

Single-cell RNA-seq workshop: tutorial

Raymond Louie, PhD

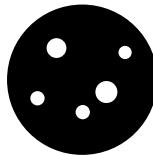
Sara Ballouz, PhD

Overview of the Single Cell tutorial

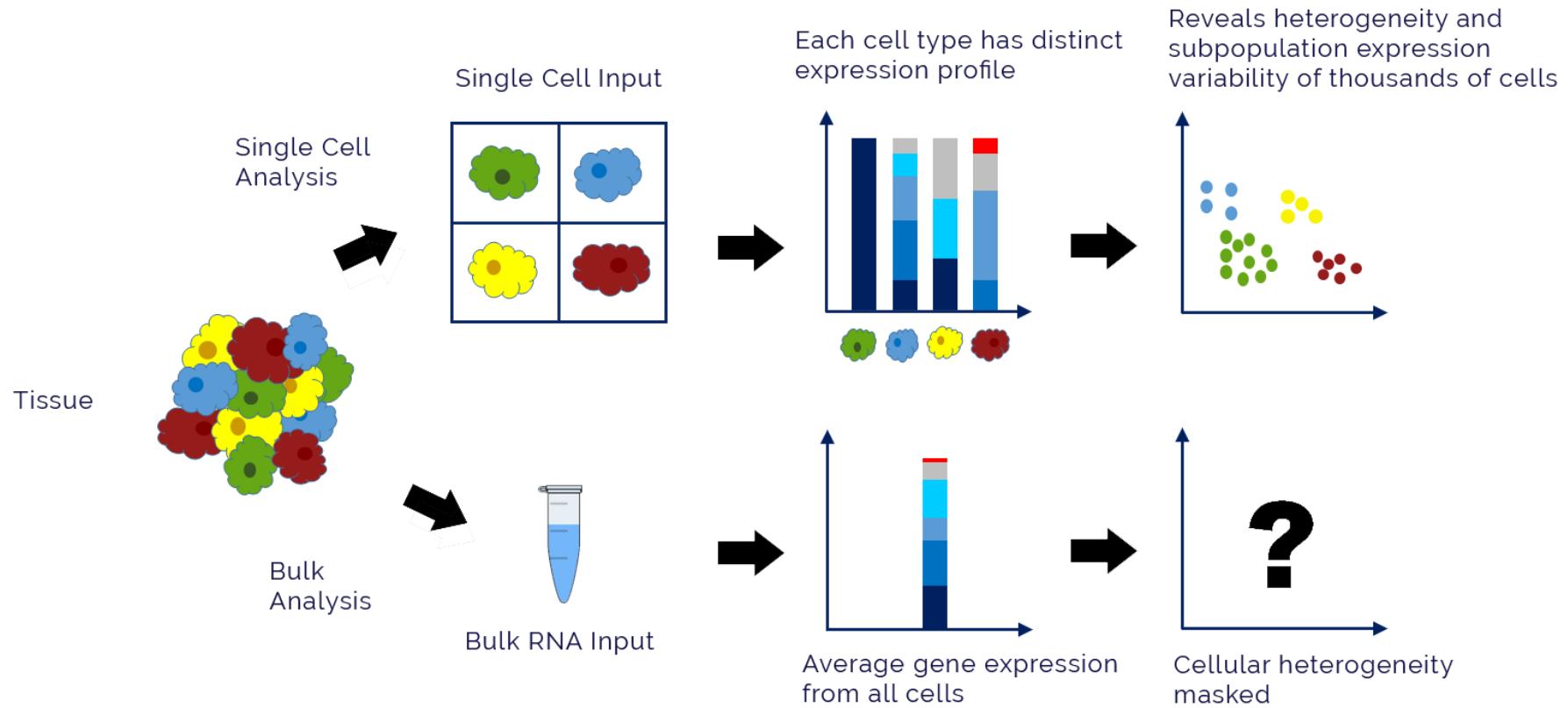
1. Introduction to single cell
2. Setting up Rstudio, data and count matrix
3. Pre-processing
4. Application 1: Cell annotation
5. Application 2: Case vs Control

Overview of the Single Cell tutorial

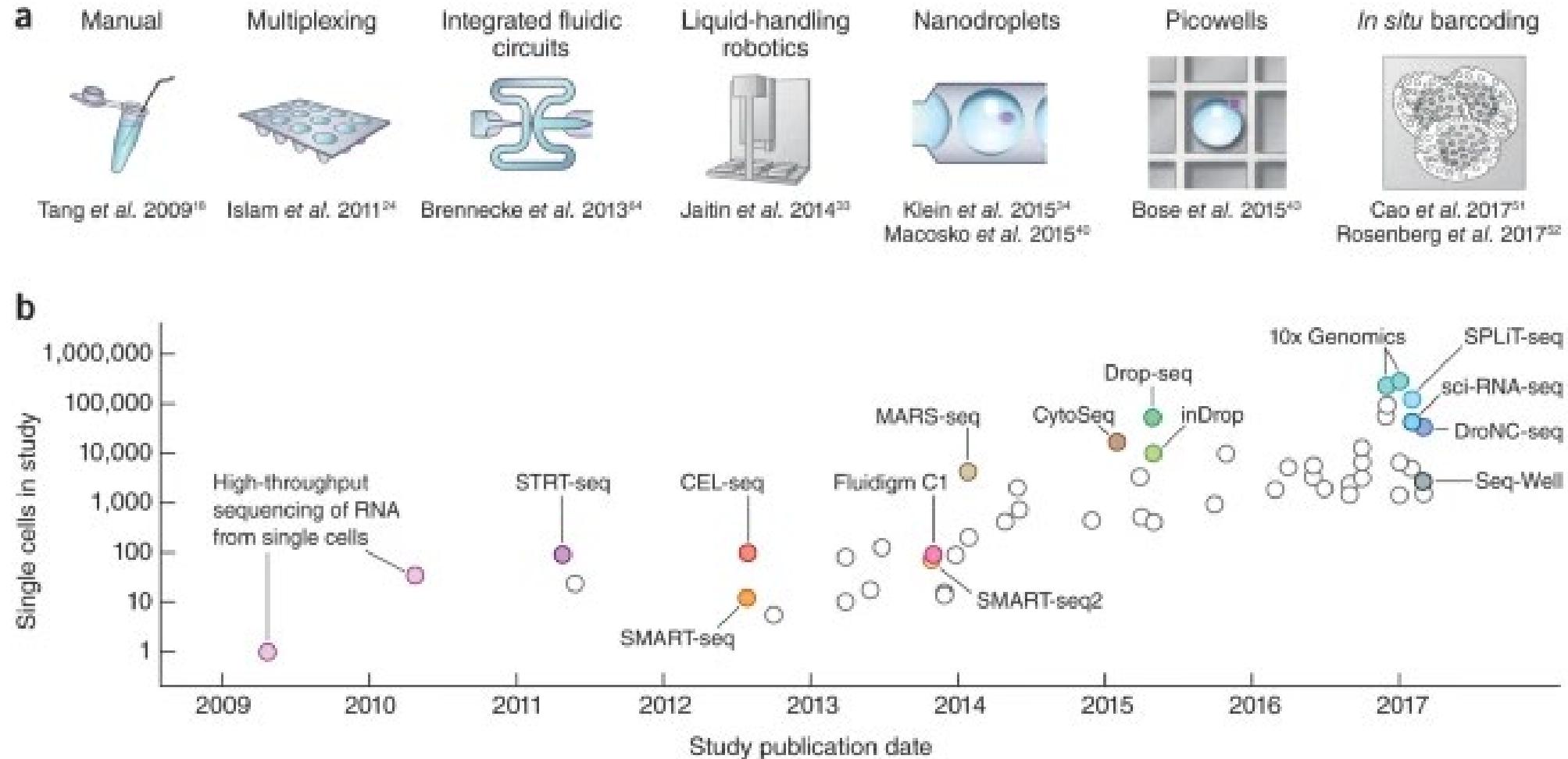
1. Introduction to single cell
2. Setting up Rstudio, data and count matrix
3. Pre-processing
4. Application 1: Cell annotation
5. Application 2: Case vs Control



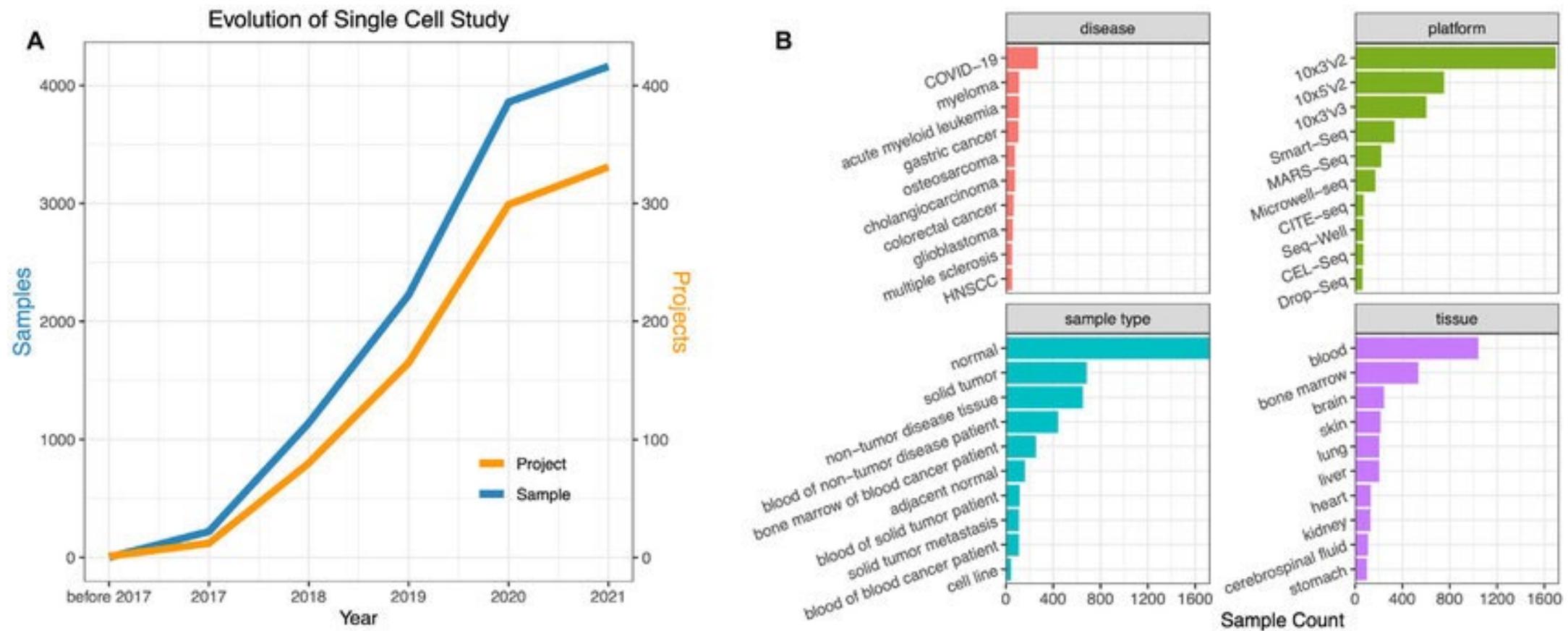
Overview of single cell



Exponential increase in the number of cells



Increase in single-cell samples (2022)



Increase in single-cell tools (2021)



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/>

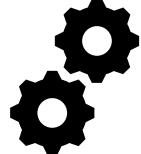


Overview

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!

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

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

1

Hi there!

Sara Ballouz has invited you to join Posit Cloud. By clicking on the link below, you can join the space that has been shared with you. Note that you will first be prompted to create an Posit Cloud account.

Click the link below to sign up now:

https://login.posit.cloud/invite?code=CGS89wYgumq4bRN-l67us4eMhihZDaU2Hslfp1sF&space_name=single+cell+workshop

- The Posit Cloud Team

2

posit



You have been invited to join a space on Posit Cloud by sara.ballouz@gmail.com.

Please log in or sign up to continue.

Already have an account?
Log In

Sign Up

3

single cell workshop Menu ▾
Kirby Institute, UNSW, Sydney

All Content ▾ (1)

