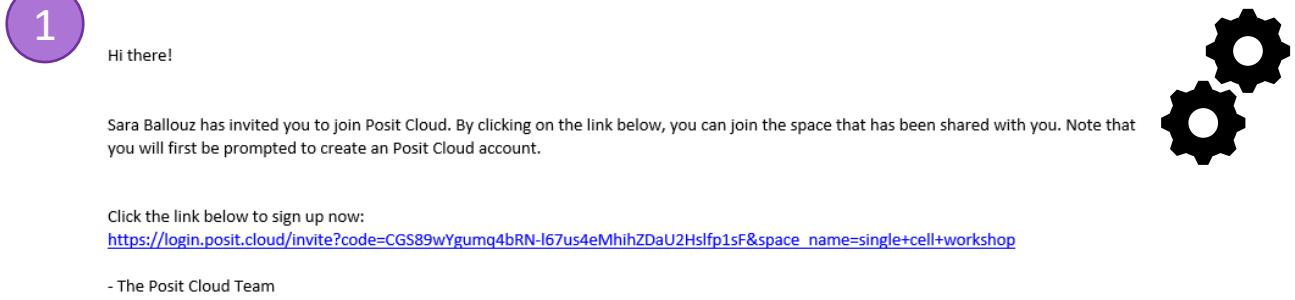


- For this workshop, we will be using **RStudio** on their cloud platform (now called posit https://posit.cloud/) through your **web browser**.
- Rstudio/posit is an IDE (integrated development environment) for **R** (and python).
- R is a **statistical programming language** that we will be using today.
- This link will only be available for a month after this workshop. However, feel free to download and export all the files you need!

If you want to repeat this again at work/home, we will provide a link at the end to download and install Rstudio on your own, along with relevant packages, data and scripts/code.

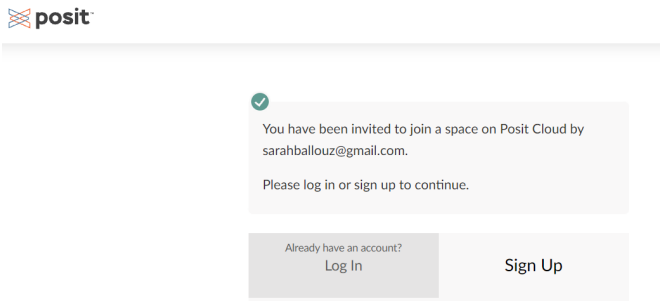
Open the link sent to your email. **If you did not receive this, let us know!**

1



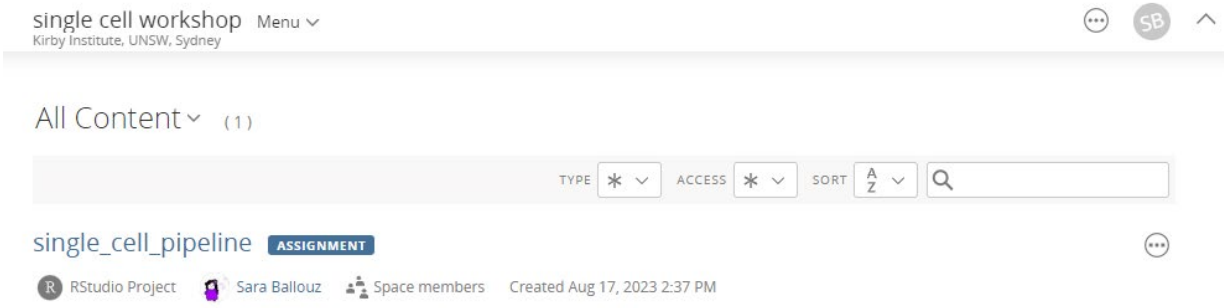
Sign up if you do not already have an account, otherwise log in.

2



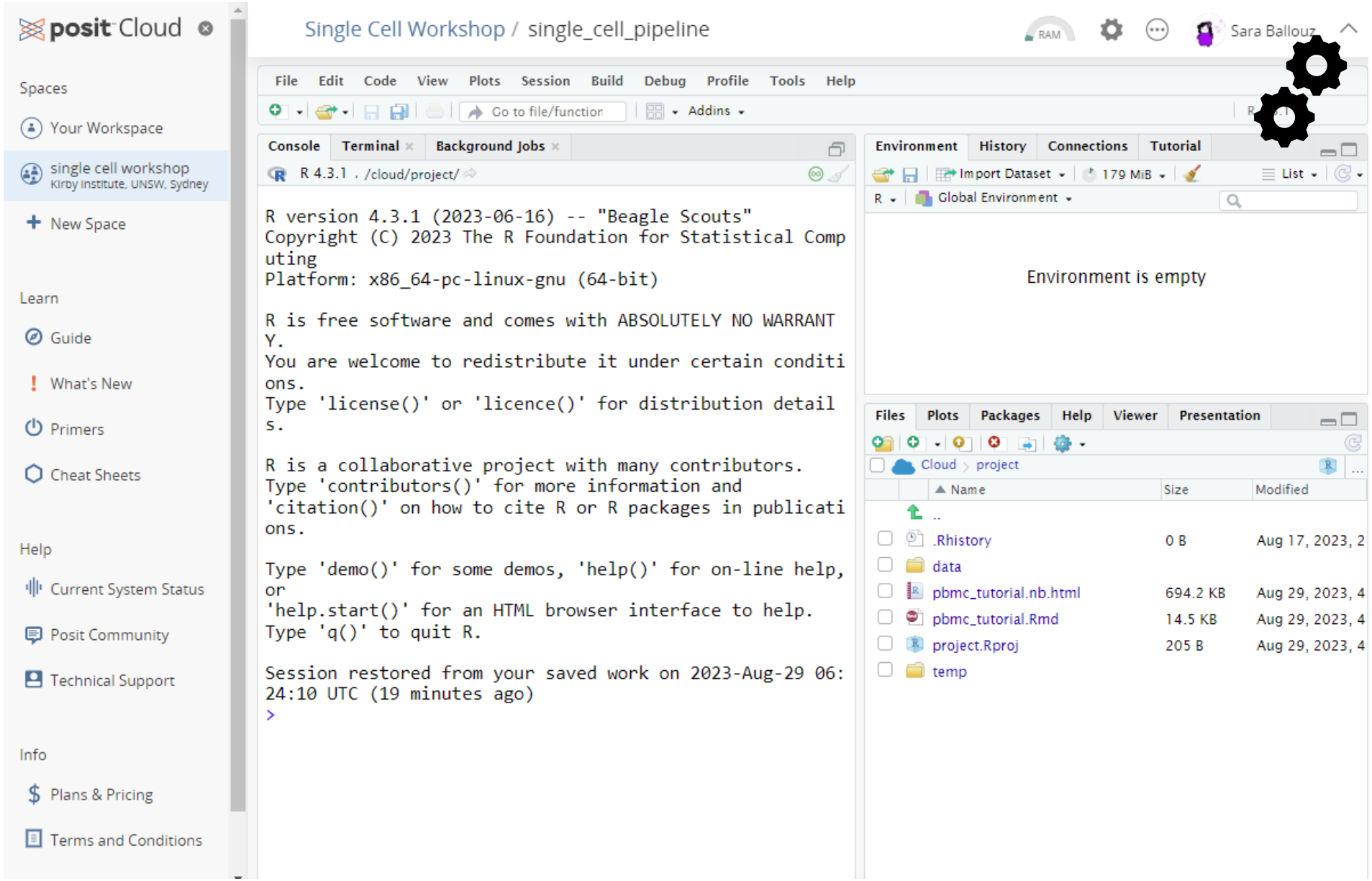
Once you've successfully logged in, you should be able to view the single_cell_pipeline "assignment" in the "single cell workshop" space

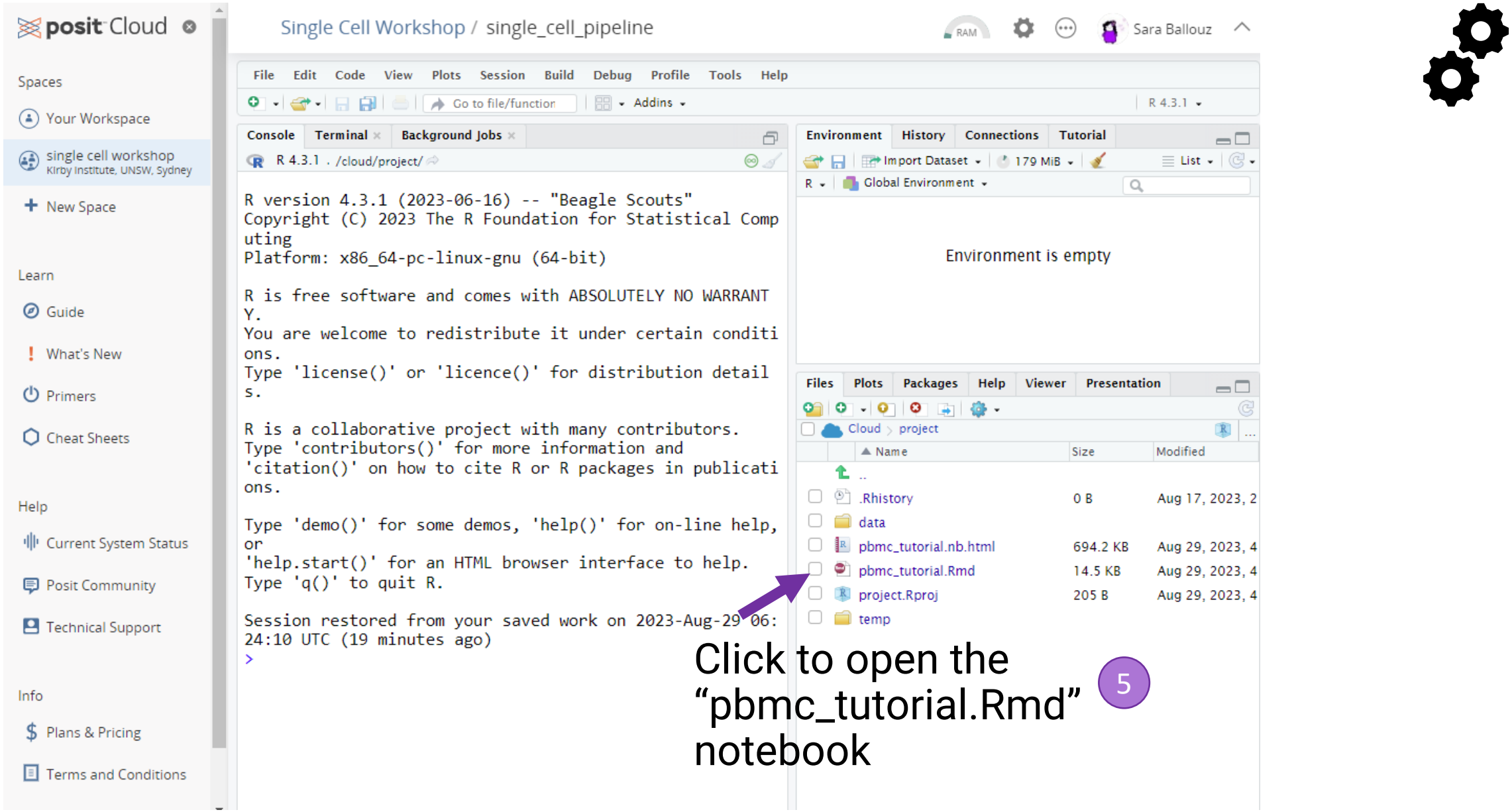
3



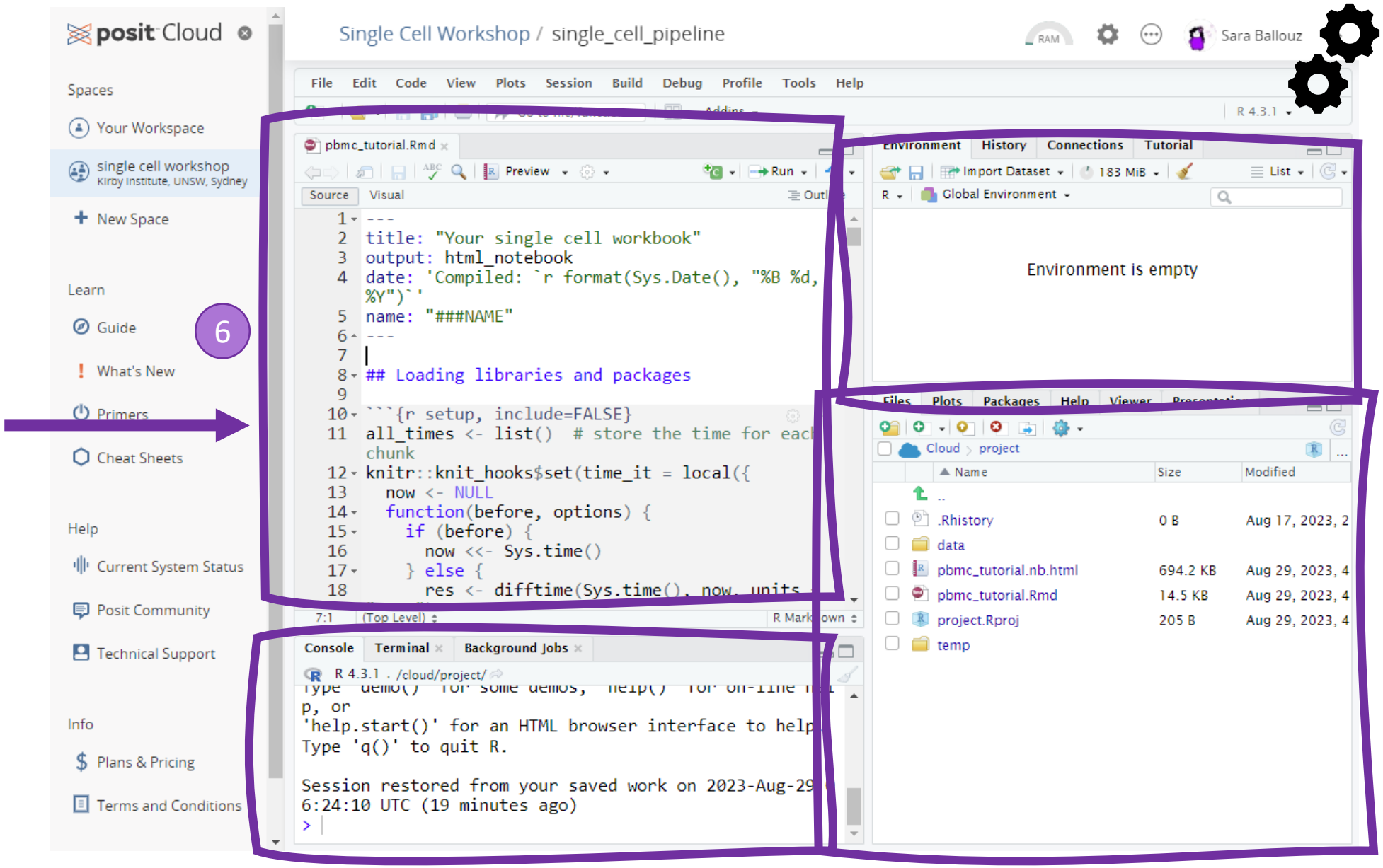
4

Selecting the “single_cell_pipeline” should open up a project and the Rstudio environment




posit Cloud
Spaces
Your Workspace
single cell workshop
Kirby Institute, UNSW, Sydney
New Space
Learn
Guide
What's New
Primers
Cheat Sheets
Help
Current System Status
Posit Community
Technical Support
Info
Plans & Pricing
Terms and Conditions
Single Cell Workshop / single_cell_pipeline
RAM
Sara Ballouz
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
R 4.3.1
Console
Terminal
Background Jobs
R 4.3.1 . /cloud/project/
R version 4.3.1 (2023-06-16) -- "Beagle Scouts"
Copyright (C) 2023 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
Session restored from your saved work on 2023-Aug-29 06:24:10 UTC (19 minutes ago)
>
Environment
History
Connections
Tutorial
Import Dataset
179 MiB
List
R
Global Environment
Environment is empty
Files
Plots
Packages
Help
Viewer
Presentation
Cloud
project
Name
Size
Modified
..
.Rhistory 0 B Aug 17, 2023, 2
data
pbmc_tutorial.nb.html 694.2 KB Aug 29, 2023, 4
pbmc_tutorial.Rmd 14.5 KB Aug 29, 2023, 4
project.Rproj 205 B Aug 29, 2023, 4
temp
Click to open the “pbmc_tutorial.Rmd” notebook
5

Notice there are four windows. We will work mostly here



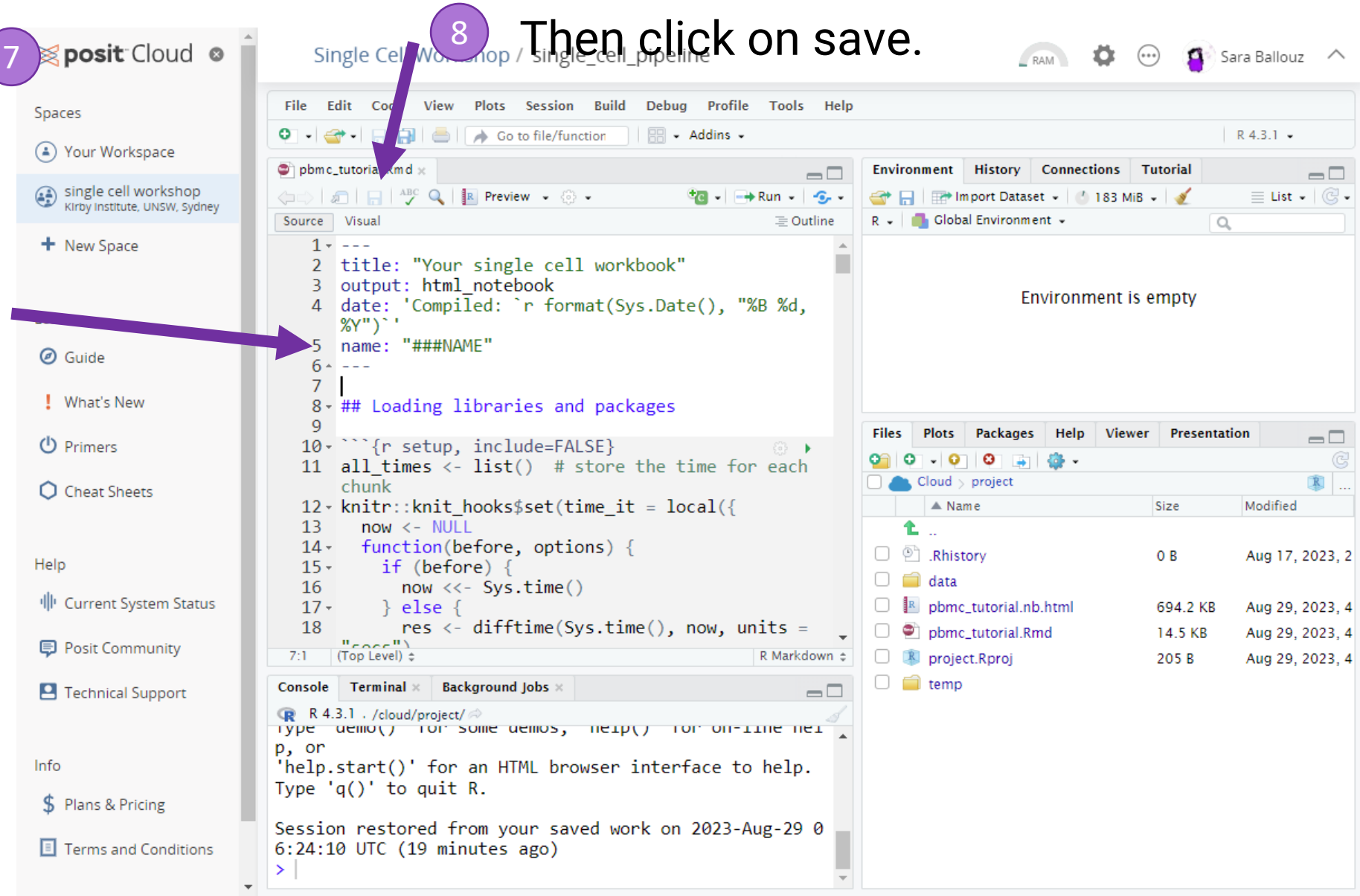
Edit the document by replacing ###NAME with your name.

Remember to keep the “ ”

7

8

Then click on save.



posit Cloud

Spaces

Your Workspace

single cell workshop
Kirby Institute, UNSW, Sydney

New Space

Learn

Guide

What's New

Primers

Cheat Sheets

Help

Current System Status

Posit Community

Technical Support

Info

Plans & Pricing

Terms and Conditions

Single Cell Workshop / single_cell_pipeline

RAM

Sara Ballouz

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins | R 4.3.1

pbmc_tutorial.Rmd

Preview | Run | Source | Visual | Outline

```
1  ---
2  title: "Your single cell workbook"
3  output: html_notebook
4  date: 'Compiled: `r format(Sys.Date(), "%B %d, %Y")`'
5  name: "Sara"
6  ---
7
8  ## Loading libraries and packages
9
10 ```{r setup, include=FALSE}
11 all_times <- list()  # store the time for each chunk
12 knitr::knit_hooks$set(time_it = local({
13   now <- NULL
14   function(before, options) {
15     if (before) {
16       now <<- Sys.time()
17     } else {
18       res <- difftime(Sys.time(), now, units = "secs")
```

5:12 | Your single cell workbook | R Markdown

Console | Terminal | Background Jobs

R 4.3.1 · /cloud/project/

```
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

Session restored from your saved work on 2023-Aug-29 06:24:10 UTC (19 minutes ago)
>
```

Environment | History | Connections | Tutorial

Import Dataset | 216 MiB | List

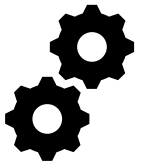
R | Global Environment

Environment is empty

Files | Plots | Packages | Help | Viewer | Presentation

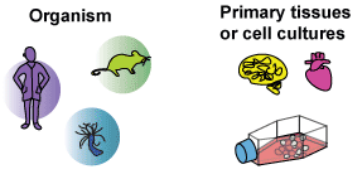
Cloud › project

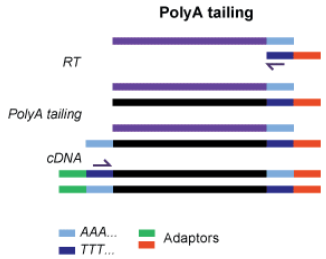
Name	Size	Modified
..		
.Rhistory	0 B	Aug 17, 2023, 2
data		
pbmc_tutorial.nb.html	694.2 KB	Aug 29, 2023, 4
pbmc_tutorial.Rmd	14.5 KB	Aug 29, 2023, 4
project.Rproj	205 B	Aug 29, 2023, 4
temp		

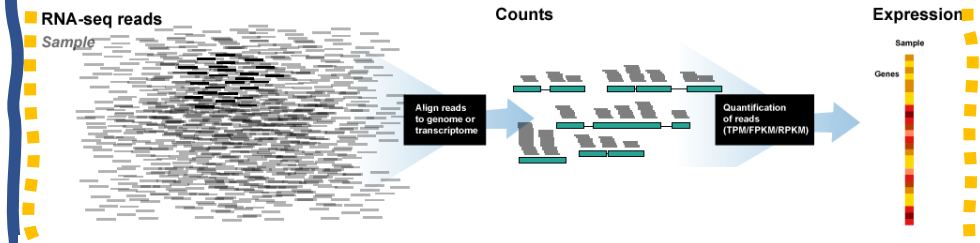


UNSW SYDNEY

Background: pre-count pipeline overview







Samples

Capture cells and isolate RNA

Library preparation

Fragment RNA into similar lengths, convert to cDNA, amplify

Sequence reads

Generate sequence for millions of fragments

QC + alignment + quantification

Match fragments against genome (mapping)
Count the number of reads per gene

“Wet”-lab component

“Dry”-lab component

UNSW SYDNEY

Obtaining the count matrix

10x GENOMICS

Run cellranger count

To generate single cell feature counts for a single library, run `cellranger count` with the following arguments. For a complete listing of the arguments accepted, see the Command Line Argument Reference below, or run `cellranger count --help`. **Cell Ranger must not be used for Single Cell Multiome Analysis.** For Single Cell Multiome ATAC + Gene Expression libraries, use Cell Ranger ARC.

Starting in Cell Ranger 7.0, the expected number of cells can either be auto-estimated or specified with `--expect-cells` (e.g., to replicate a previous analysis), see Gene Expression algorithm overview. If needed, automated cell calling can be overridden with the `--force-cells` option.

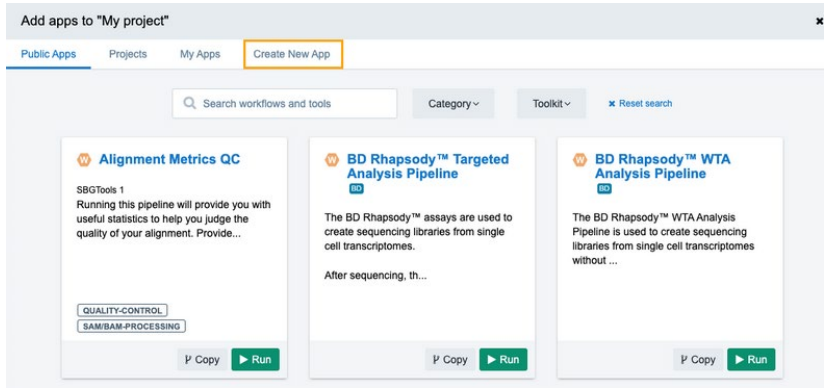
For help on which arguments to use to target a particular set of FASTQs, consult Specifying Input FASTQ Files for 10x Genomics pipelines.

After determining these input arguments and customizing the code in red, run `cellranger`:

```
$ cd /home/jdoe/runs
$ cellranger count --id=sample345 \
                   --transcriptome=/opt/refdata-gex-GRCh38-2020-A \
                   --fastqs=/home/jdoe/runs/HAWT7ADXX/outs/fastq_path \
                   --sample=mysample \
                   --localcores=8 \
                   --localmem=64
```

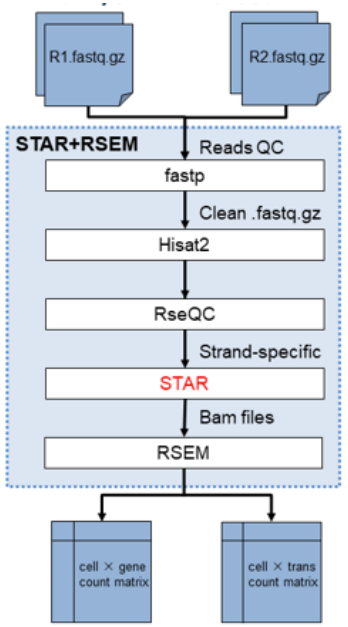
cellranger

BD Biosciences



Seven Bridges

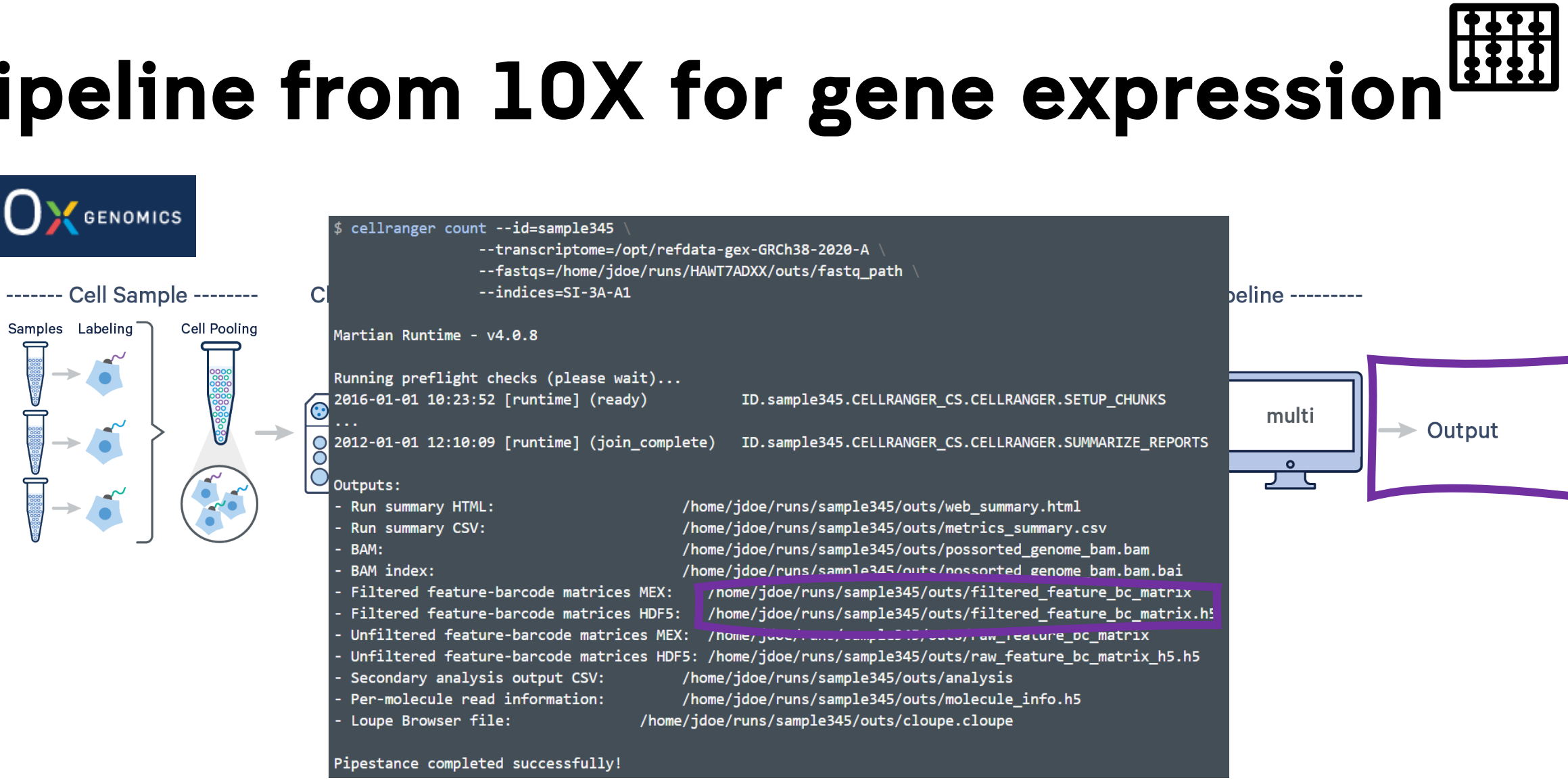
Smart-SEQ

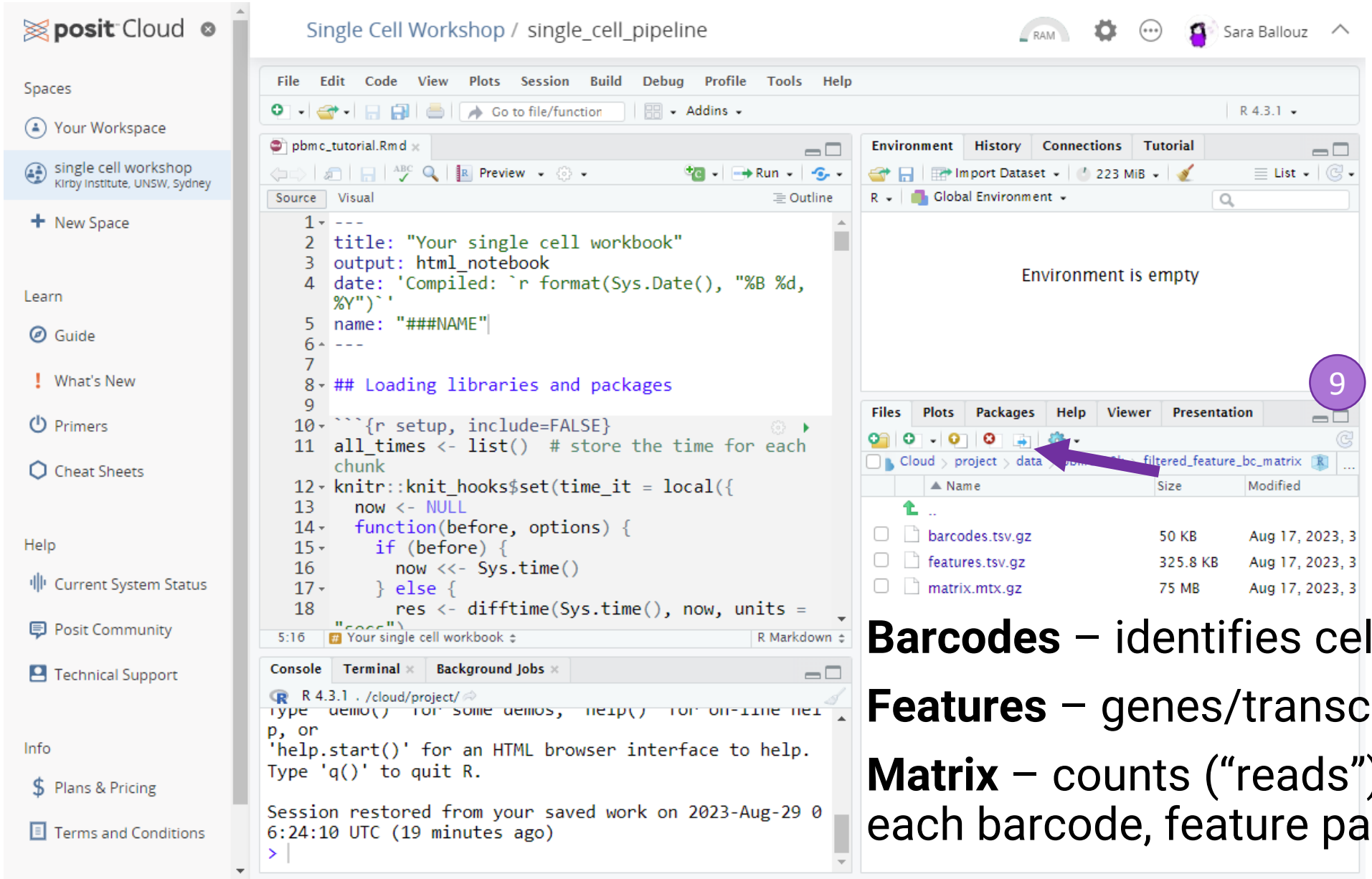


https://broadinstitute.github.io/warp/docs/Pipelines/Smart-seq2_Single_Sample_Pipeline/README/

UNSW SYDNEY

Pipeline from 10X for gene expression





Barcodes – identifies cells

Features – genes/transcripts

Matrix – counts (“reads”) for each barcode, feature pair

The gene expression count matrix

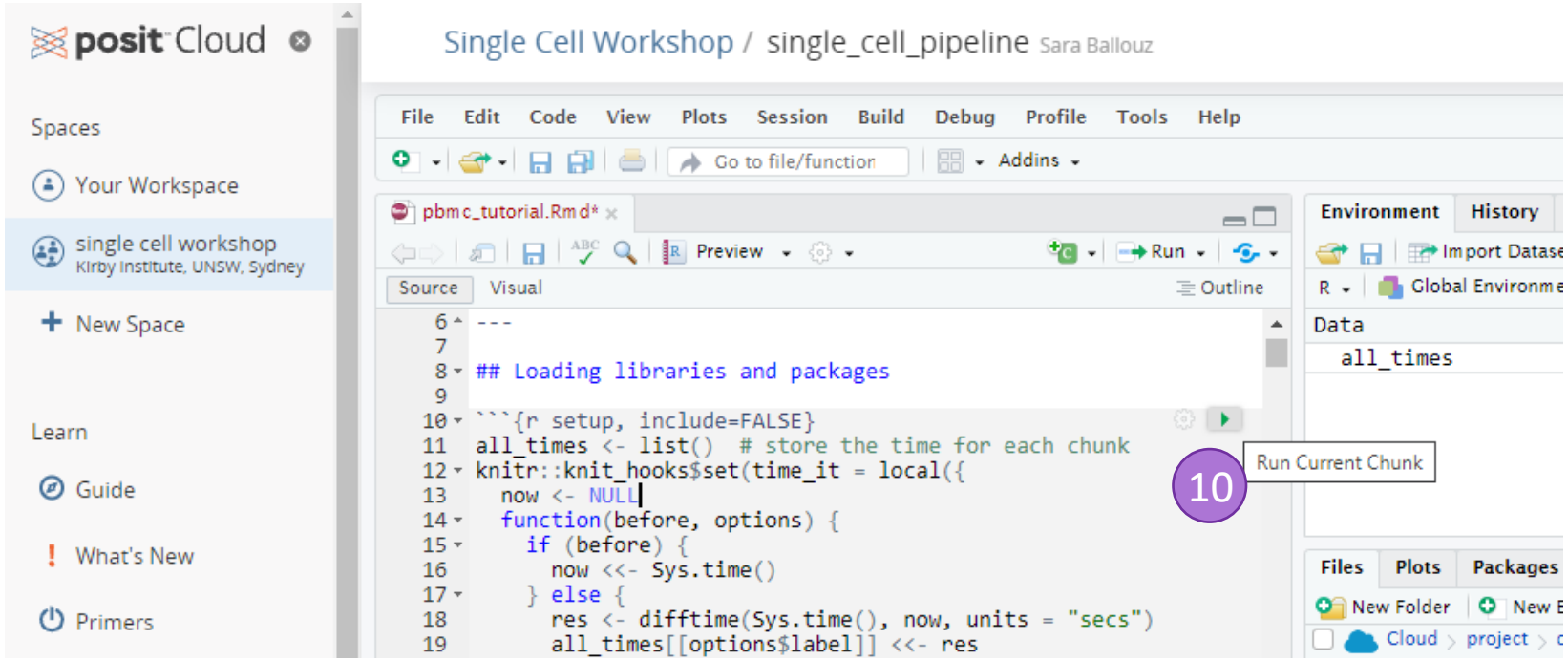
barcodes.tsv.gz

	Cell1	Cell2	...	CellN
Gene1	3	2	.	13
Gene2	2	3	.	1
Gene3	1	14	.	18
...	.	.	.	.
...	.	.	.	.
...	.	.	.	.
GeneM	25	0	.	0

features.tsv.gz

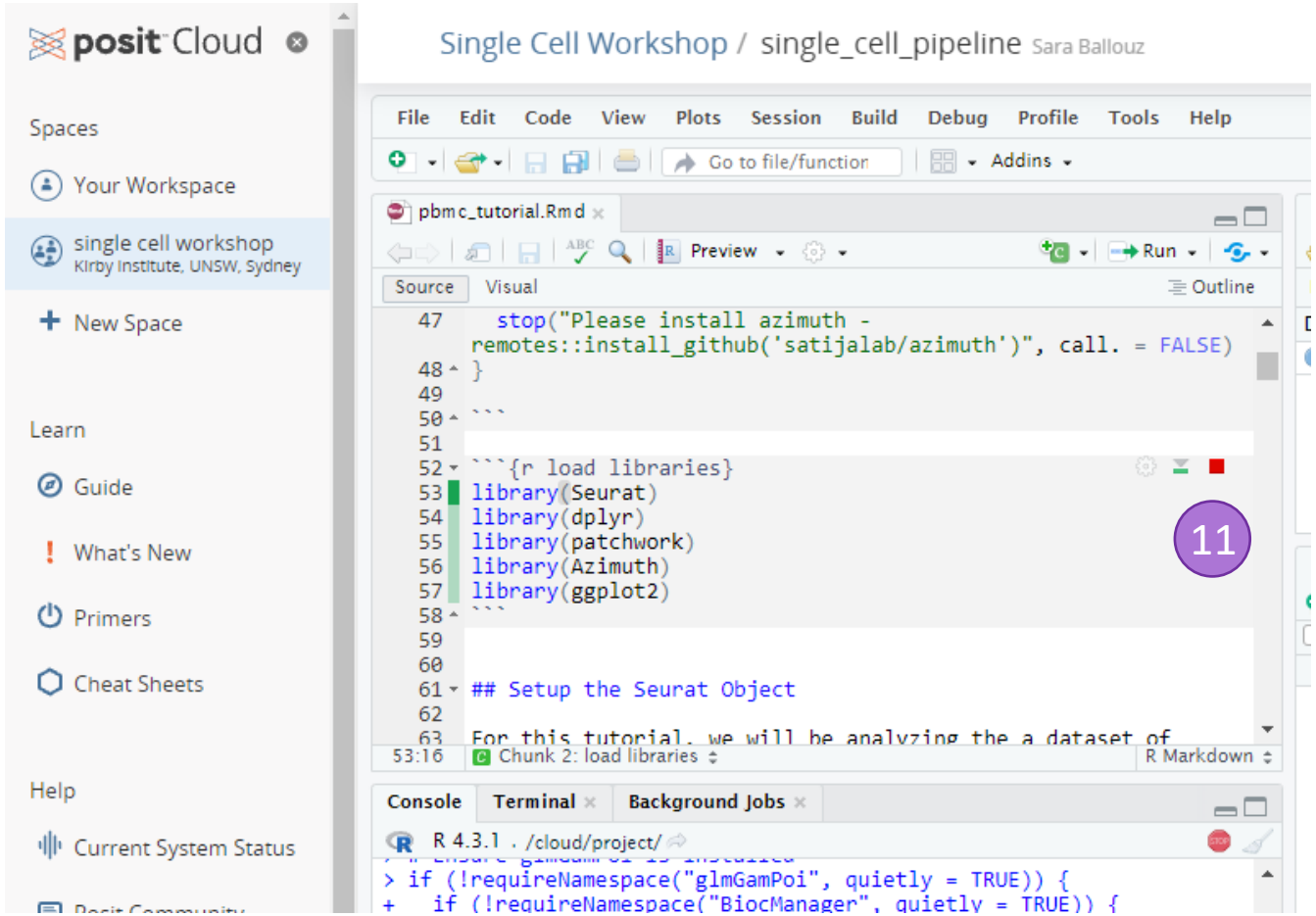
Matrix.mtx.gz

Setting up environment



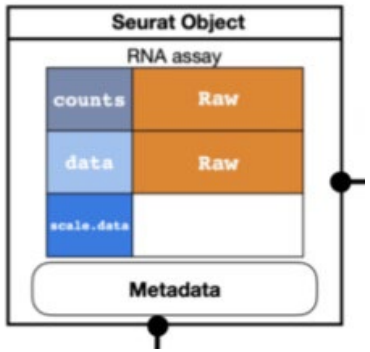
Run code (or "chunks") by clicking the arrow/play button

Loading in libraries



Load in libraries. These are the R packages that have all the functions we need to run our analyses.