TYPE * ACCESS * SORT A Z ⌂

single_cell_pipeline ASSIGNMENT

RStudio Project Sara Ballouz Space members Created Aug 17, 2023 2:37 PM



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

4

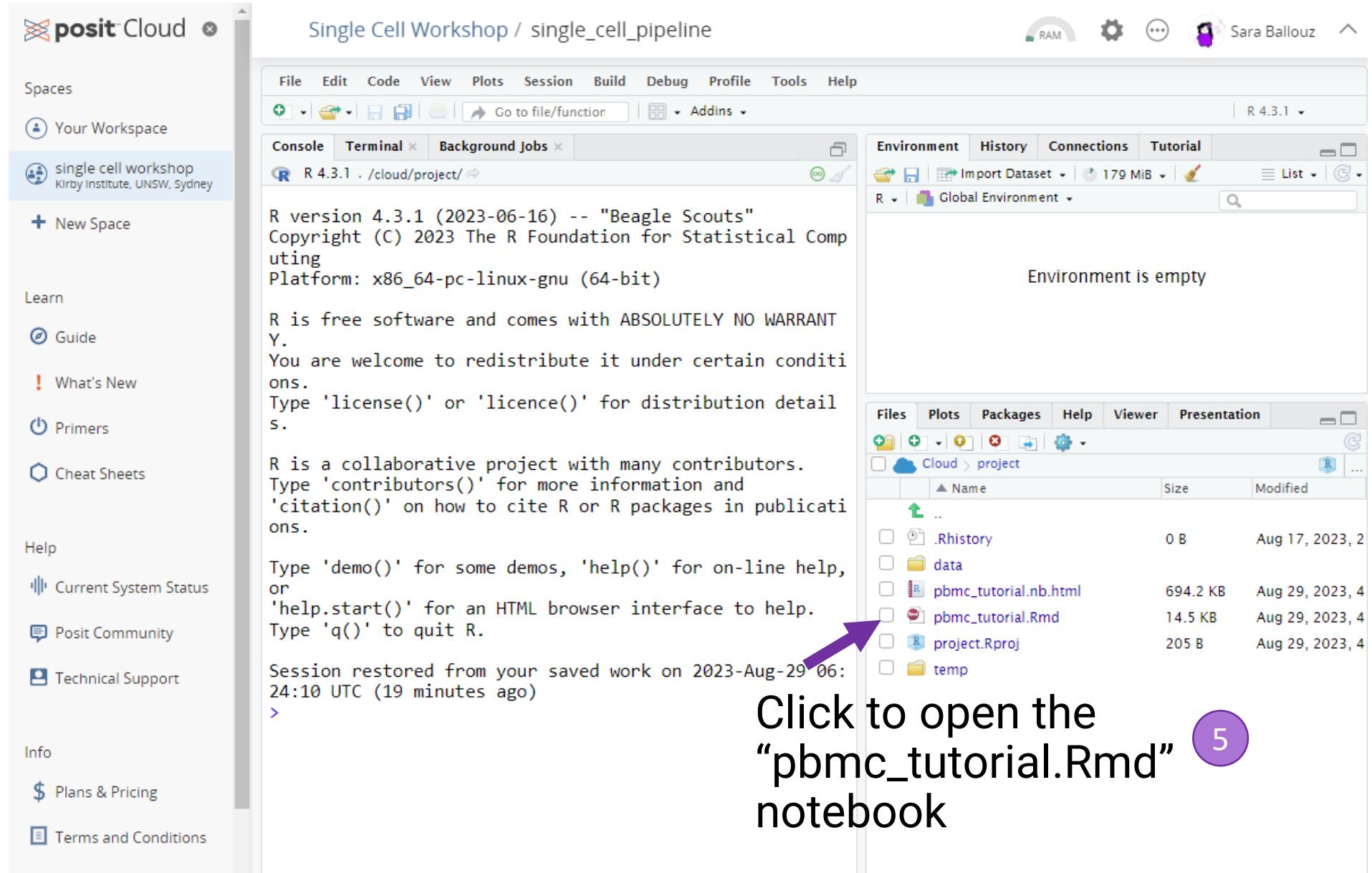
The screenshot shows the RStudio interface running in posit Cloud. The left sidebar displays the 'Your Workspace' section with a single item: 'single cell workshop Kirby Institute, UNSW, Sydney'. The main workspace shows the R console output for a new session:

```
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)  
>
```

The top right corner of the RStudio window has three black gear icons. The bottom right corner of the slide features the UNSW Sydney logo.



UNSW
SYDNEY



Click to open the
“pbmc_tutorial.Rmd”
notebook



Notice there
are four
windows.
We will work
mostly here

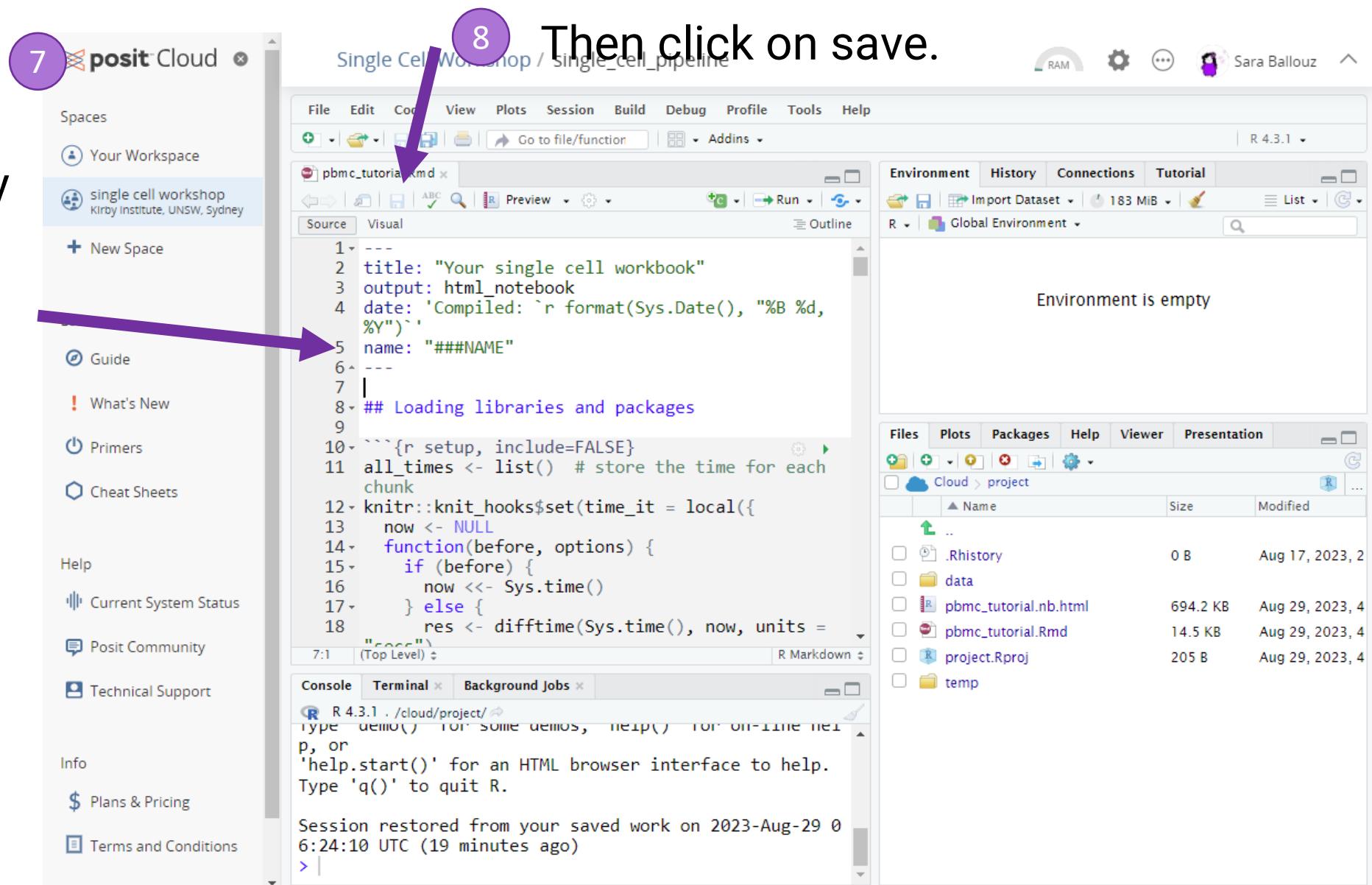
The screenshot shows the posit Cloud RStudio interface with four main windows:

- Source Editor:** The central window displays the R Markdown file `pbmc_tutorial.Rmd`. The code includes setup instructions, library loading, and a function to calculate execution time.
- Environment:** The top right window shows the global environment, which is currently empty.
- Files:** The bottom right window lists the contents of the current project directory, including files like `.Rhistory`, `data`, and `pbmc_tutorial.nb.html`.
- Console:** The bottom left window shows the R console output, indicating a session was restored from saved work on 2023-Aug-29 at 6:24:10 UTC.

A purple arrow points from the text "We will work mostly here" to the Source Editor window. A purple circle with the number "6" is positioned near the Learn section of the sidebar.

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

*Remember
to keep the
“ ”*





posit Cloud

Single Cell Workshop / single_cell_pipeline

File Edit Code View Plots Session Build Debug Profile Tools Help

RAM Addins R 4.3.1

pbmc_tutorial.Rmd x

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")  
19 cat(res)  
20 }  
21 })  
22 )  
23 .Rhistory  
24 data  
25 pbmc_tutorial.nb.html  
26 pbmc_tutorial.Rmd  
27 project.Rproj  
28 temp
```

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		

Console Terminal Background Jobs

R 4.3.1 ./cloud/project/
Type demo() for some demos, help() for on-line hel-
p, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

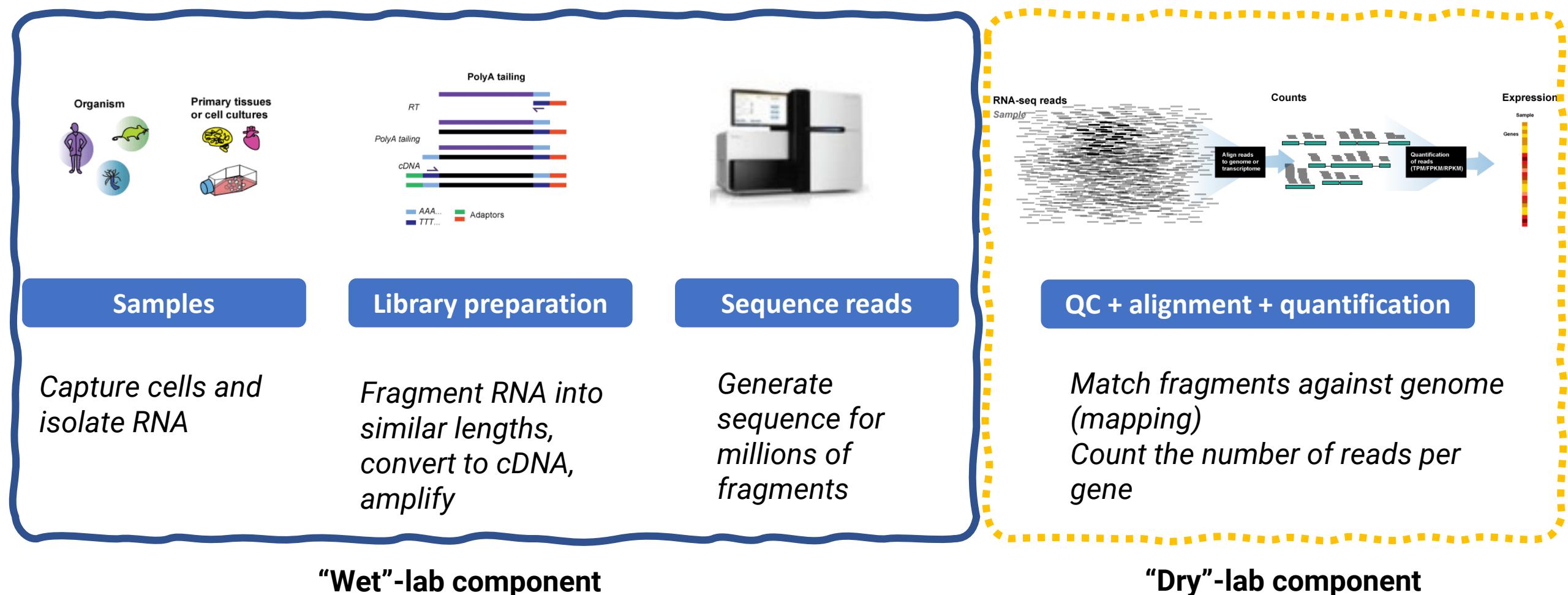
Session restored from your saved work on 2023-Aug-29 0
6:24:10 UTC (19 minutes ago)

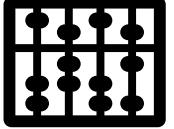
>



UNSW
SYDNEY

Background: pre-count pipeline overview





Obtaining the count matrix



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](#).

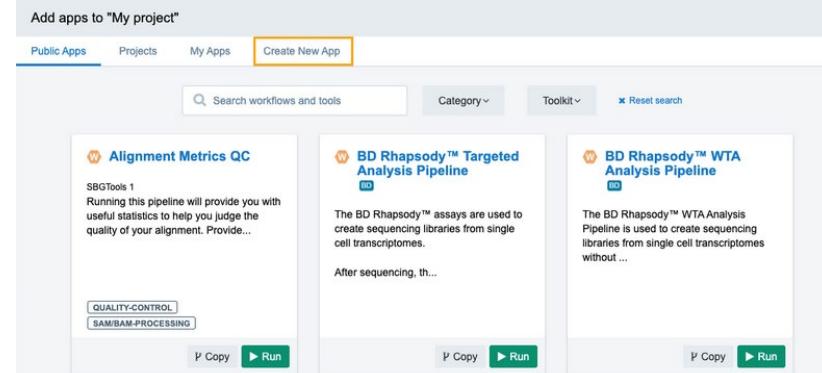
Starting in Cell Ranger 7.0, the expected number of cells can either be auto-estimated or specified with `--expect-cells` (e.g., to replicate a previous analysis), see [Gene Expression algorithm overview](#). If needed, automated cell calling can be overridden with the `--force-cells` option.

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`:

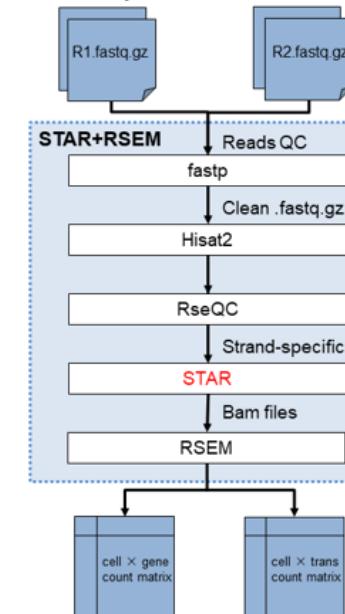
```
$ cd /home/jdoe/runs
$ cellranger count --id=sample345 \
  --transcriptome=/opt/refdata-gex-GRCh38-2020-A \
  --fastqs=/home/jdoe/runs/HANTADXX/outs/fastq_path \
  --sample=mysample \
  --localcores=8 \
  --localmem=64
```

cellranger

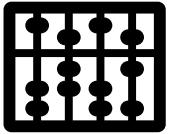


Seven Bridges

Smart-SEQ



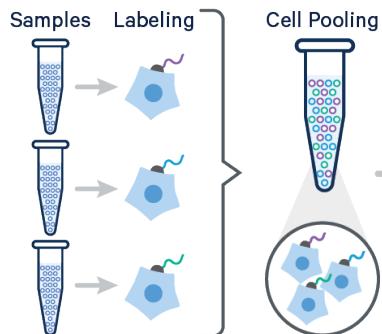
https://broadinstitute.github.io/warp/docs/Pipelines/Smart-seq2_Single_Sample_Pipeline/README/



Pipeline from 10X for gene expression



----- Cell Sample -----



```
$ cellranger count --id=sample345 \
    --transcriptome=/opt/refdata-gex-GRCh38-2020-A \
    --fastqs=/home/jdoe/runs/HAWT7ADXX/outs/fastq_path \
    --indices=SI-3A-A1

Cell Ranger Count Pipeline - v4.0.8

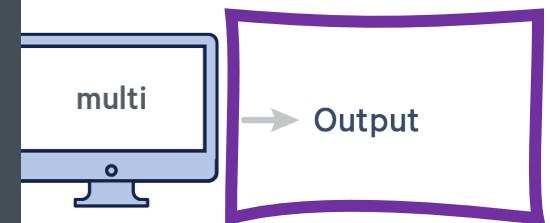
Martian Runtime - v4.0.8

Running preflight checks (please wait)...
2016-01-01 10:23:52 [runtime] (ready)           ID.sample345.CELLRANGER_CS.CELLRANGER.SETUP_CHUNKS
...
2012-01-01 12:10:09 [runtime] (join_complete)     ID.sample345.CELLRANGER_CS.CELLRANGER.SUMMARIZE_REPORTS

Outputs:
- Run summary HTML:                      /home/jdoe/runs/sample345/outs/web_summary.html
- Run summary CSV:                        /home/jdoe/runs/sample345/outs/metrics_summary.csv
- BAM:                                    /home/jdoe/runs/sample345/outs/posorted_genome_bam.bam
- BAM index:                             /home/jdoe/runs/sample345/outs/posorted_genome_bam.bam.bai
- Filtered feature-barcode matrices MEX:   /home/jdoe/runs/sample345/outs/filtered_feature_bc_matrix
- Filtered feature-barcode matrices HDF5:  /home/jdoe/runs/sample345/outs/filtered_feature_bc_matrix.h5
- Unfiltered feature-barcode matrices MEX: /home/jdoe/runs/sample345/outs/raw_feature_bc_matrix
- Unfiltered feature-barcode matrices HDF5: /home/jdoe/runs/sample345/outs/raw_feature_bc_matrix.h5.h5
- Secondary analysis output CSV:          /home/jdoe/runs/sample345/outs/analysis
- Per-molecule read information:         /home/jdoe/runs/sample345/outs/molecule_info.h5
- Loupe Browser file:                   /home/jdoe/runs/sample345/outs/cloupe.cloupe

Pipelinstance completed successfully!
```

----- Pipeline -----

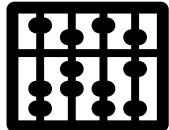


UNSW
SYDNEY

Single Cell Workshop / single_cell_pipeline



Sara Ballouz



File Edit Code View Plots Session Build Debug Profile Tools Help

pbmc_tutorial.Rmd x Go to file/function Addins R 4.3.1

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: "##NAME"  

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")  

19     }  

20   })  

21 })  

22```  

23  

24# Your single cell workbook
```

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 0
6:24:10 UTC (19 minutes ago)

Environment History Connections Tutorial

Import Dataset 223 MiB List C

R Global Environment

Environment is empty

9

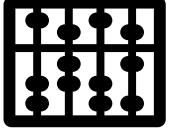
Files Plots Packages Help Viewer Presentation

Cloud > project > data > pbmc3k.h5ad filtered_feature_bc_matrix

Name	Size	Modified
..		
barcodes.tsv.gz	50 KB	Aug 17, 2023, 3
features.tsv.gz	325.8 KB	Aug 17, 2023, 3
matrix.mtx.gz	75 MB	Aug 17, 2023, 3

Barcodes – identifies cells
Features – genes/transcripts
Matrix – counts (“reads”) for each barcode, feature pair



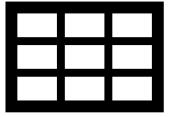


The gene expression count matrix

	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

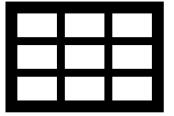


Setting up environment

The screenshot shows the RStudio interface within the posit Cloud environment. The left sidebar displays 'Your Workspace' and 'single cell workshop' under 'Spaces'. The main window shows a file named 'pbmc_tutorial.Rmd' with R code for setting up a pipeline. A purple circle with the number 10 is overlaid on the 'Run Current Chunk' button in the toolbar.

```
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$set(time_it = local({  
13   now <- NULL  
14   function(before, options) {  
15     if (before) {  
16       now <- Sys.time()  
17     } else {  
18       res <- difftime(Sys.time(), now, units = "secs")  
19       all_times[[options$label]] <- res
```

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



Loading in libraries

The screenshot shows the posit Cloud RStudio interface. On the left, the sidebar includes 'Spaces' (Your Workspace, single cell workshop), 'New Space', 'Learn' (Guide, What's New, Primers, Cheat Sheets), and 'Help' (Current System Status, Posit Community). The main area displays an R Markdown file titled 'pbmc_tutorial.Rmd' under 'Single Cell Workshop / single_cell_pipeline' by Sara Ballouz. The code in the 'Source' tab shows:

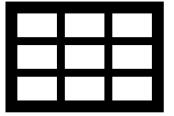
```
47 stop("Please install azimuth -  
remotes::install_github('satijalab/azimuth')", call. = FALSE)  
48 }  
49  
50 `--  
51 `--{r load libraries}  
52 library(Seurat)  
53 library(dplyr)  
54 library(patchwork)  
55 library(Azimuth)  
56 library(ggplot2)  
57 `--  
58  
59  
60  
61 ## Setup the Seurat Object  
62  
63 For this tutorial, we will be analyzing the a dataset of  
53:16 C Chunk 2: load libraries
```

A purple circle with the number '11' is overlaid on the code area. Below the code editor is the 'Console' tab, which shows:

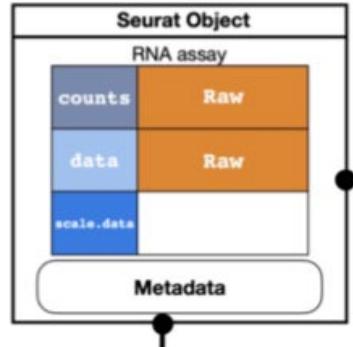
```
R 4.3.1 . /cloud/project/  
> if (!requireNamespace("glmGamPoi", quietly = TRUE)) {  
+   if (!requireNamespace("BiocManager", quietly = TRUE)) {
```

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



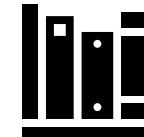


What are Seurat objects?

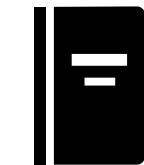


The Seurat object is a **representation** of single-cell expression data for R; each Seurat object revolves around a set of cells and consists of one or more Assay objects, or individual representations of expression data (eg. RNA-seq, ATAC-seq, etc).

Think of it as a bookshelf, with books.

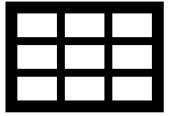


Each book is an assay.
e.g., it holds a book on the RNA-seq assay.



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





Setup the Seurat Object

Single Cell Workshop / single_cell_pipeline Sara Ballouz

File Edit Code View Plots Session Build Debug Profile Tools Help

pbmc_tutorial.Rmd x

Source Visual

```
number of molecules for each feature (i.e. gene; row) that are detected in each cell (column).  
79  
80  
81 -```{r init}  
82 # Load the PBMC dataset  
83 pbmc.data <- Read10X(data.dir =  
84 "data/pbmc10k/filtered_feature_bc_matrix/")  
85 # Initialize the Seurat object with the raw (non-normalized data).  
86 pbmc <- CreateSeuratObject(counts = pbmc.data$`Gene Expression`, project = "pbmc10k", min.cells = 3, min.features = 200)  
87 pbmc  
88 -```
```

53:16 Chunk 2: load libraries R Markdown

Console Terminal Background Jobs

12

13

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.



UNSW
SYDNEY

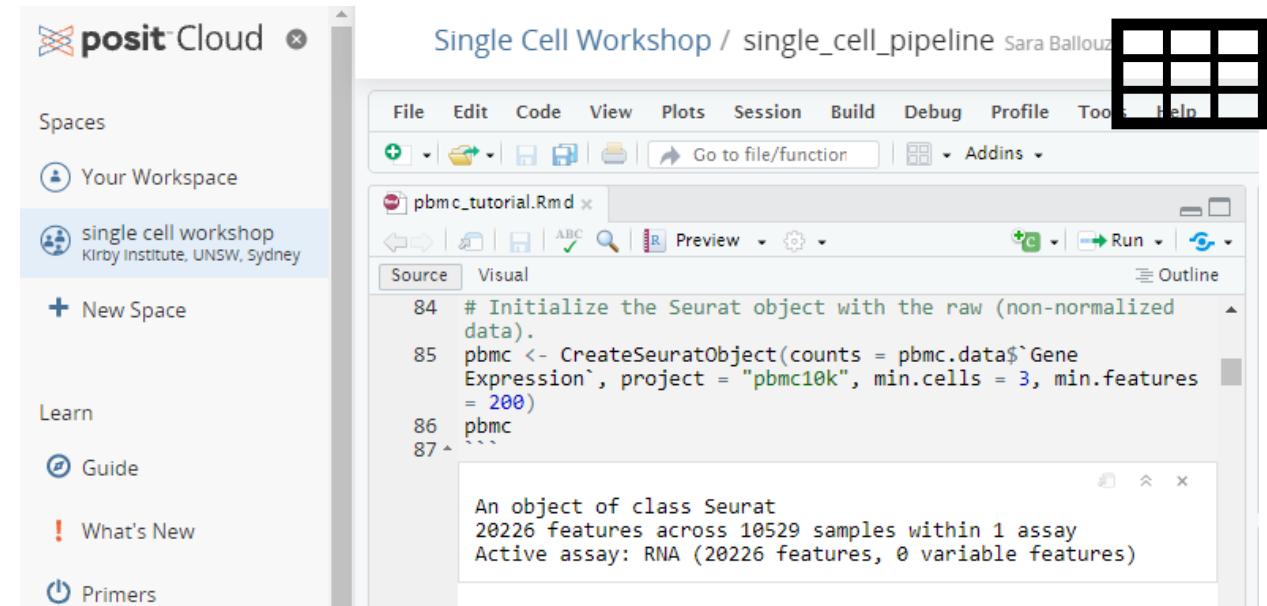
Data loaded!

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.



The screenshot shows the posit Cloud interface. On the left, there's a sidebar with 'Spaces' (Your Workspace, single cell workshop - Kirby Institute, UNSW, Sydney selected), 'New Space', 'Learn', 'Guide', 'What's New', and 'Primers'. The main area is a code editor with tabs for 'Source' and 'Visual'. The 'Source' tab shows R code for initializing a Seurat object from a pbmc dataset. The 'Visual' tab shows the output of the code execution, which is an object of class Seurat with 20226 features across 10529 samples. A status bar at the bottom indicates 'Active assay: RNA (20226 features, 0 variable features)'.

```
84 # Initialize the Seurat object with the raw (non-normalized  
85 pbmc <- CreateSeuratObject(counts = pbmc.data$`Gene  
Expression`, project = "pbmc10k", min.cells = 3, min.features  
= 200)  
86 pbmc  
87
```

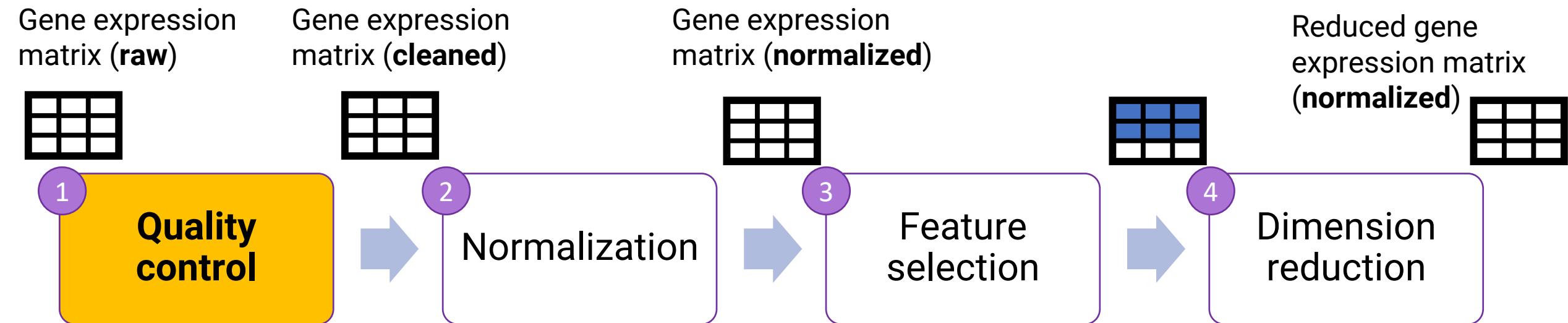
An object of class Seurat
20226 features across 10529 samples within 1 assay
Active assay: RNA (20226 features, 0 variable features)