What are Seurat objects?



The Seurat object is a **representation** of single-cell expression data for R; each Seurat object revolves around a set of cells and consists of one or more Assay objects, or individual representations of expression data (eg. RNA-seq, ATAC-seq, etc).

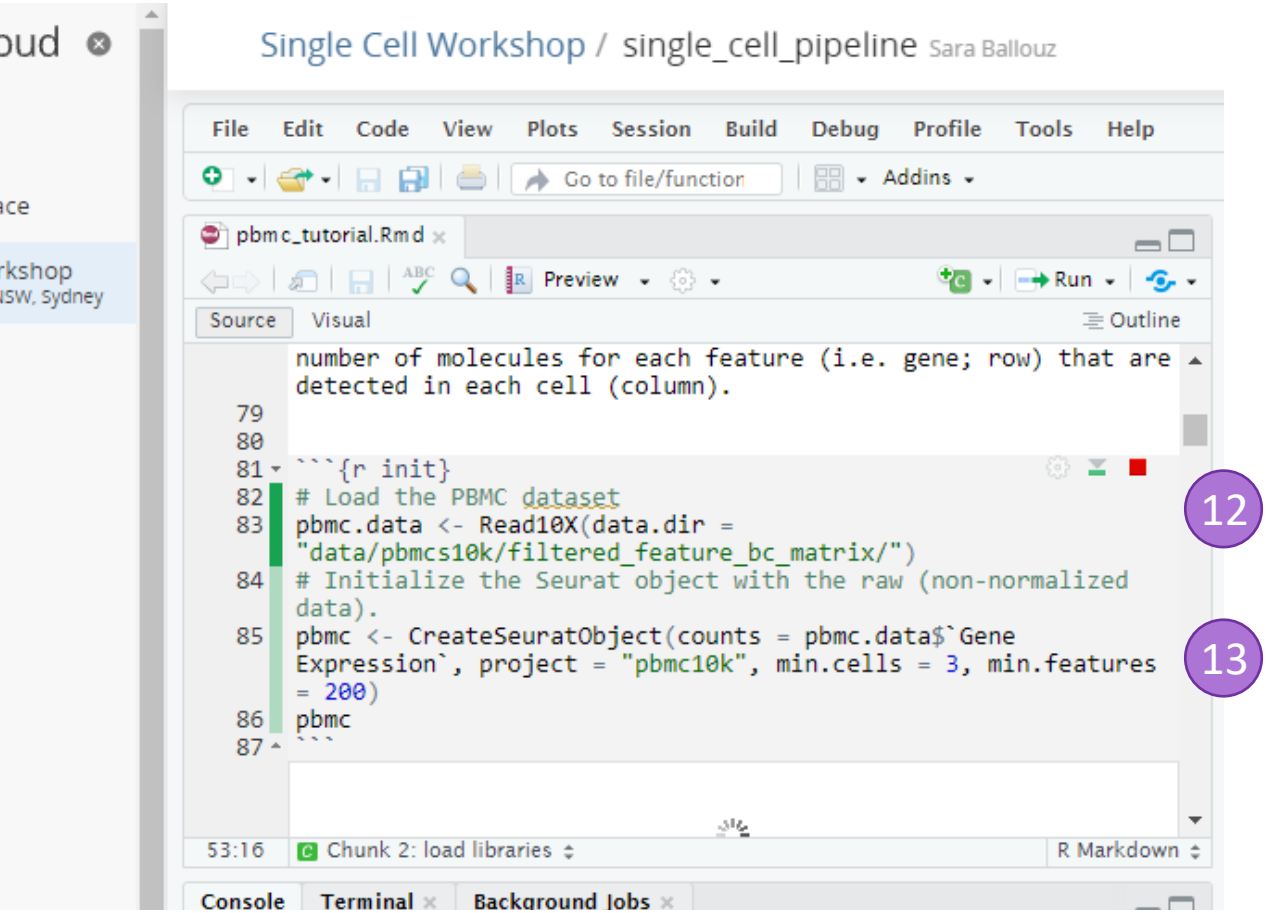
Think of it as a bookshelf, with books.

Each book is an assay.
e.g., it holds a book on the RNA-seq assay.

That book has a page on the raw counts data, one on metadata, etc.

https://search.r-project.org/CRAN/refmans/SeuratObject/html/Seurat-class.html

Setup the Seurat Object



Load in the data by running the “init” chunk. This runs the Read10X function. Note, we’ve specified the output **folder that holds the filtered counts matrix**.

This chunk also creates the Seurat object. It assigns the data to the counts slot, names the project “pbmc10k”, only keeps features with at least 3 cell counts, and cells with at least 200 genes/features.

You can adjust those thresholds based on your experiment.

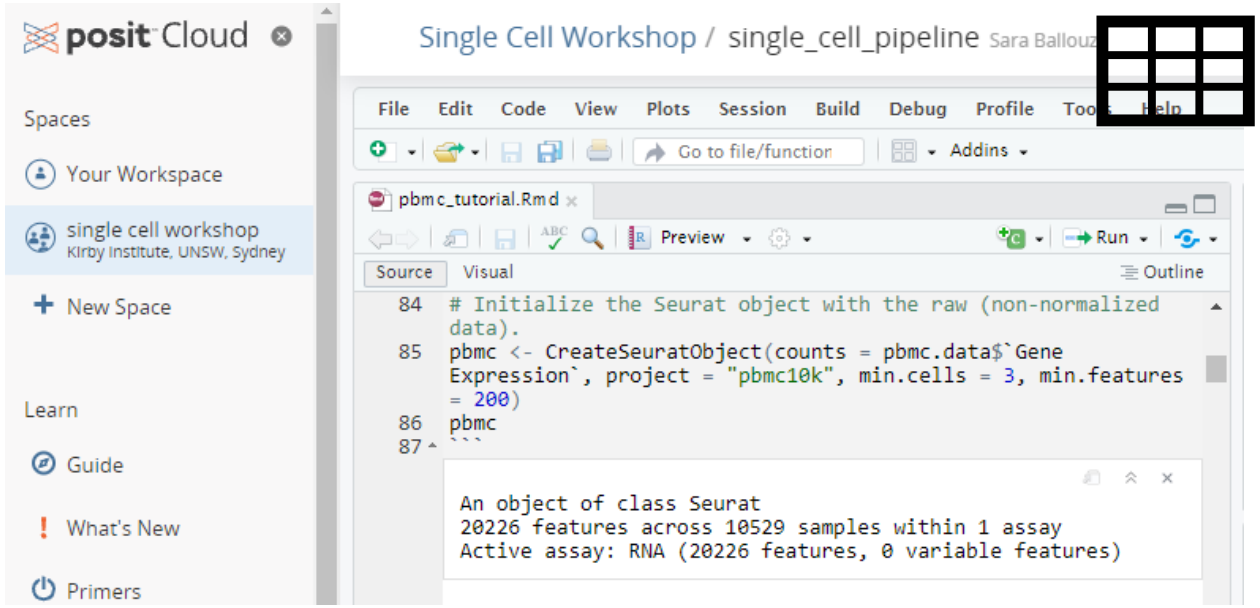
Data loaded!

Hopefully, your data should be loaded and your object created. Feel free to explore!

In the Console terminal below your notebook, type in the name of your object and follow it with an @ symbol ie:

pbmc@

The different slots should appear. Pick one and press enter.

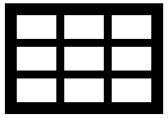


Overview

1. Introduction to single cell
2. Setting up Rstudio, data and count matrix
3. **Pre-processing**
4. Application 1: Cell annotation
5. Application 2: Case vs Control

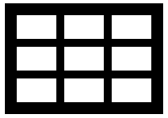
Preprocessing pipeline

Gene expression matrix (**raw**)



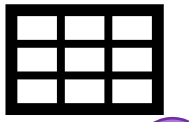
1 **Quality control**

Gene expression matrix (**cleaned**)

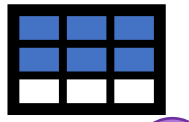


2 Normalization

Gene expression matrix (**normalized**)

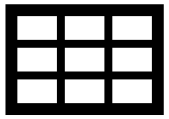


3 Feature selection



4 Dimension reduction

Reduced gene expression matrix (**normalized**)



Worked example

	Cell 1	Cell 2	Cell 3	Cell 4
Gene 1	0	0	3	10
Gene 2	24	0	41	12
(MT gene) Gene 3	175	284	93	162
Gene 4	0	0	0	0
Gene 5	36	0	32	21

Number of genes:
Cell 1: 3
Cell 2: 1

for each cell, total number of genes with more than 0 expression

Total count:
Cell 1: 24 + 175 + 36
Cell 2: 284

for each cell, sum of all the gene expression counts

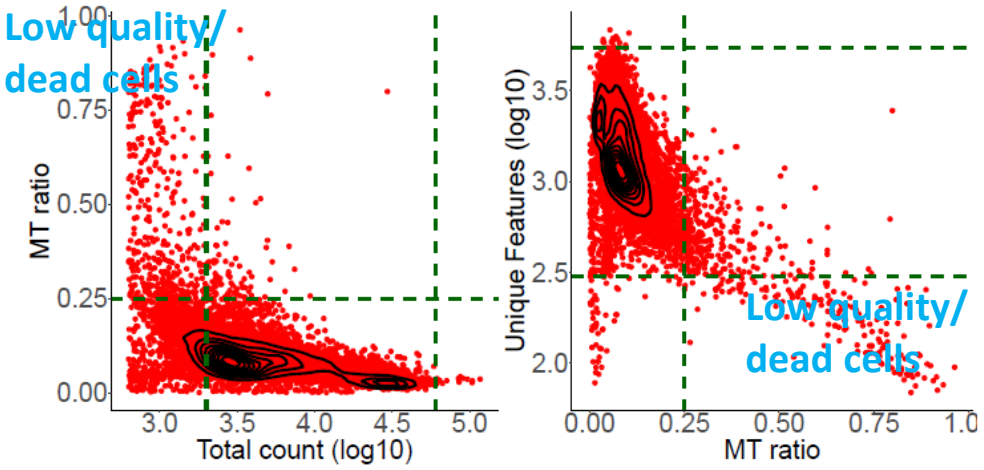
Mitochondrial (MT) ratio
Cell 1: 175/(175+24+36)
Cell 2: 1

for each cell, fraction of expression from mitochondrial genes

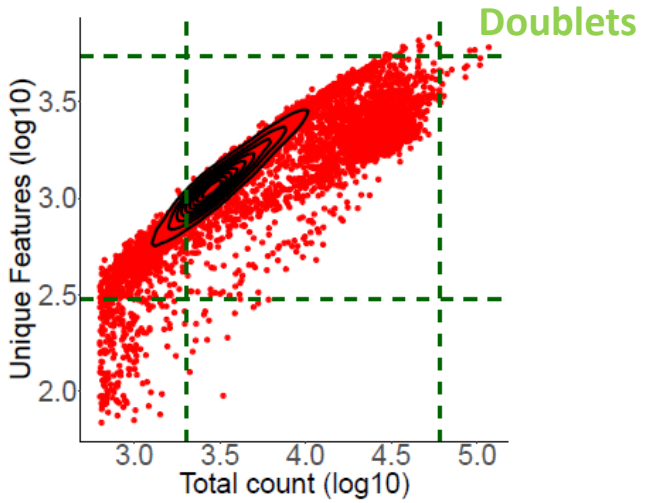
Preprocessing: quality control

1. **Quality control** → 2. Normalization → 3. Feature selection → 4. Dimension reduction

- **Low quality/dead cells: high MT ratio / low total count / low number of genes**

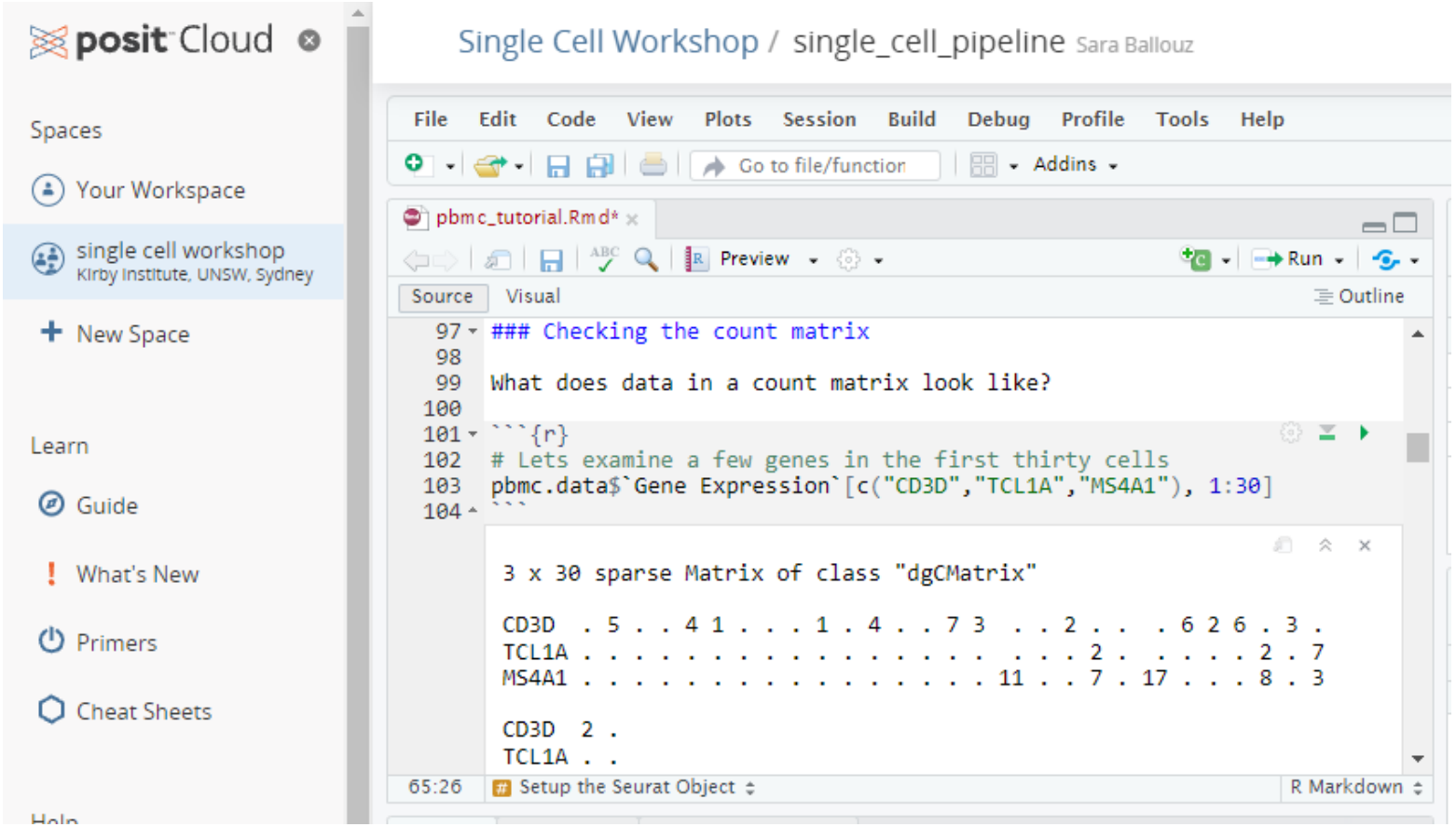


- **Doublets: high total count/ high number of genes**



UNSW SYDNEY

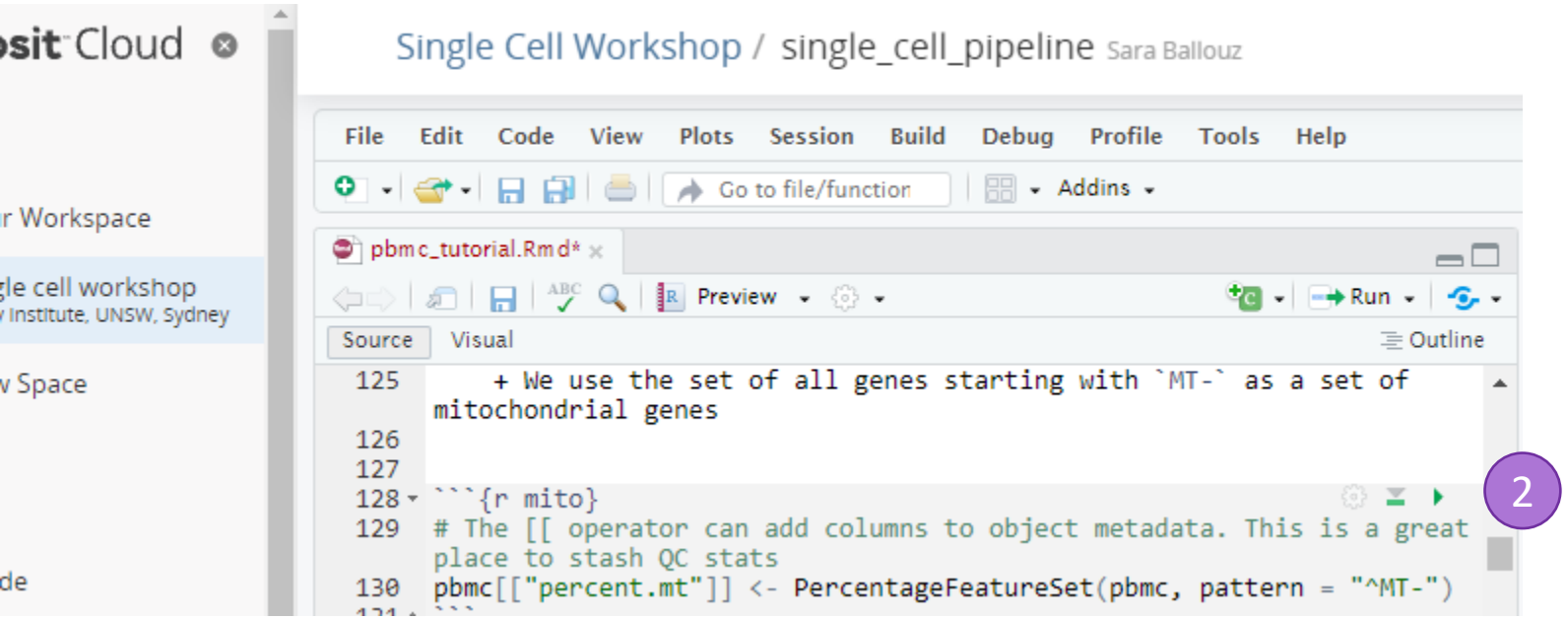
Count matrix check



1 Run the chunk. This should show you the first few genes and cells.

Note “.” are 0 counts.

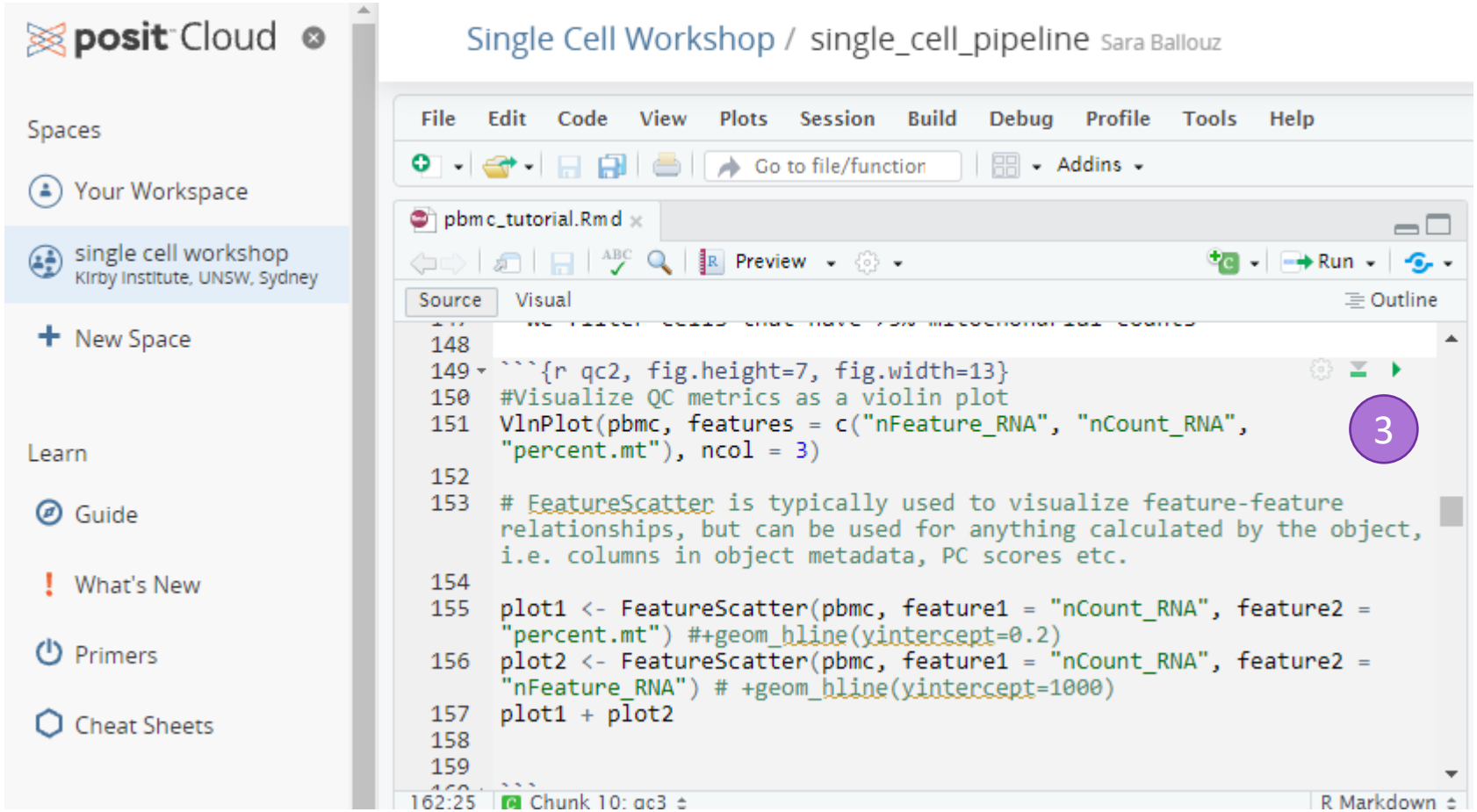
MT-ratio



Run the chunk to calculate the mitochondrial ratio. Note, we look for the genes that start with “MT-”. This is dependent on the gene IDs and species.

We store it in a slot called “percent.mt”

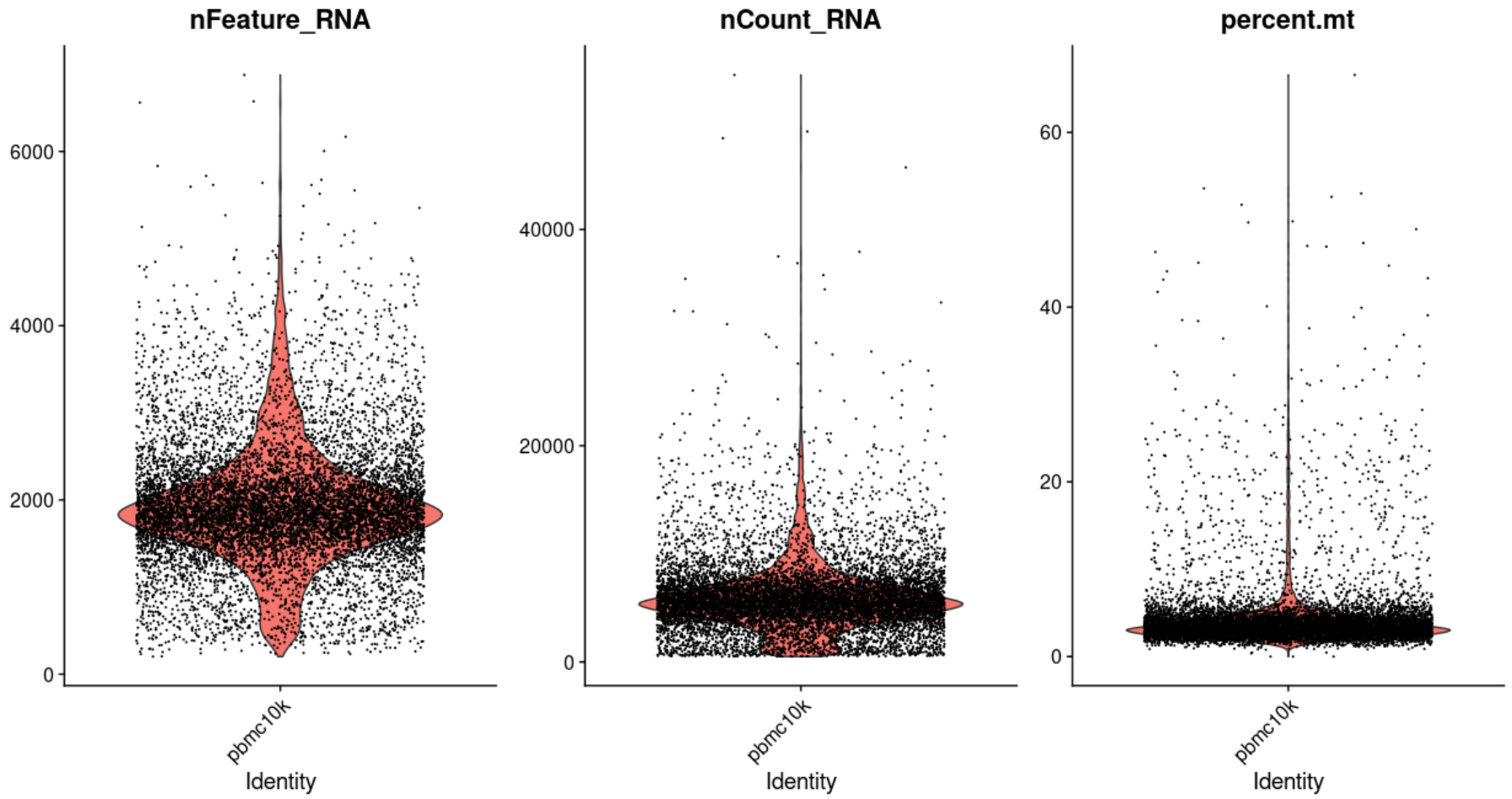
Visualizing QC metrics

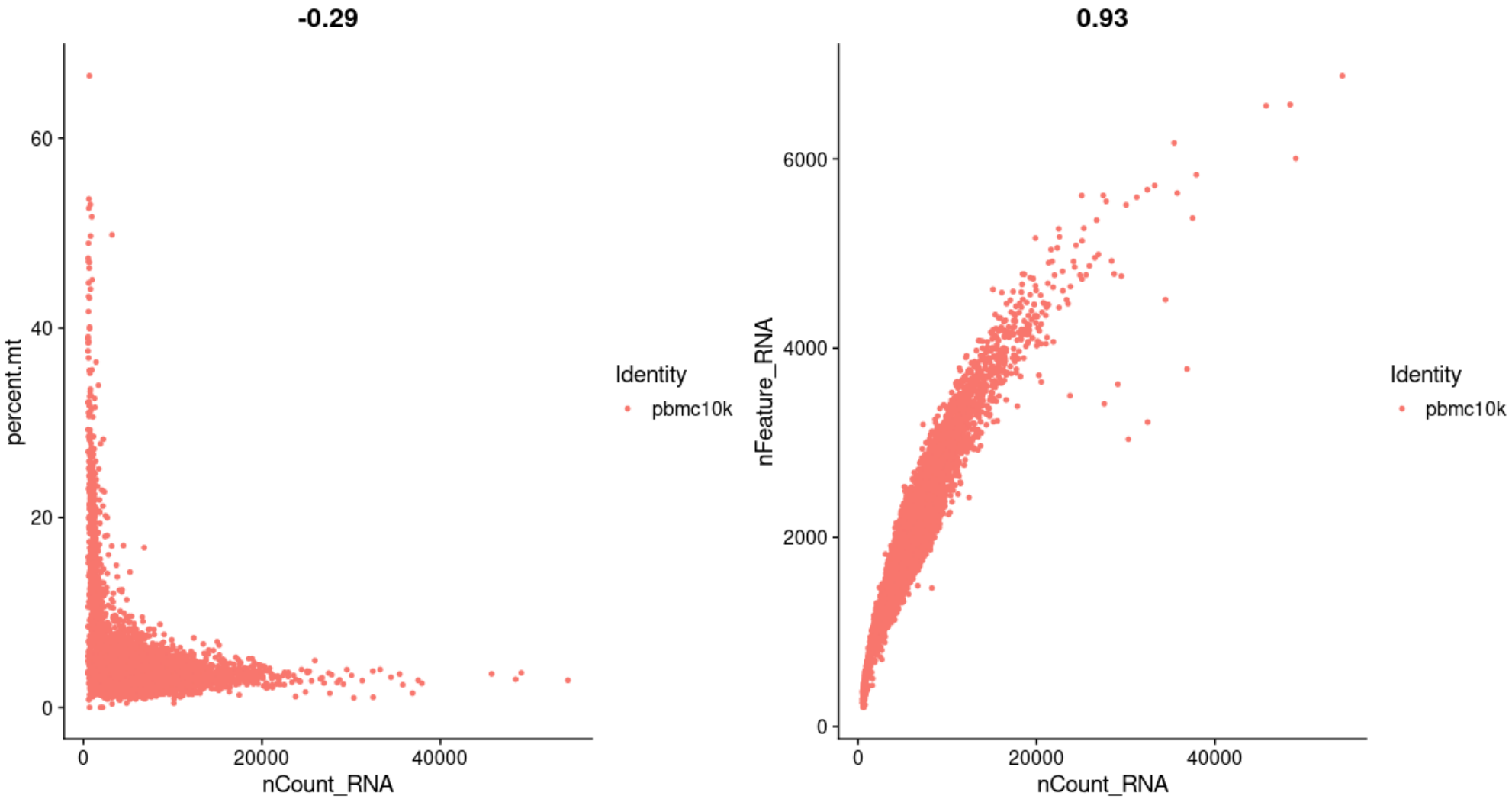


Run the chunk to plot our QC metrics.

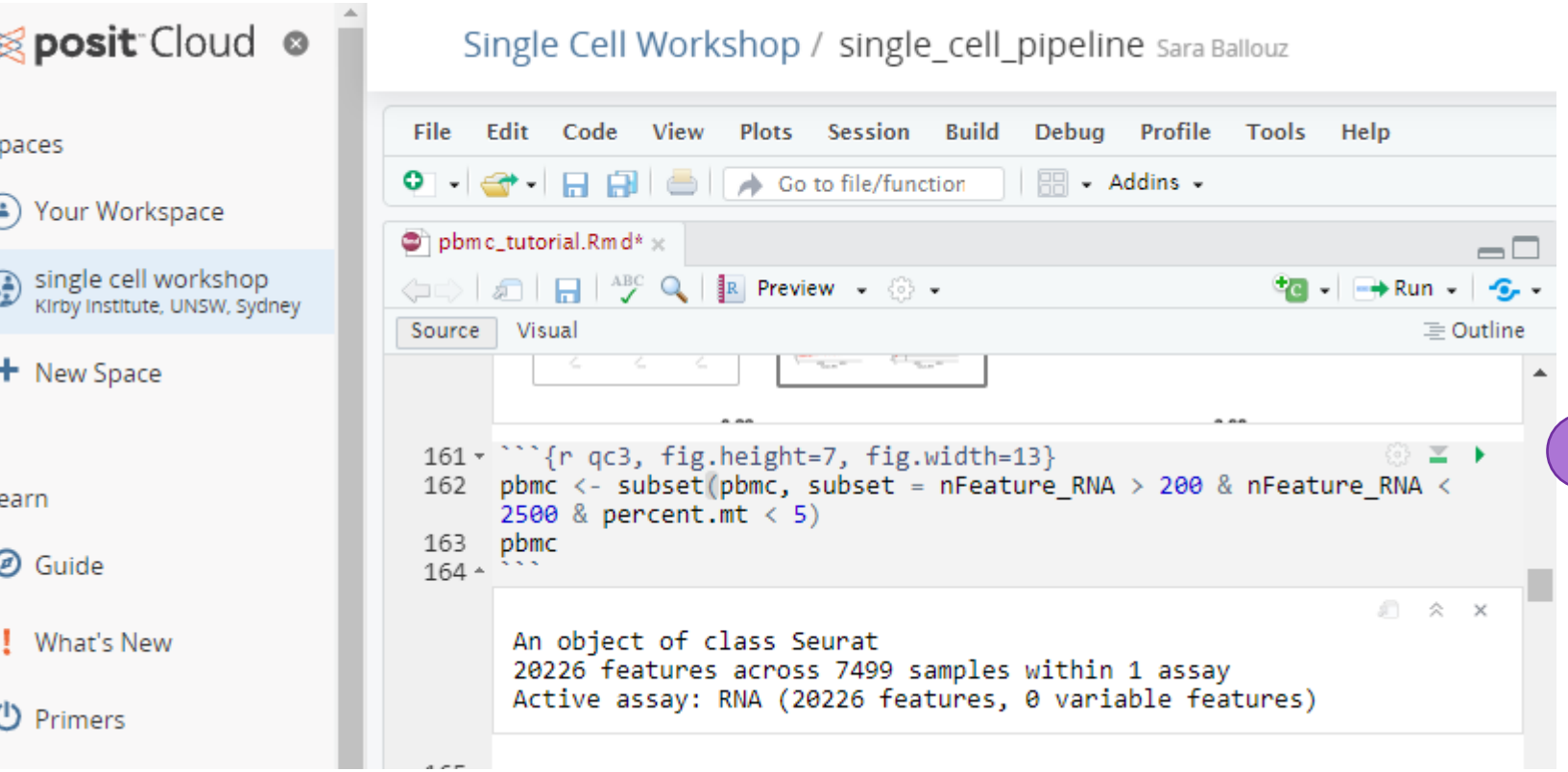
This includes a violin plot of the number of genes (**nFeature_RNA**), Total counts (**nCount_RNA**) and MT-ratio (**percent.mt**).

Additionally, we plot these metrics in a scatter plot.


nFeature_RNA
6000
4000
2000
0
pbmc10k
Identity
nCount_RNA
40000
20000
0
pbmc10k
Identity
percent.mt
60
40
20
0
pbmc10k
Identity


-0.29
percent.mt
60
40
20
0
0
20000
40000
nCount_RNA
Identity
pbmc10k
0.93
nFeature_RNA
6000
4000
2000
0
0
20000
40000
nCount_RNA
Identity
pbmc10k

Filtering away poor-quality cells



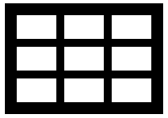
3

Once we've decided on our thresholds for the individual metrics, we filter away the poor-quality cells.

Using these defaults, we should have **7499** cells remaining.

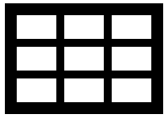
Preprocessing pipeline

Gene expression matrix (**raw**)



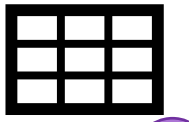
1 Quality control

Gene expression matrix (**cleaned**)

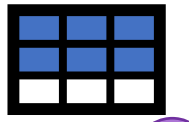


2 **Normalization**

Gene expression matrix (**normalized**)

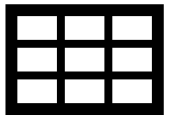


3 Feature selection



4 Dimension reduction

Reduced gene expression matrix (**normalized**)

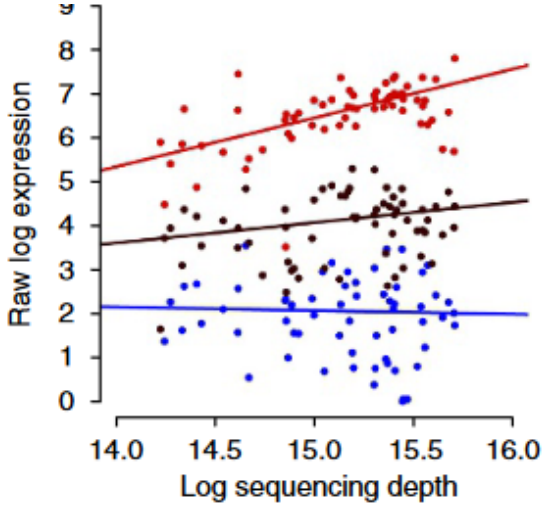


Normalization

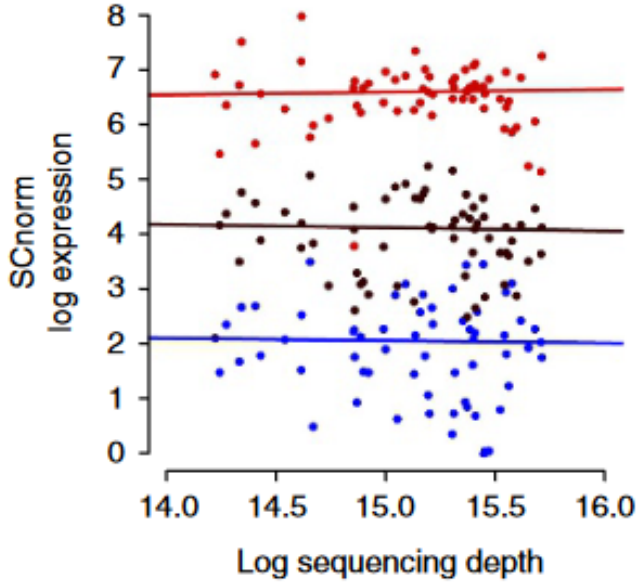
- Difference in total counts due to sequencing coverage–> same cell type may have different gene expression abundance

- Normalization or “rescaling” to ensure correct relative gene expression abundances between cells

Raw data



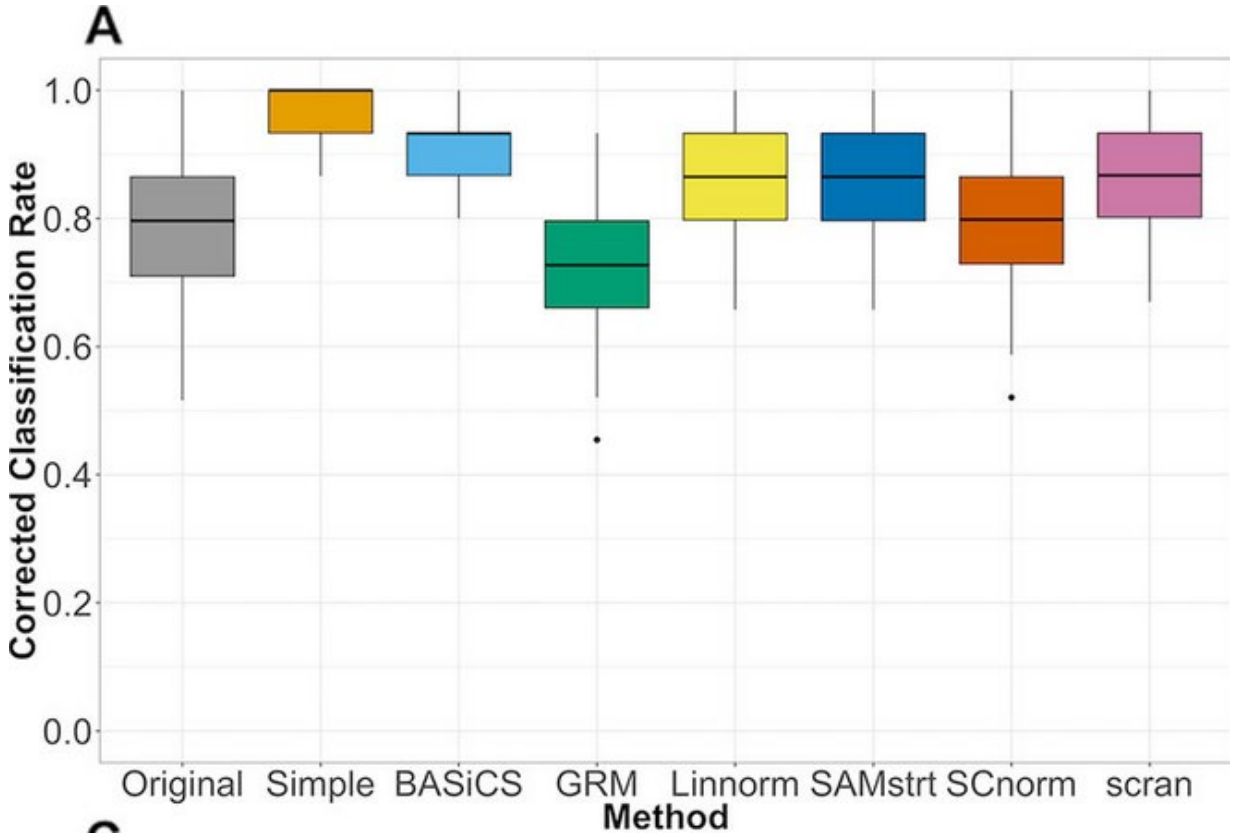
Normalized data



https://www.nature.com/articles/nmeth.4263

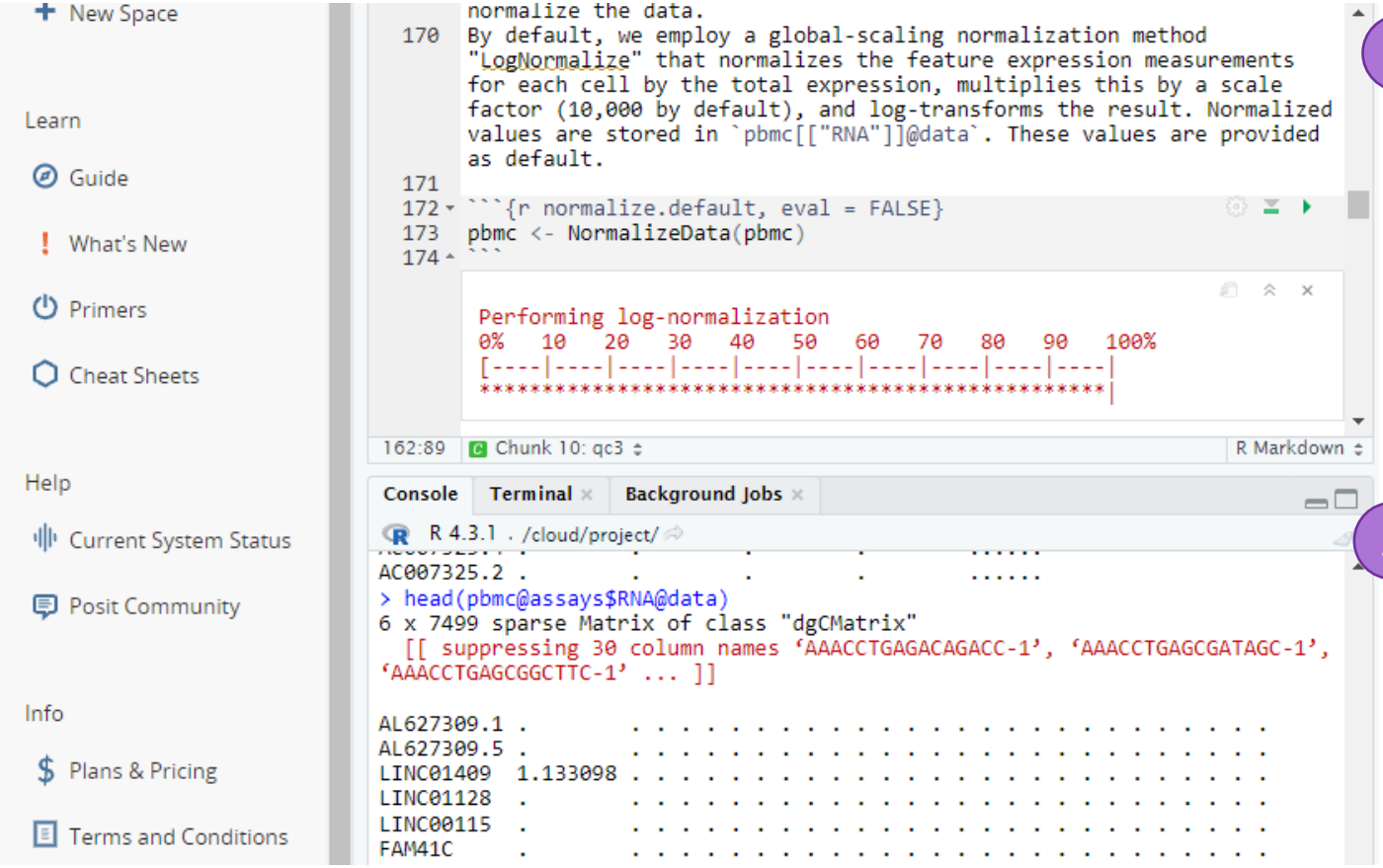
UNSW SYDNEY

Normalization (different methods)



https://www.frontiersin.org/articles/10.3389/fgene.2020.00041/full

Normalization

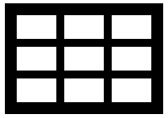


4 Run the code chunk, this should perform the log-normalization, and add a new slot to our object called data.

5 Check it out by typing head(**pbmc@assays$RNA@data)** and then hitting enter in the console window.

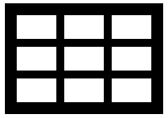
Preprocessing pipeline

Gene expression matrix (**raw**)



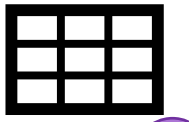
1 Quality control

Gene expression matrix (**cleaned**)

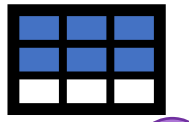


2 Normalization

Gene expression matrix (**normalized**)

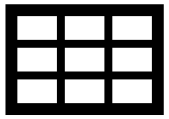


3 **Feature selection**



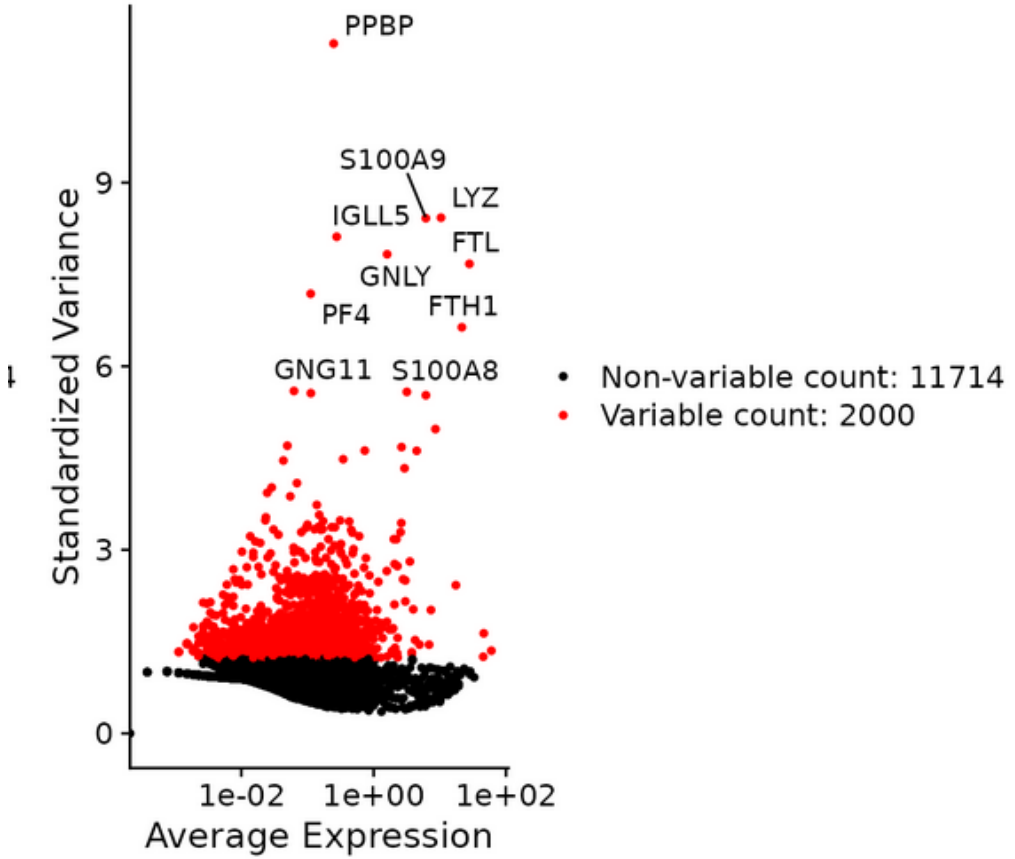
4 Dimension reduction

Reduced gene expression matrix (**normalized**)



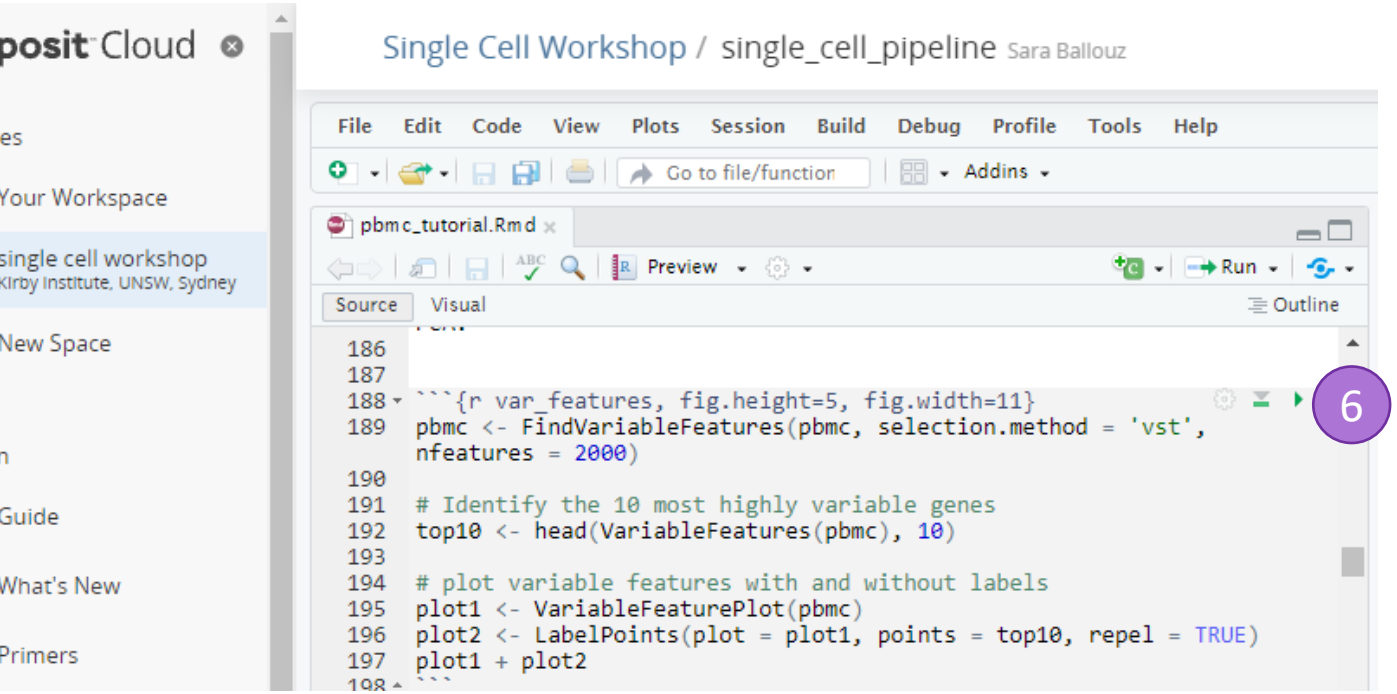
Feature selection

- We are interested in differences between cells, which cannot be captured by genes which are similarly expressed
- Choose genes with highest standardized variance (z-score)



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

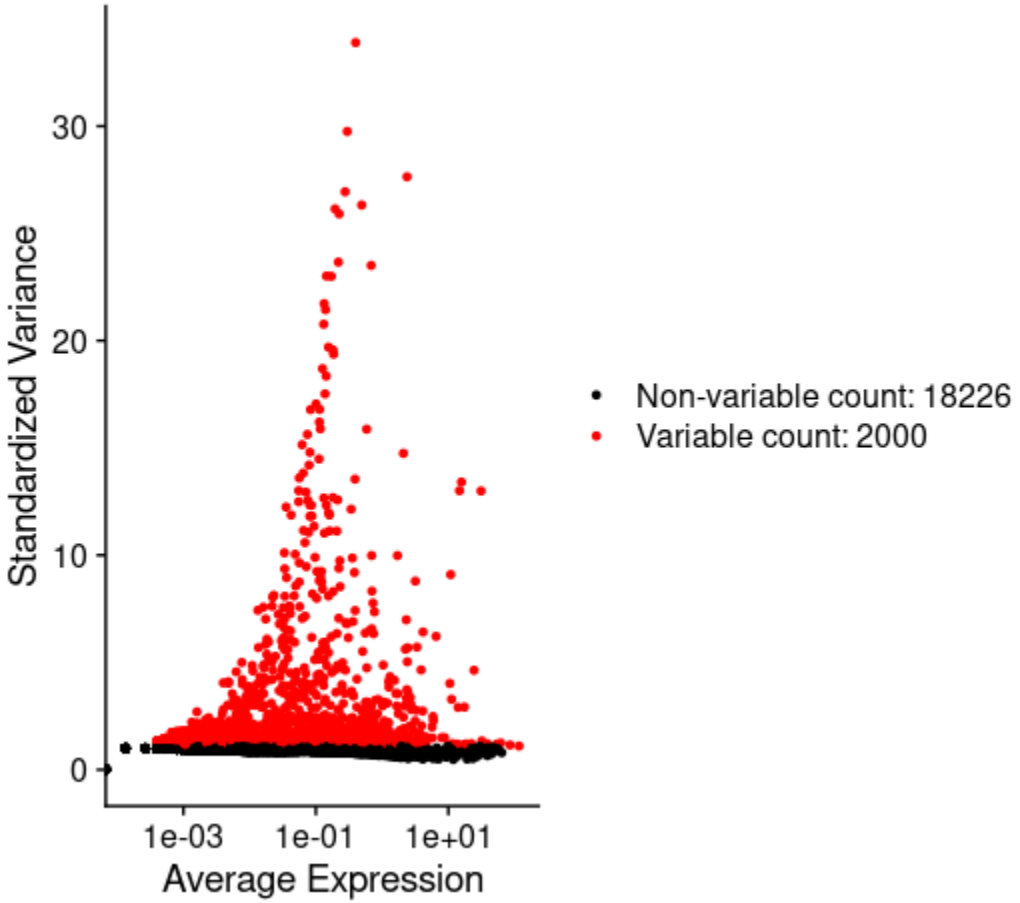
Feature selection

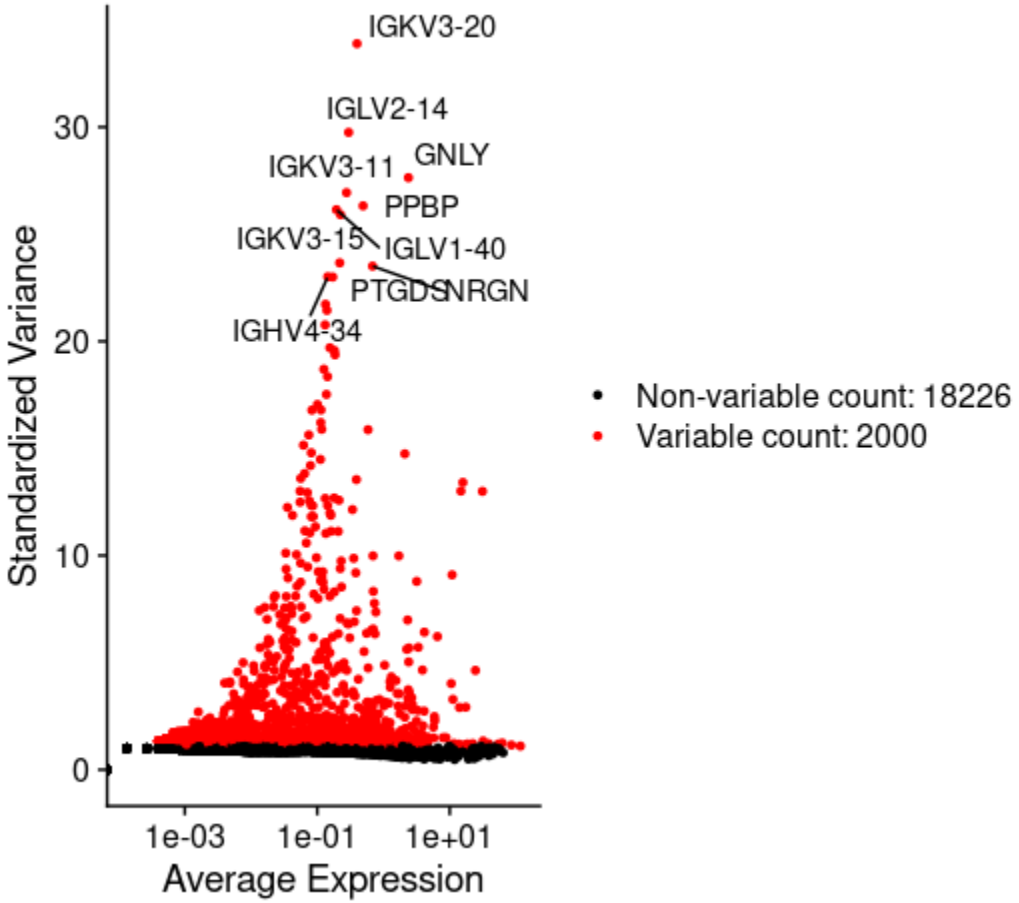


What is notable about the top 10?

Run the code chunk, which will use the "VST" method (variance stabilisation transformation) to pick the most variable genes/features (we asked for 2000 genes).

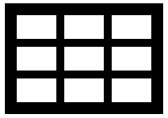
We can then plot the average expression against the standardized variance to show these genes and label the top 10.


Standardized Variance
30
20
10
0
1e-03
1e-01
1e+01
Average Expression
Non-variable count: 18226
Variable count: 2000


IGKV3-20
IGLV2-14
IGKV3-11
GNLY
PPBP
IGKV3-15
IGLV1-40
PTGDS
NRGN
IGHV4-34
Standardized Variance
30
20
10
0
1e-03
1e-01
1e+01
Average Expression
Non-variable count: 18226
Variable count: 2000

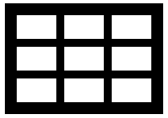
Preprocessing pipeline

Gene expression matrix (**raw**)



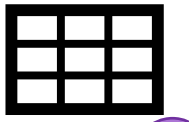
1 Quality control

Gene expression matrix (**cleaned**)

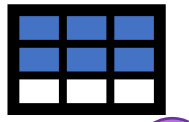


2 Normalization

Gene expression matrix (**normalized**)

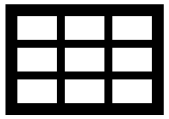


3 Feature selection



4 **Dimension reduction**

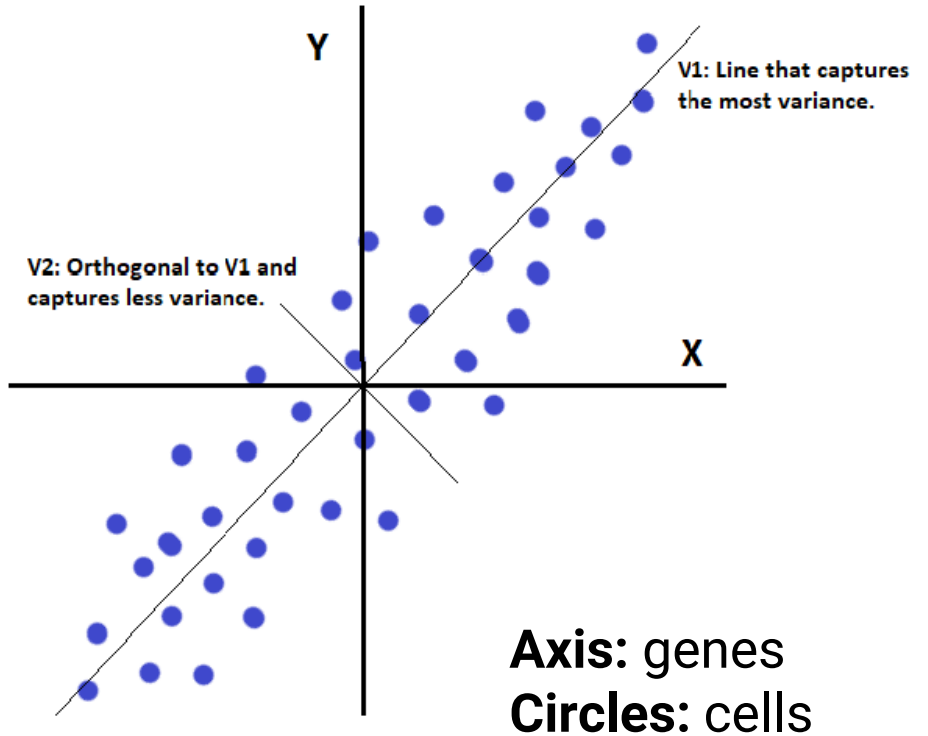
Reduced gene expression matrix (**normalized**)



Dimension reduction

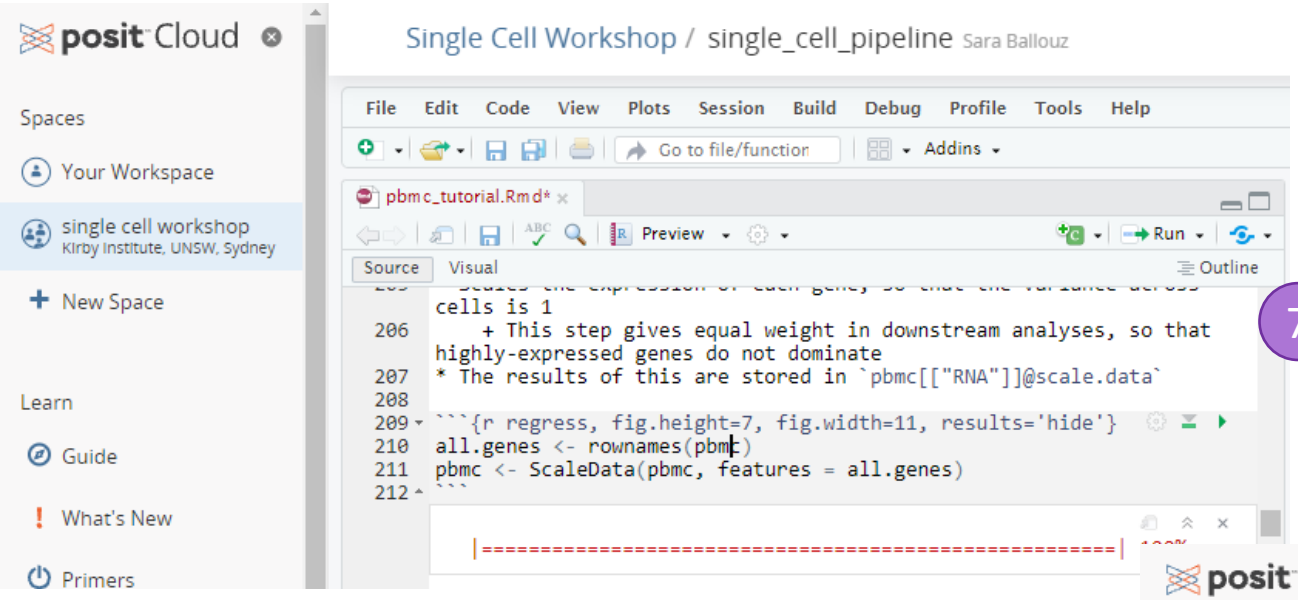
- **Problems:**
  - Noisy data
  - Computational processing (many genes, many cells)

- **Solution**: Dimension reduction
  - Reduces to a few dimensions which capture the “most important” information



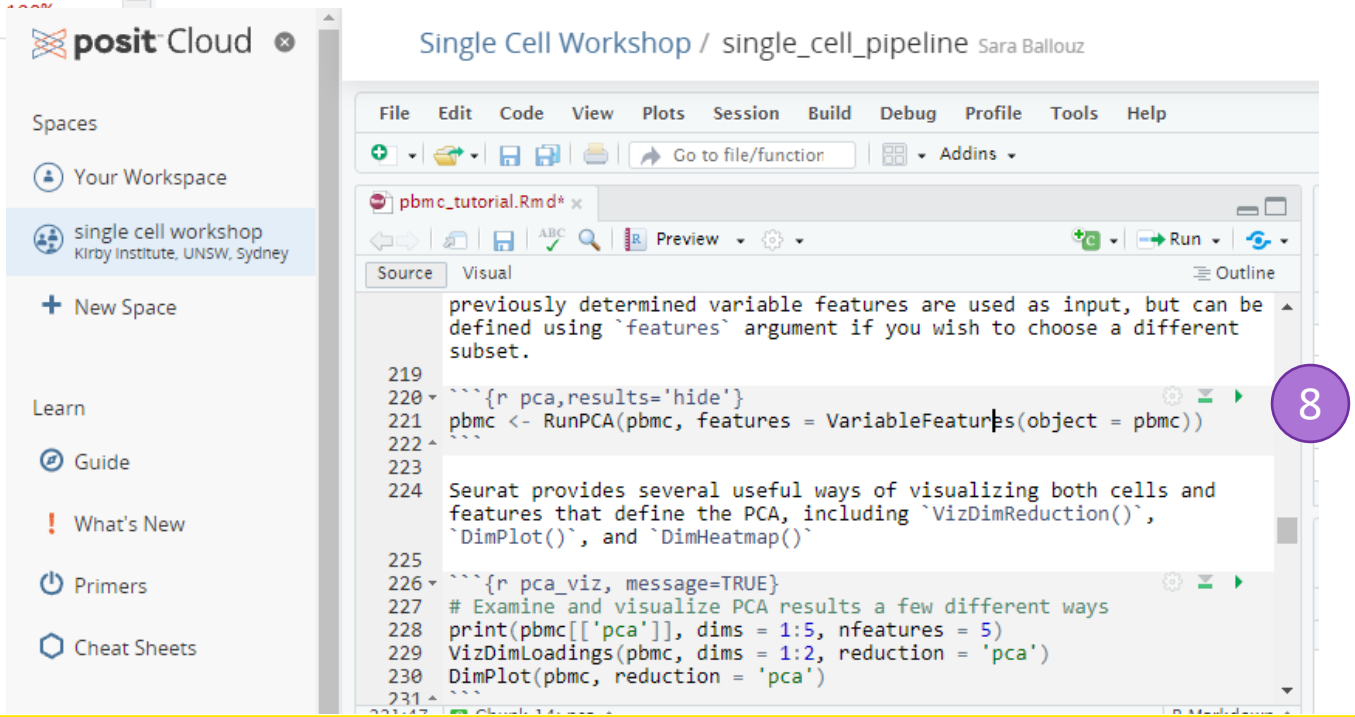
Axis: genes
Circles: cells

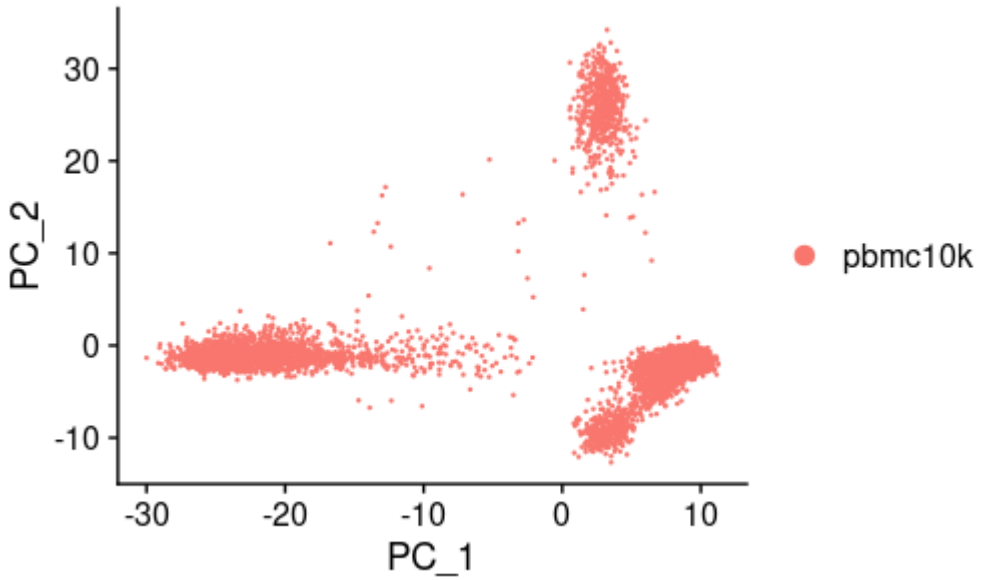
Choose the line which captures most variance

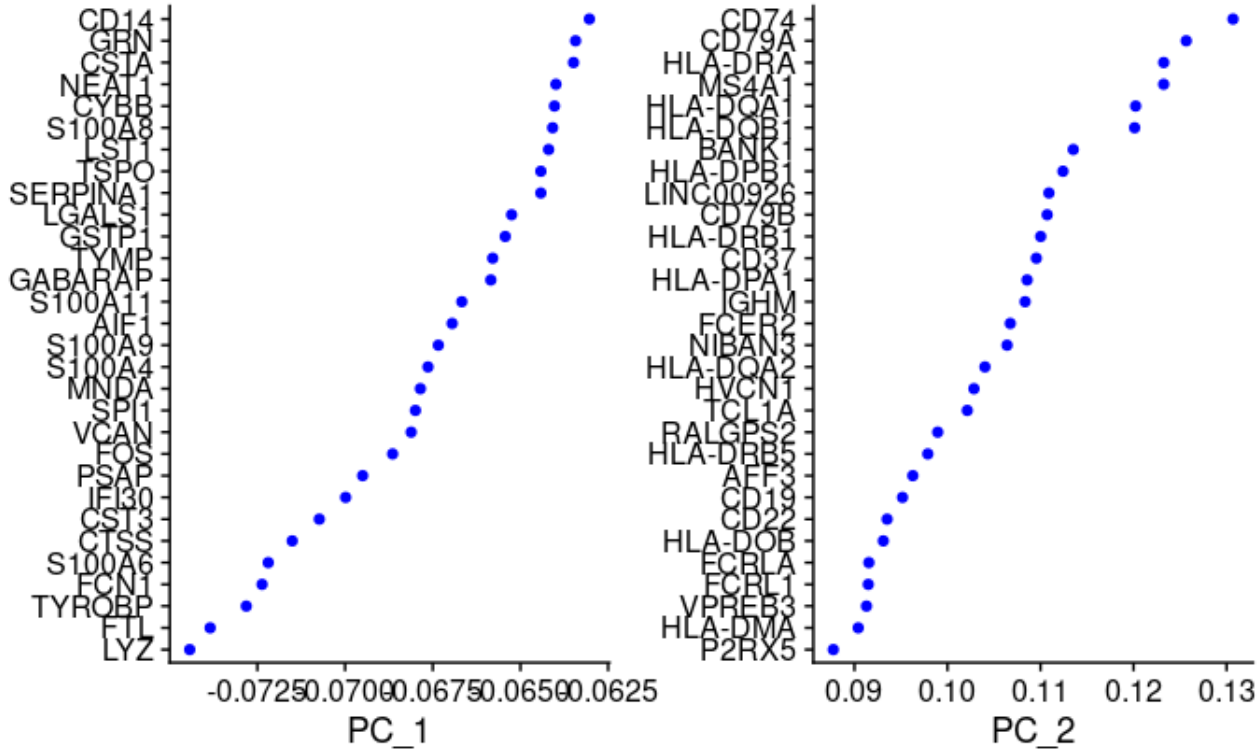


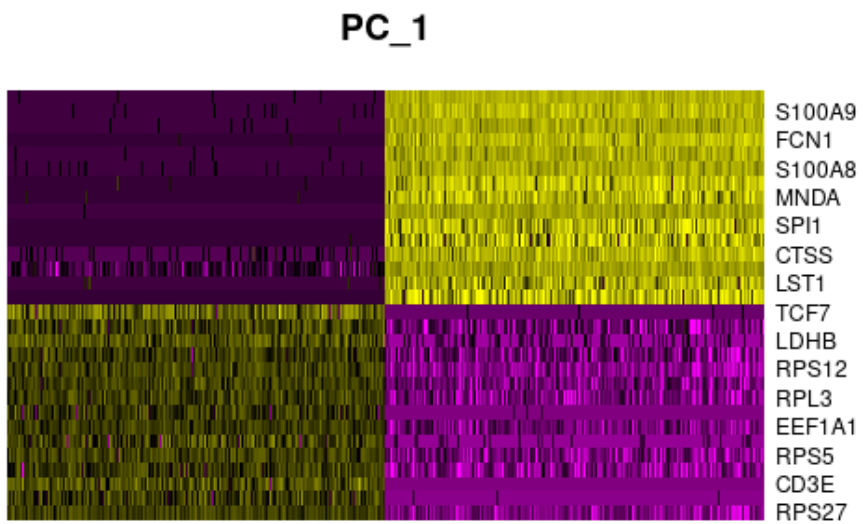
First, we scale the data. This is to transform it prior to running any dimension reduction. This data is added to the “scale.data” slot.

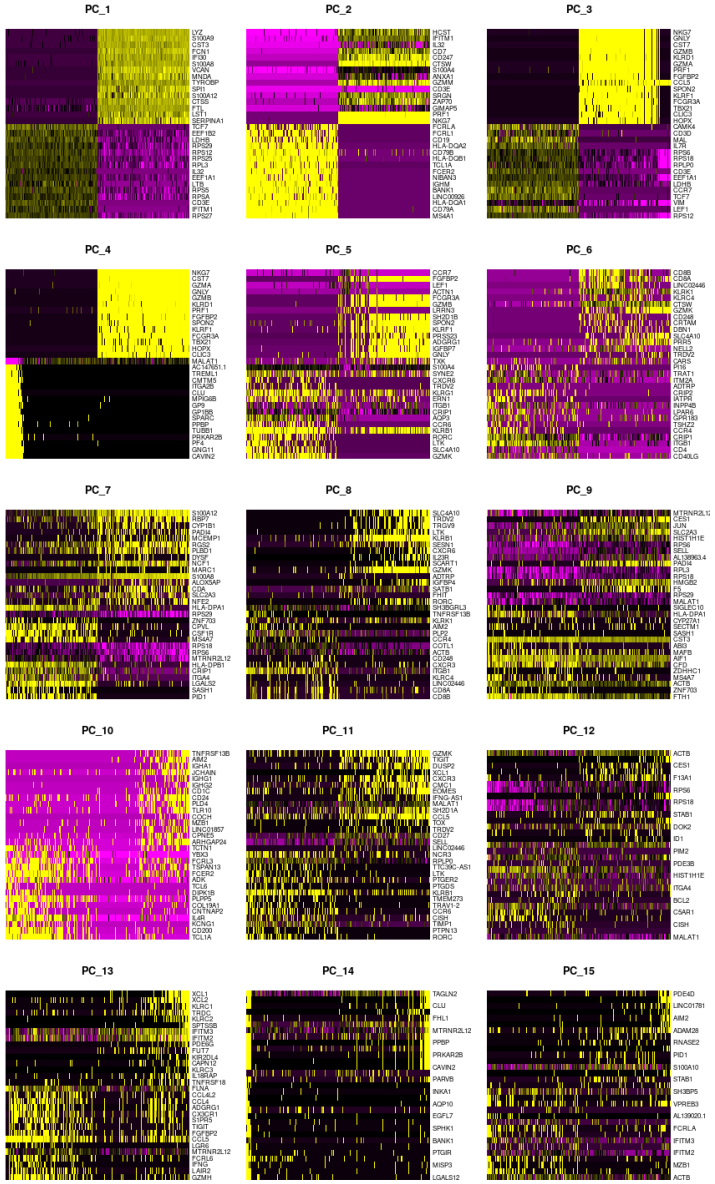
Next, we run PCA and visualise the data, looking at the first two principal components (PCs).