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



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: $175 / (175 + 24 + 36)$

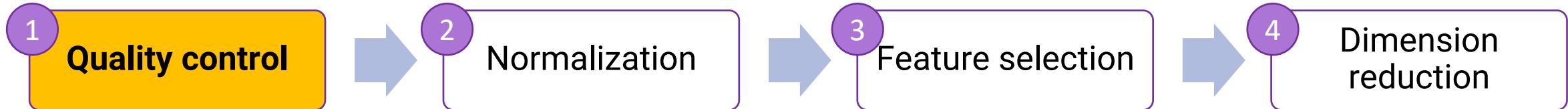
Cell 2: 1

for each cell, fraction of expression from mitochondrial genes

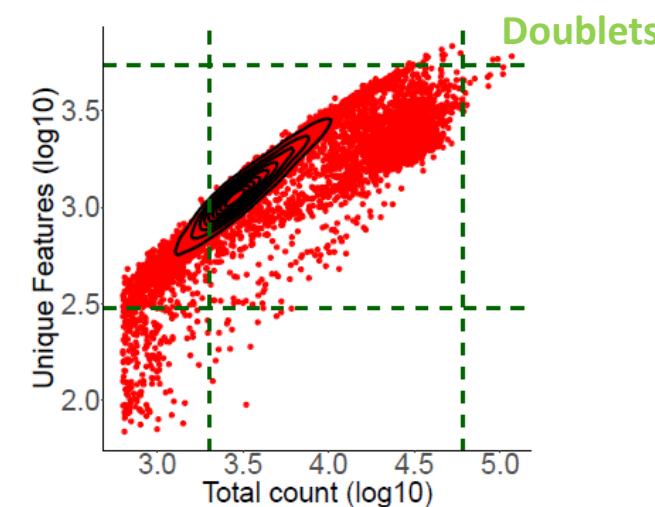
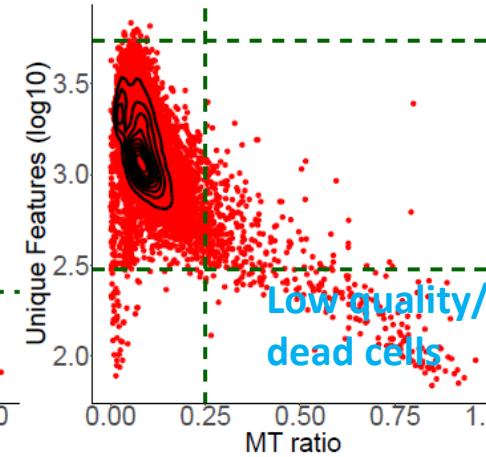
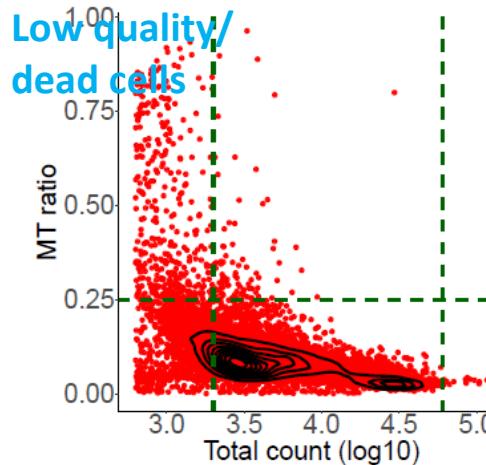


UNSW
SYDNEY

Preprocessing: quality control



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



Count matrix check



The screenshot shows the RStudio IDE interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. Below the menu is a toolbar with various icons for file operations like Open, Save, Print, and Go to file/function. A dropdown menu for Addins is also present. The main workspace shows an R Markdown file named "pbmc_tutorial.Rmd*". The code editor displays the following R code:

```
97  ##> ### Checking the count matrix
98
99  What does data in a count matrix look like?
100
101  ````{r}
102  # Lets examine a few genes in the first thirty cells
103  pbmc.data$`Gene Expression`[c("CD3D","TCL1A","MS4A1"), 1:30]
104  ````
```

A modal window is open below the code editor, showing the output of the R code. The output is a sparse matrix object:

```
3 x 30 sparse Matrix of class "dgCMatrix"
CD3D . 5 . . 4 1 . . . 1 . 4 . . 7 3 . . 2 . . . 6 2 6 . 3 .
TCL1A . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2 . 7
MS4A1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11 . . 7 . 17 . . . . 8 . 3

CD3D 2 .
TCL1A . .
```

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

Note “.” are 0 counts.



MT-ratio



Single Cell Workshop / single_cell_pipeline Sara Ballouz

File Edit Code View Plots Session Build Debug Profile Tools Help

pbmc_tutorial.Rmd*

Source Visual

```
125 + We use the set of all genes starting with `MT-` as a set of
mitochondrial genes
126
127
128 ````{r mito}
129 # The [[ operator can add columns to object metadata. This is a great
place to stash QC stats
130 pbmc[["percent.mt"]] <- PercentageFeatureSet(pbmc, pattern = "MT-")
131 ````
```

2

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”



UNSW
SYDNEY

Visualizing QC metrics



posit Cloud

Spaces

- Your Workspace
- single cell workshop Kirby Institute, UNSW, Sydney
- New Space

Learn

- Guide
- What's New
- Primers
- Cheat Sheets

Single Cell Workshop / single_cell_pipeline Sara Ballouz

File Edit Code View Plots Session Build Debug Profile Tools Help

pbmc_tutorial.Rmd

Source Visual

```
148
149 `r qc2, fig.height=7, fig.width=13}
150 #Visualize QC metrics as a violin plot
151 VlnPlot(pbmc, features = c("nFeature_RNA", "nCount_RNA",
152 "percent.mt"), ncol = 3)
153 # FeatureScatter is typically used to visualize feature-feature
154 relationships, but can be used for anything calculated by the object,
155 i.e. columns in object metadata, PC scores etc.
156 plot1 <- FeatureScatter(pbmc, feature1 = "nCount_RNA", feature2 =
157 "percent.mt") #+geom_hline(yintercept=0.2)
158 plot2 <- FeatureScatter(pbmc, feature1 = "nCount_RNA", feature2 =
159 "nFeature_RNA") # +geom_hline(yintercept=1000)
160 plot1 + plot2
161
162:25
```

Chunk 10: ac3

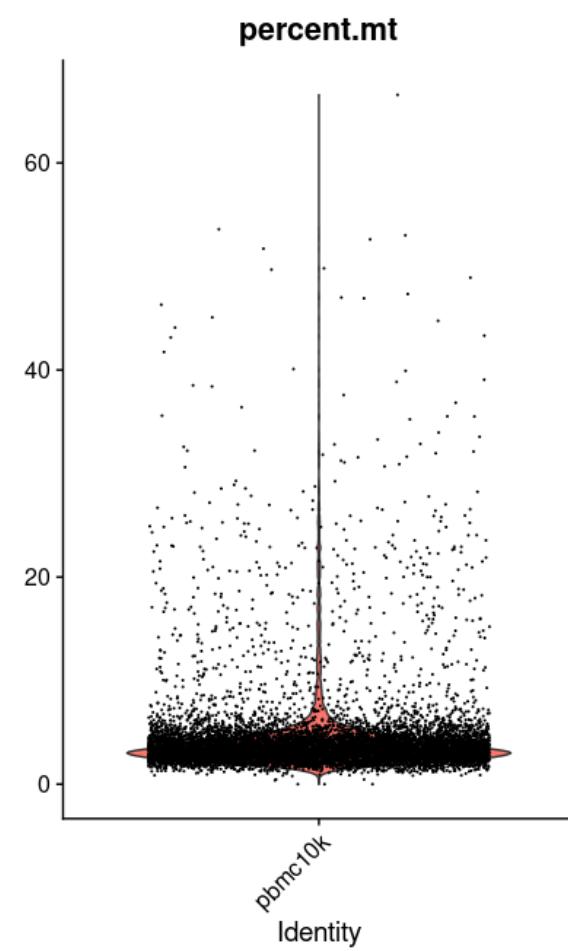
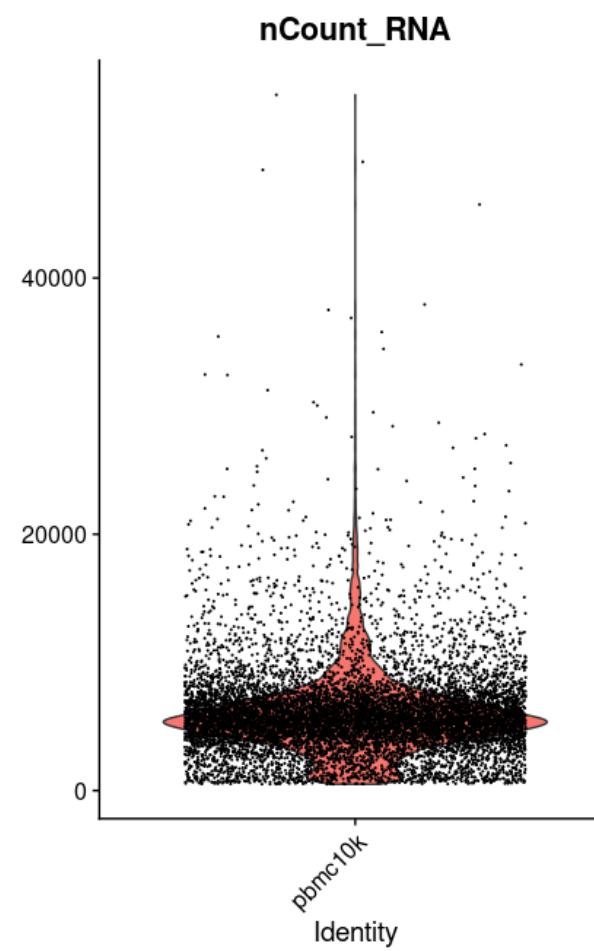
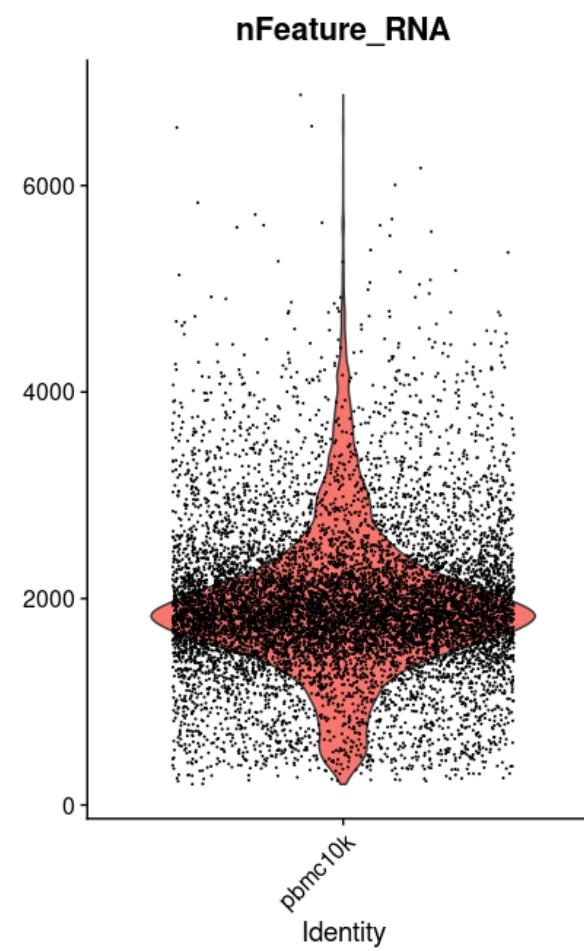
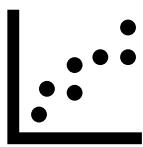
R Markdown

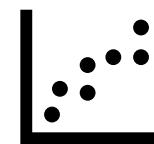
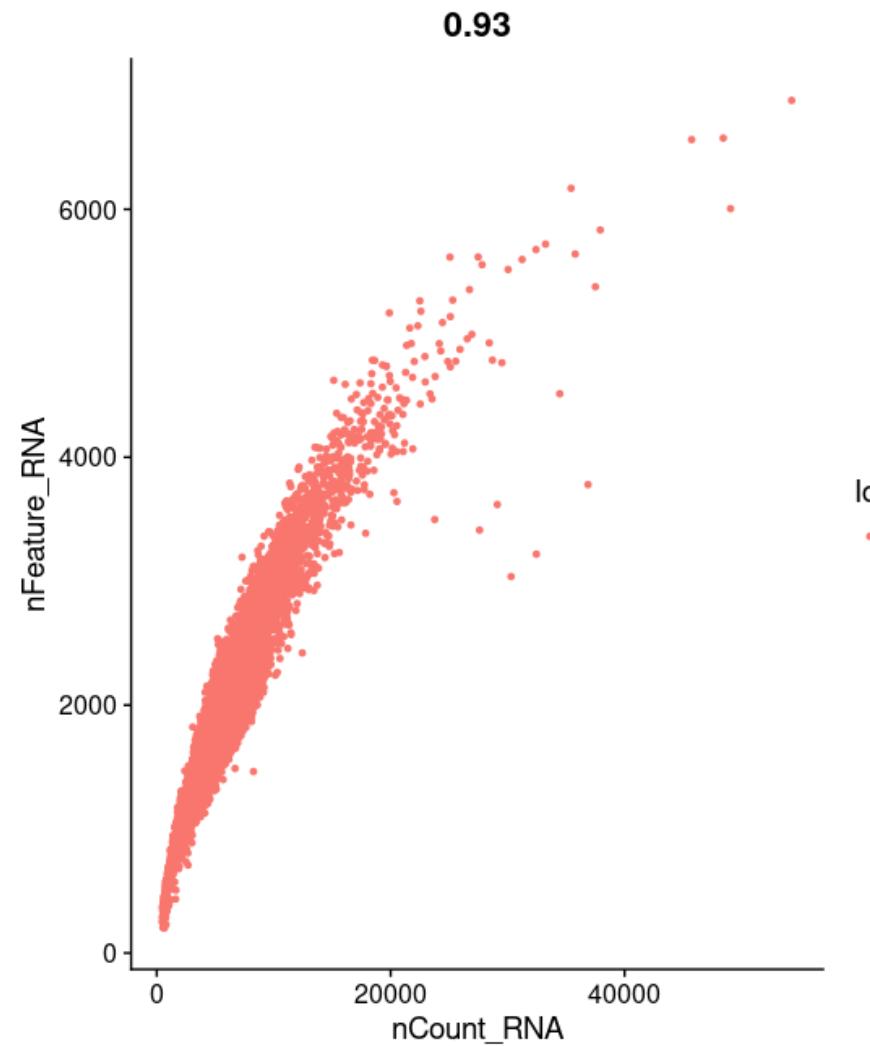
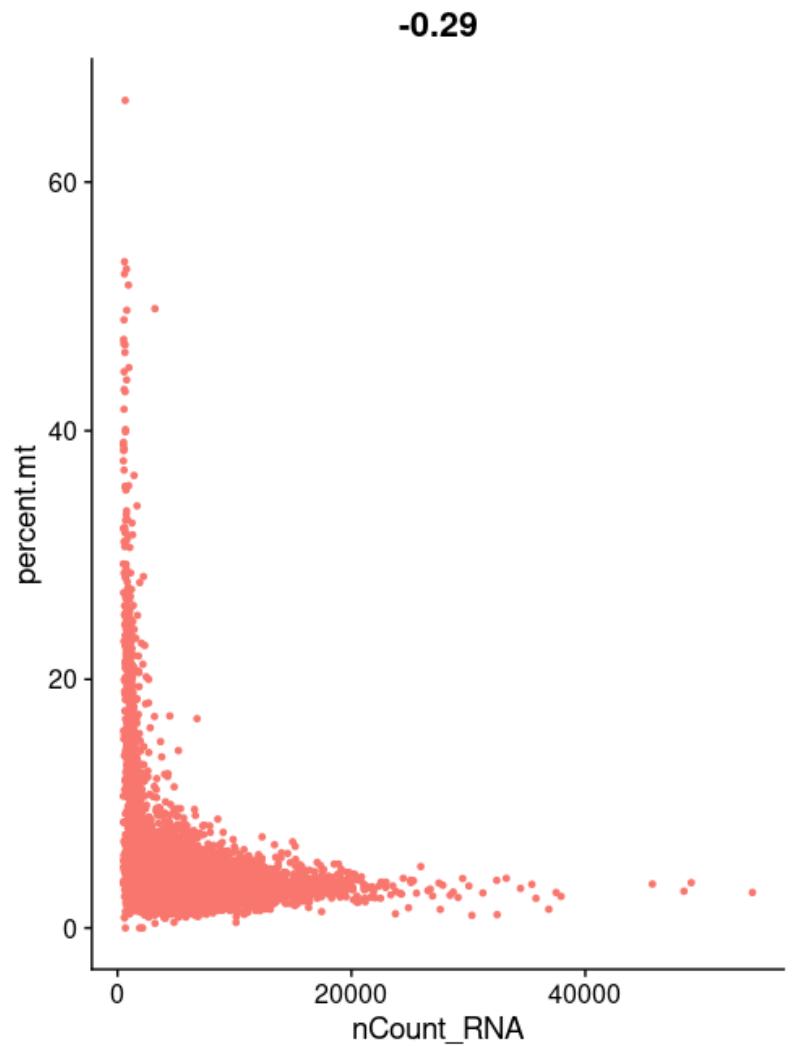
Run the chunk to plot our QC metrics.