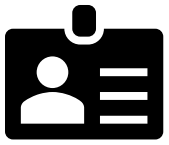

30
20
10
0
-10
PC_2
-30
-20
-10
0
10
PC_1
pbmc10k


CD14
GRN
CSTA
NEAT1
CYBB
S100A8
TST1
TSPO
SERPINA1
LGALS1
GSTP1
TYMP
GABARAP
S100A11
AIF1
S100A9
S100A4
MNDA
SPI1
VCAN
FOS
PSAP
IFI30
CST3
CTSS
S100A6
FCN1
TYROBP
FTL
LYZ
-0.0725
-0.0700
-0.0675
-0.0650
-0.0625
PC_1
CD74
CD79A
HLA-DRA
MS4A1
HLA-DQA1
HLA-DQB1
BANK1
HLA-DPB1
LINC00926
CD79B
HLA-DRB1
CD37
HLA-DPA1
IGHM
FCER2
NIBAN3
HLA-DQA2
HVCN1
TCL1A
RALGPS2
HLA-DRB5
AFF3
CD19
CD22
HLA-DOB
ECRLA
FCRL1
VPREB3
HLA-DMA
P2RX5
0.09
0.10
0.11
0.12
0.13
PC_2

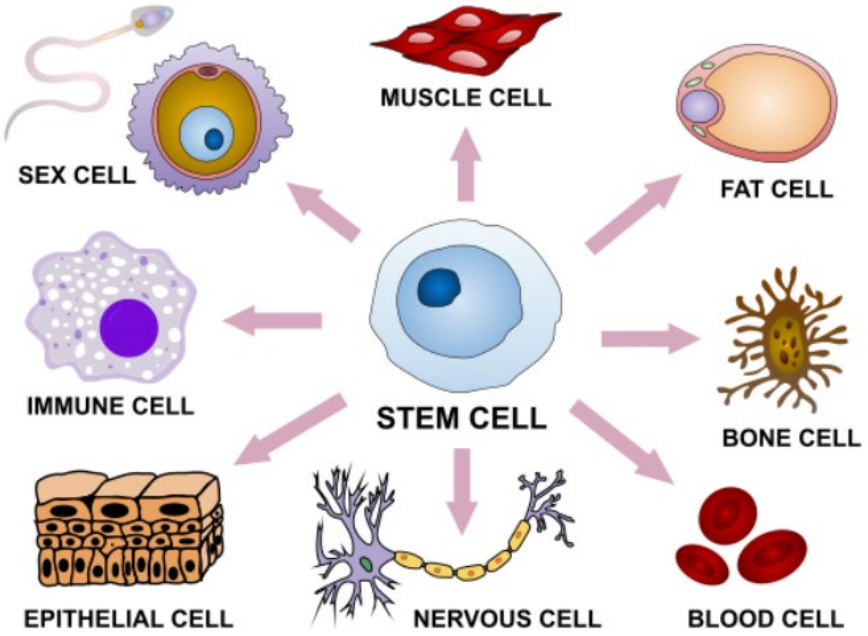

PC_1
S100A9
FCN1
S100A8
MNDA
SPI1
CTSS
LST1
TCF7
LDHB
RPS12
RPL3
EEF1A1
RPS5
CD3E
RPS27

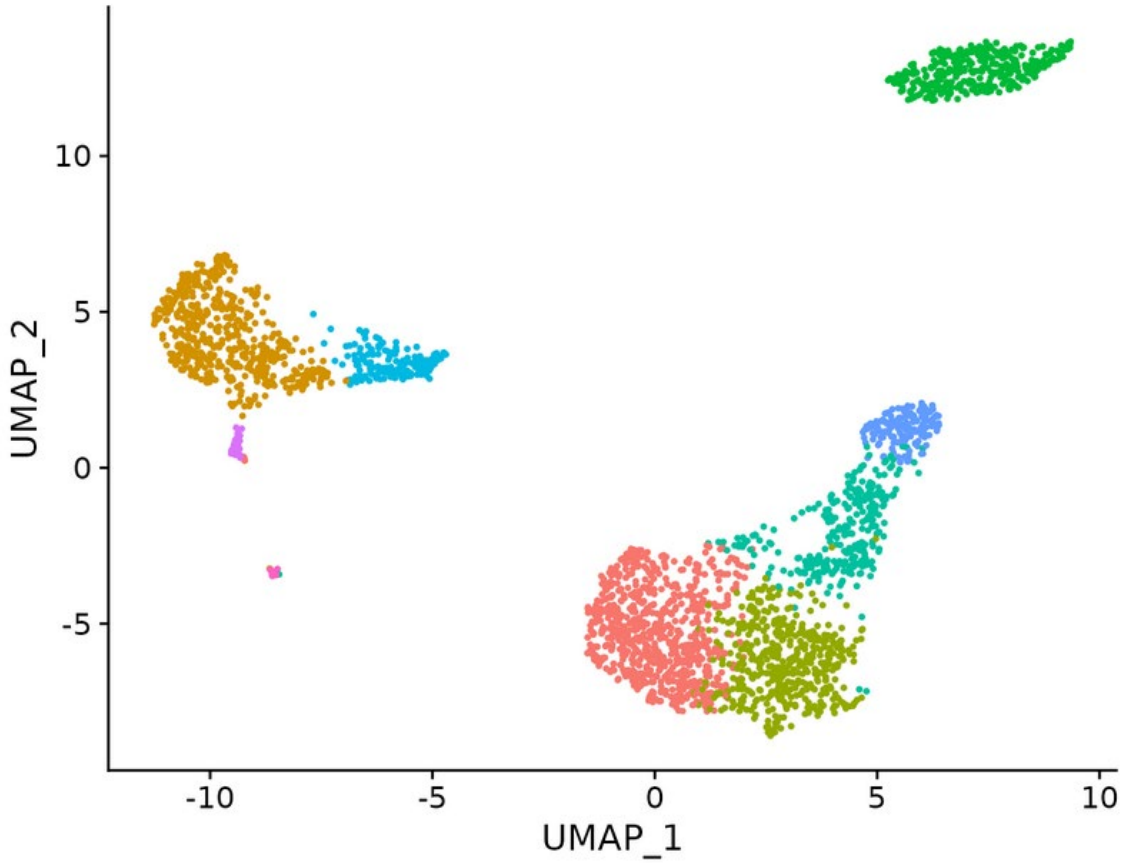

PC_1
PC_2
PC_3
PC_4
PC_5
PC_6
PC_7
PC_8
PC_9
PC_10
PC_11
PC_12
PC_13
PC_14
PC_15

Overview

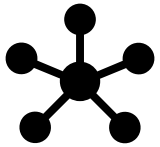
1. Introduction to single cell
2. Setting up Rstudio, data and count matrix
3. Pre-processing
4. **Application 1: Cell annotation** 
5. Application 2: Case vs Control

Cell annotation



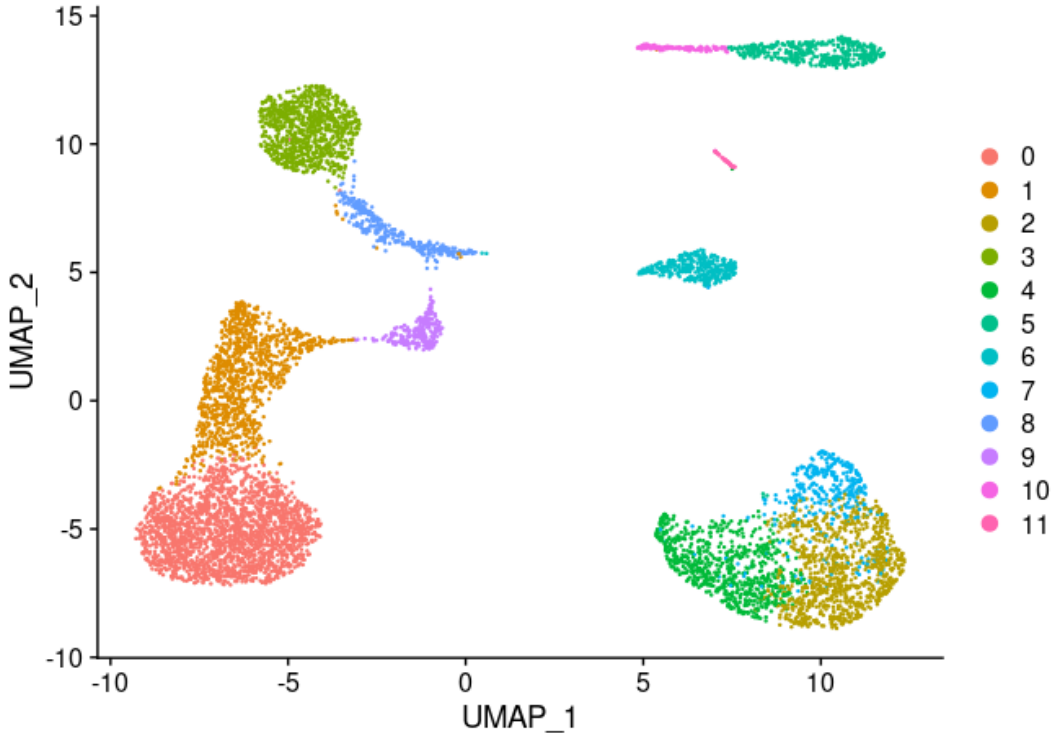


Cell annotation pipeline



1 **Clustering** → 2 Marker genes → 3 Visualisation

Unsupervised clustering



- Unsupervised machine learning algorithm which groups cells which are transcriptionally similar in the same cluster

Many unsupervised clustering algorithms (2021)

Estimate number of cell types

Inter- and intra-cluster similarities

1. scLCA
2. CIDR
3. SHARP
4. RaceID
5. SINCERA

Eigenvector-based metrics

9. SIMLR
10. Spectrum
11. SC3

Community detection

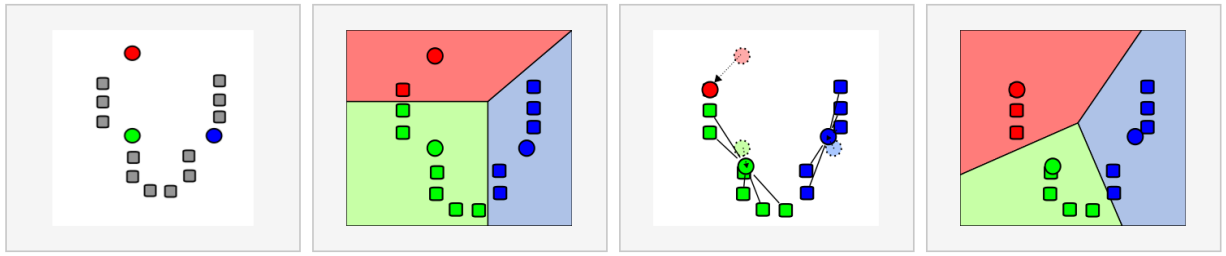
6. ACTIONet
7. Monocle3
8. Seurat

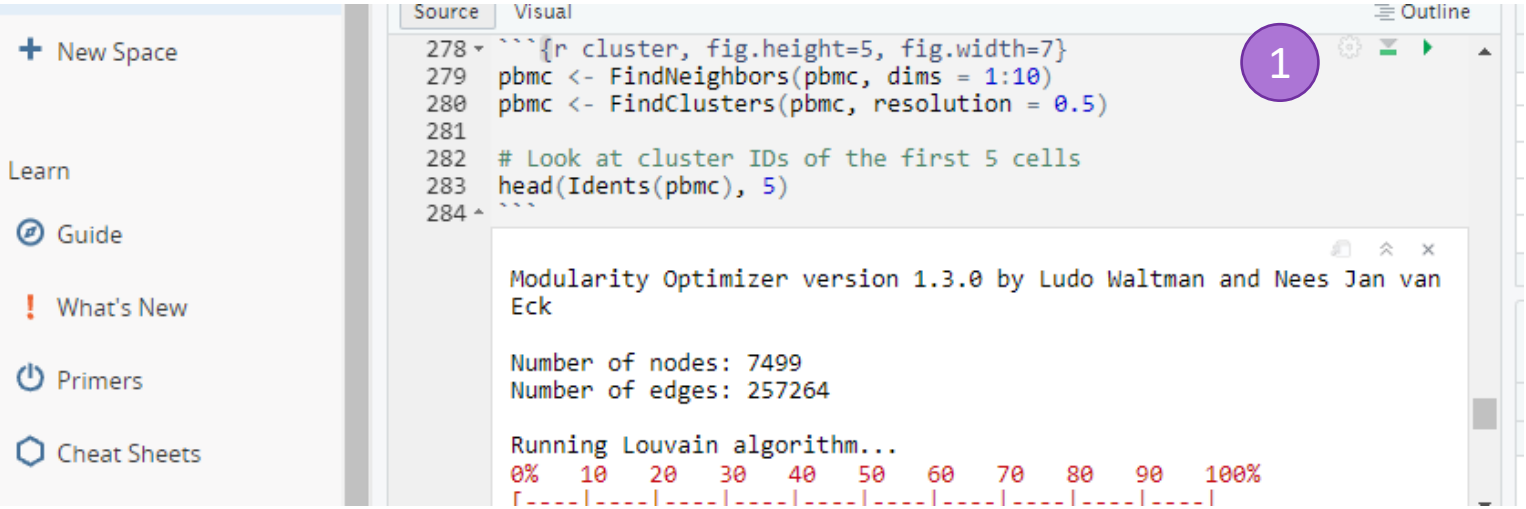
Stability metric

12. densityCut
13. scCCESS-Kmeans
14. scCCESS-SIMLR

https://genomebiology.biomedcentral.com/articles/10.1186/s13059-022-02622-0

Unsupervised clustering





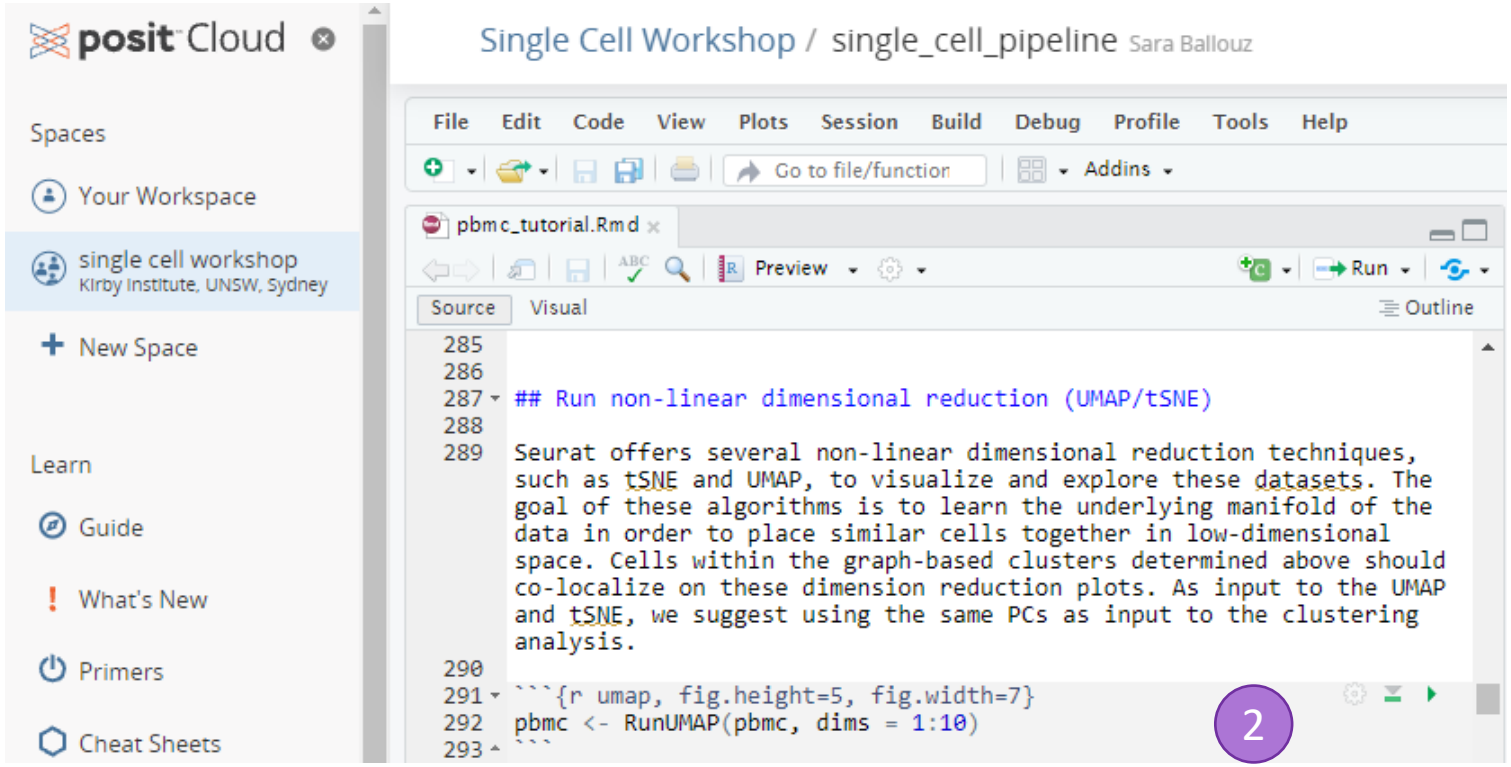
First, we look at "neighbours" of the cells, and build neighbourhoods. This is based on calculating Euclidean distances between cells in PCA space.

We then define the clusters; many methods exist to do this. The Louvain algorithm is used here. We also pick a "resolution".

https://biocellgen-public.svi.edu.au/mig_2019_scrnaseq-workshop/clustering-and-cell-annotation.html

UNSW SYDNEY

Visualising cells




```{r umapplot, fig.height=5, fig.width=7}
# note that you can set `label = TRUE` or use the
LabelClusters function to help label individual clusters
DimPlot(pbmc, reduction = 'umap')
```

Uniform Manifold Approximation and Projection (UMAP)

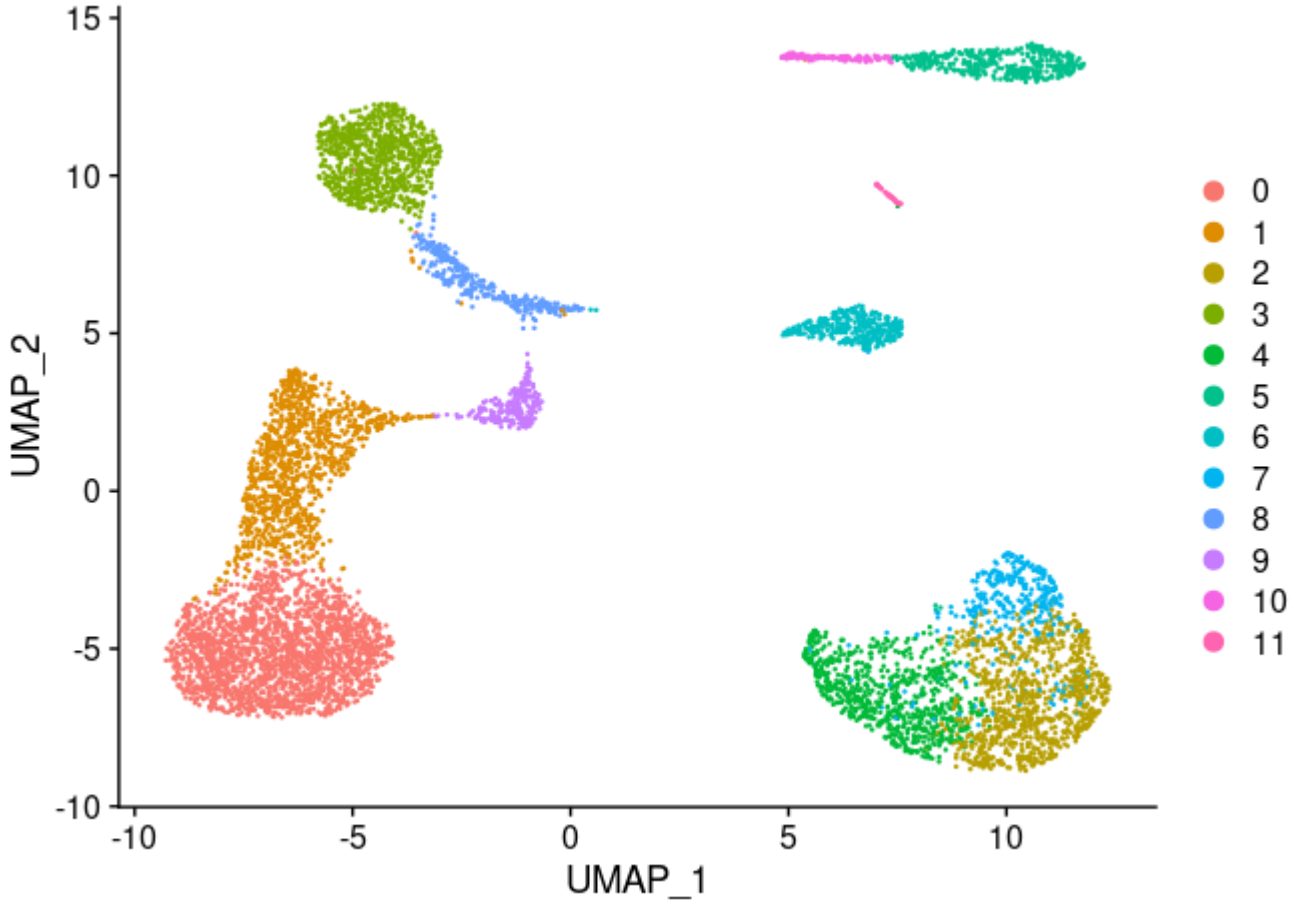
To visualise these clusters, we run another dimensional reduction algorithm. This time, we will use **UMAP**.

And then plot with the DimPlot function.

https://umap-learn.readthedocs.io/en/latest/how_umap_works.html

UNSW SYDNEY

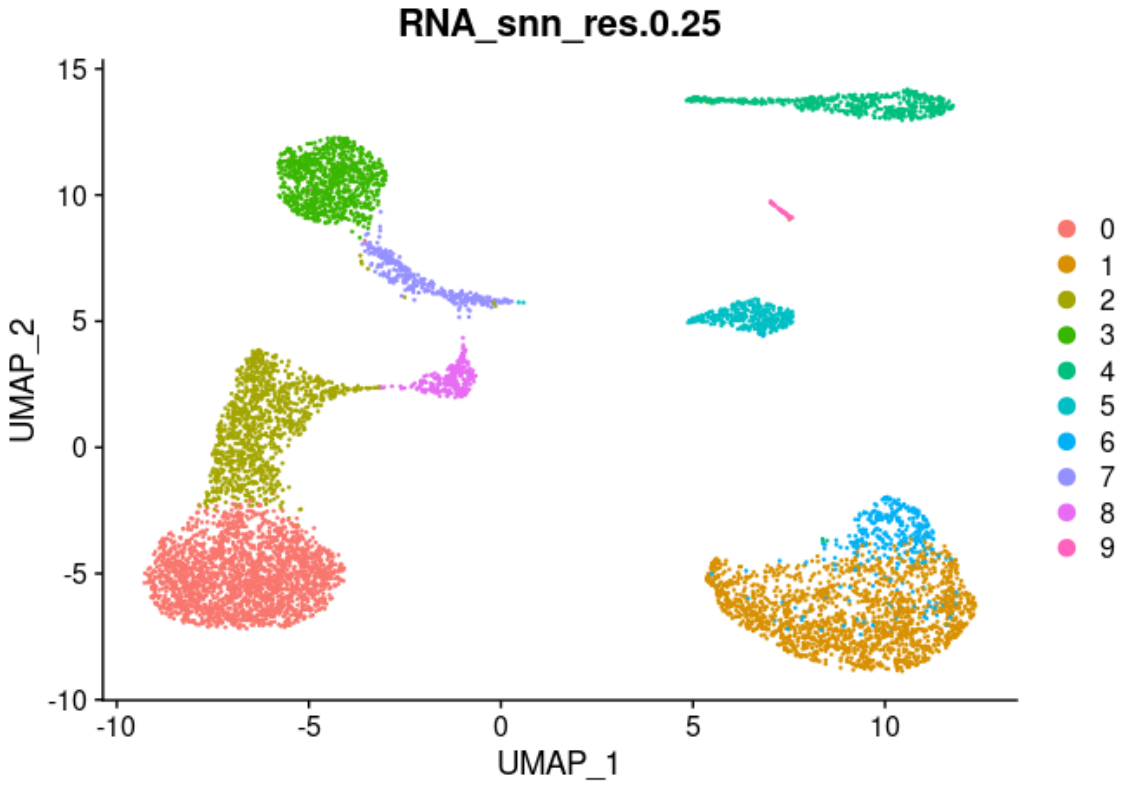
Visualisation: UMAP



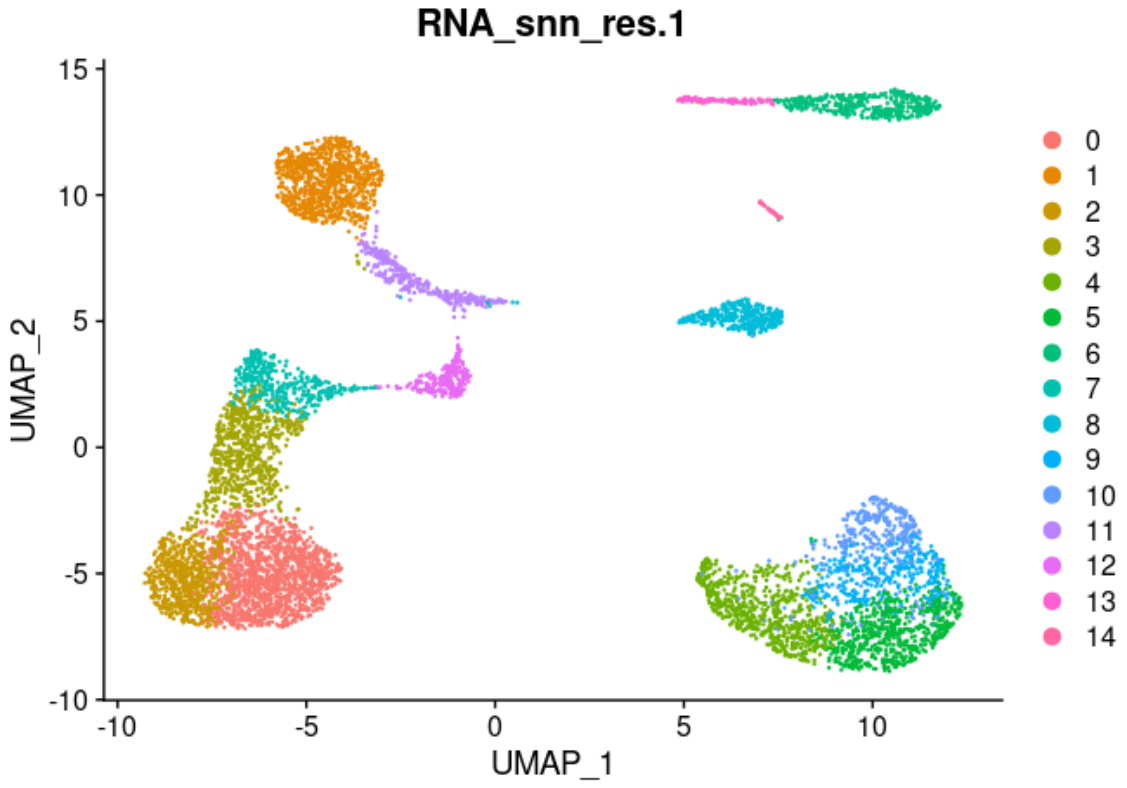
Note the number of clusters

Unsupervised clustering: resolutions

Low resolution



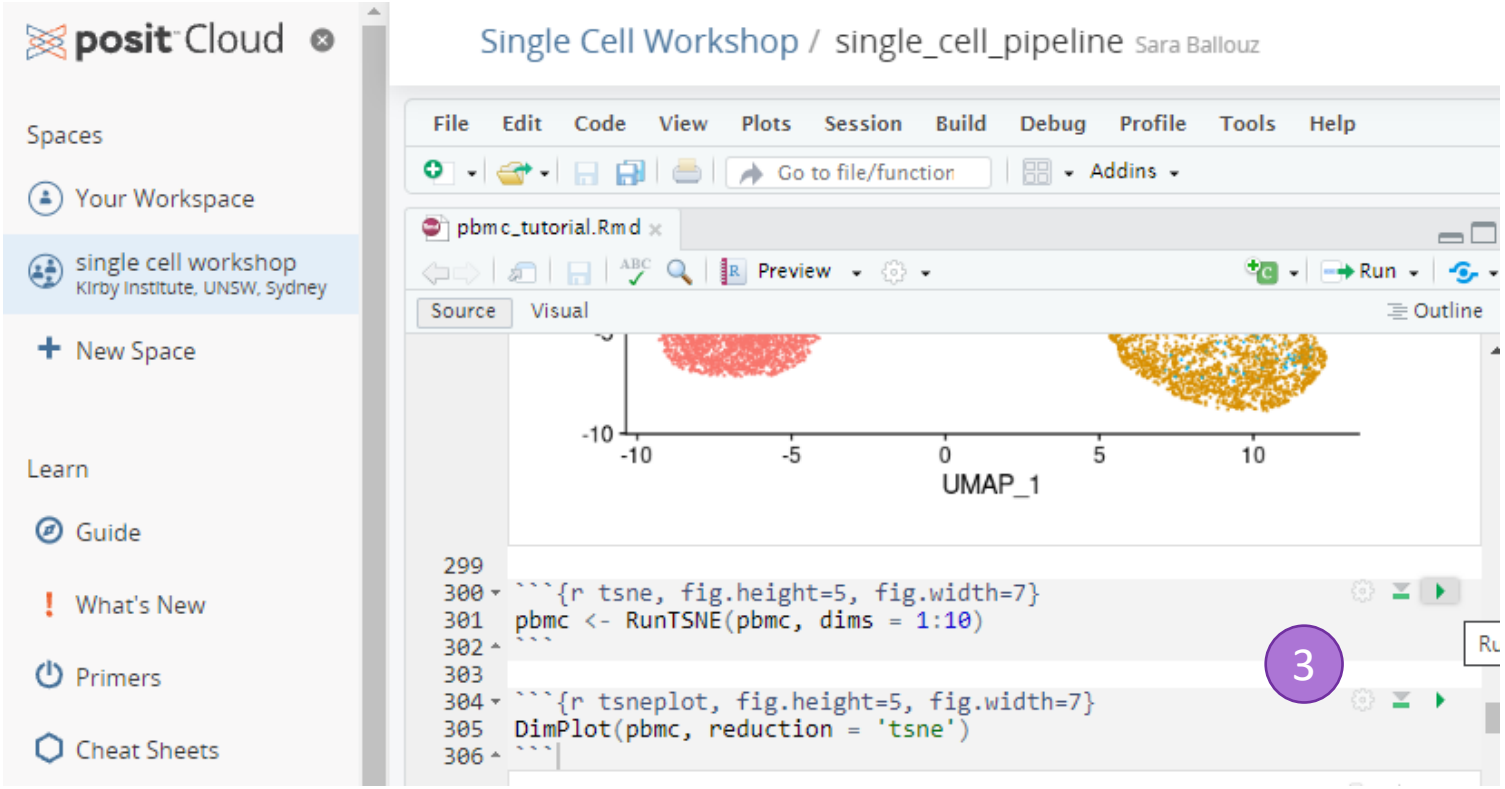
High resolution



Note the number of clusters

UNSW SYDNEY

Visualisation: tSNE



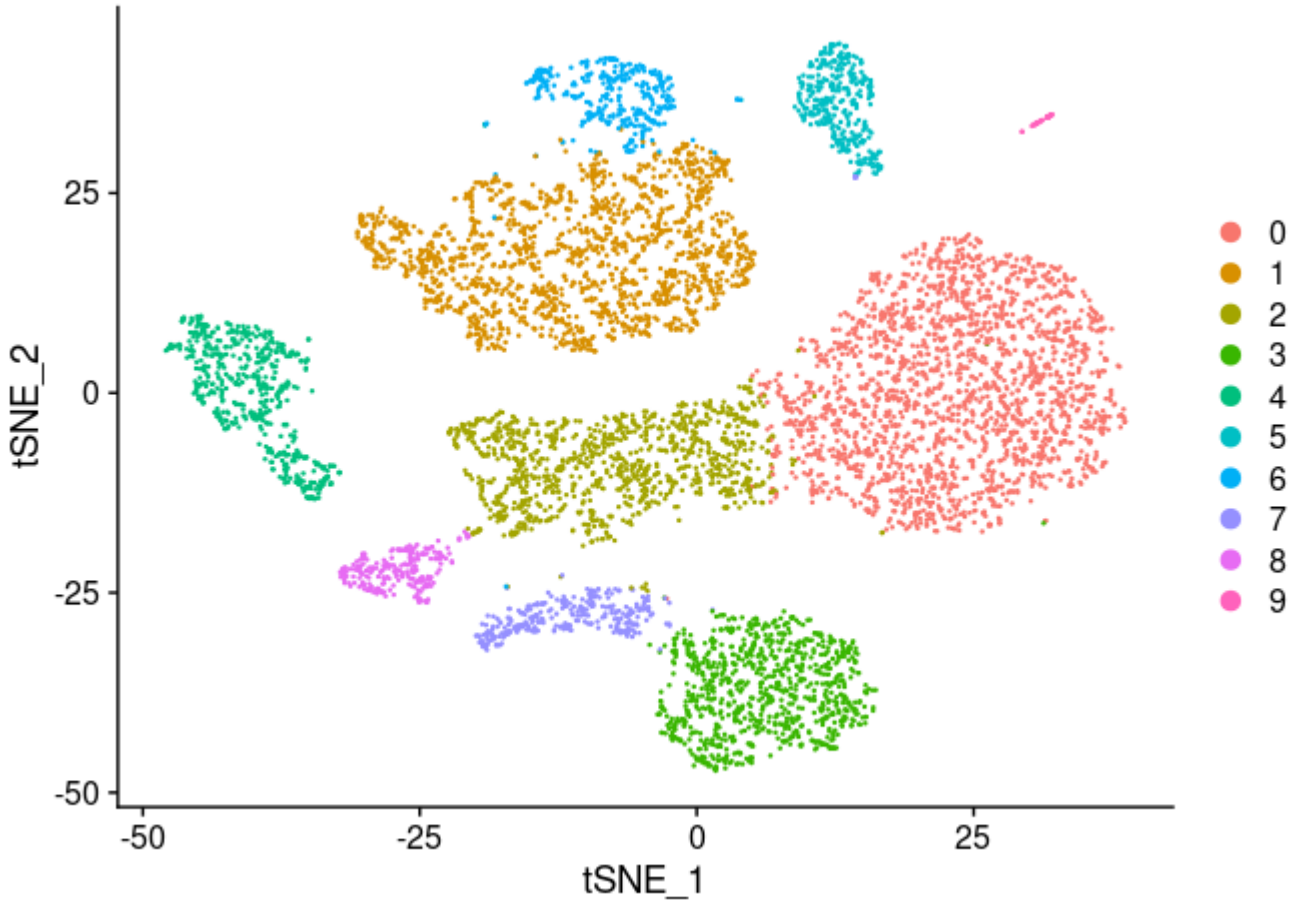
t-distributed Stochastic Neighbor Embedding (t-SNE)

Another dimension reduction method is **tSNE**. We can use this and plot the cells, by specifying the "reduction" method in the function.

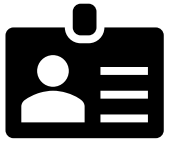
https://towardsdatascience.com/t-sne-clearly-explained-d84c537f53a

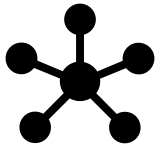
UNSW SYDNEY

Visualisation: tSNE



Cell annotation pipeline





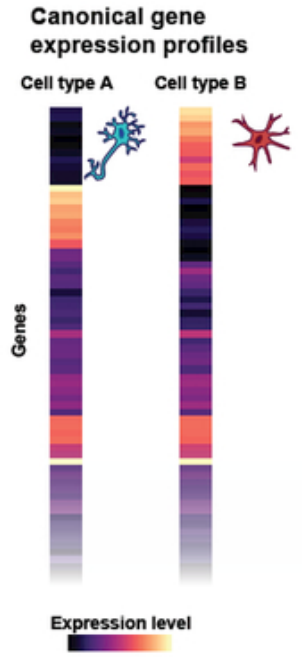
1 Clustering

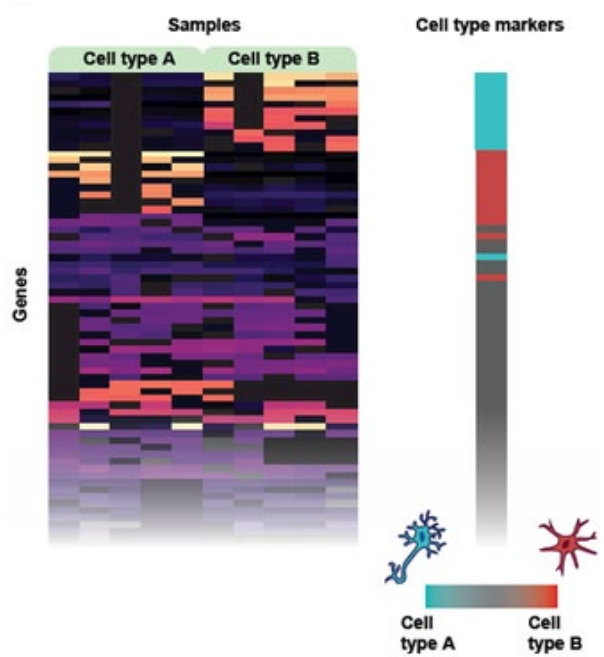
2 **Marker genes**

3 Visualisation

Marker genes: differential expression

- To find marker genes, we need to find the genes which are highly expressed in one cluster but not in others

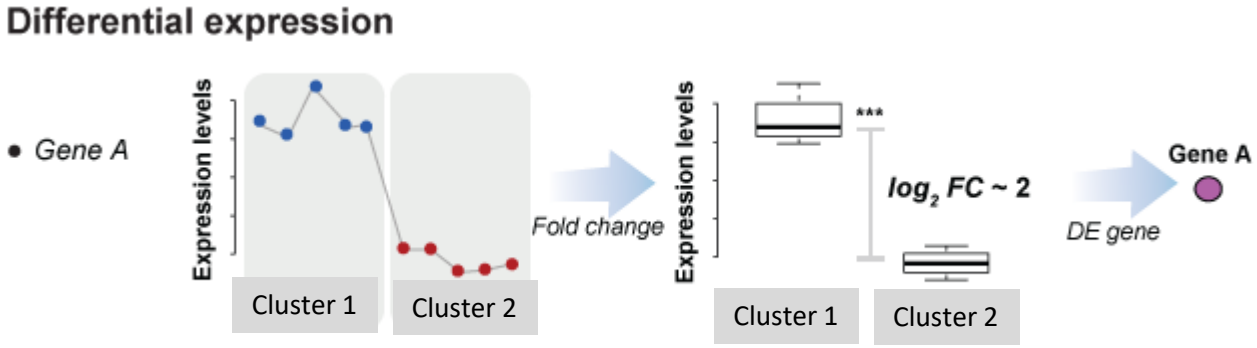




https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6195469/

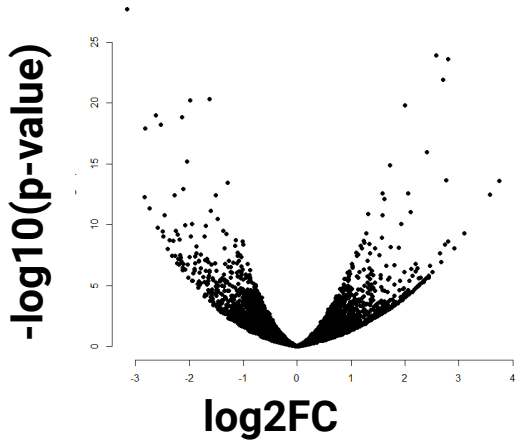
Differential expression: overview

Statistical analysis to discover **quantitative changes** in **expression levels** between **clusters**

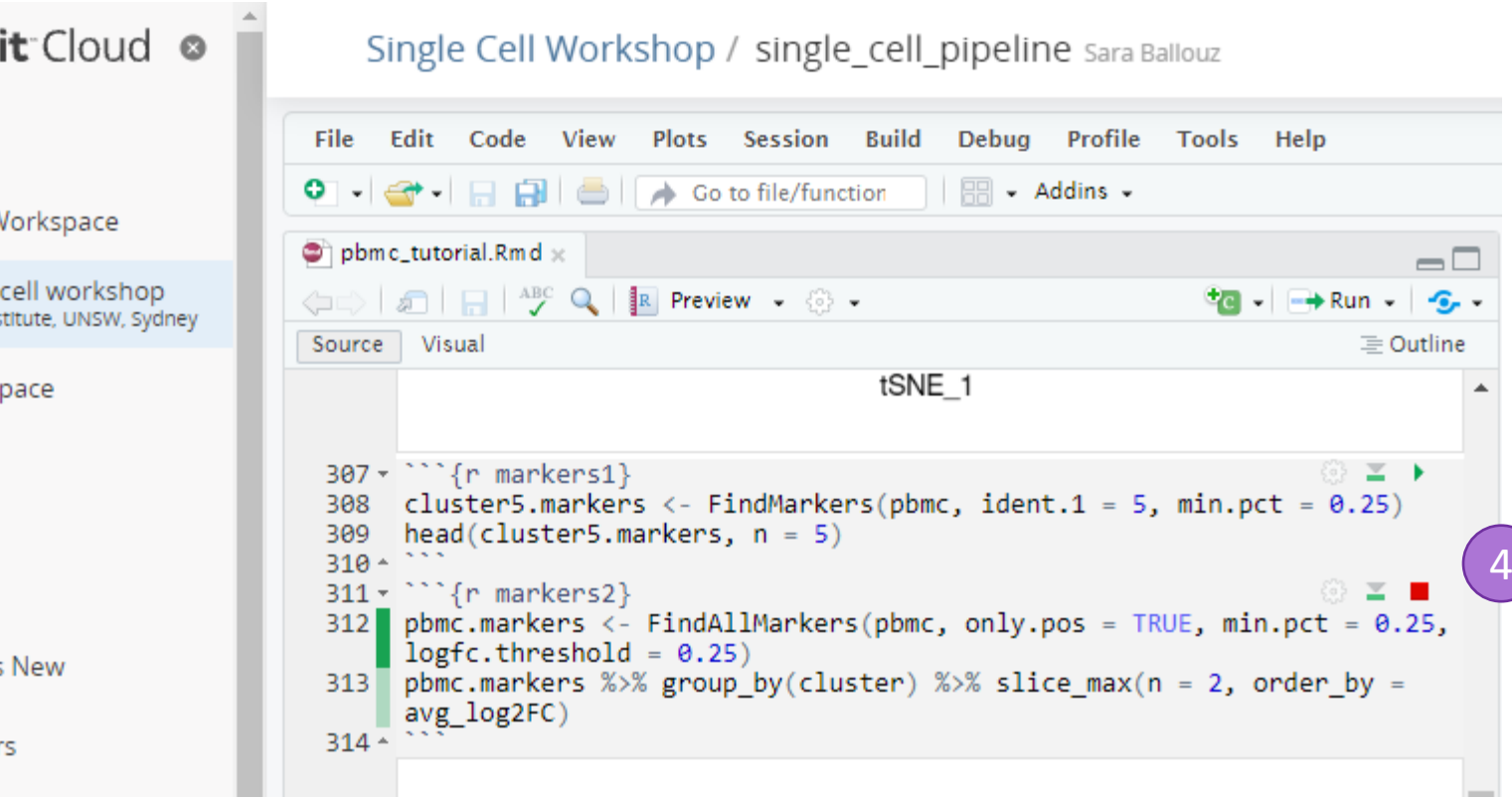


$$log2FC = \log_2\left(\frac{Average\ expression\ cluster\ 1}{Average\ expression\ cluster\ 2}\right)$$

Volcano plot



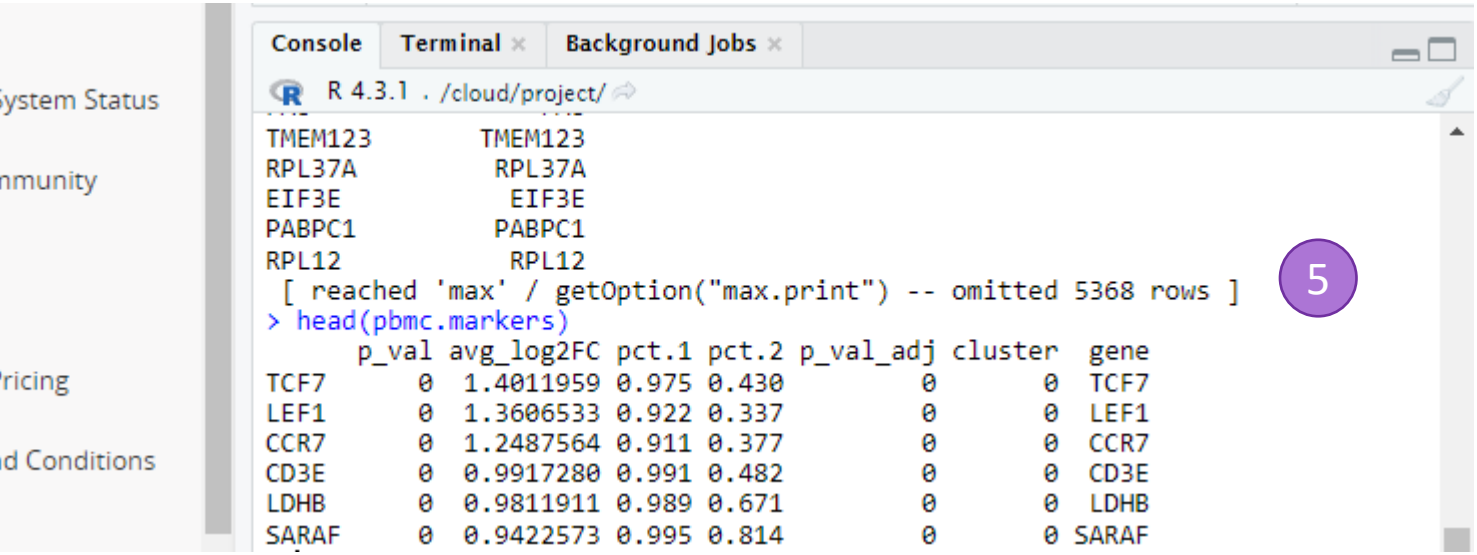
Differential gene expression analysis



To identify markers, we run the *FindAllMarkers* function. This compares each cluster to all other clusters. We select that at least 25% of the cells express the genes tested. We pick a threshold for the average log2 fold change (0.25).

Then we order the results by fold change.

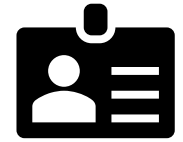
Differential gene expression analysis

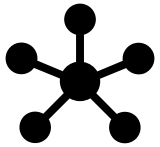


Take a closer look at the output, type into the console and press enter: **head(pbmc.markers)**

- Metrics used to determine cluster genes for Cluster 0
  - **p_val:** statistical significance between Cluster 0 vs rest
  - **FC:** Average gene expression difference in Cluster 0 vs other clusters
  - **pct.1:** Percentage of cells expressing gene in Cluster 0
  - **pct.2:** Percentage of cells expressing gene in all other clusters

Cell annotation pipeline





1 Clustering

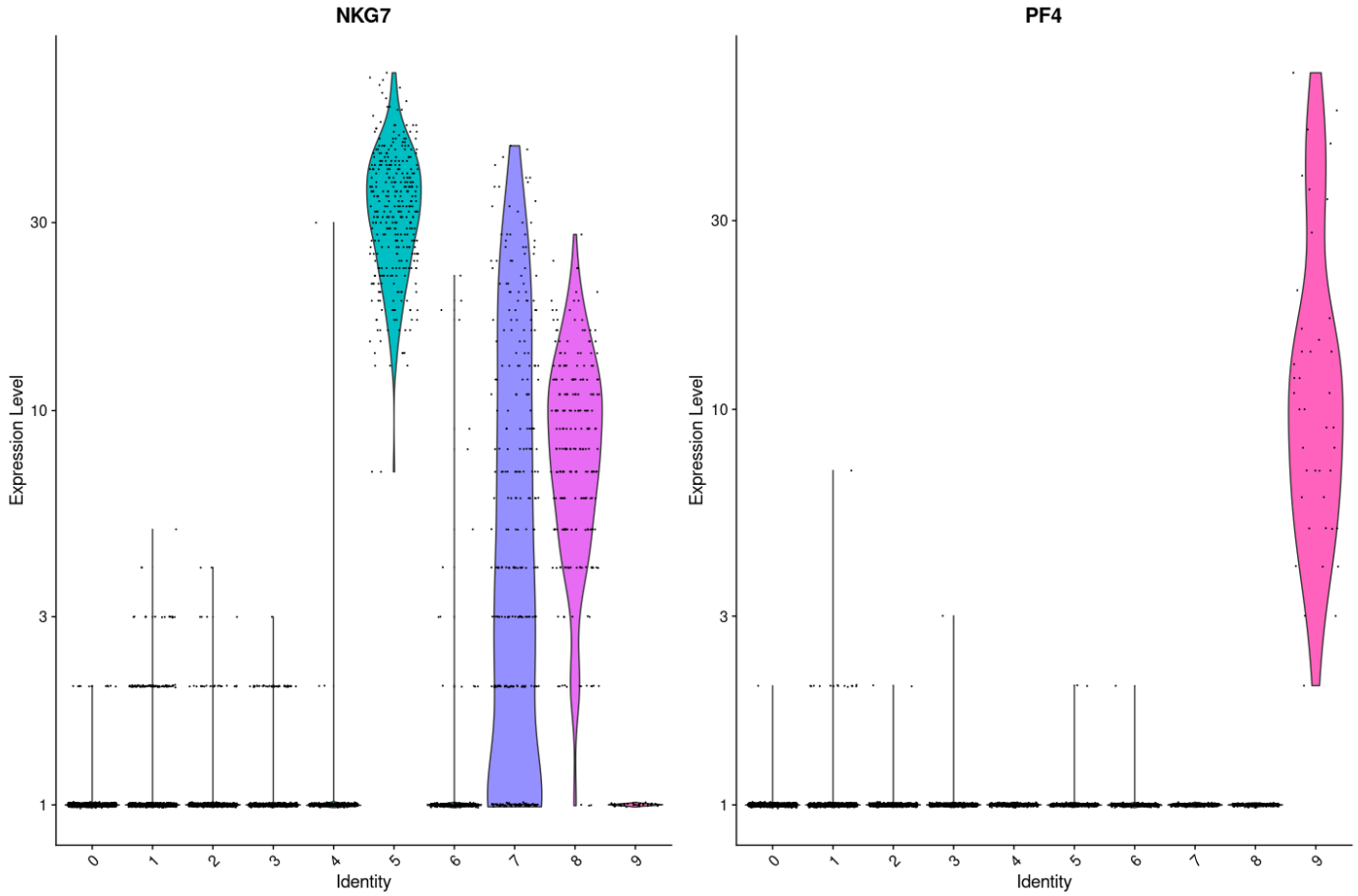
2 Marker genes

3 **Visualisation**

Visualisation

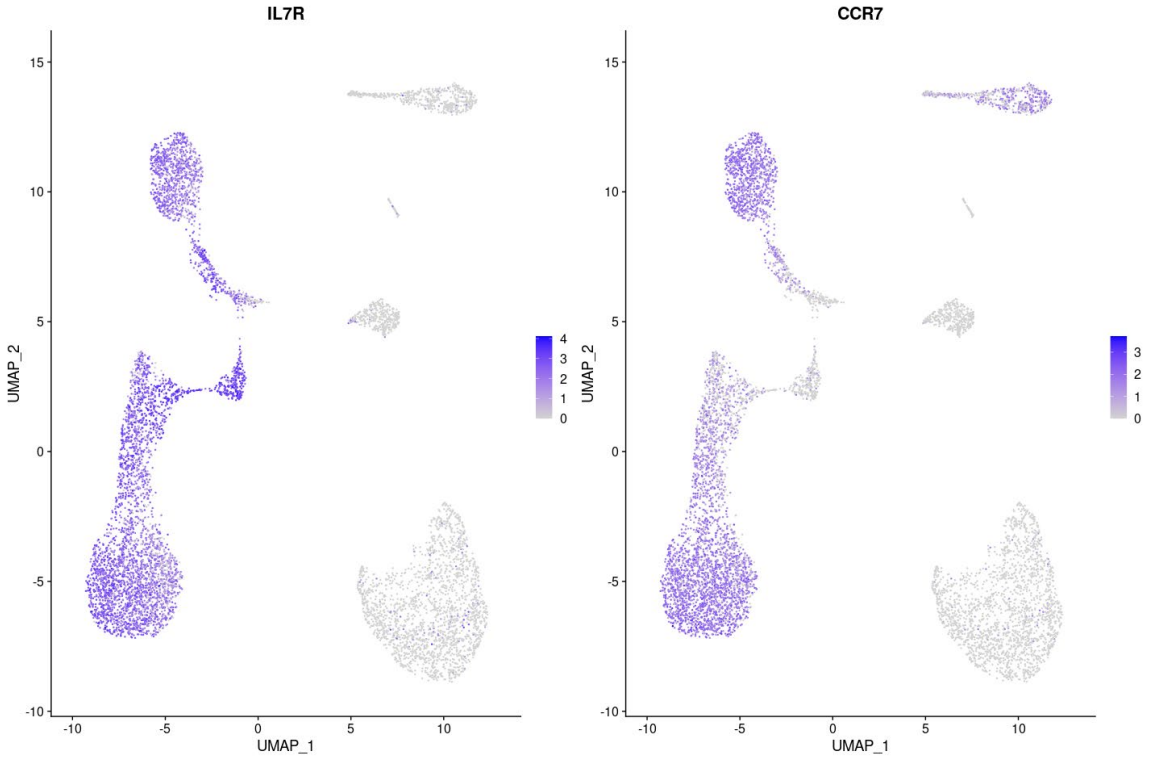
- Visualise marker genes as a sanity check
  - Violin plots
  - UMAP/tSNE
  - Heatmap
  - Dot plots

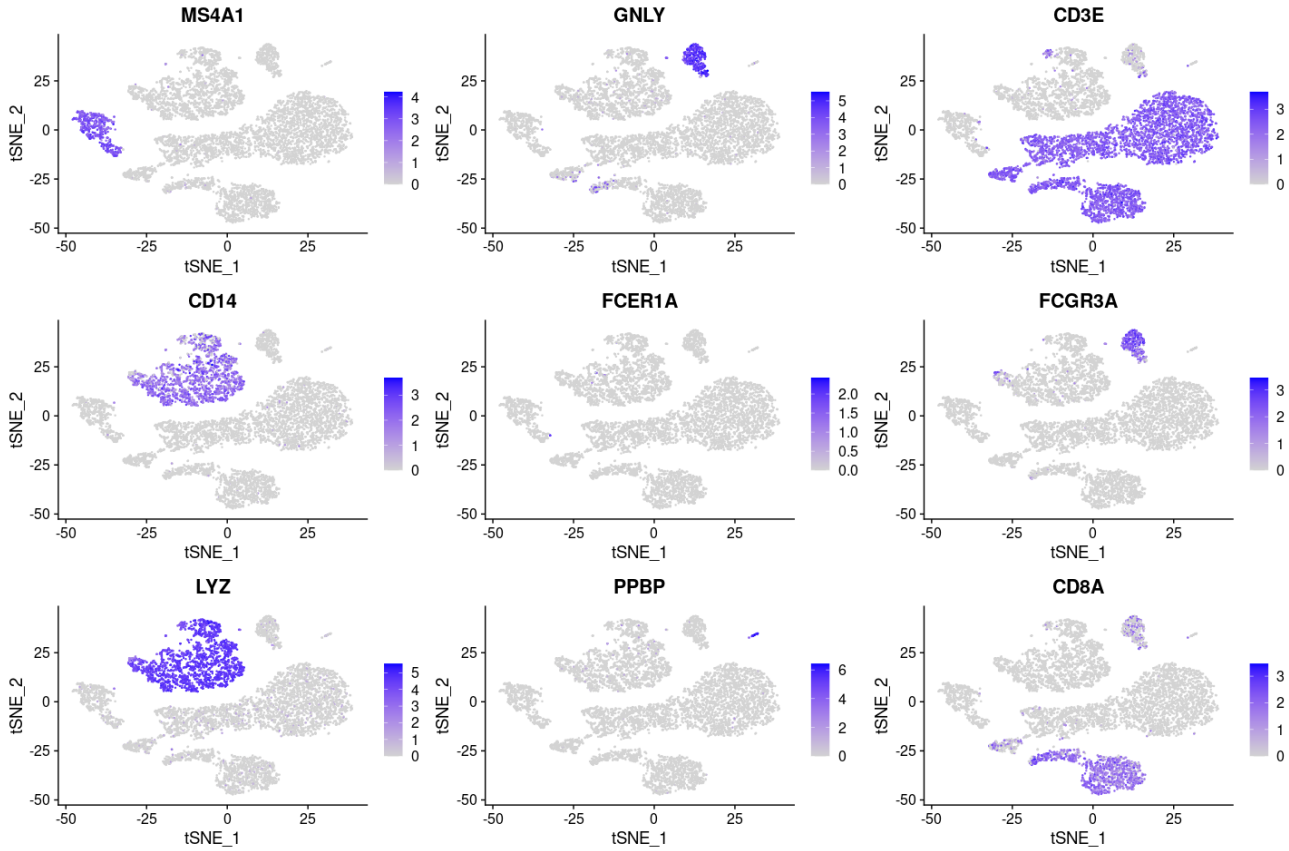
Visualisation: Violin plots



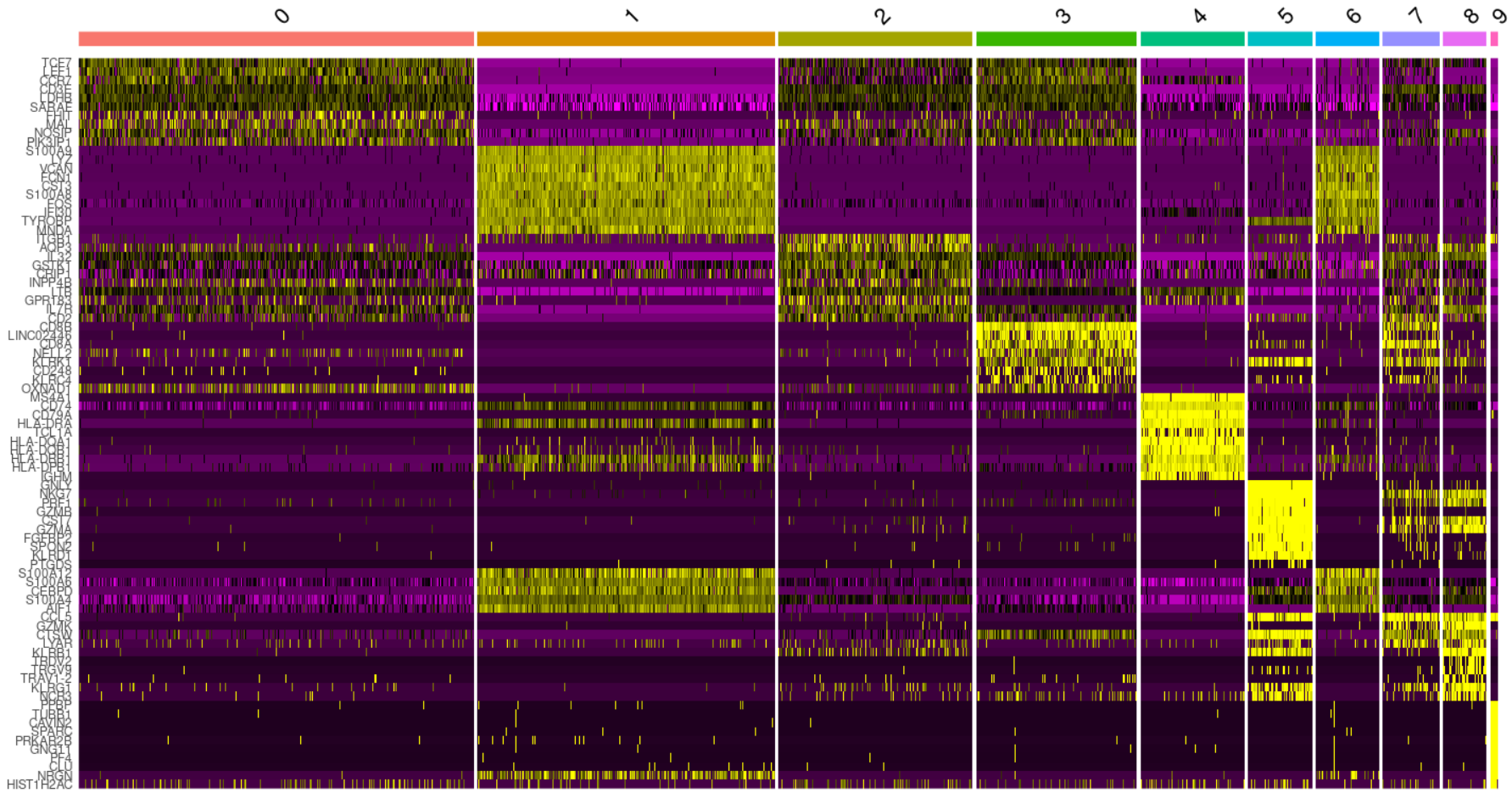
- Each dot is a cell
- X-axis shows clusters
- Y-axis shows gene expression level

Visualisation: UMAP/tSNE





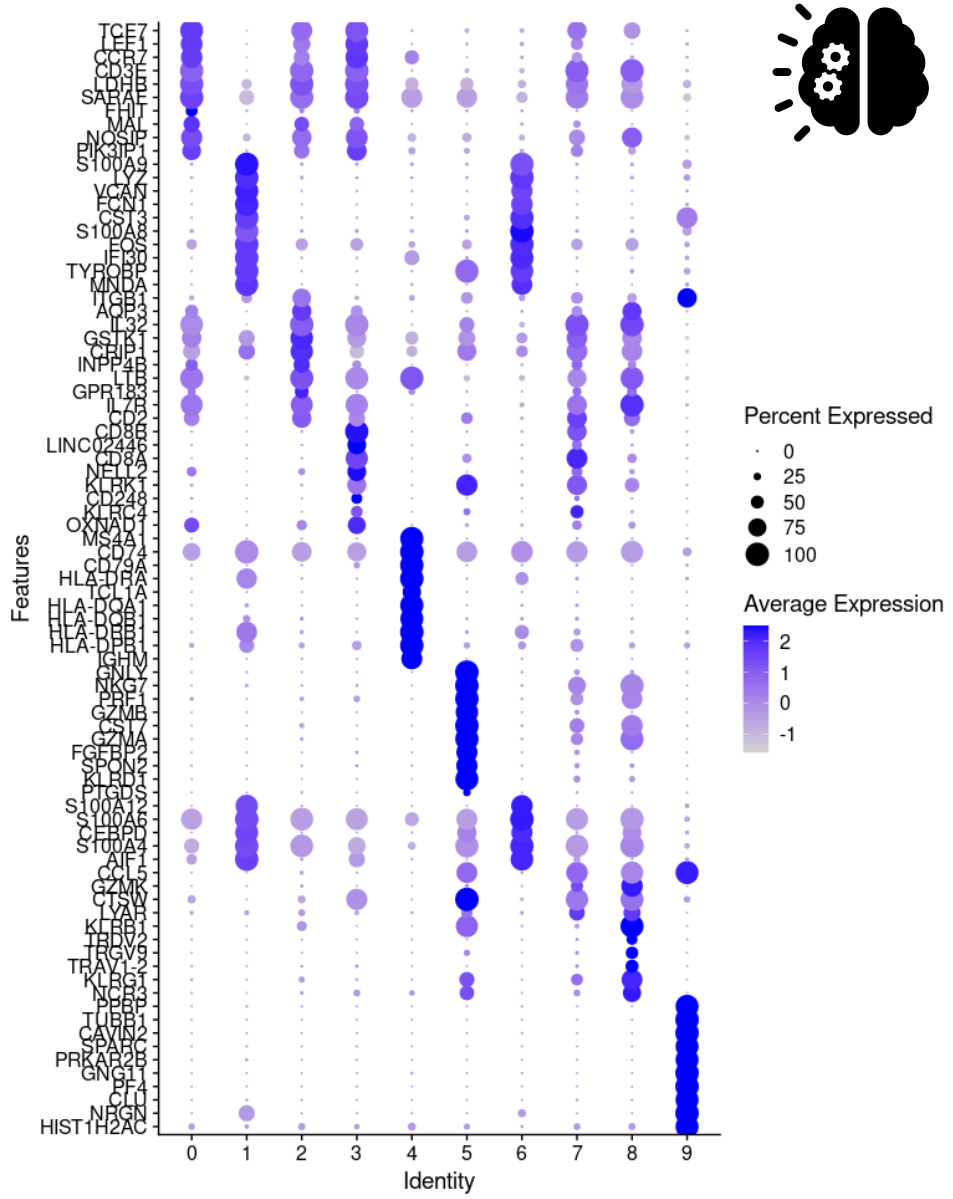
Visualisation: Heatmaps



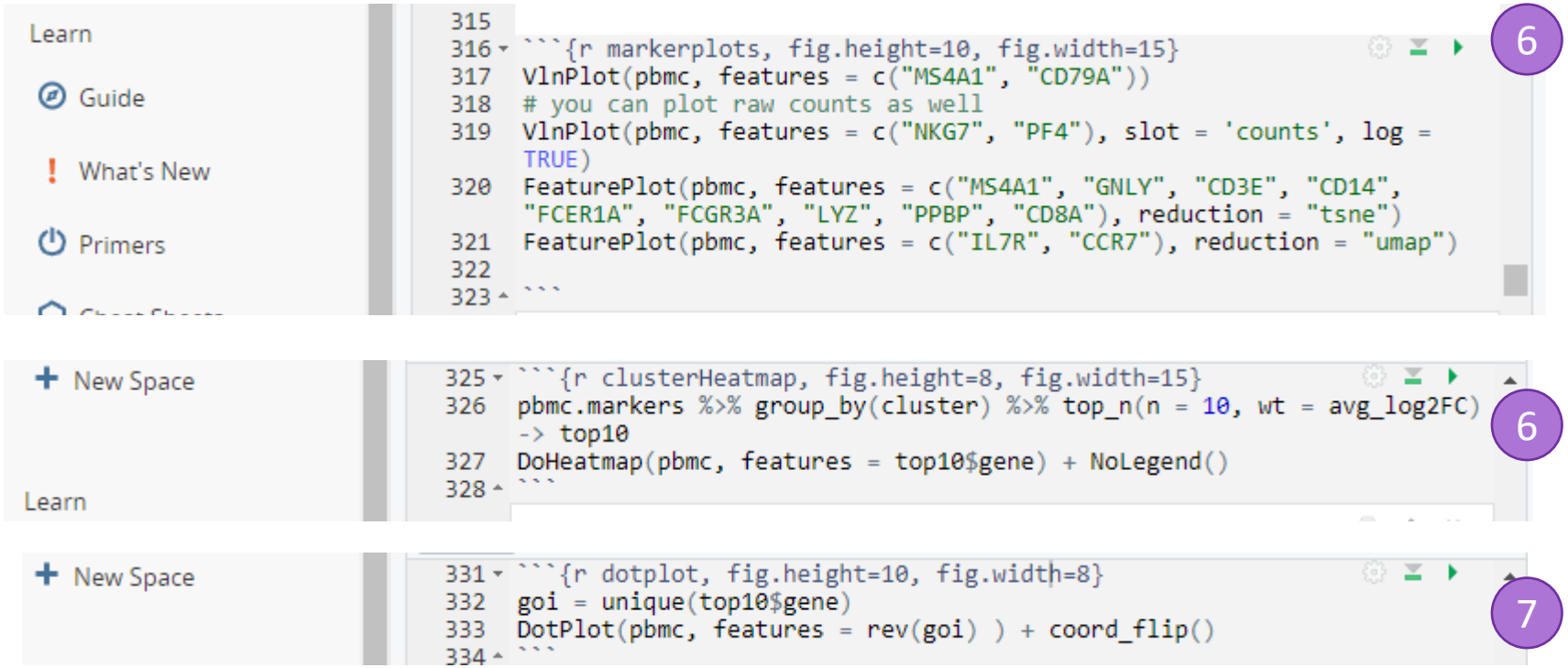
- Each row is a gene
- Each column is a cell
- Bright colours indicate higher expression of the gene

Visualisation: Dotplots

- X-axis is a cluster
- Y-axis is a gene
- Colours indicate higher expression of the gene
- Size of dot reflects percentage of cells in that cluster that express gene



Visualisations




```
315
316 ```{r markerplots, fig.height=10, fig.width=15}
317 VlnPlot(pbmc, features = c("MS4A1", "CD79A"))
318 # you can plot raw counts as well
319 VlnPlot(pbmc, features = c("NKG7", "PF4"), slot = 'counts', log =
    TRUE)
320 FeaturePlot(pbmc, features = c("MS4A1", "GNLY", "CD3E", "CD14",
    "FCER1A", "FCGR3A", "LYZ", "PPBP", "CD8A"), reduction = "tsne")
321 FeaturePlot(pbmc, features = c("IL7R", "CCR7"), reduction = "umap")
322
323 ```
```

6

Violin and feature plots. Here we specified different “features” (genes) to plot.

```
325 ```{r clusterHeatmap, fig.height=8, fig.width=15}
326 pbmc.markers %>% group_by(cluster) %>% top_n(n = 10, wt = avg_log2FC)
    -> top10
327 DoHeatmap(pbmc, features = top10$gene) + NoLegend()
328 ```
```

6

Heatmap of top 10 markers per cluster.

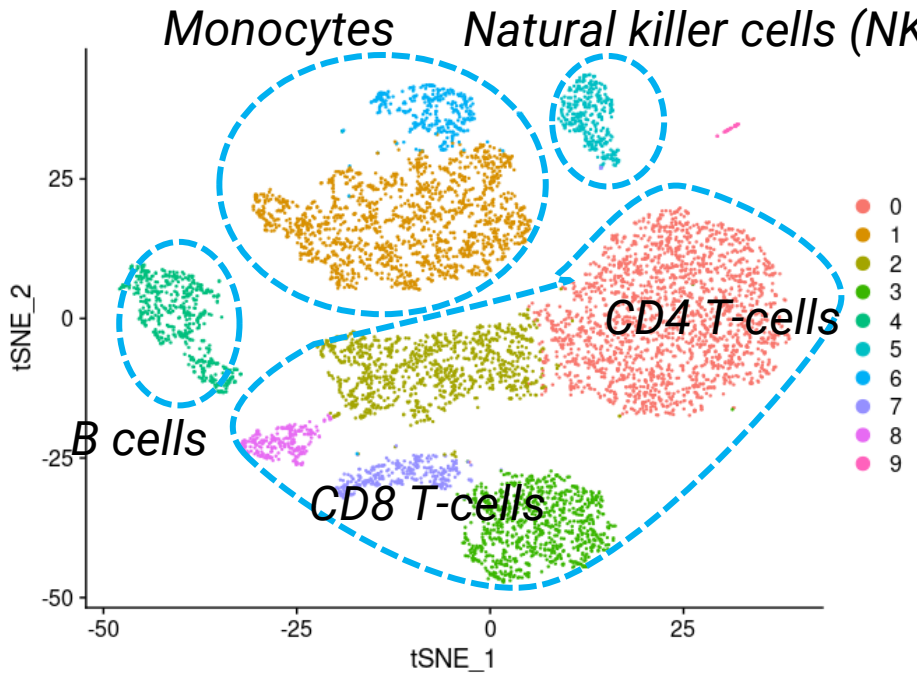
```
331 ```{r dotplot, fig.height=10, fig.width=8}
332 goi = unique(top10$gene)
333 DotPlot(pbmc, features = rev(goi) ) + coord_flip()
334 ```
```

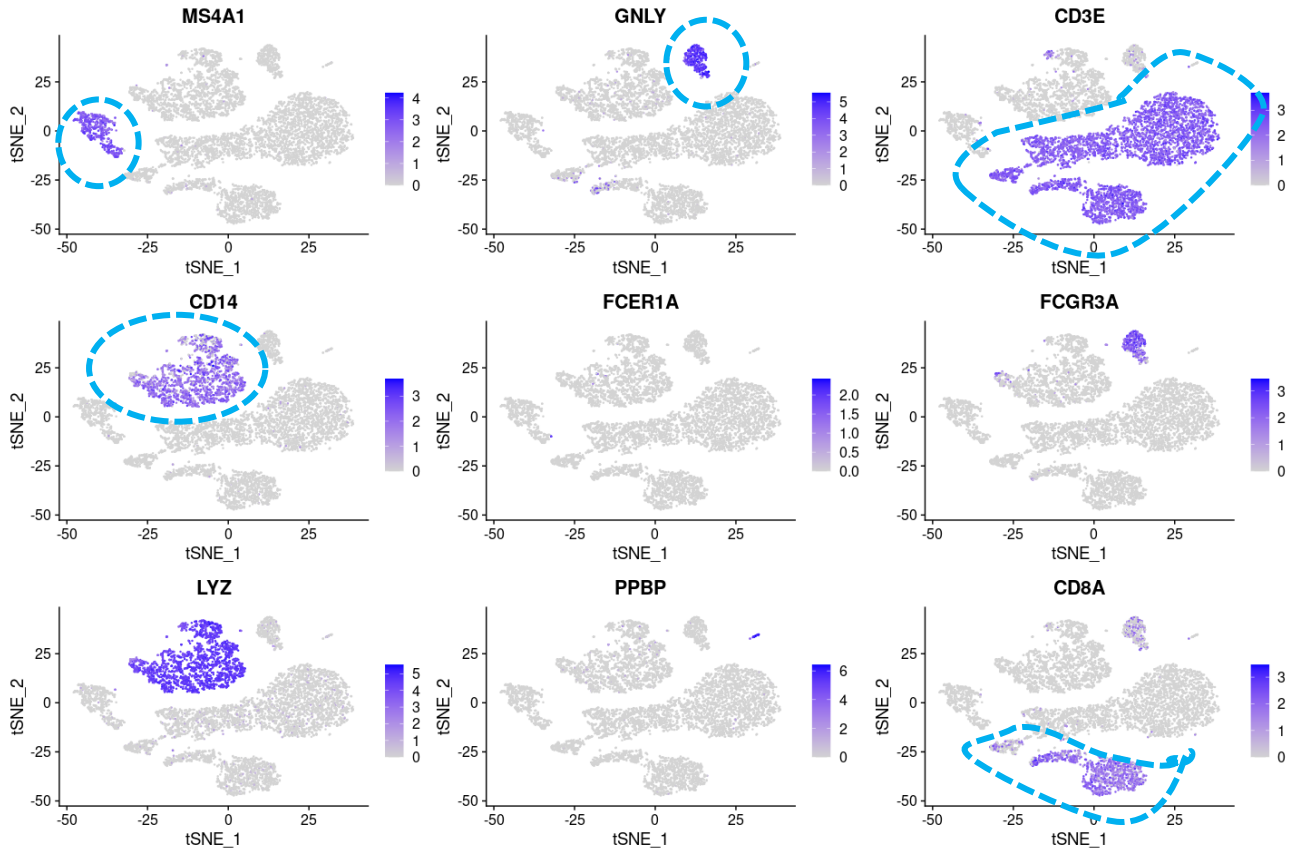
7

Dotplot of top 10 markers per cluster.

Cell annotation – manual

- Use known marker genes to identify clusters.



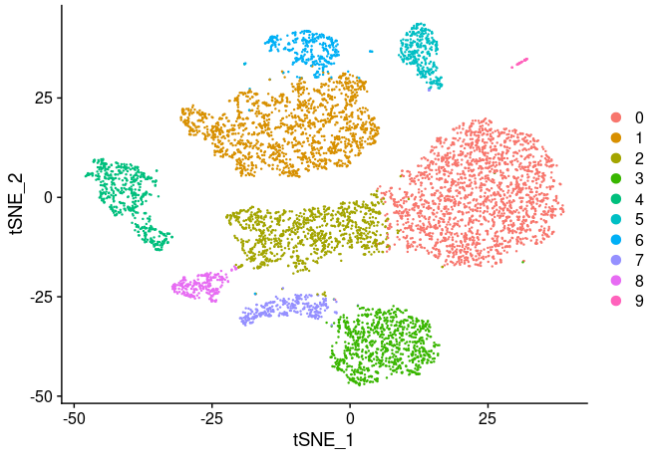


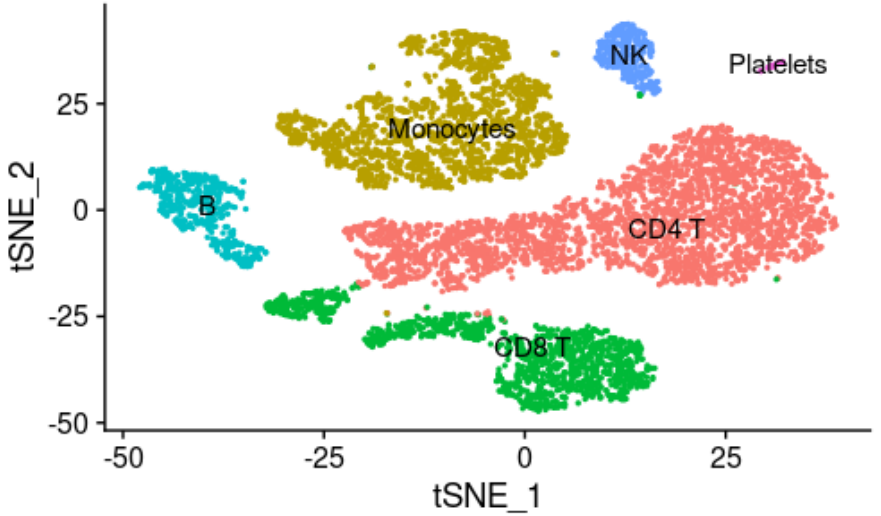
Cell annotation – manual

```
355
356 ```{r manual}
357 new.cluster.ids <- c("CD4 T", "Monocytes", "CD4 T", "CD8 T", "B",
    "NK", "Monocytes", "CD8 T", "CD8 T", "Platelets")
358 names(new.cluster.ids) <- levels(pbmc)
359 pbmc <- RenameIdents(pbmc, new.cluster.ids)
360 DimPlot(pbmc, reduction = 'tsne', label = TRUE, pt.size = 0.5) +
    NoLegend()
361 ```
```

7

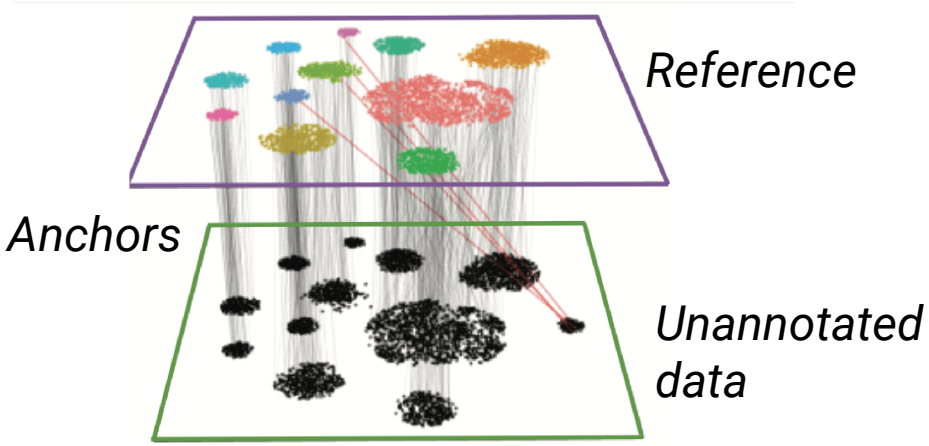
Match each cluster with a cell type, and rename the cells. Plot.

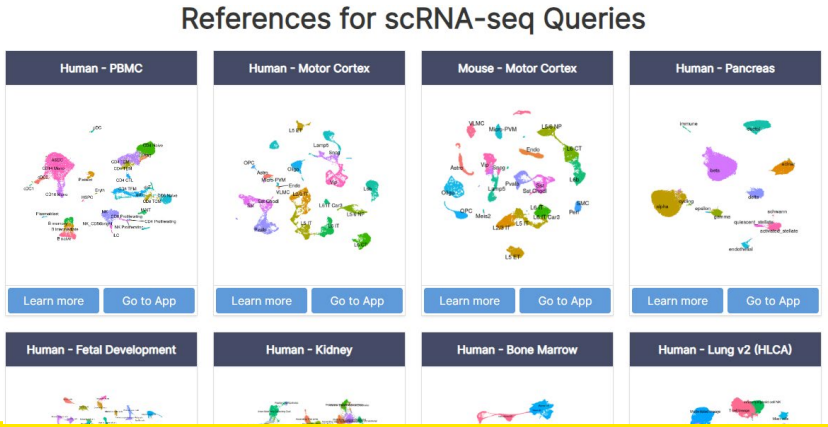




Cell annotation - automatic

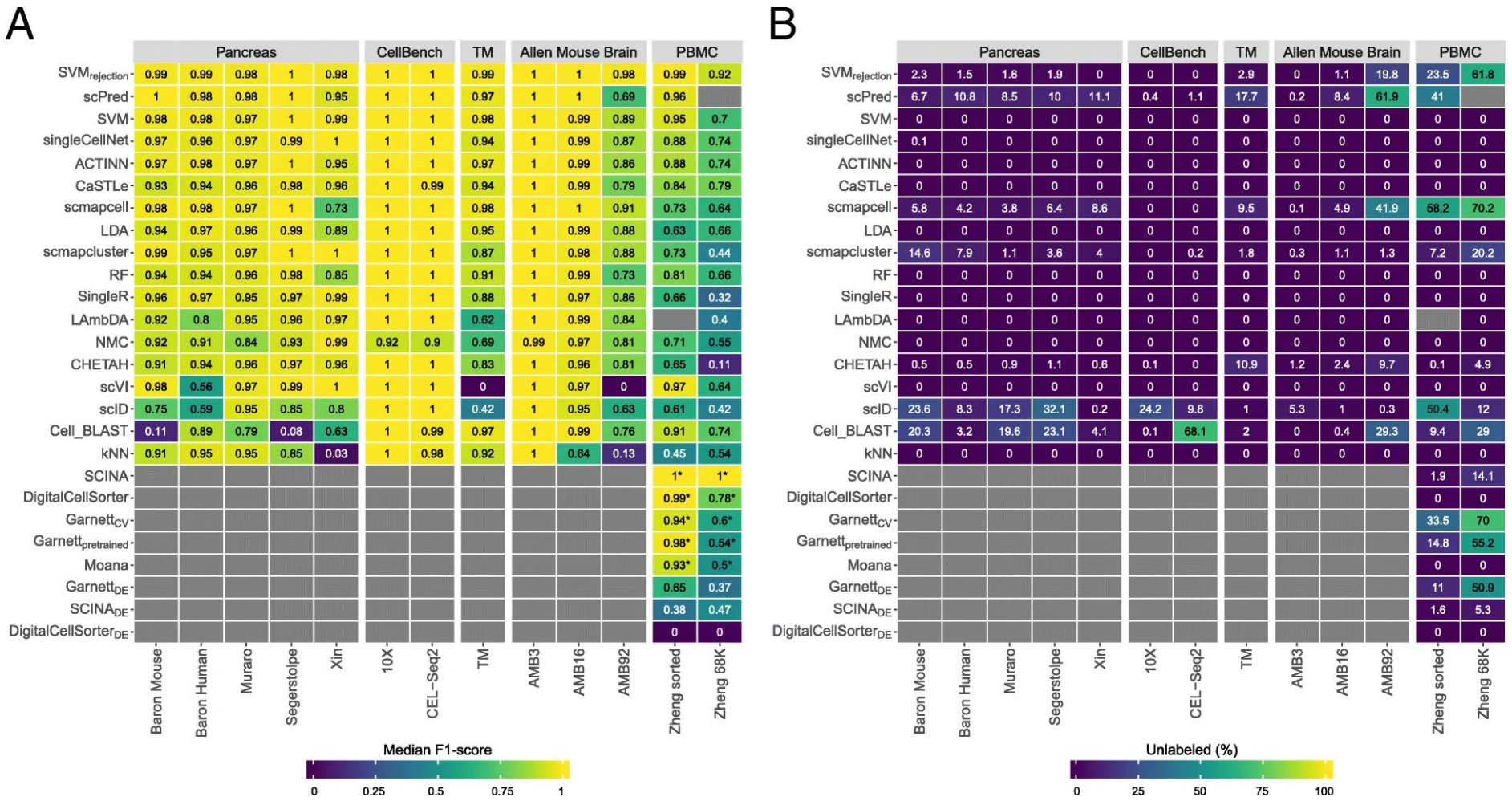
- Use a reference dataset and compare clusters to reference clusters. Transfer the label of the reference cluster most similar.
- Online tool to do this is **Azimuth**, which has several references, including PBMCs.
- Many other annotation tools!