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.







Filtering away poor-quality cells



posit Cloud

Your Workspace

single cell workshop
Kirby Institute, UNSW, Sydney

New Space

Learn

Guide

What's New

Primers

Single Cell Workshop / single_cell_pipeline Sara Ballouz

File Edit Code View Plots Session Build Debug Profile Tools Help

pbmcTutorial.Rmd*

Source Visual

```
161 `r qc3, fig.height=7, fig.width=13}
162 pbmc <- subset(pbmc, subset = nFeature_RNA > 200 & nFeature_RNA <
2500 & percent.mt < 5)
163 pbmc
164 ```

An object of class Seurat
20226 features across 7499 samples within 1 assay
Active assay: RNA (20226 features, 0 variable features)
```

3

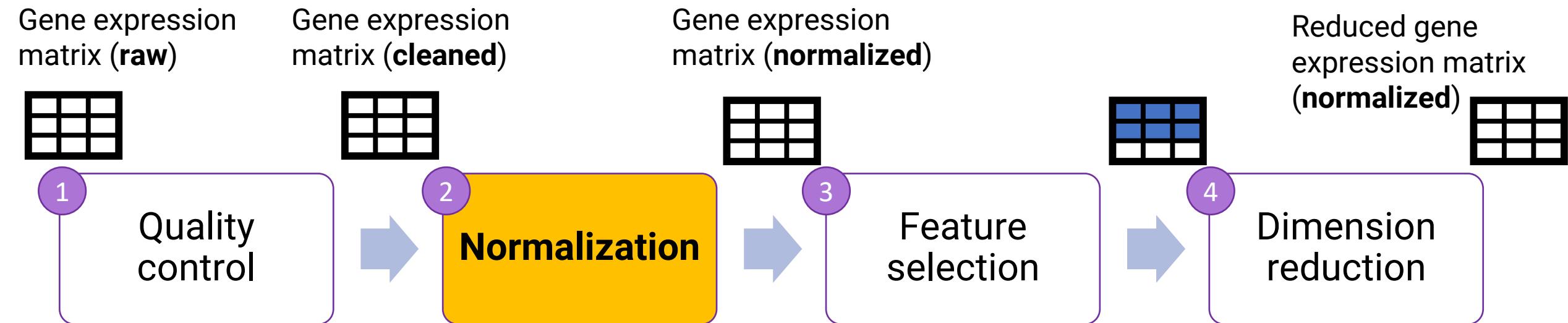
Once we've decided on our thresholds for the individual metrics, we filter away the poor-quality cells.

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



UNSW
SYDNEY

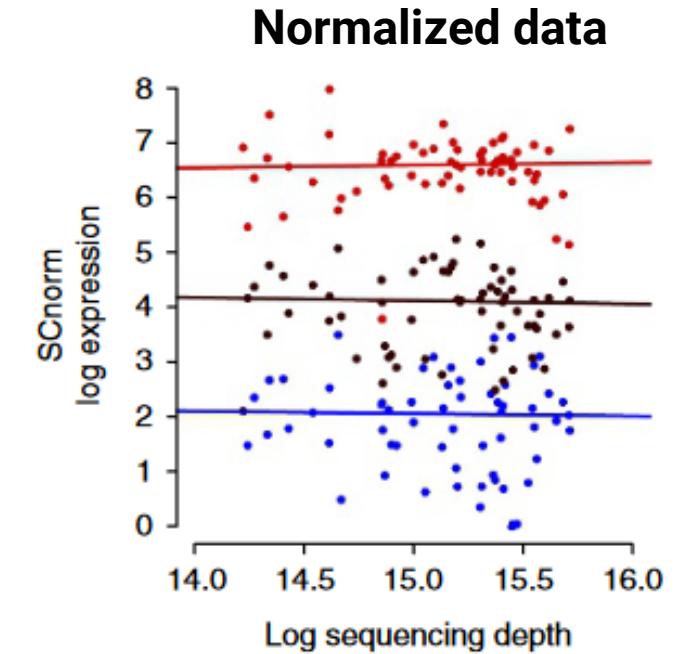
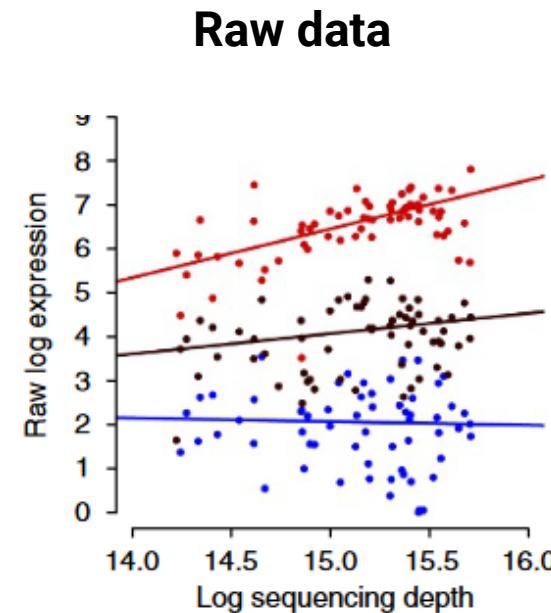
Preprocessing pipeline



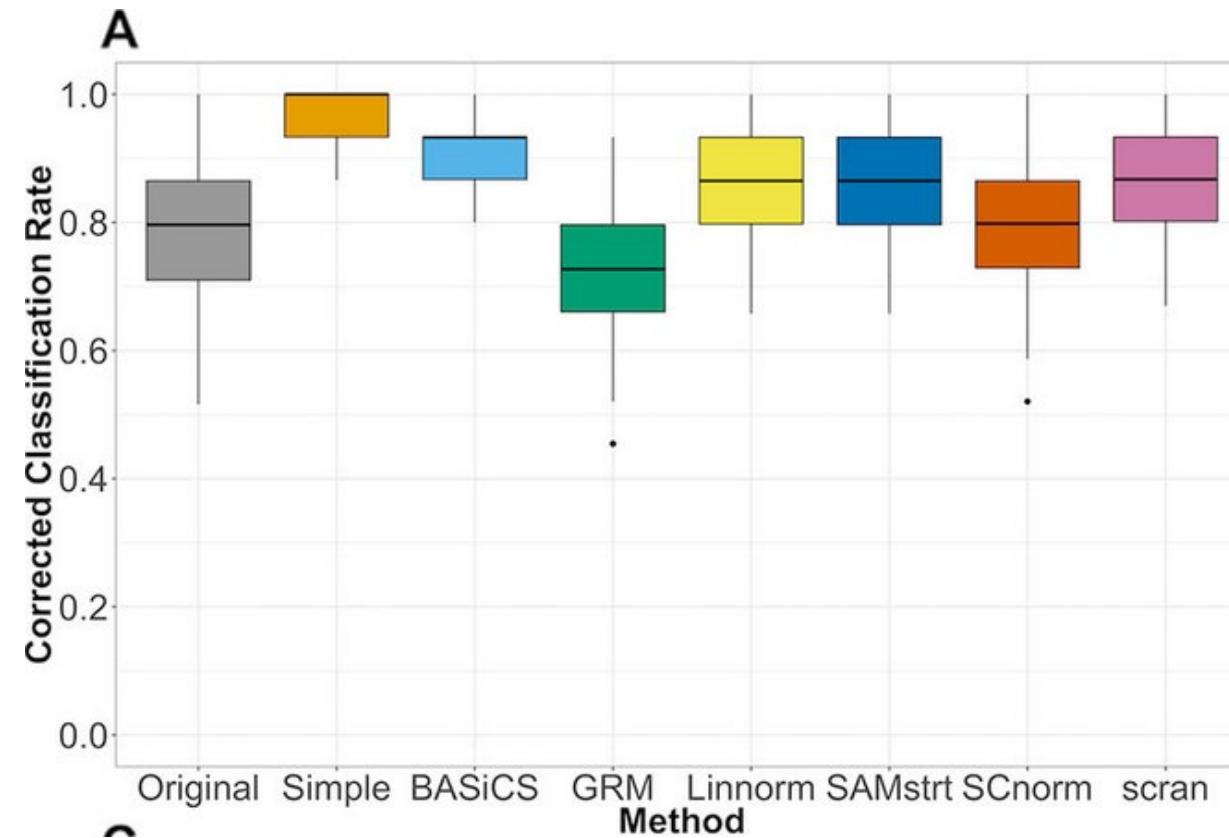
Normalization



- 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



Normalization (different methods)



Normalization



```
normalize the data.  
170 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. Normalized  
values are stored in `pbmc[["RNA"]][@data` . These values are provided  
as default.  
171  
172 ``{r normalize.default, eval = FALSE}  
173 pbmc <- NormalizeData(pbmc)  
174 ``
```

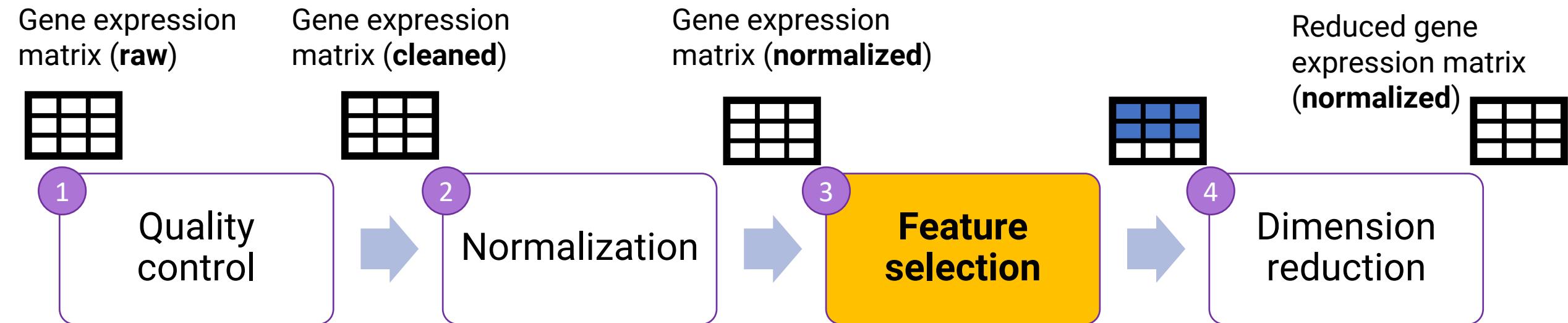
4

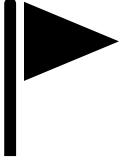
Run the code chunk, this should perform the log-normalization, and add a new slot to our object called `data`.