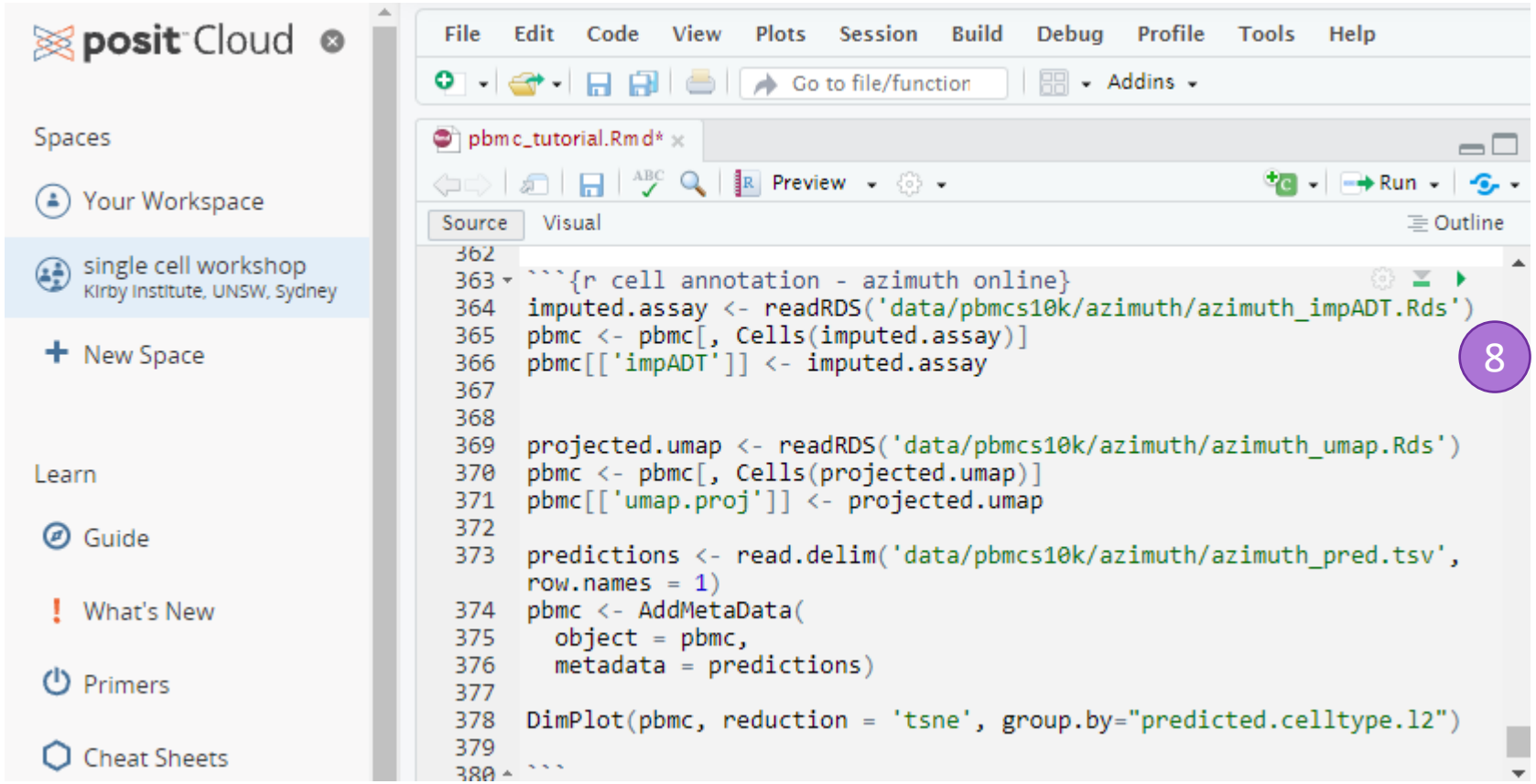
https://azimuth.hubmapconsortium.org/

Annotation tools



https://link.springer.com/article/10.1186/s13059-019-1795-z/tables/1

Cell annotation – Azimuth demo



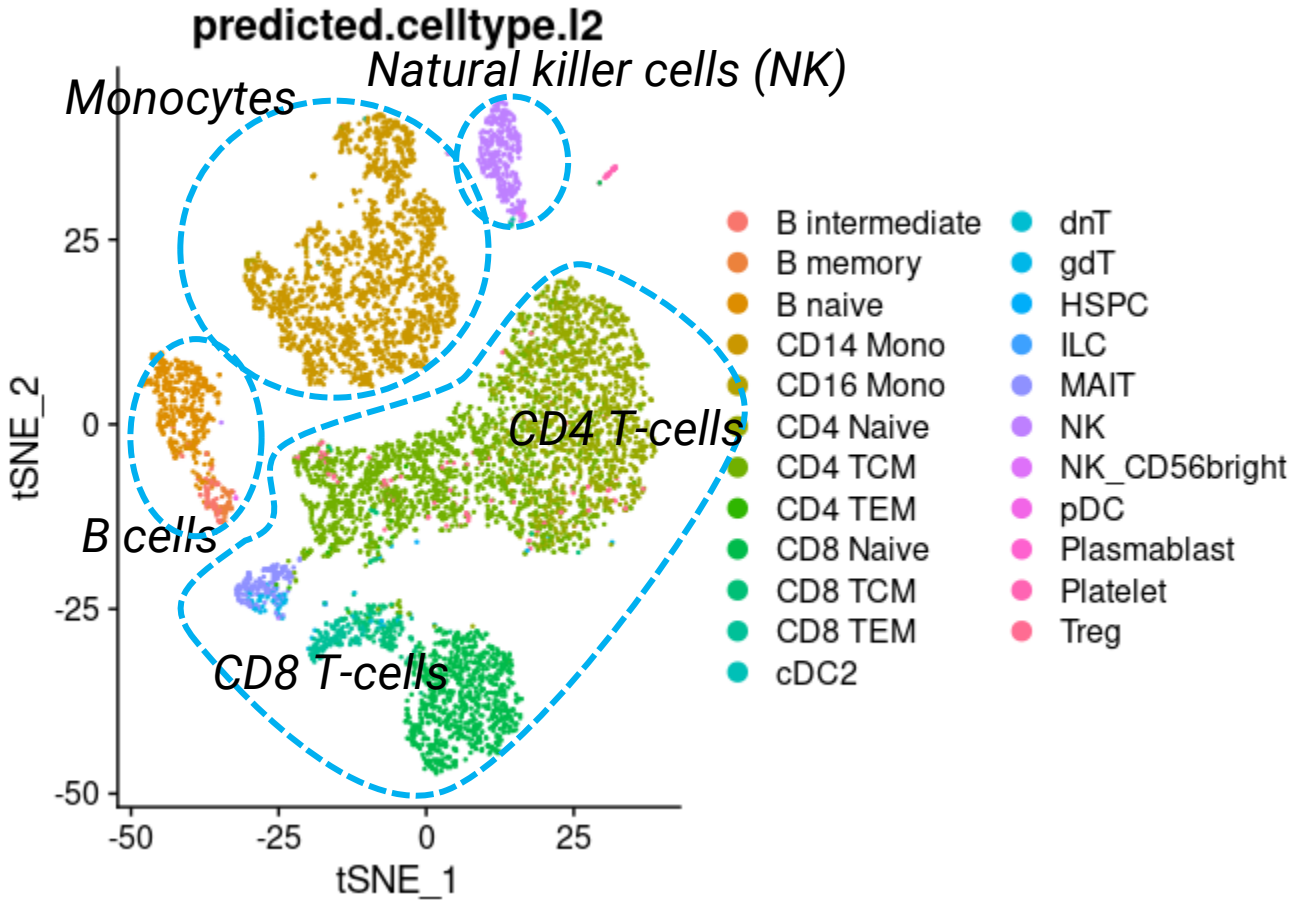
Upload and store the predictions, then plot “predicted.celltype.l2”

Manual versus automatic

Pretty good match!

More granular resolution can be selected with “l3”

And broader cell-types with “l1”



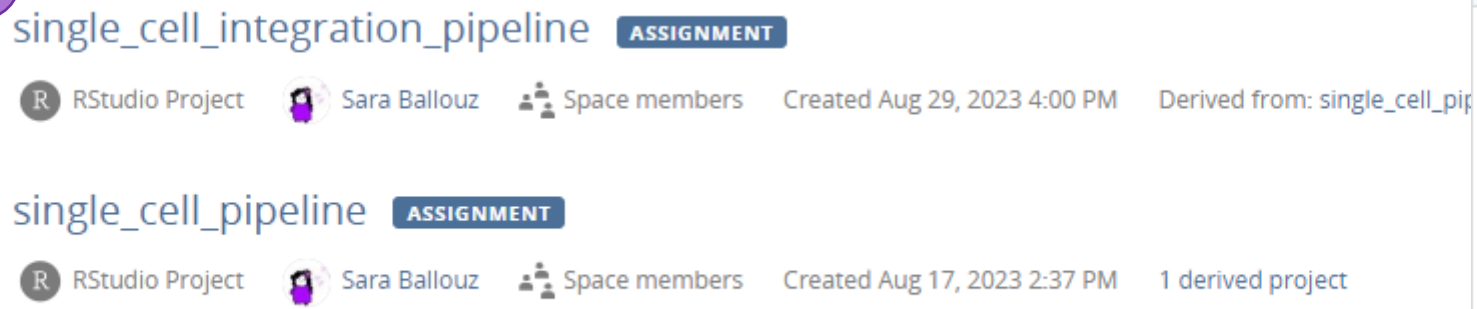
Overview

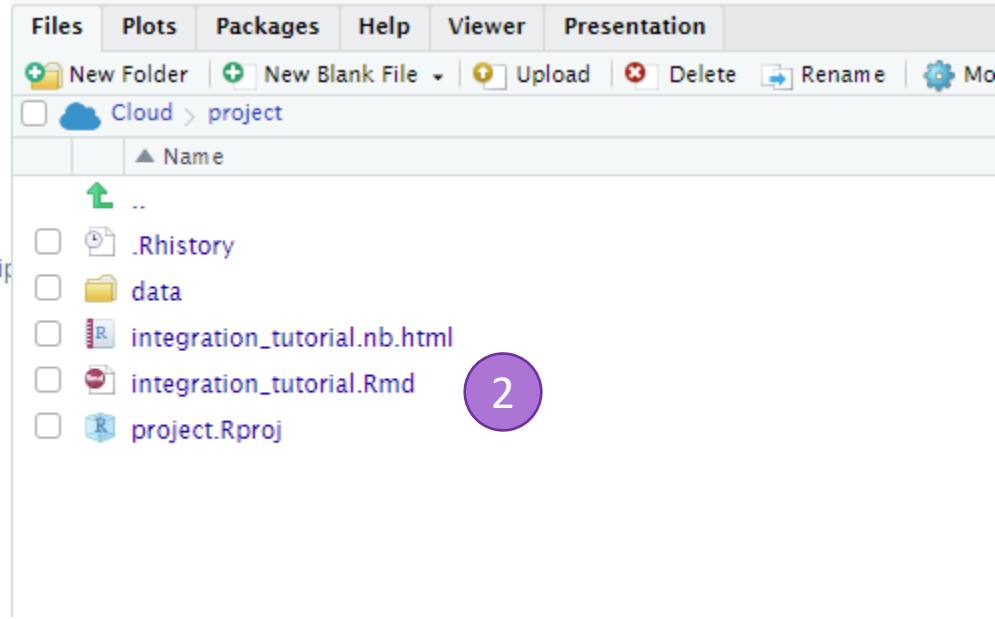
1. Introduction to single cell
2. Setting up Rstudio, data and count matrix
3. Pre-processing
4. Application 1: Cell annotation
5. **Application 2: Case vs Control**

Switch!

- Before we get started, save and close your project.
- And now open the **single_cell_integration_pipeline** workshop.
- Open the “integration_tutorial.Rmd” file

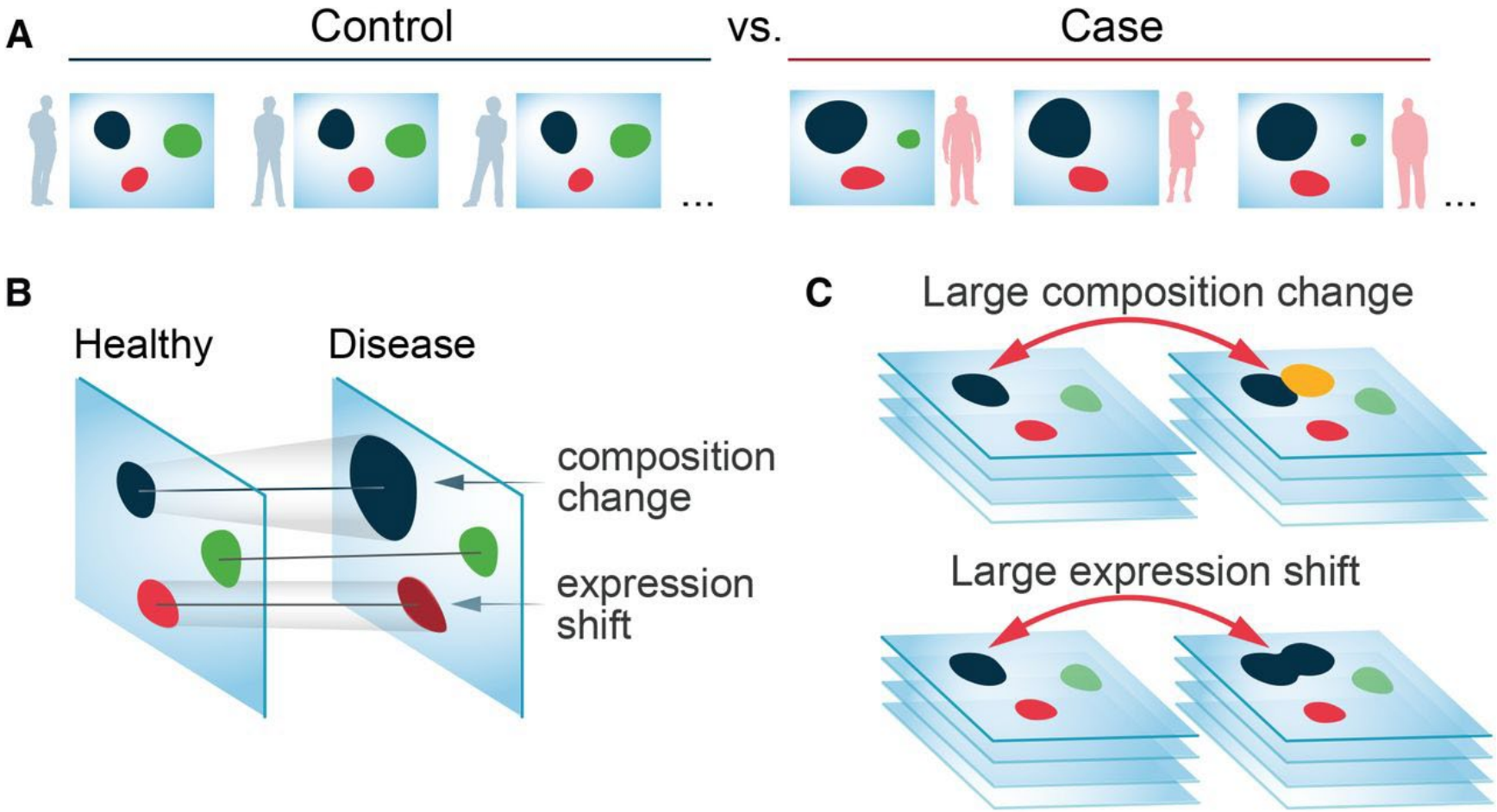
1





2

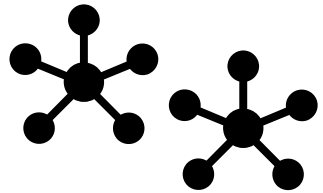
Analysis case versus control



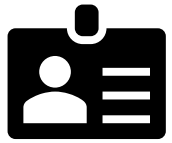
https://www.biorxiv.org/content/10.1101/2022.03.15.484475v1.full

UNSW SYDNEY

Case versus control pipeline



1 **Integration**

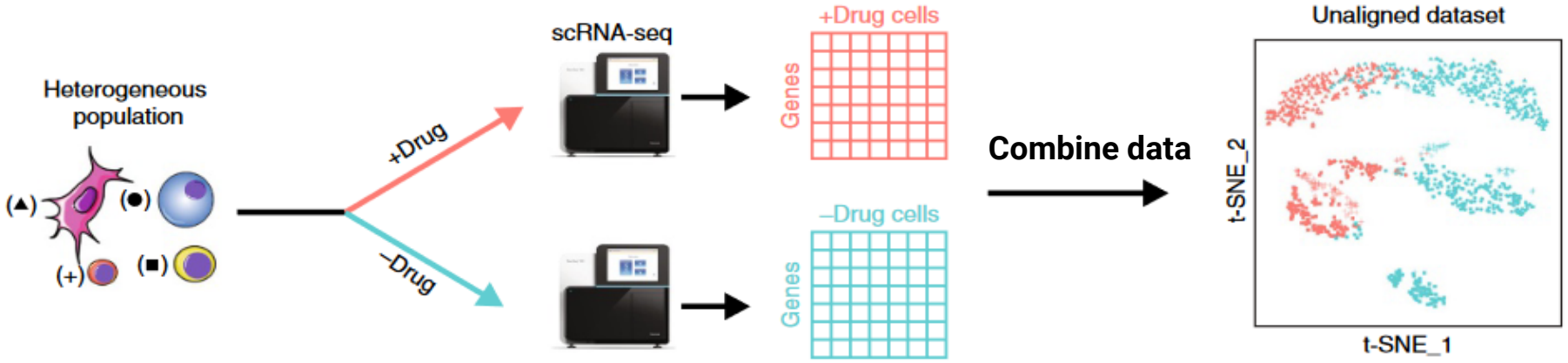


2 Cell annotation

3 Differential

- Composition
- Expression

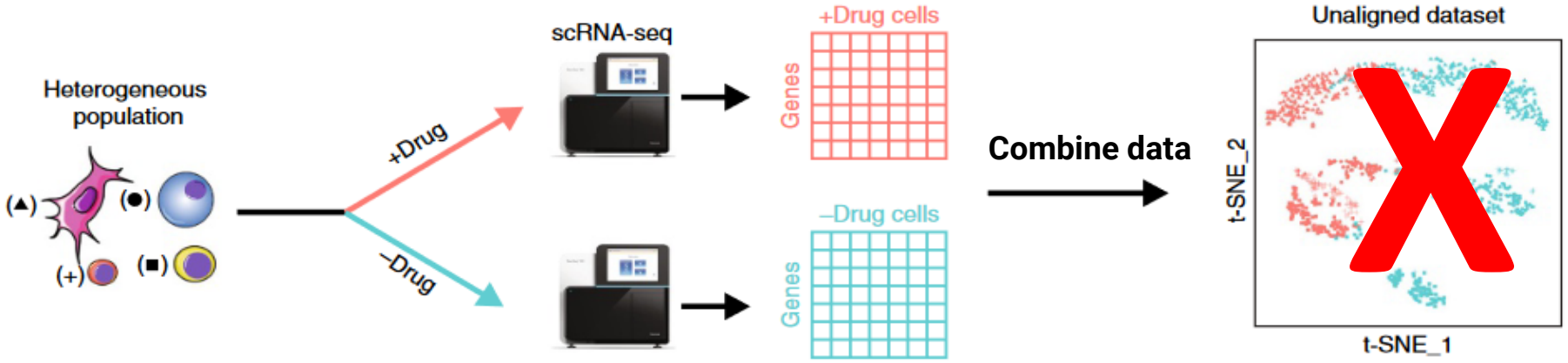
Two different samples may have batch effects



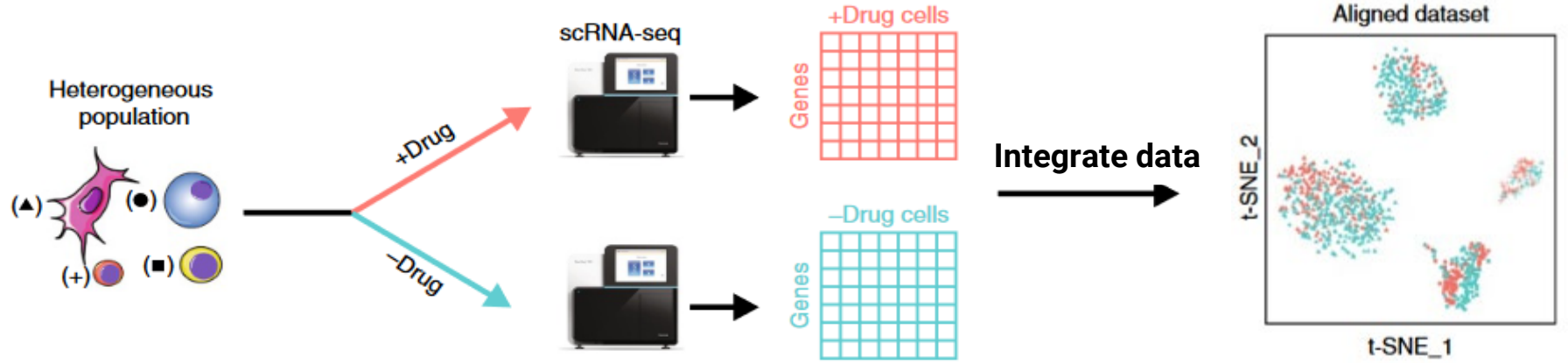
Cluster differences not due to biology

https://www.nature.com/articles/nbt.4096

Two different samples may have batch effects



Cluster differences not due to biology

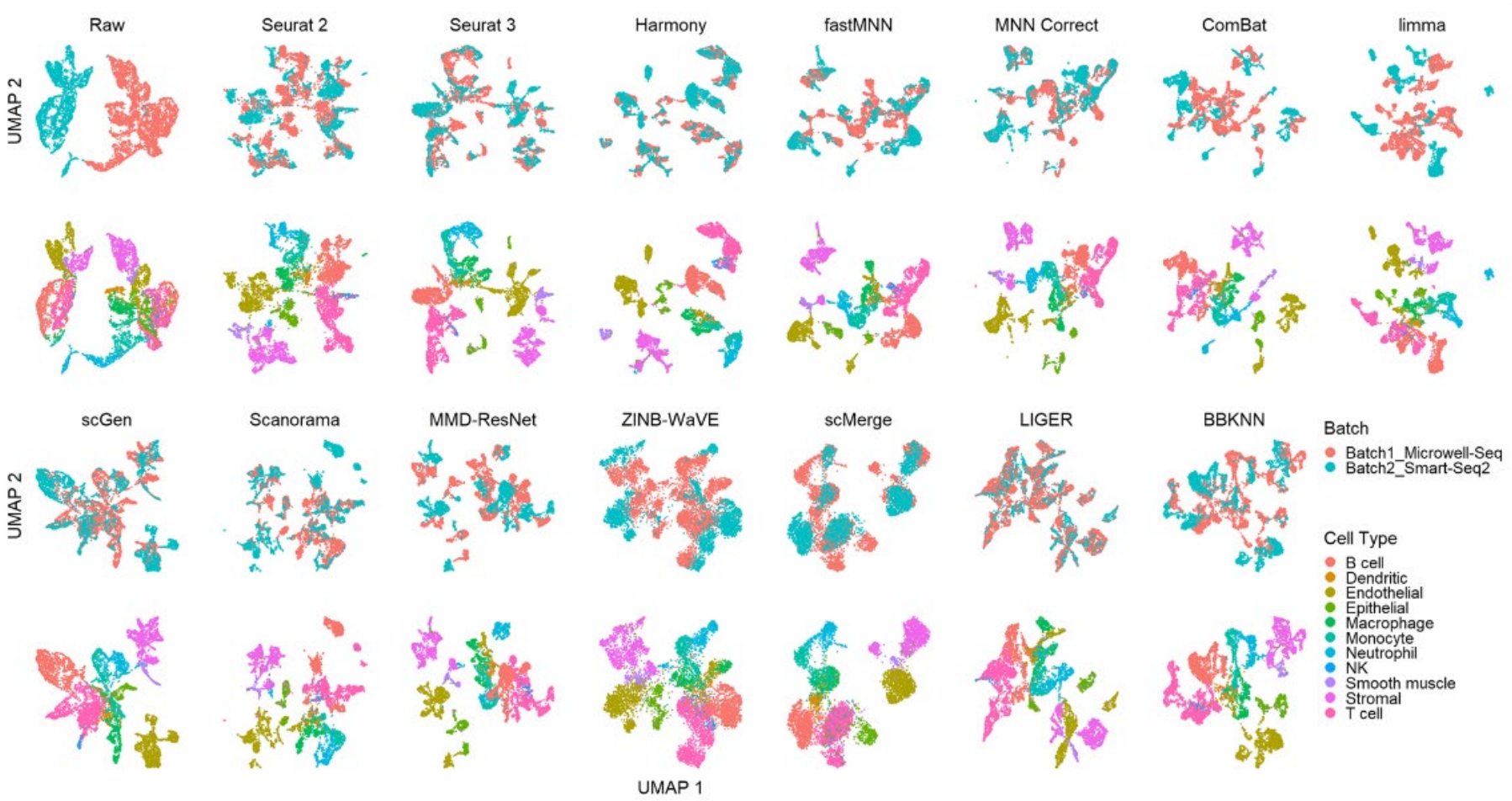


Cluster differences due to biology

- Test using differential analysis

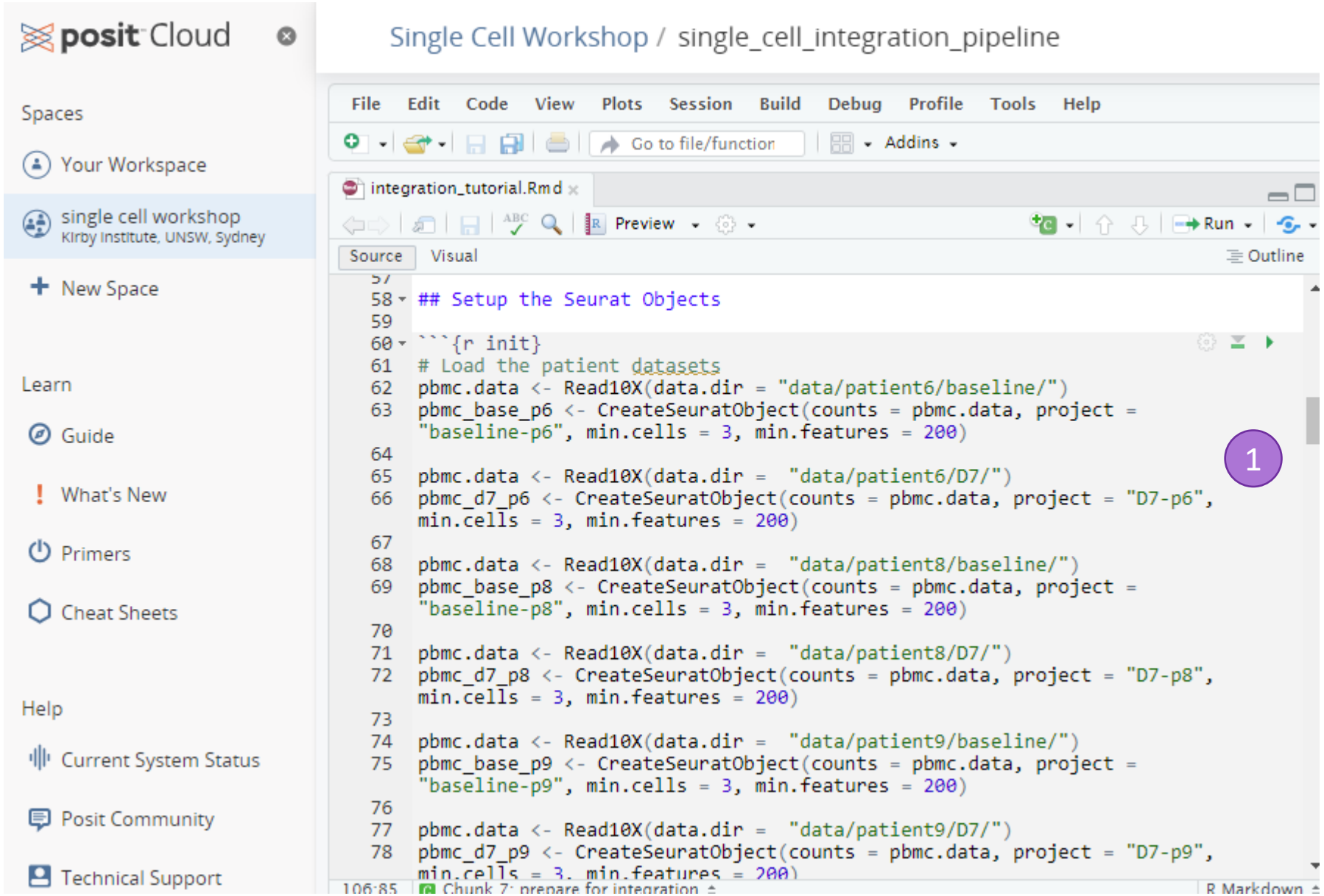
https://www.nature.com/articles/nbt.4096

Many batch integration methods



https://genomebiology.biomedcentral.com/articles/10.1186/s13059-019-1850-9

UNSW SYDNEY



Load in data from three cancer patients: baseline and then day 7 post infusion.

```r
```{r mito, fig.height=7, fig.width=13}
pbmc.list = list(pbmc_base_p6, pbmc_d7_p6, pbmc_base_p8, pbmc_d7_p8,
pbmc_base_p9, pbmc_d7_p9)
names(pbmc.list) = c("Baseline-p6", "D7-p6","Baseline-p8",
"D7-p8","Baseline-p9", "D7-p9")
```

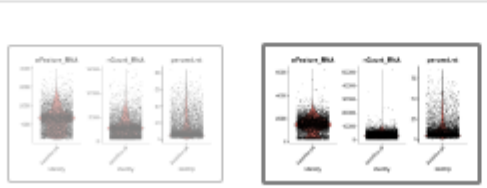
2

To simplify our work, we join all the Seurat objects as a list. And label each element in the list.

We “loop” through the list to perform all the regular steps – this is the MT-ratio calculation.

3

```r
for( i in 1:length(pbmc.list)){
  pbmc.list[[i]][["percent.mt"]] <- PercentageFeatureSet(pbmc.list[[i]],
pattern = "^MT-")
}
```
```{r }
#Visualize QC metrics as a violin plot
VlnPlot( pbmc.list[[1]], features = c("nFeature_RNA", "nCount_RNA",
"percent.mt"), ncol = 3)
VlnPlot( pbmc.list[["Baseline-p9"]], features = c("nFeature_RNA",
"nCount_RNA", "percent.mt"), ncol = 3)
```


```

```r
```{r subsetting}
for( i in 1:length(pbmc.list)){
  pbmc.list[[i]] <- subset(pbmc.list[[i]], subset = nFeature_RNA > 200 & nFeature_RNA < 2500 & percent.mt < 7)
}
```
```

3 We subset the individual datasets jointly again. We've changed the MT threshold.

```r
```{r prepare for integration}
pbmc.list <- lapply(X = pbmc.list, FUN = SCTransform)
features <- SelectIntegrationFeatures(object.list = pbmc.list, nfeatures = 3000)
pbmc.list <- PrepSCTIntegration(object.list = pbmc.list, anchor.features = features)
```
```

4 We prepare the data for integration first by normalising. This time we are using *SCTransform* instead of log-normalising.

5

Then we selecting features that we will use to integrate. These are generally features in common across all the datasets. Creates new slot with these features.

```{r anchors}
pbmc.anchors <- FindIntegrationAnchors(object.list = pbmc.list, normalization.method = "SCT",
anchor.features = features)
```

6 Once the objects are prepared, we identify “anchor” cells.

```{r integrate data}
pbmc.combined.sct <- IntegrateData(anchorset = pbmc.anchors, normalization.method = "SCT")
```

7 We build an integrated object, using the anchors and specifying the normalisation method.

```{r clustering etc}
pbmc.combined.sct <- RunPCA(pbmc.combined.sct, verbose = FALSE)
pbmc.combined.sct <- RunUMAP(pbmc.combined.sct, reduction = "pca", dims = 1:30)
#pbmc.combined.sct <- RunTSNE(pbmc.combined.sct, reduction = "pca", dims = 1:30)
pbmc.combined.sct <- FindNeighbors(pbmc.combined.sct, dims = 1:10)
pbmc.combined.sct <- FindClusters(pbmc.combined.sct, resolution = 0.5)
```

8 The usual steps including clustering and dimension reduction.

```{r plots and metadata}
DimPlot(pbmc.combined.sct, reduction = "umap")
DimPlot(pbmc.combined.sct, reduction = "umap", group.by = "orig.ident")


pbmc.combined.sct[["patients"]] = factor(gsub("baseline-|D7-", "", pbmc.combined.sct$orig.ident
))
pbmc.combined.sct[["type"]] = factor(gsub("-p\\d", "", pbmc.co[illegible]$orig.ident ))

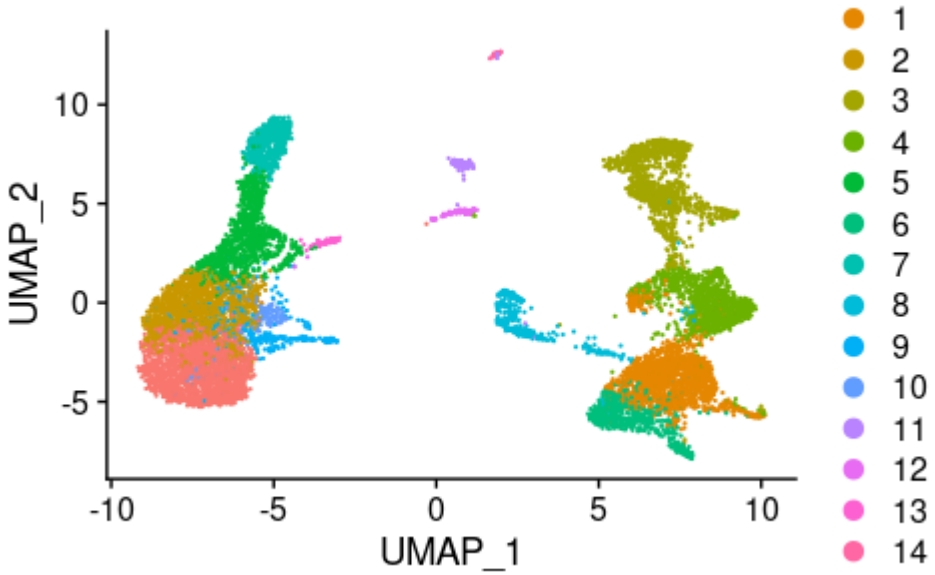

DimPlot(object = pbmc.combined.sct, reduction = "umap", group.by = "patients")
DimPlot(object = pbmc.combined.sct, reduction = "umap", group.by = "type")

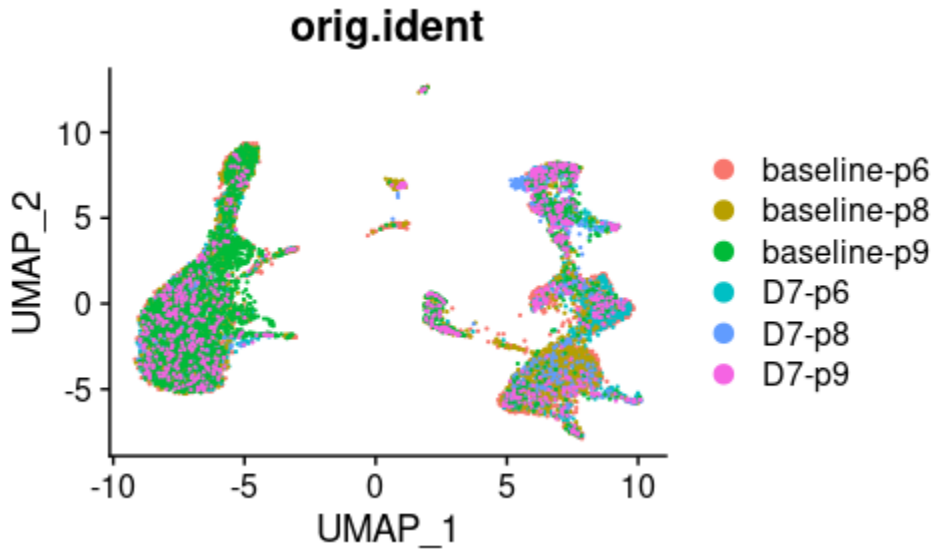
```

Sanity checks

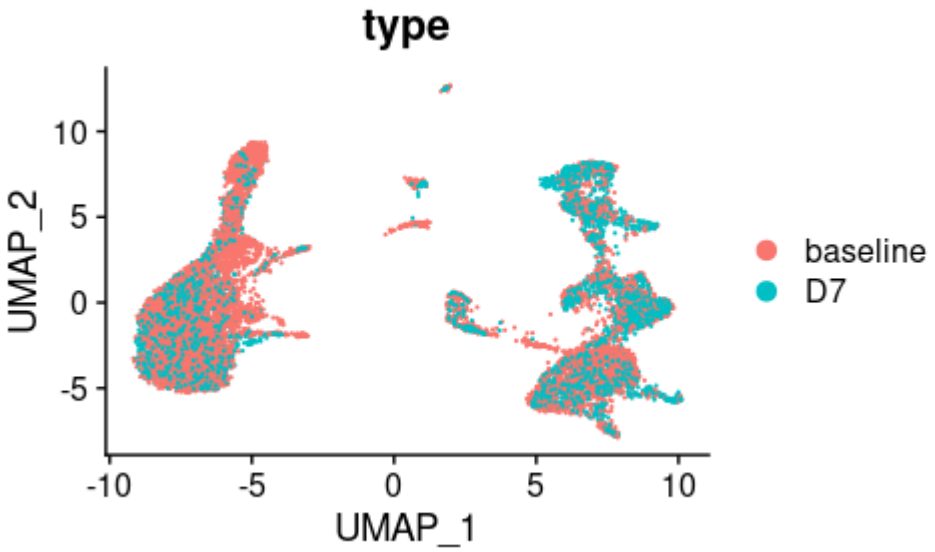
9 And finally, a (few) plots, and adding additional metadata.

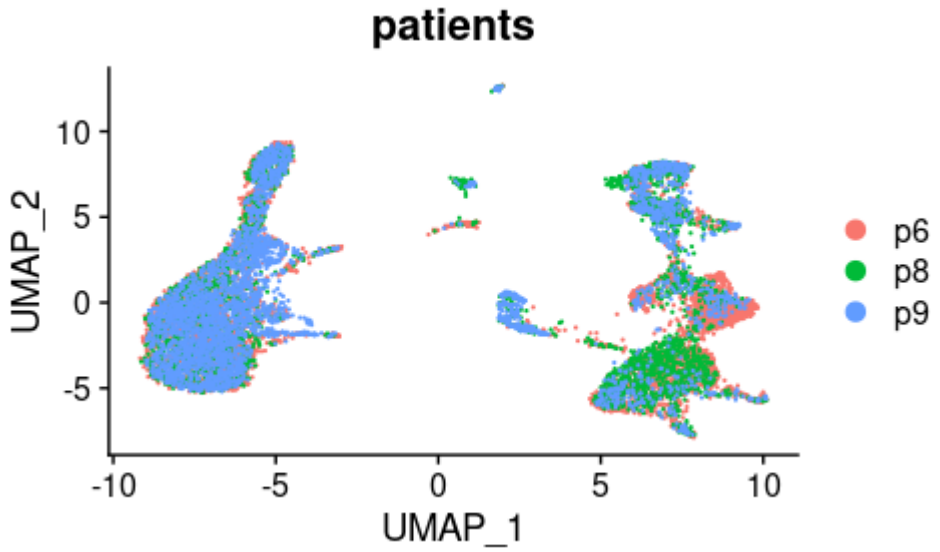
Sanity checks



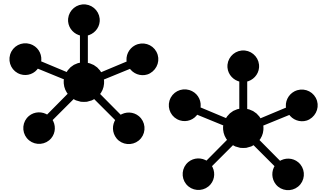


Sanity checks

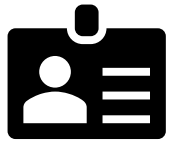




Case versus control pipeline



1 Integration



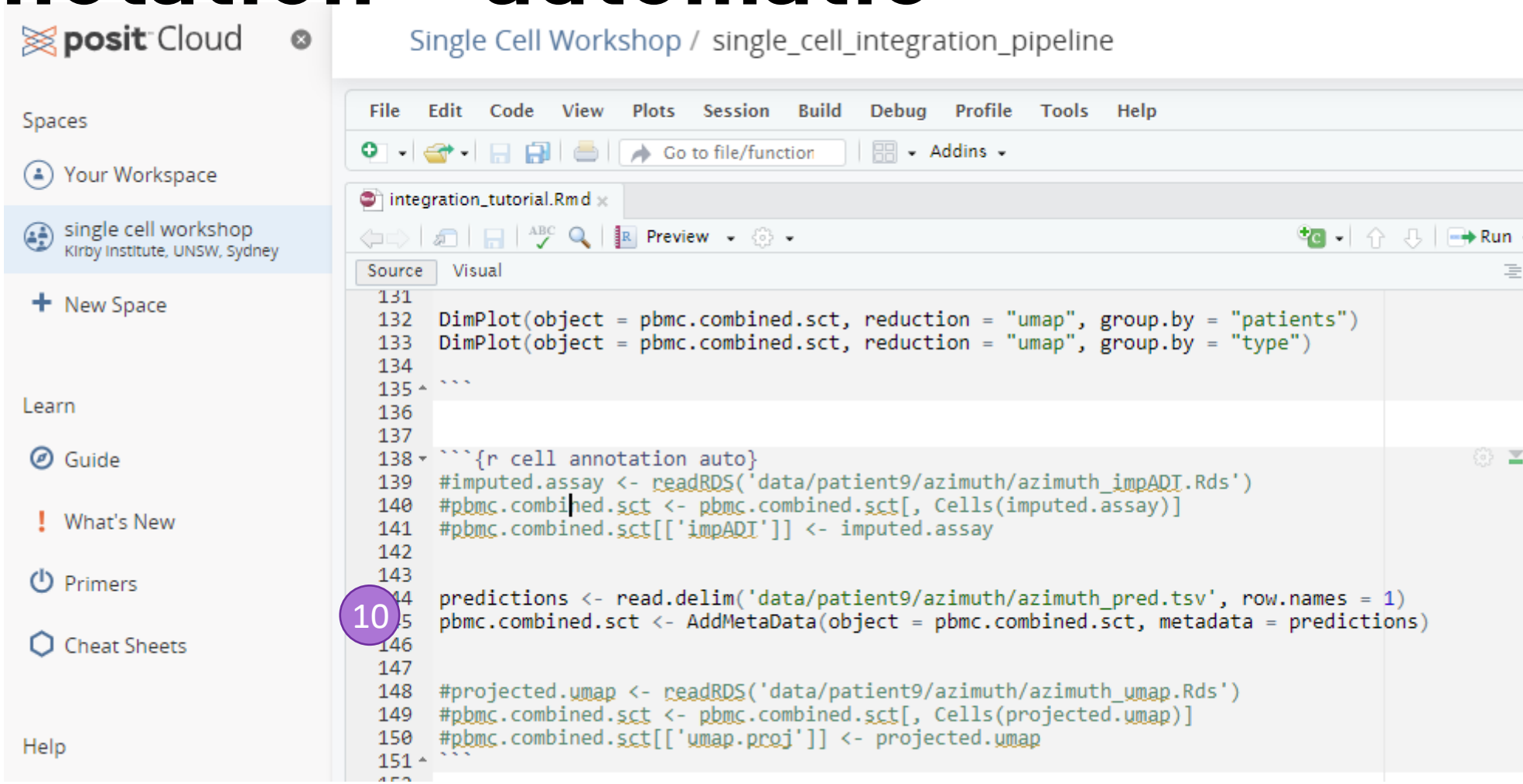
2 **Cell annotation**

3 Differential

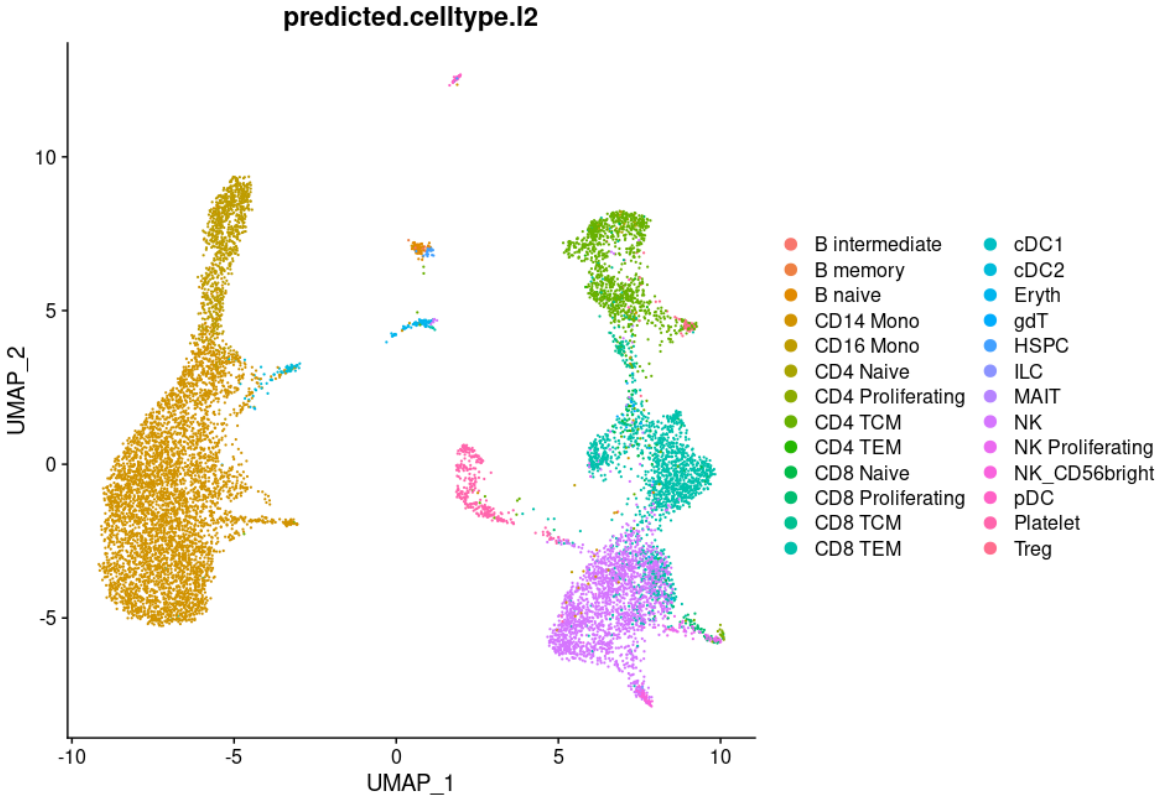
- Composition
- Expression

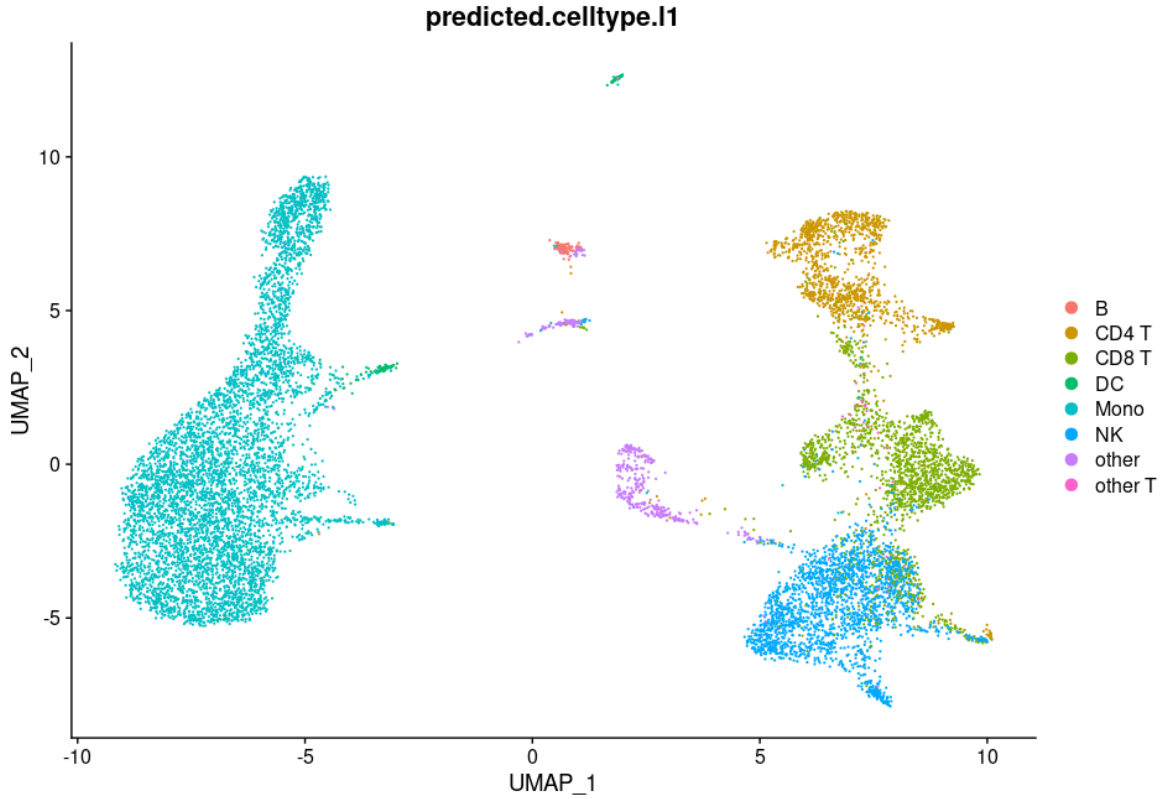
Cell annotation – automatic

Already prepped, import data as before. This time, we are only importing the predictions.

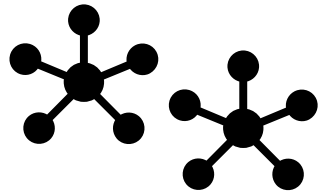


```
153
154 ```{r fig.width=10, fig.height=7}
155 DimPlot(pbmc.combined.sct, reduction = "umap", group.by = "predicted.celltype.l2")
156 DimPlot(pbmc.combined.sct, reduction = "umap", group.by = "predicted.celltype.l1")
157 ```
158
```

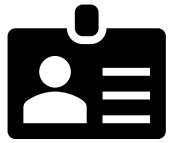




Case versus control pipeline



1 Integration

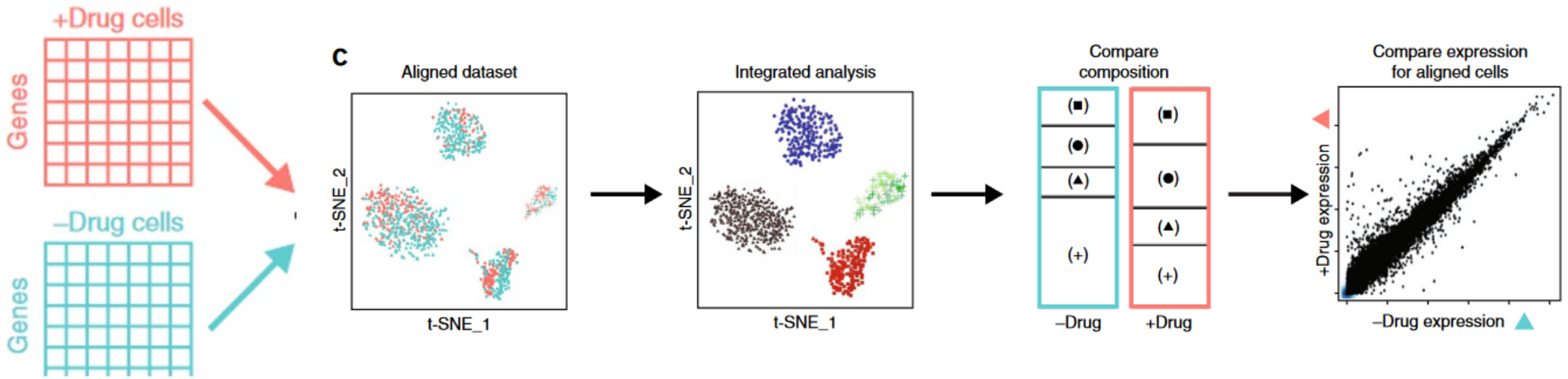


2 Cell annotation

3 **Differential**

- **Composition**
- **Expression**

Two types of comparison



Cluster frequency **Gene expression**

https://www.nature.com/articles/nbt.4096

UNSW SYDNEY

Comparing cluster proportions

````
```{r proportions per sample}
freq = plyr::count(cbind(as.character(pbmc.combined.sct$predicted.celltype.l1),
as.character(pbmc.combined.sct$orig.ident) ))
freq_table <- tidyr::spread(freq, key=1, value=3)
freq_table[is.na(freq_table)] = 0

frac_table = t((freq_table[,-1])/rowSums(freq_table[,-1] , na.rm=T)) * 100

colnames(frac_table) = freq_table[,1]
my_color_palette <- scales::hue_pal()(8)
barplot( frac_table , col=my_color_palette )

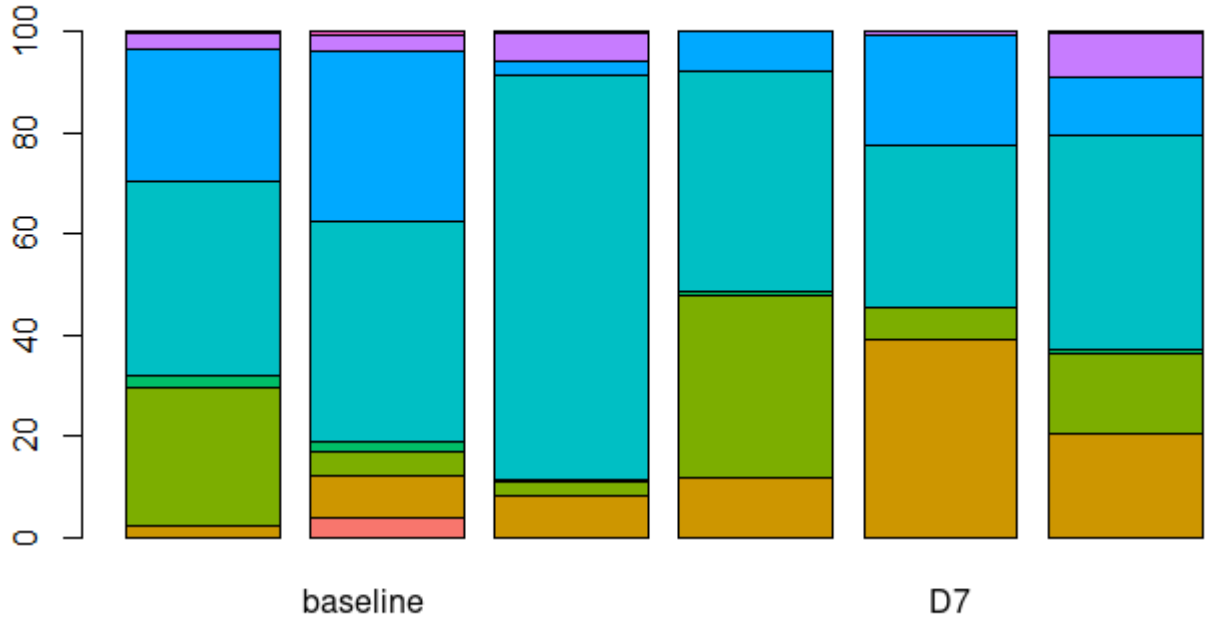
```
````

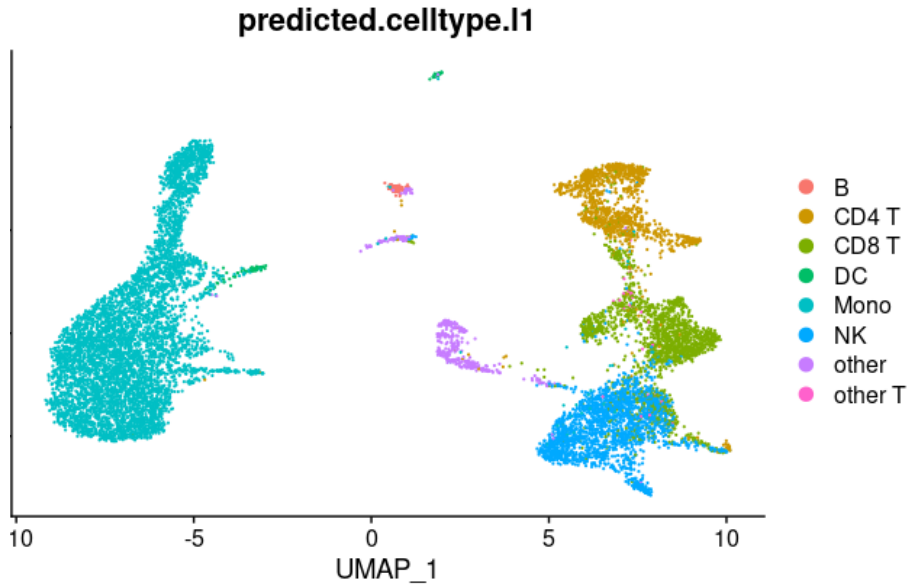
11

Calculate the frequency of each cell type per condition per sample.

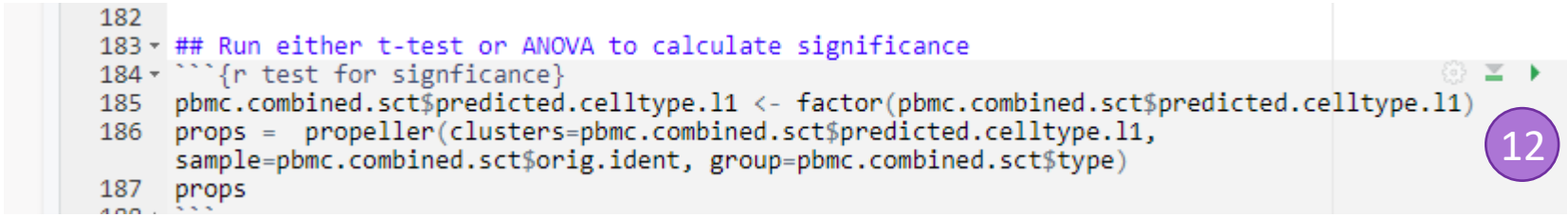
Plot as a stacked barplot using the color palette from Seurat.

Comparing cluster proportions





Testing for significance



Because we only have 6 samples, we will not be able to reach “statistical significance”.

Here, we use the *propeller* function in the “speckle” package to run a t-test on the proportions.

Testing for significance

Clusters	Average proportions	Baseline proportions	Day 7 proportions	Ratio	Tstatistic	P.Value	FDR
CD4 T	10.60%	6.41%	23.78%	0.27	-1.6506787	0.118	0.576
other	3.62%	4.06%	3.17%	1.28	1.5372704	0.144	0.576
CD8 T	15.49%	11.59%	19.39%	0.60	-0.8561149	0.405	0.616
Mono	49.89%	53.78%	39.25%	1.37	0.6985715	0.495	0.616
DC	1.31%	1.56%	0.65%	2.40	0.6874163	0.502	0.616
other T	0.27%	0.36%	0.14%	2.59	0.6702436	0.512	0.616
B	0.77%	1.33%	0.00%	NA	0.627633	0.539	0.616
NK	18.04%	20.91%	13.62%	1.54	0.199156	0.845	0.845

Differential gene expression

```
## Prepare data
Since we ran SCTransform, we need to recorrect our data before running the FindMarkers()
function. Here we are using the default wilcox-test for differential expression but there are
many other approaches that can be used.
```{r find markers/DE, fig.height=8, fig.width=15}
pbmc.combined.sct = PrepSCTFindMarkers(pbmc.combined.sct)
filt = pbmc.combined.sct@meta.data$predicted.celltype.l1 == "CD8 T"

b_v_d7.markers <- FindMarkers(pbmc.combined.sct[,filt], assay="SCT", ident.1 = "D7",  ident.2 =
"baseline", group.by = "type", min.pct = 0.25, logfc.threshold = 0, recorrect_umi = FALSE)

```
```

12

Because the data is integrated, we need to adjust and recorrect the expression data.

Then we run *FindMarkers*, specifying our cases and controls. Since we've recorrected our counts, we set that as "false".

https://satijalab.org/seurat/reference/prepsctfindmarkers

UNSW SYDNEY

Differential gene expression

```
198
199 ```{r volcano plot}
200 plot(b_v_d7.markers$avg_log2FC, -log10(b_v_d7.markers$p_val) ,xlab="Average log2FC",
    ylab="-log10(p-value)", pch=19, main = "baseline vs D7")
201 abline(h=-log10(0.05/dim(b_v_d7.markers)[1]), lwd=2, col=4, lty=2)
202 abline(v=c(1,-1), lty=3, lwd=2, col=4)
203 abline(v=c(0.5,-0.5), lty=2, col=4)
204
205 head( b_v_d7.markers[  abs(b_v_d7.markers$avg_log2FC) > 1 & b_v_d7.markers$p_val_adj < 0.05, ] )

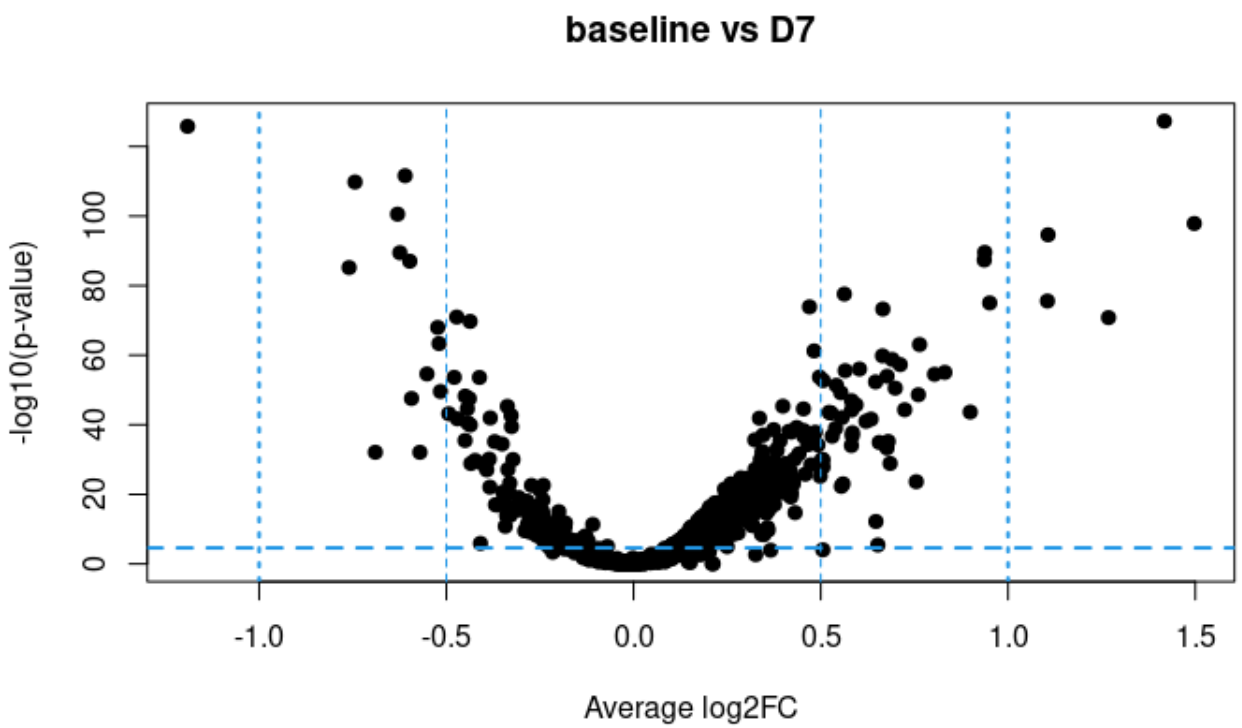
206 ```
```

13

We plot a volcano plot using the **average log2FC** on the x-axis, and the **–log10(p-value)** on the y-axis.

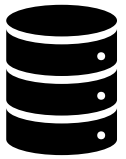
We added some thresholds to the fold change and the adjusted p-values consider.

Gene biomarkers (volcano plot)



	p_val	avg_log2FC	pct.1	pct.2	p_val_adj
CD74	5.165184e-128	1.417385	0.974	0.882	8.699202e-124
MALAT1	1.630997e-126	-1.191078	0.993	1.000	2.746925e-122
GZMB	1.389634e-98	1.497207	0.856	0.741	2.340421e-94
HLA-DRA	2.466791e-95	1.106755	0.700	0.260	4.154569e-91
HLA-DRB5	2.660964e-76	1.105121	0.757	0.435	4.481595e-72
LGALS1	1.584336e-71	1.268224	0.885	0.710	2.668340e-67

R, RStudio and Seurat installation



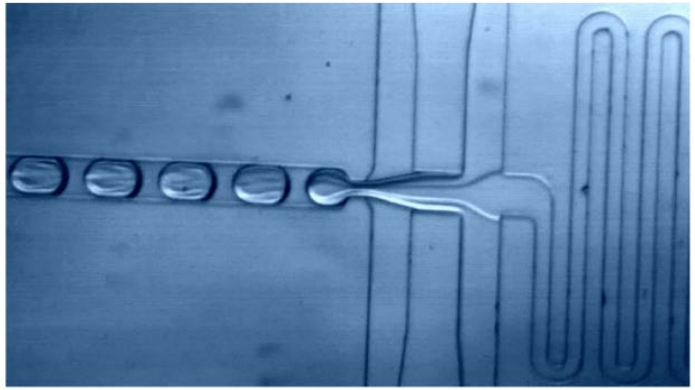
- R and RStudio installation (R needs to be downloaded and installed first):
  - https://cran.r-project.org/bin/windows/base/ (R)
  - https://posit.co/products/open-source/rstudio/ (Rstudio)

- Seurat installation
  - https://satijalab.org/seurat/articles/install

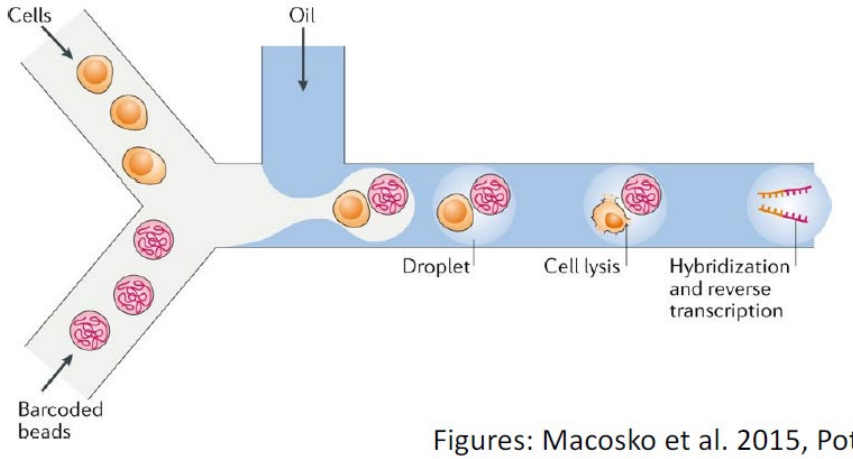
- Other additional packages:
  - Use the command on the command prompt: install.packages(“package_name”)

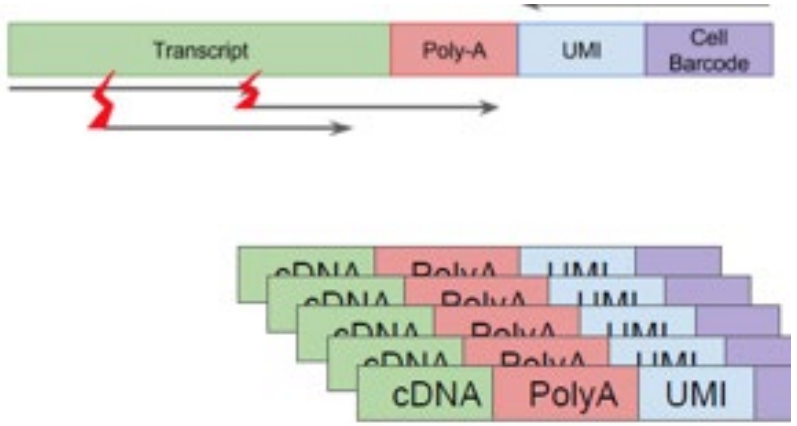
Supplementary

Droplet based



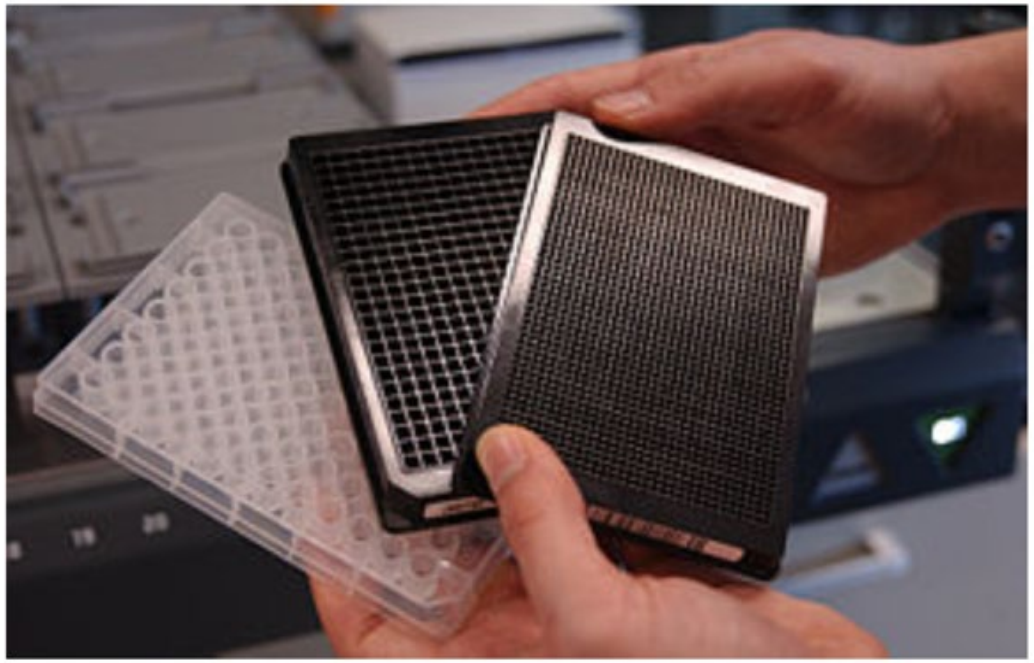
- **Con:** Captures sequences at end of transcript (can't detect transcript isoforms)
- **Pro:** High number of cells
- Useful for analysing heterogeneity and detection of novel cell types



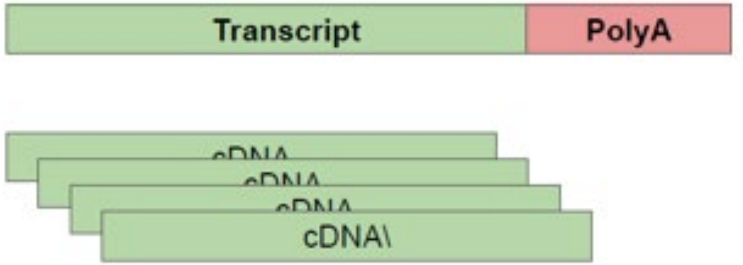


Figures: Macosko et al. 2015, Potter SS. 2018

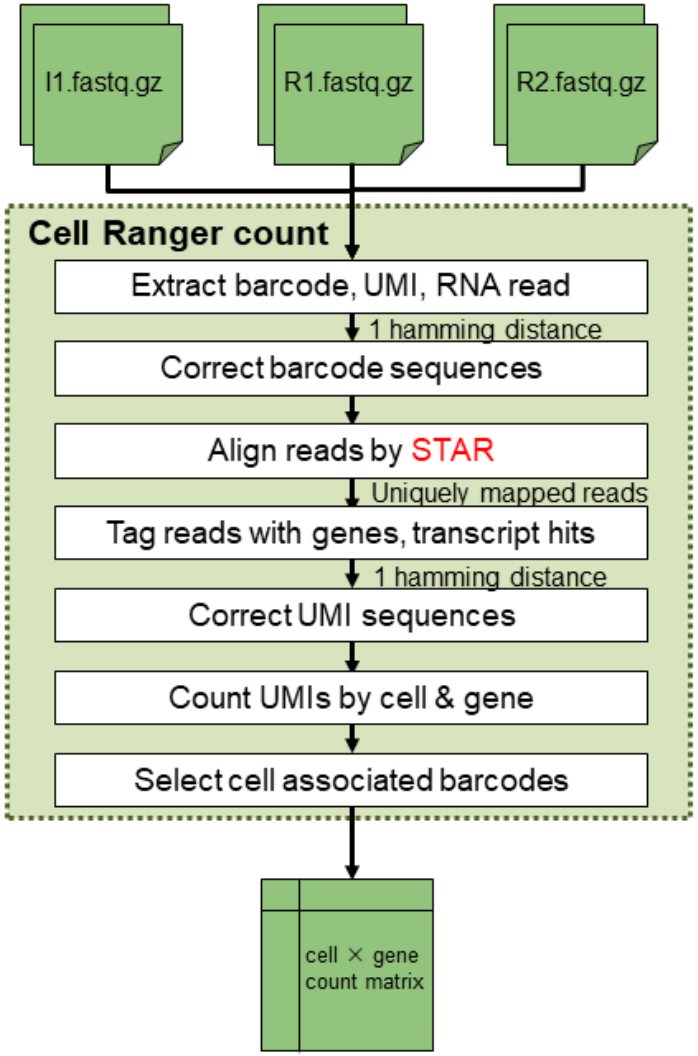
Plate-based



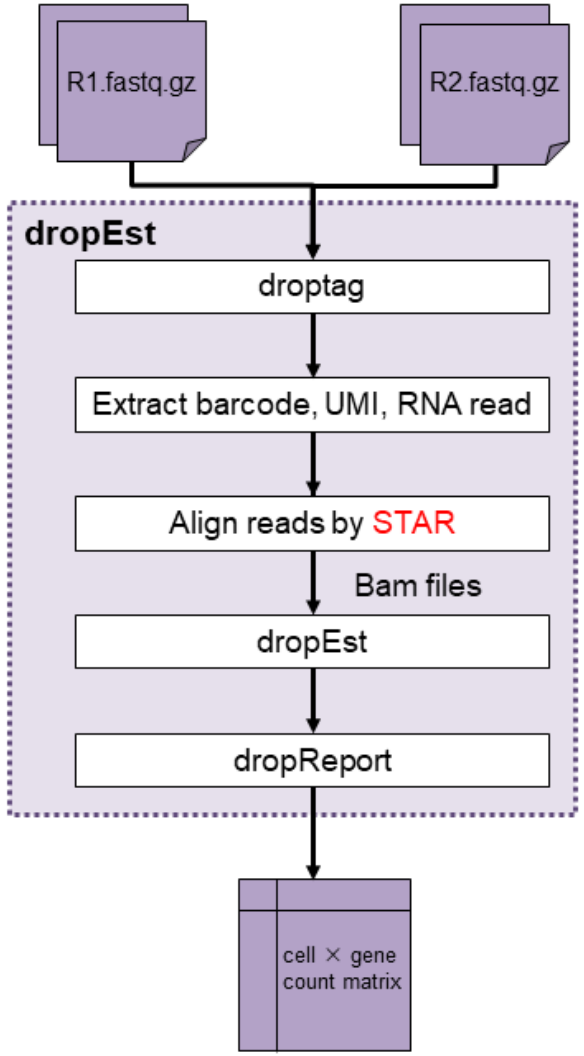
- **Pro:** Captures entire transcript
- **Con:** Low number of cells
- Useful for questions involving isoforms, mutations, SNVs



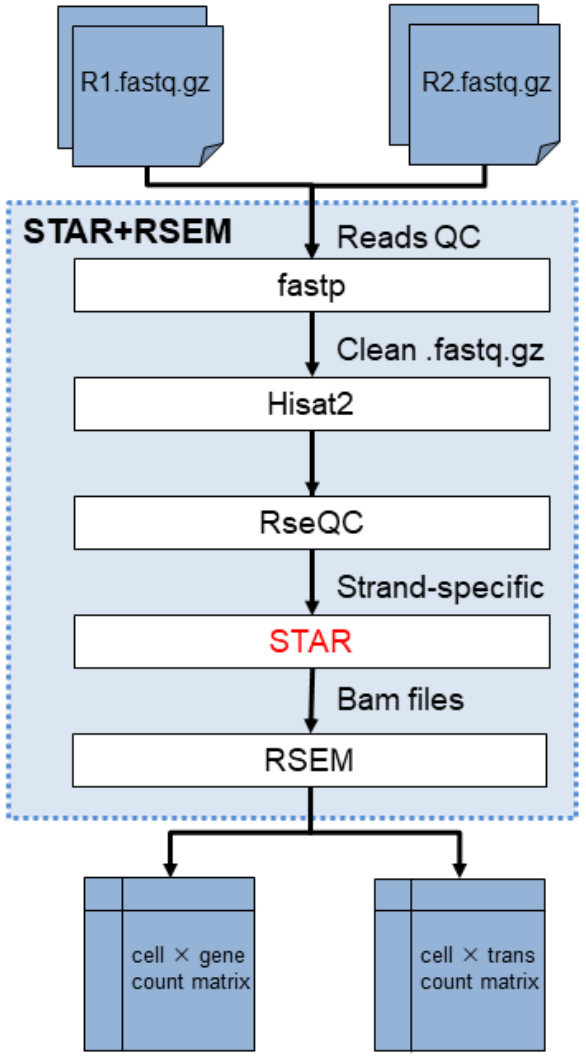
➢ 10X



➢ Drop-seq & inDrop

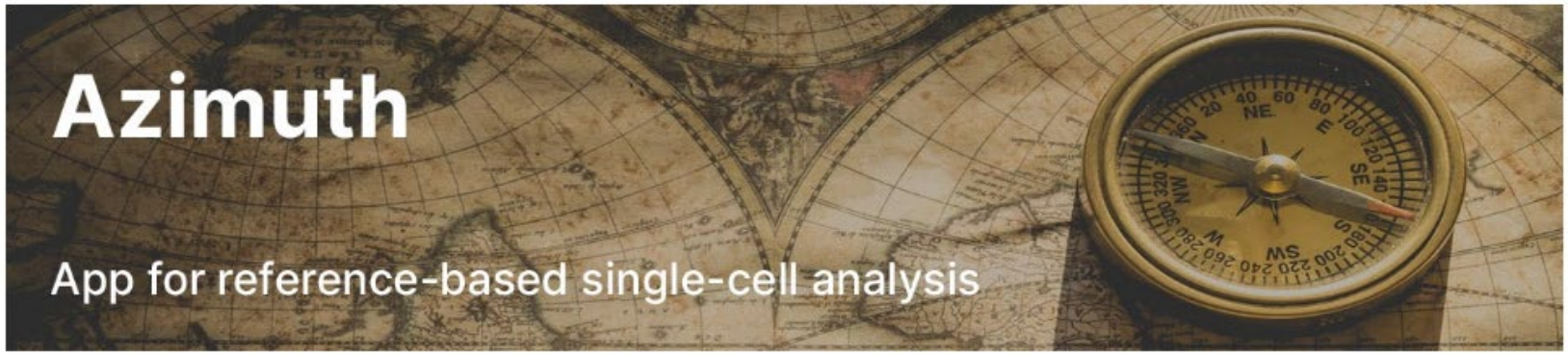


➢ Bulk /SMART-based



HuBMAP

Home References



Azimuth is a web application that uses an annotated reference dataset to **automate the processing, analysis, and interpretation of a new single-cell RNA-seq or ATAC-seq experiment**. Azimuth leverages a **'reference-based mapping'** pipeline that inputs a counts matrix and performs normalization, visualization, cell annotation, and differential expression (biomarker discovery). All results can be explored within the app, and easily downloaded for additional downstream analysis.