5

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

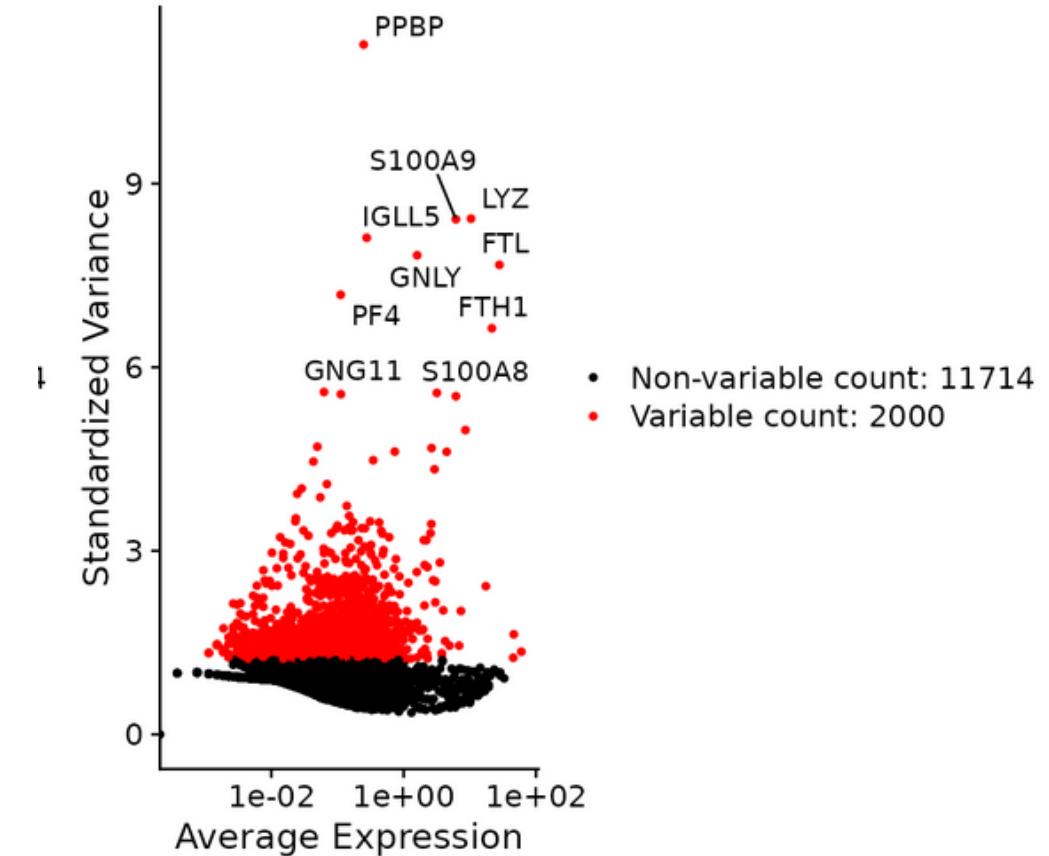
Preprocessing pipeline

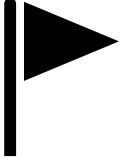




Feature selection

- 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)





Feature selection

The screenshot shows the RStudio interface with the title bar "Single Cell Workshop / single_cell_pipeline Sara Ballouz". The left sidebar includes "Your Workspace" and "single cell workshop Kirby Institute, UNSW, Sydney". The main area displays an R Markdown file "pbmc_tutorial.Rmd" with the following code:

```
186  
187  
188 `r var_features, fig.height=5, fig.width=11}  
189 pbmc <- FindVariableFeatures(pbmc, selection.method = 'vst',  
nfeatures = 2000)  
190  
191 # Identify the 10 most highly variable genes  
192 top10 <- head(VariableFeatures(pbmc), 10)  
193  
194 # plot variable features with and without labels  
195 plot1 <- VariableFeaturePlot(pbmc)  
196 plot2 <- LabelPoints(plot = plot1, points = top10, repel = TRUE)  
197 plot1 + plot2  
198`
```

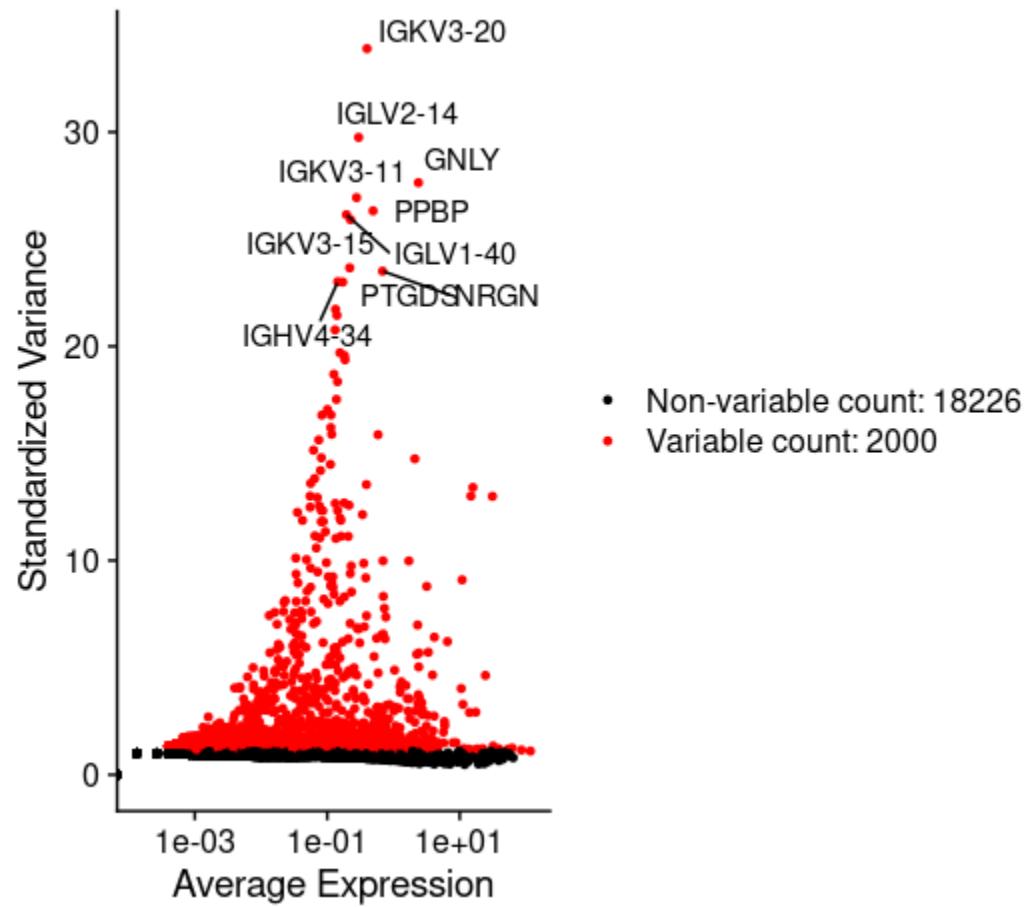
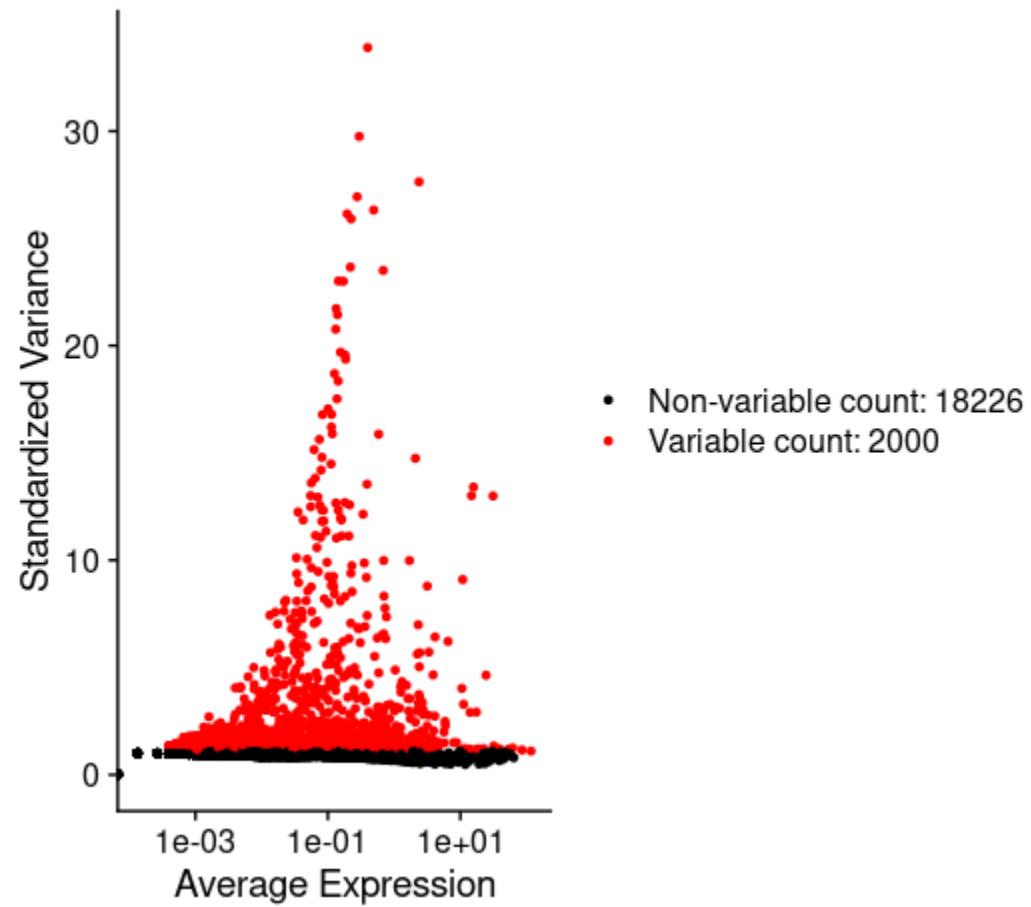
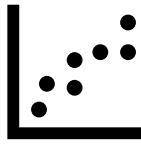
A purple circle with the number "6" is overlaid on the right side of the code editor.

What is notable about the top 10?

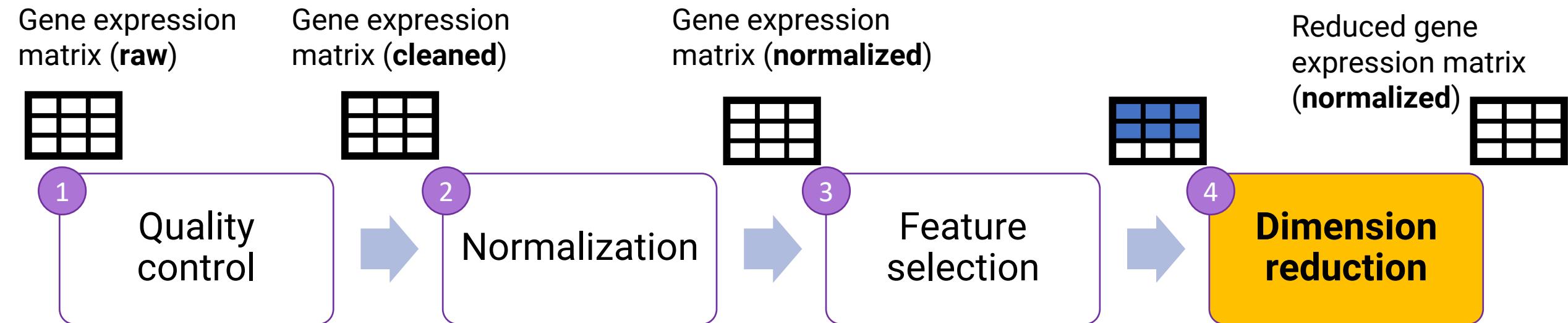
Run the code chunk, which will use the “VST” method (variance stabilisation transformation) to pick the most variable genes/features (we asked for 2000 genes).

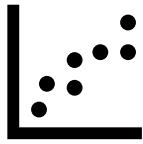
We can then plot the average expression against the standardized variance to show these genes and label the top 10.





Preprocessing pipeline





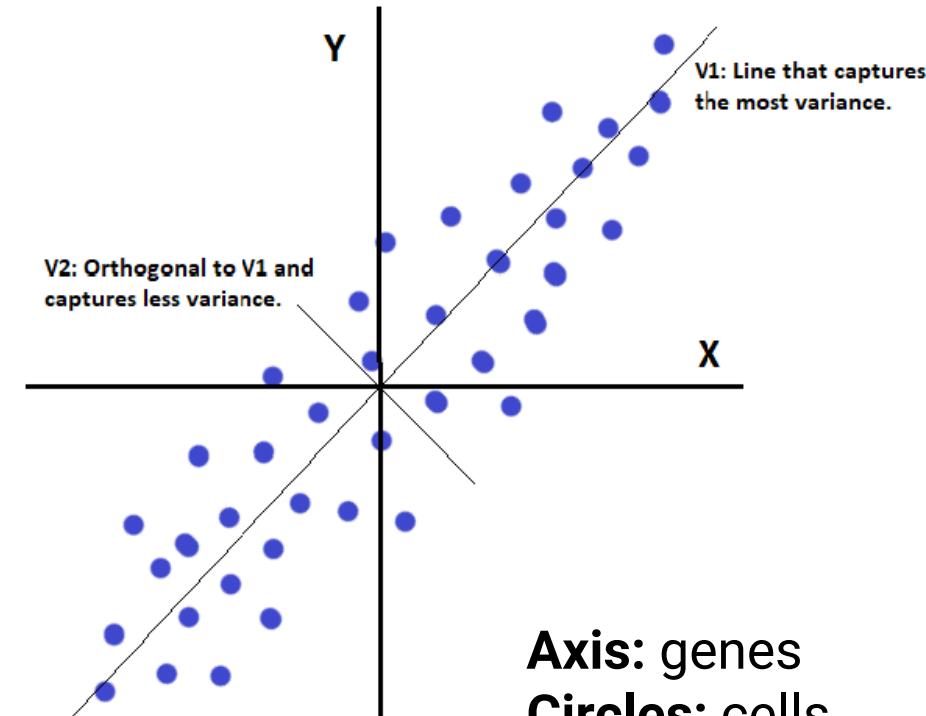
Dimension reduction

- **Problems:**

- Noisy data
- Computational processing
(many genes, many cells)

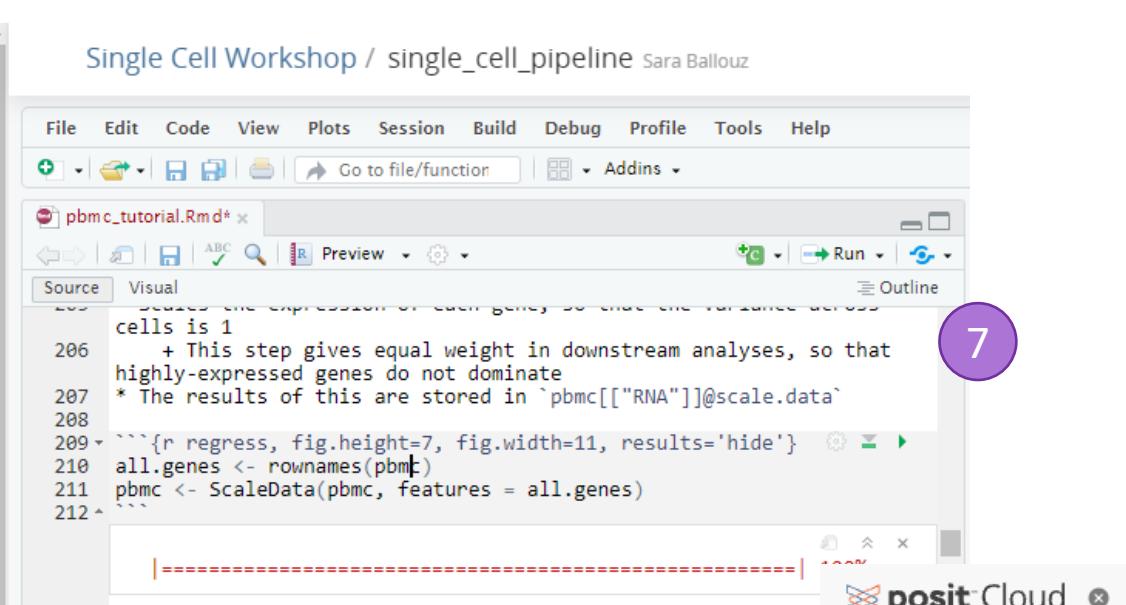
- **Solution:** Dimension reduction

- Reduces to a few dimensions which capture the “most important” information



Choose the line which captures most variance

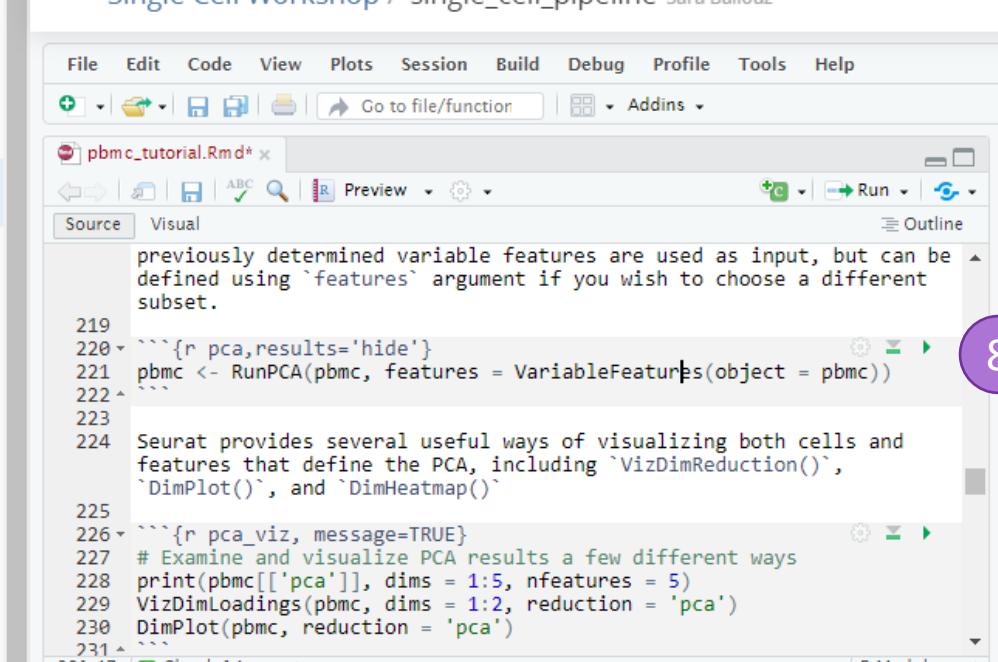
First, we scale the data. This is to transform it prior to running any dimension reduction. This data is added to the “scale.data” slot.



```

1  scaled the expression of each gene, so that the variance across
2  cells is 1
3      + This step gives equal weight in downstream analyses, so that
4  highly-expressed genes do not dominate
5  * The results of this are stored in `pbmc[["RNA"]][@scale.data]`
6
7  ````{r regress, fig.height=7, fig.width=11, results='hide'}
8  all.genes <- rownames(pbm)
9  pbmc <- ScaleData(pbm, features = all.genes)
10 ````
```

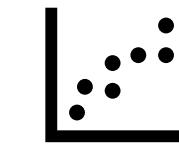
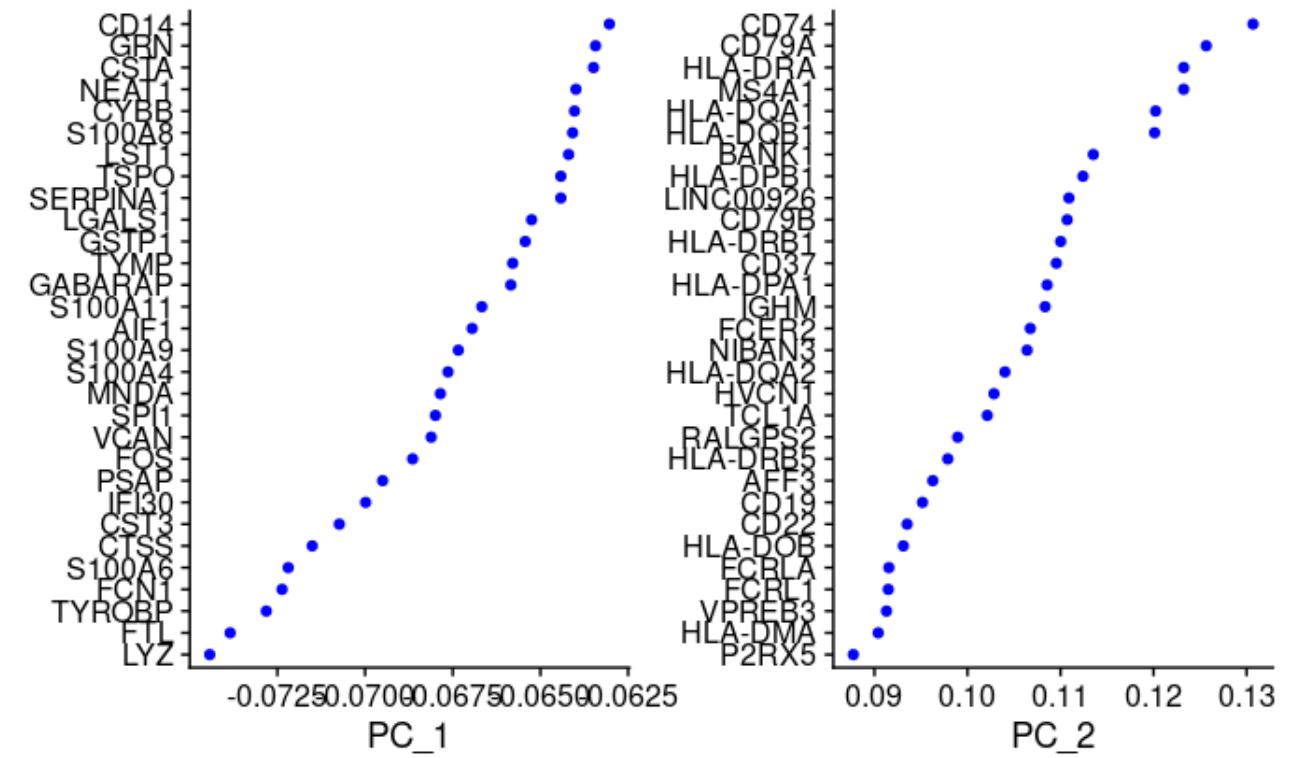
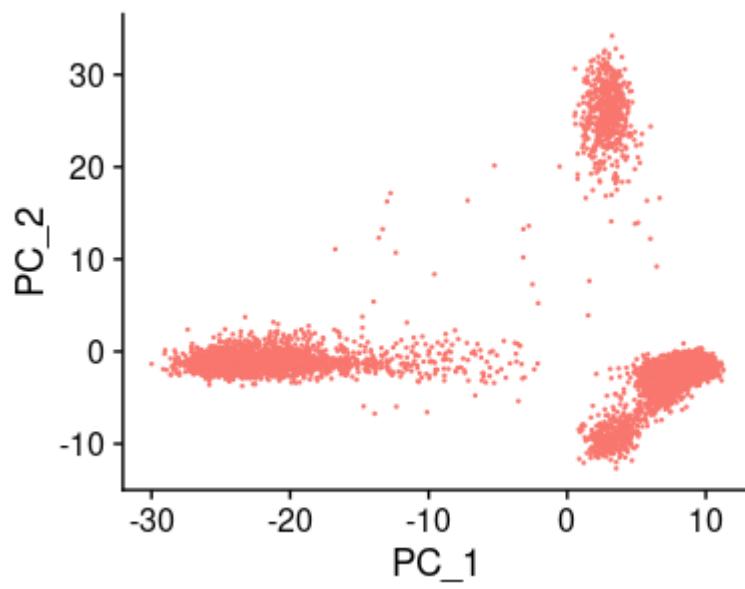
Next, we run PCA and visualise the data, looking at the first two principal components (PCs).

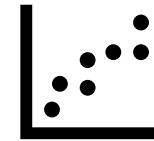
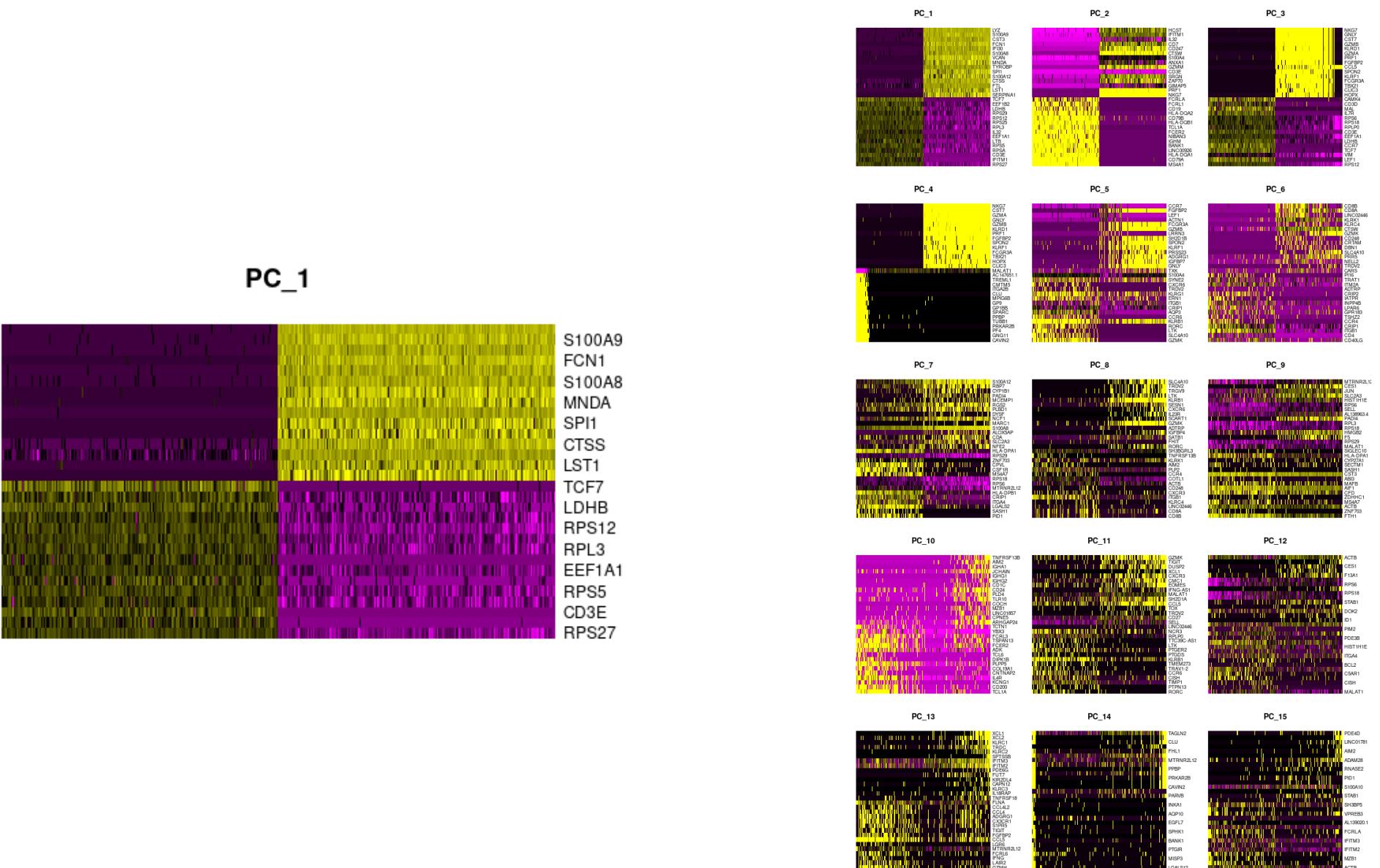


```

1  previously determined variable features are used as input, but can be
2  defined using `features` argument if you wish to choose a different
3  subset.
4
5  ````{r pca,results='hide'}
6  pbmc <- RunPCA(pbm, features = VariableFeatures(object = pbmc))
7
8
9  Seurat provides several useful ways of visualizing both cells and
10 features that define the PCA, including `VizDimReduction()`,
11 `DimPlot()`, and `DimHeatmap()`
12
13 ````{r pca_viz, message=TRUE}
14 # Examine and visualize PCA results a few different ways
15 print(pbm[["pca"]], dims = 1:5, nfeatures = 5)
16 VizDimLoadings(pbm, dims = 1:2, reduction = 'pca')
17 DimPlot(pbm, reduction = 'pca')
18 ````
```