The development of Azimuth is led by the New York Genome Center Mapping Component as part of the NIH Human Biomolecular Atlas Project (HuBMAP). Thirteen molecular reference maps are currently available, with more coming soon.

References for scRNA-seq Queries

Human - PBMC	Human - Motor Cortex	Mouse - Motor Cortex	Human - Pancreas

0

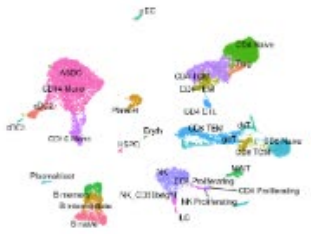
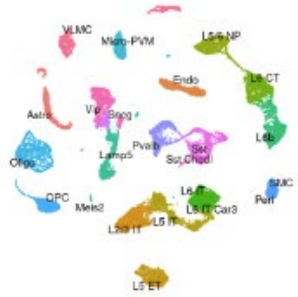
Go to the azimuth weblink.

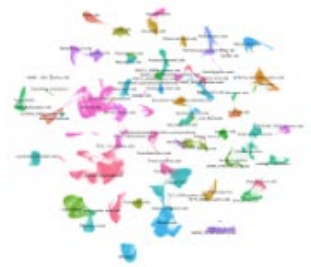
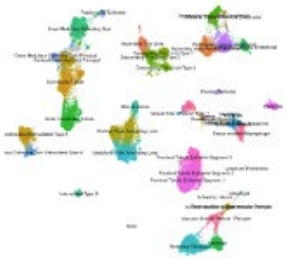
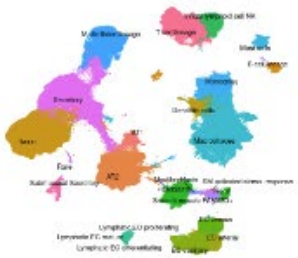
Azimuth demo

UNSW SYDNEY

Biomolecular Atlas Project (HuBMAP). Thirteen molecular reference maps are currently available, with more coming soon.

References for scRNA-seq Queries

Human - PBMC	Human - Motor Cortex	Mouse - Motor Cortex	Human - Pancreas
			
Learn more / Go to App	Learn more / Go to App	Learn more / Go to App	Learn more / Go to App

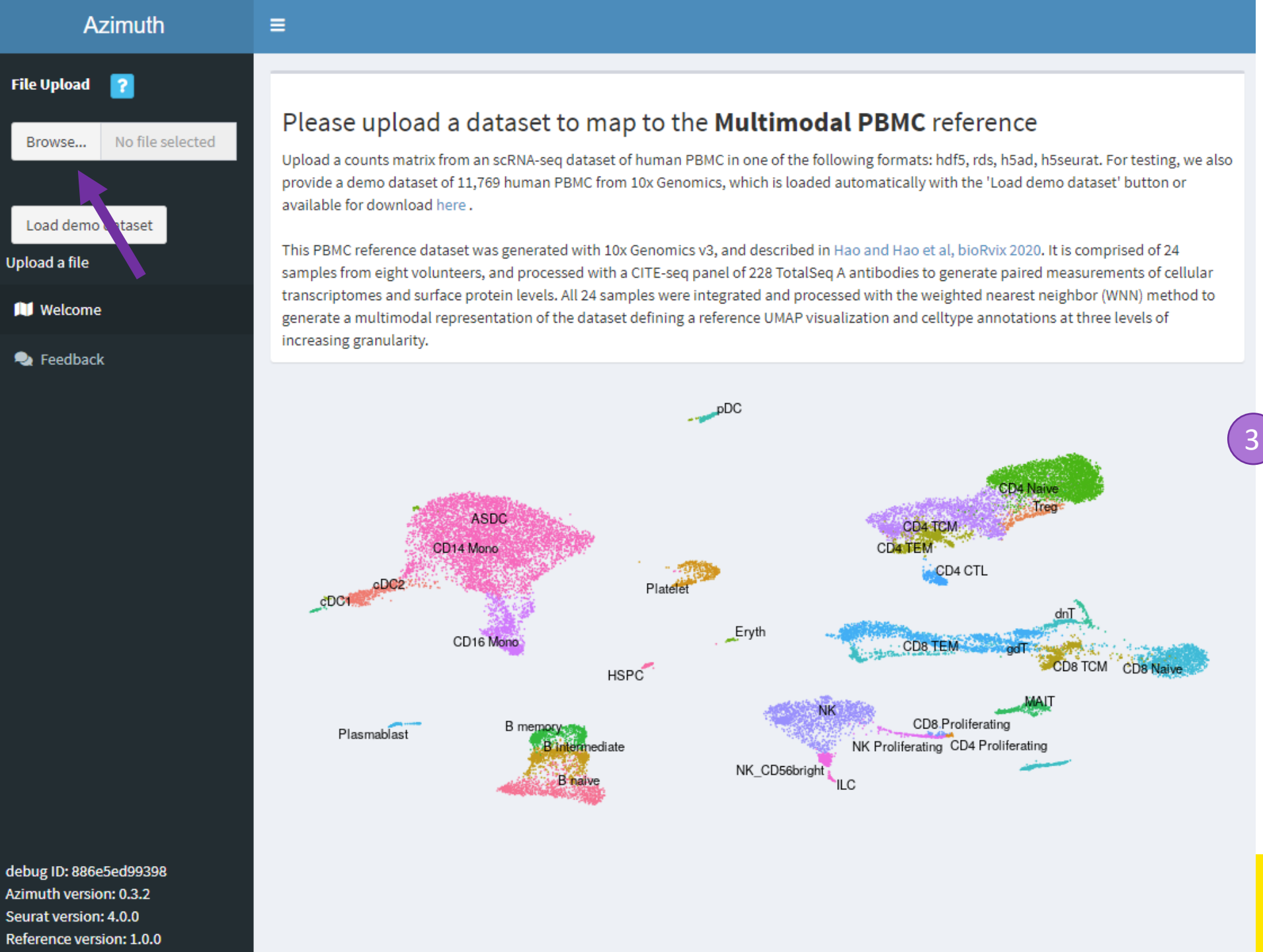
Human - Fetal Development	Human - Kidney	Human - Bone Marrow	Human - Lung v2 (HLCA)
			
Learn more / Go to App	Learn more / Go to App	Learn more / Go to App	Learn more / Go to App

1

Scroll down to the Human – PBMC reference box, and select “Go to App”

Human reference

UNSW SYDNEY



3

Click on browse, and select your dataset. Make sure it is in either ".rds" or ".h5" format.

Data upload

UNSW SYDNEY

Azimuth

File Upload

Browse... pbmc10k.rds

Load demo dataset

Upload a file

Welcome

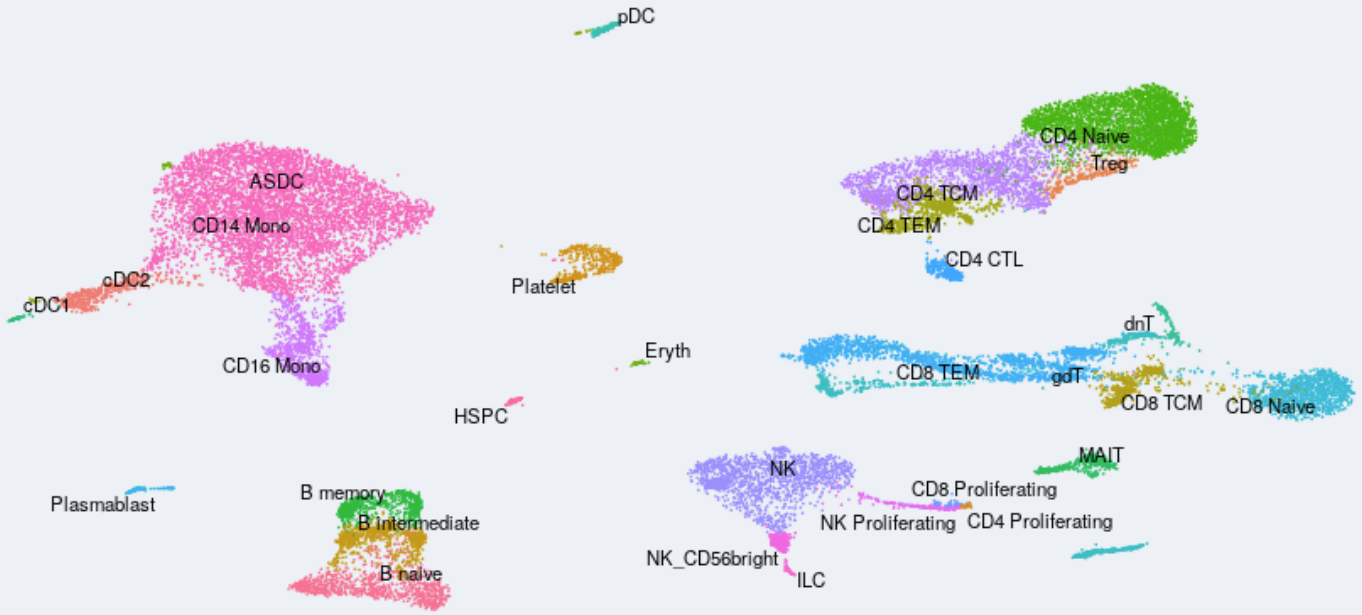
Feedback

debug ID: 886e5ed99398
Azimuth version: 0.3.2
Seurat version: 4.0.0
Reference version: 1.0.0

Please upload a dataset to map to the **Multimodal PBMC** reference

Upload a counts matrix from an scRNA-seq dataset of human PBMC in one of the following formats: hdf5, rds, h5ad, h5seurat. For testing, we also provide a demo dataset of 11,769 human PBMC from 10x Genomics, which is loaded automatically with the 'Load demo dataset' button or available for download here .

This PBMC reference dataset was generated with 10x Genomics v3, and described in Hao and Hao et al, bioRvix 2020. It is comprised of 24 samples from eight volunteers, and processed with a CITE-seq panel of 228 TotalSeq A antibodies to generate paired measurements of cellular transcriptomes and surface protein levels. All 24 samples were integrated and processed with the weighted nearest neighbor (WNN) method to generate a multimodal representation of the dataset defining a reference UMAP visualization and celltype annotations at three levels of increasing granularity.



Load …

UNSW SYDNEY

Azimuth

File Upload

Browse... pbmc10k.rds

Upload complete

Load demo dataset

7499 cells uploaded

Welcome

Preprocessing

Feedback

debug ID: 886e5ed99398
Azimuth version: 0.3.2
Seurat version: 4.0.0
Reference version: 1.0.0

QC Filters

min nCount_RNA	max nCount_RNA
510	12445

min nFeature_RNA	max nFeature_RNA
284	2499

min percent.mt	max percent.mt
0	5

7499 cells remain after current filters

Transfer Options

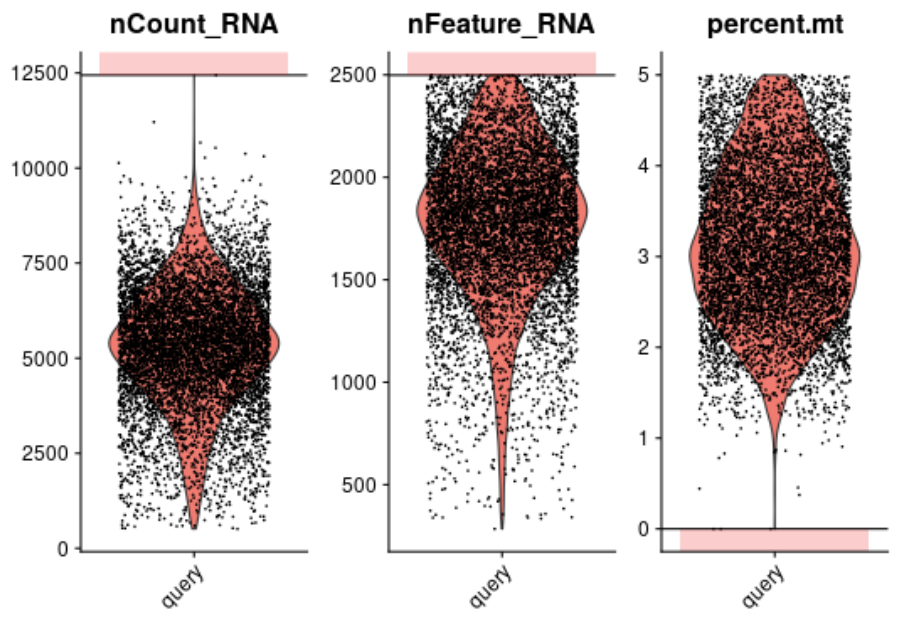
Reference Metadata to Transfer

celltype.l2

Map cells to reference

7499
cells uploaded

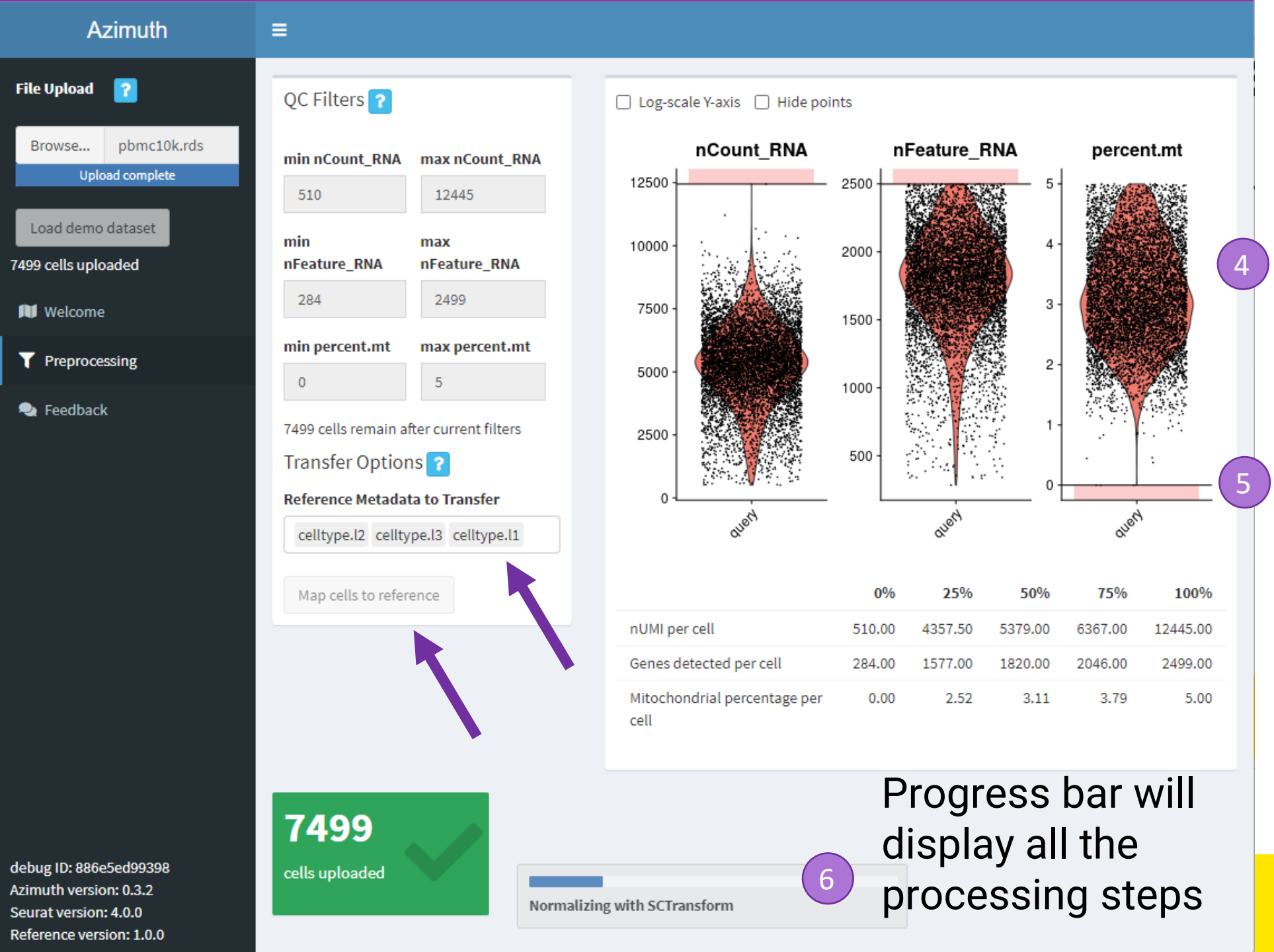
Log-scale Y-axis Hide points



	0%	25%	50%	75%	100%
nUMI per cell	510.00	4357.50	5379.00	6367.00	12445.00
Genes detected per cell	284.00	1577.00	1820.00	2046.00	2499.00
Mitochondrial percentage per cell	0.00	2.52	3.11	3.79	5.00

Success!

UNSW SYDNEY



4

5

6

Select the celltype level you wish to predict. For PBMCs, we can choose from level 1 to level 3.

Then click on the “Map cells to reference” box

Progress bar will display all the processing steps

Select metadata

UNSW SYDNEY

Azimuth

File Upload

Browse... pbmc10k.rds

Upload complete

Load demo dataset

7499 cells uploaded
7499 cells preprocessed
7499 cells mapped
in 1 minutes 2 seconds

- Welcome
- Preprocessing
- Cell Plots
- Feature Plots
- Download Results
- Feedback

7

debug ID: 886e5ed99398
Azimuth version: 0.3.2
Seurat version: 4.0.0
Reference version: 1.0.0

QC Filters

min nCount_RNA	max nCount_RNA
510	12445

min nFeature_RNA	max nFeature_RNA
284	2499

min percent.mt	max percent.mt
0	5

7499 cells remain after current filters

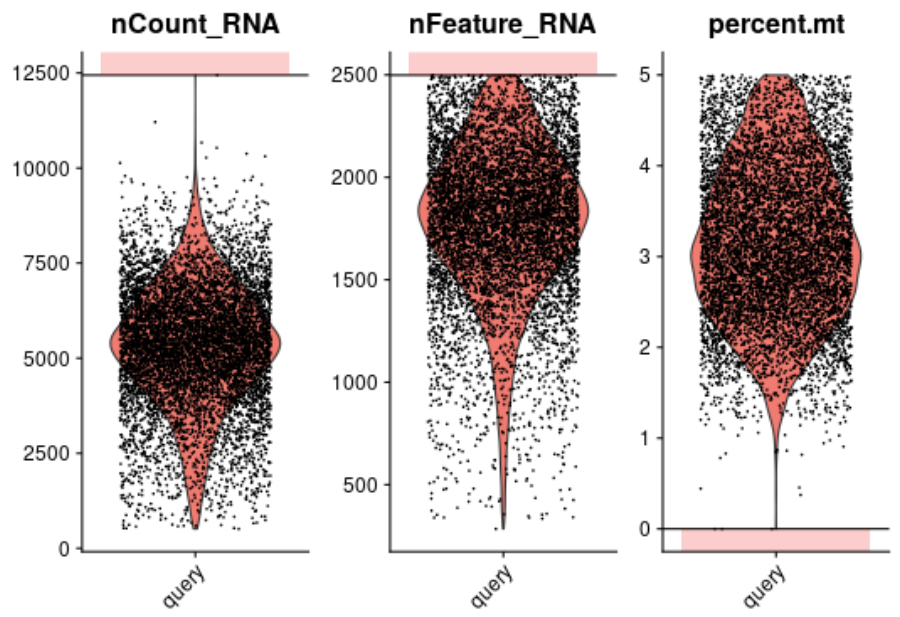
Transfer Options

Reference Metadata to Transfer

celltype.l2 celltype.l3 celltype.l1

Map cells to reference

Log-scale Y-axis Hide points



	0%	25%	50%	75%	100%
nUMI per cell	510.00	4357.50	5379.00	6367.00	12445.00
Genes detected per cell	284.00	1577.00	1820.00	2046.00	2499.00
Mitochondrial percentage per cell	0.00	2.52	3.11	3.79	5.00

7499 cells uploaded

7499 cells after filtering

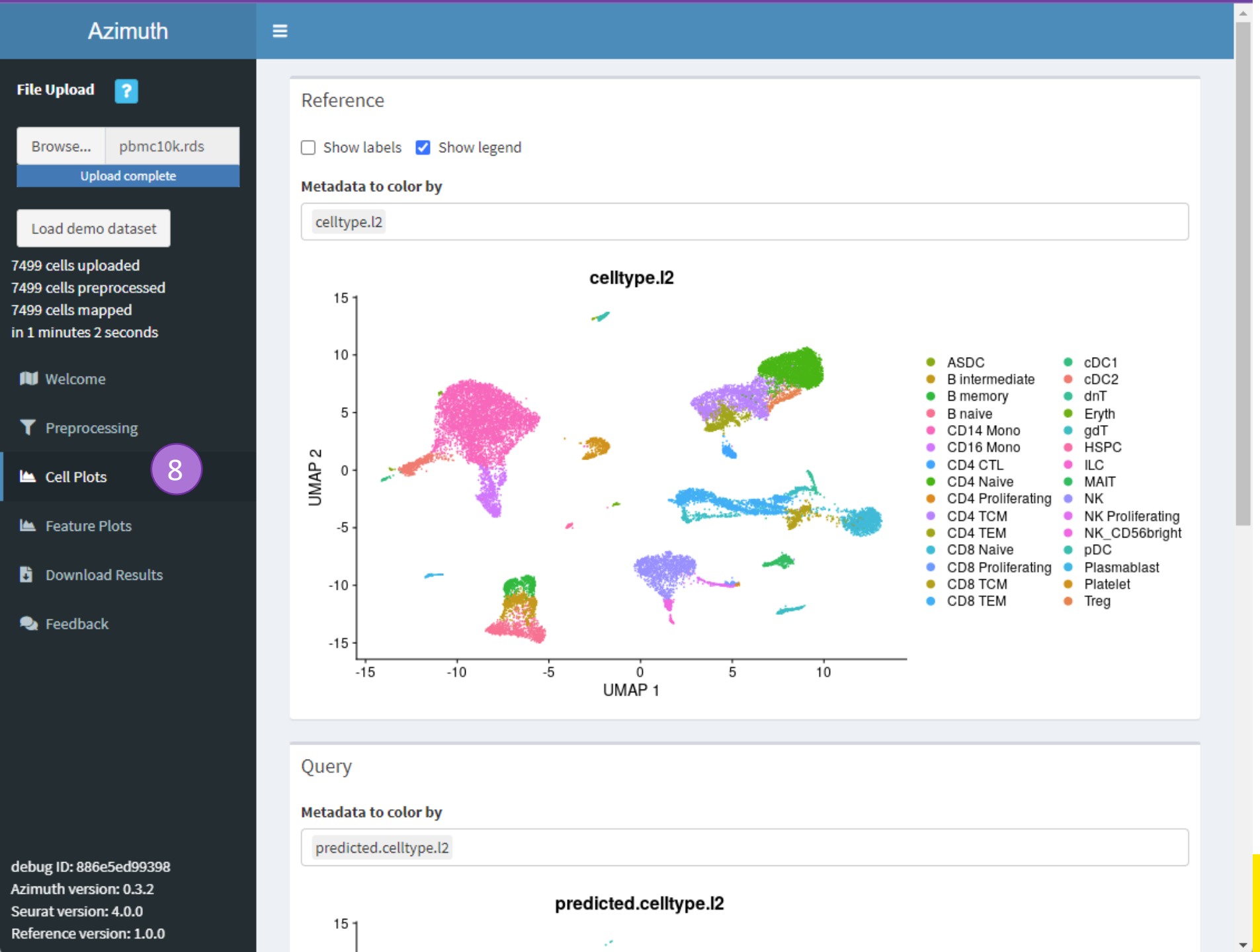
39.77% % of query cells with anchors

0.95/5 cluster preservation score

Outputs from the annotation appear in the menu bar on the left. This includes plots to visualise the cells and downloading the predictions.

Results!

UNSW SYDNEY



UMAPs of both reference and query (user) data.

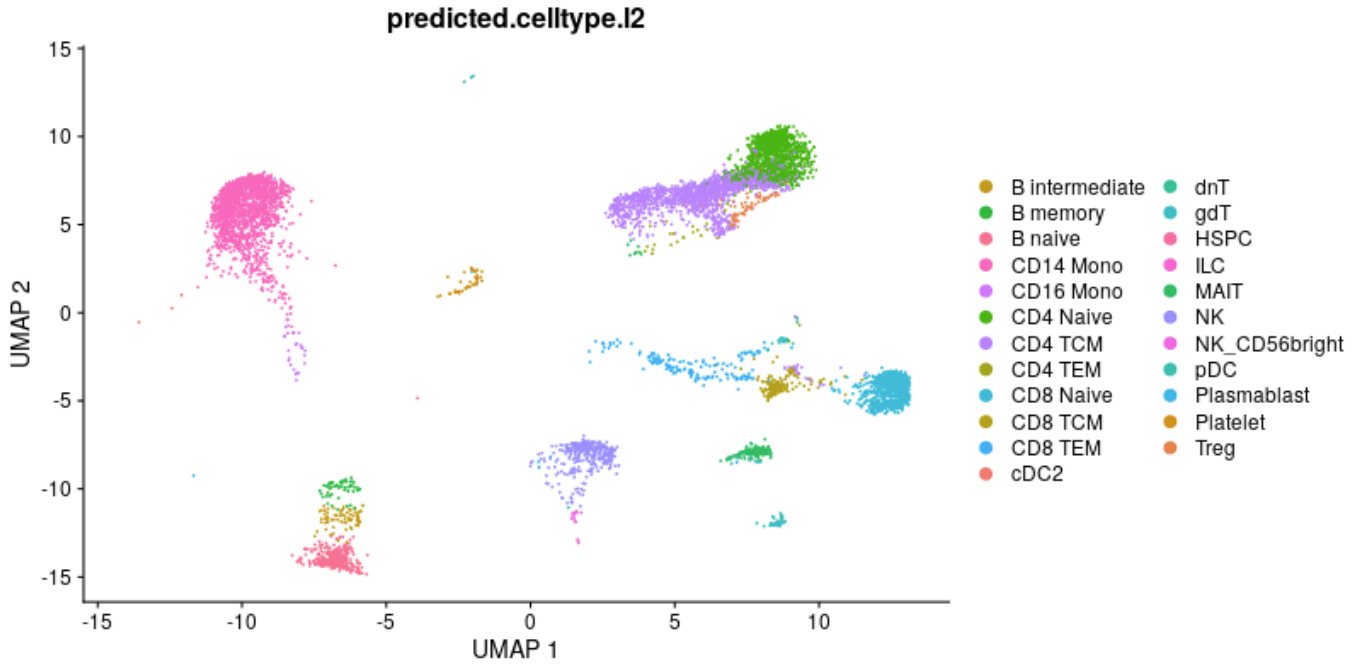
Cell plots

UNSW SYDNEY

Query

Metadata to color by

predicted.celltype.l2



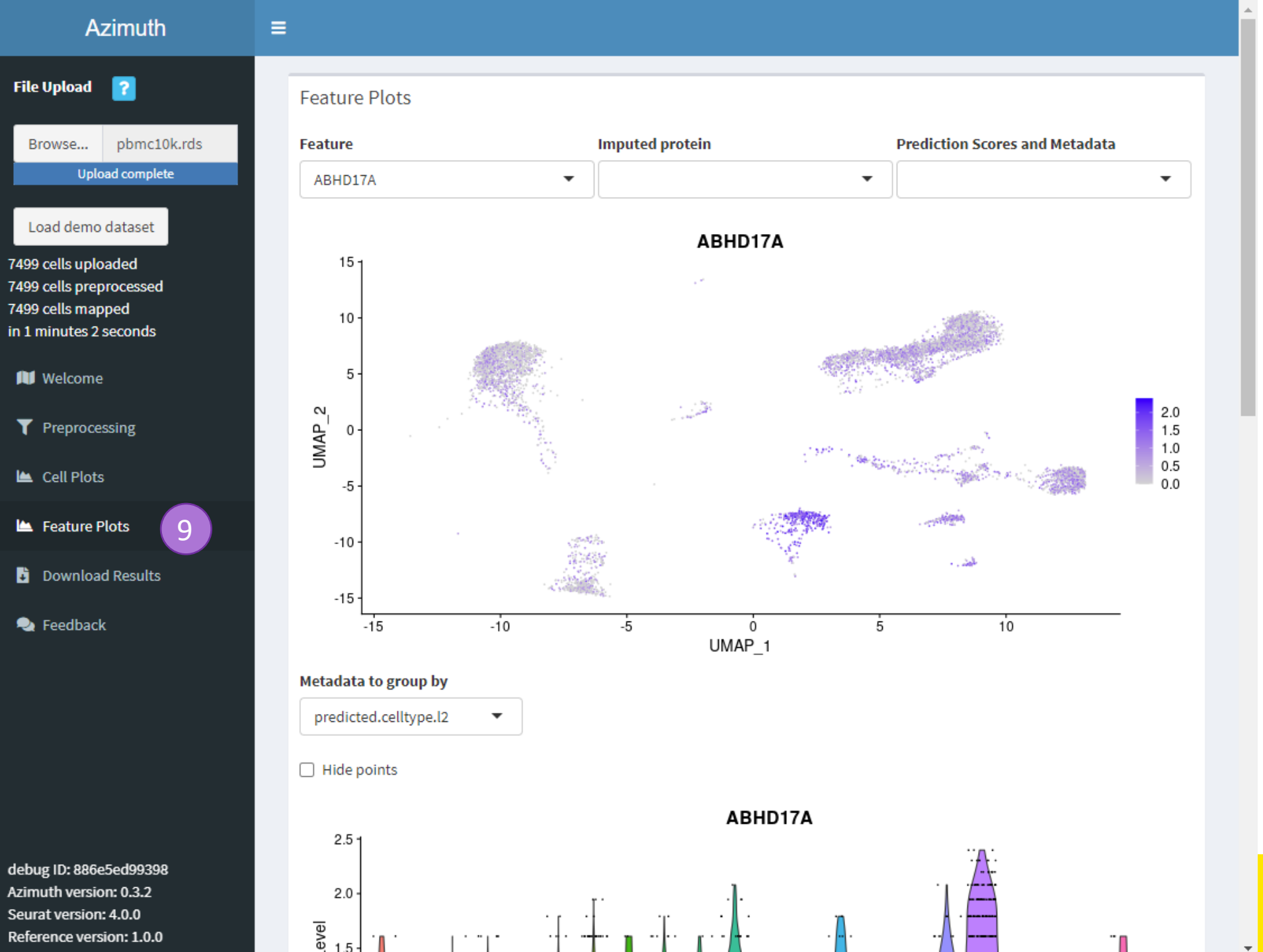
Metadata table

Table rows seurat_clusters

Table columns predicted.celltype.l1

Percentage Frequency

	B	CD4 T	CD8 T	DC	Mono	NK	other	other T
0	0	0	0	2	1590	0	0	0
1	0	1568	2	0	0	0	0	5
10	138	0	0	3	0	0	0	0
11	0	0	0	0	0	0	41	0
2	0	1021	15	0	0	0	3	2
3	0	4	865	0	0	0	0	2
4	0	574	1	0	0	0	0	0
5	417	0	0	0	0	0	0	0
6	0	4	1	1	351	1	0	0
7	0	0	5	0	0	346	0	0
8	0	12	281	0	0	1	1	9
9	0	8	1	0	0	1	0	223



UMAPs and violin plots of features.

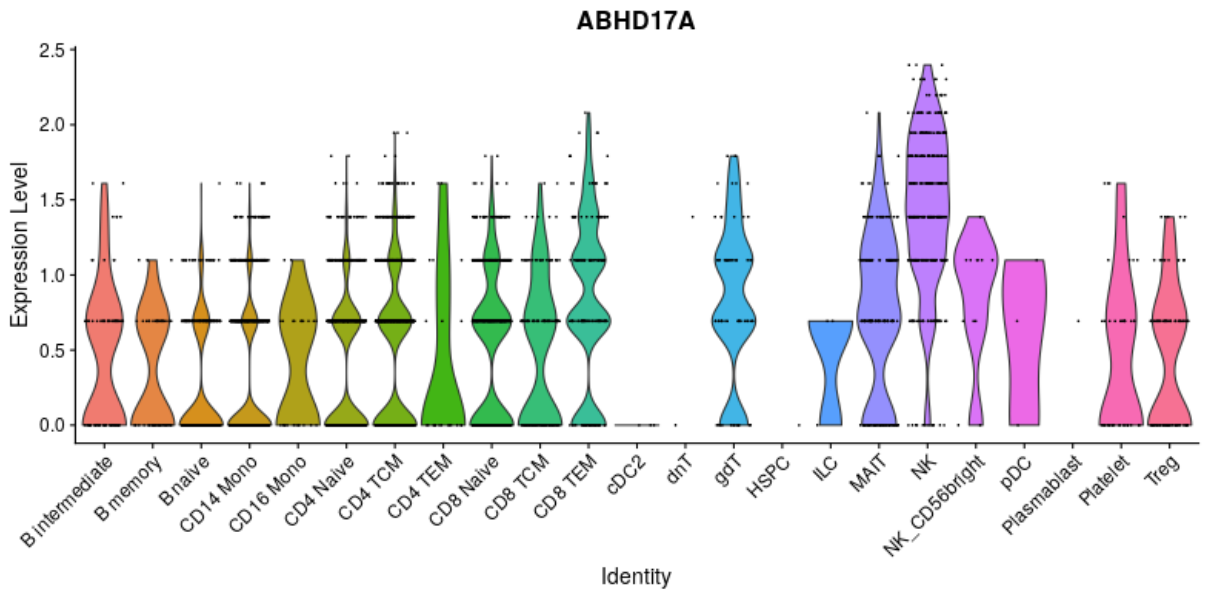
Feature plots

UNSW SYDNEY

Metadata to group by

predicted.celltype.l2

Hide points



Predicted cell type cluster biomarkers

Metadata group

celltype.l2

Predicted cell type

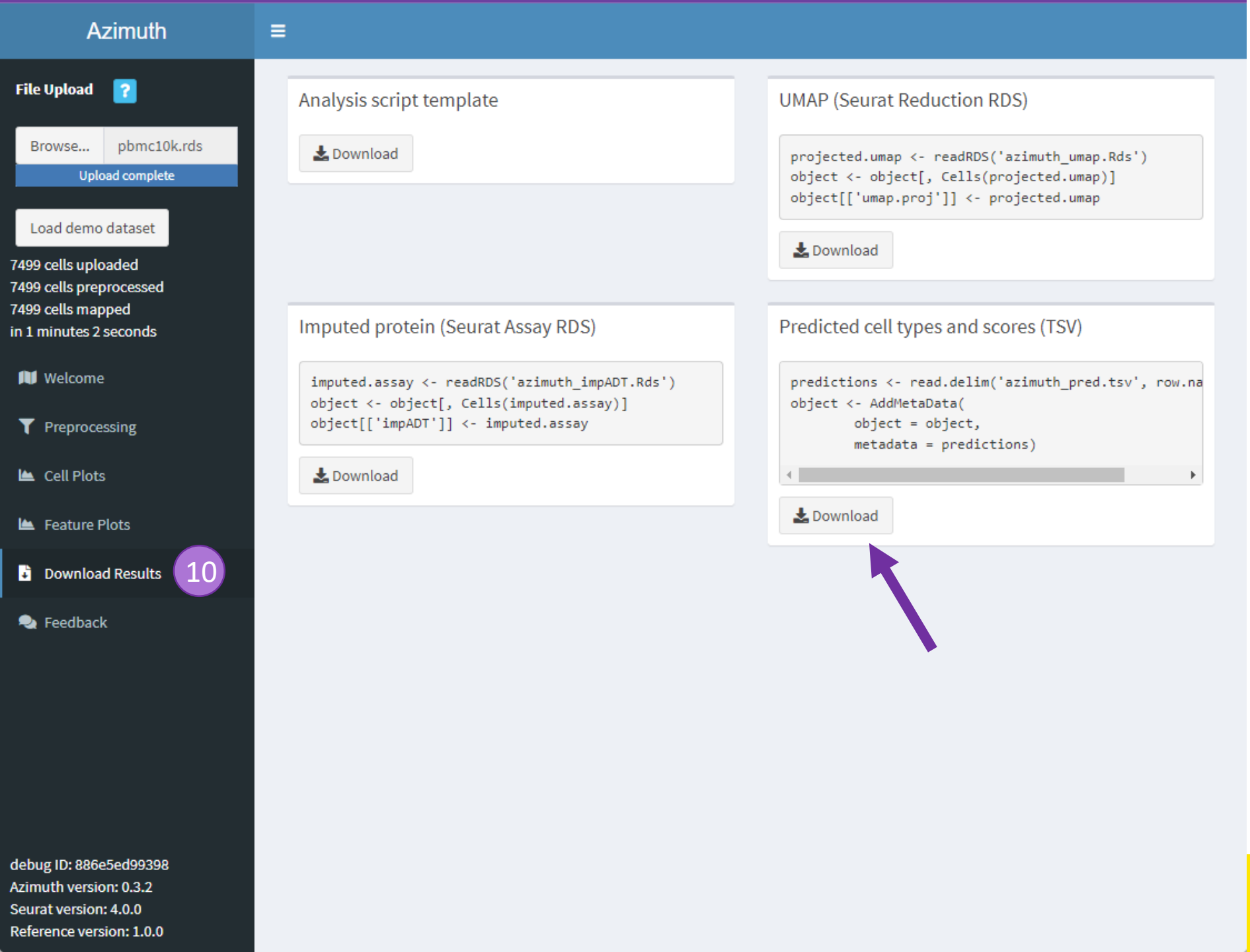
B intermediate

RNA biomarkers

	auc	padj	pct_in	pct_out
LINC01857	0.795	0	60	1.16
TNFRSF13B	0.711	0	42.7	0.471
GALNTL6	0.593	7.88e-246	18.7	0.0404
IGHG2	0.671	6.42e-241	34.7	0.445
FCRLA	0.913	5.19e-240	85.3	3.96
FCRL2	0.747	2.25e-217	50.7	1.35
CD24	0.842	8.58e-215	70.7	2.94
AIM2	0.735	5.79e-212	48	1.23
SPIB	0.878	2.08e-206	78.7	3.91
FCGR2B	0.834	7.51e-204	69.3	2.99

Imputed protein biomarkers

	auc	padj	pct_in	pct_out
CD267	0.996	2.01e-47	100	100
CD1c	0.995	2.01e-47	100	100
CD307c/FcRL3	0.994	2.38e-47	100	100
CD24	0.991	7.51e-47	100	100
CD196	0.99	1.07e-46	100	100
IgM	0.988	1.46e-46	100	100
CD79b	0.987	2.23e-46	100	100
CD19	0.986	3.18e-46	100	100
CD39	0.982	1.89e-45	100	100
CD52	0.977	1.43e-44	100	100



Download the predictions and UMAP, along with the scripts used.

Download!

UNSW SYDNEY

Merged (not integrated)