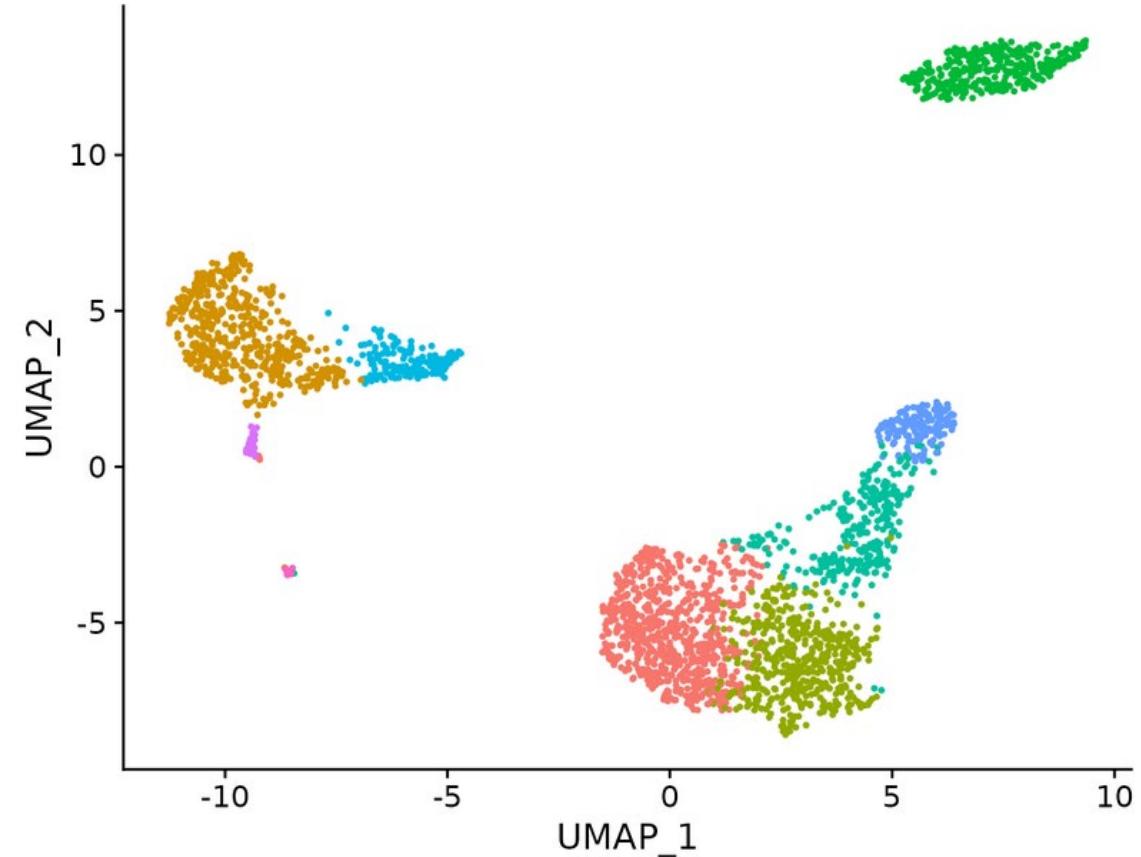
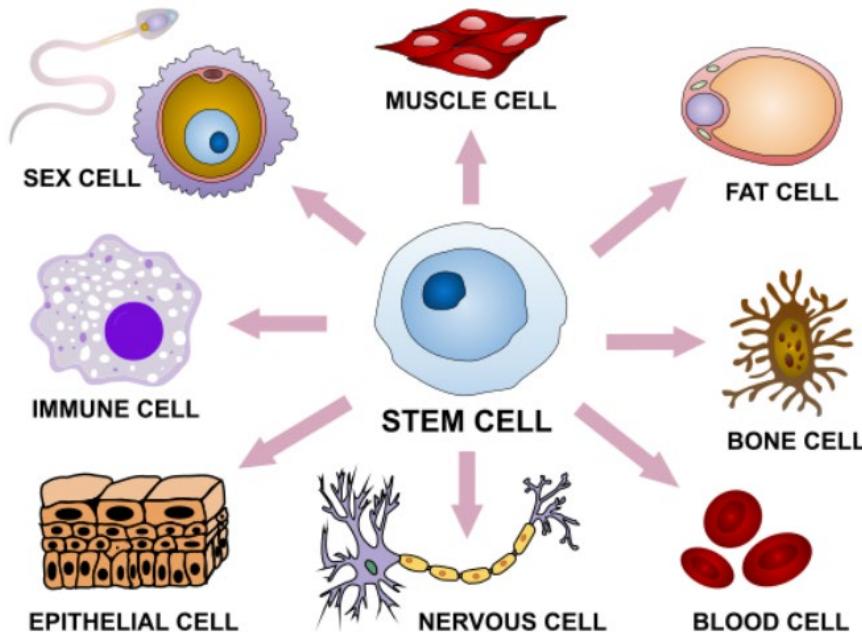
Overview

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



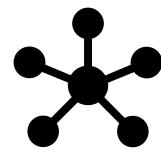


Cell annotation





Cell annotation pipeline



1

Clustering

2

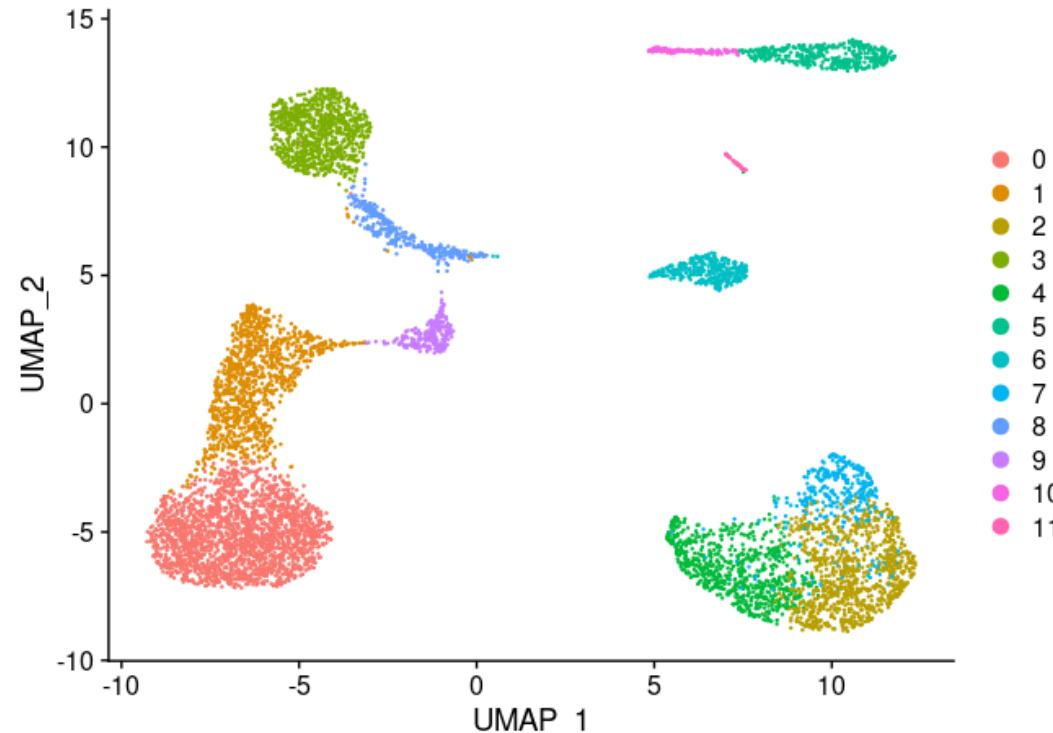
Marker genes

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 Eigenvector-based metrics

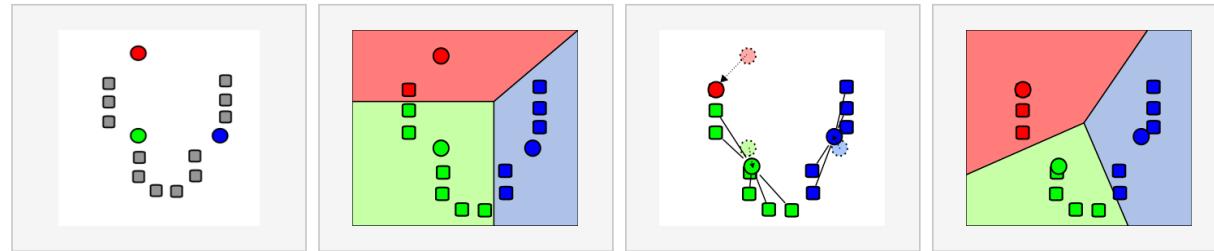
- | | |
|------------|--------------|
| 1. scLCA | 9. SIMLR |
| 2. CIDR | 10. Spectrum |
| 3. SHARP | 11. SC3 |
| 4. RacelD | |
| 5. SINCERA | |

Community detection Stability metric

- | | |
|-------------|--------------------|
| 6. ACTIONet | 12. densityCut |
| 7. Monocle3 | 13. scCCESS-Kmeans |
| 8. Seurat | 14. scCCESS-SIMLR |



Unsupervised clustering



New Space Learn Guide What's New Primers Cheat Sheets

Source Visual Outline

278 ````{r cluster, fig.height=5, fig.width=7}
279 pbmc <- FindNeighbors(pbmc, dims = 1:10)
280 pbmc <- FindClusters(pbmc, resolution = 0.5)
281
282 # Look at cluster IDs of the first 5 cells
283 head(Ids(pbmc), 5)
284 ````

Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck

Number of nodes: 7499
Number of edges: 257264

Running Louvain algorithm...
0% 10 20 30 40 50 60 70 80 90 100%

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”.



Visualising cells

posit Cloud

Spaces
Your Workspace
single cell workshop Kirby Institute, UNSW, Sydney
New Space

Learn
Guide
What's New
Primers
Cheat Sheets

Single Cell Workshop / single_cell_pipeline Sara Ballouz

File Edit Code View Plots Session Build Debug Profile Tools Help

pbmc_tutorial.Rmd x Go to file/function Addins

Source Visual

285
286
287 ## Run non-linear dimensional reduction (UMAP/tSNE)
288
289 Seurat offers several non-linear dimensional reduction techniques,
such as tSNE and UMAP, to visualize and explore these datasets. The
goal of these algorithms is to learn the underlying manifold of the
data in order to place similar cells together in low-dimensional
space. Cells within the graph-based clusters determined above should
co-localize on these dimension reduction plots. As input to the UMAP
and tSNE, we suggest using the same PCs as input to the clustering
analysis.
290
291 ```{r umap, fig.height=5, fig.width=7}
292 pbmc <- RunUMAP(pbmc, dims = 1:10)
293 ````

2

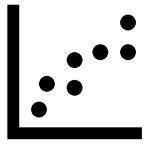
```{r umapplot, fig.height=5, fig.width=7}  
# note that you can set `label = TRUE` or use the  
LabelClusters function to help label individual clusters  
DimPlot(pbmc, reduction = 'umap')  
```

Uniform Manifold Approximation and Projection (UMAP)

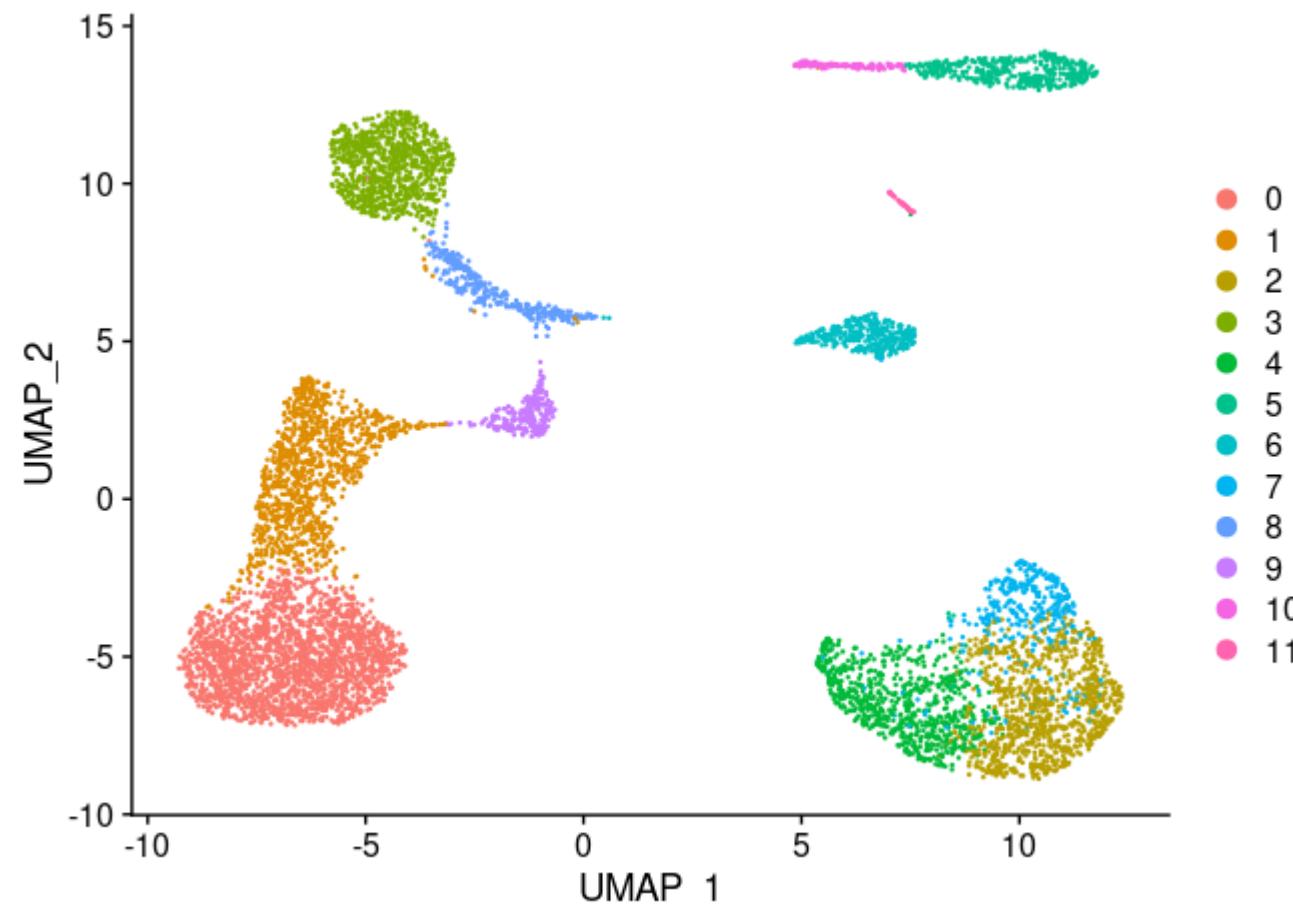
To visualise these clusters, we run another dimensional reduction algorithm. This time, we will use **UMAP**.

And then plot with the DimPlot function.

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

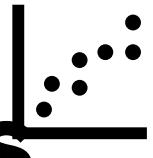


Visualisation: UMAP

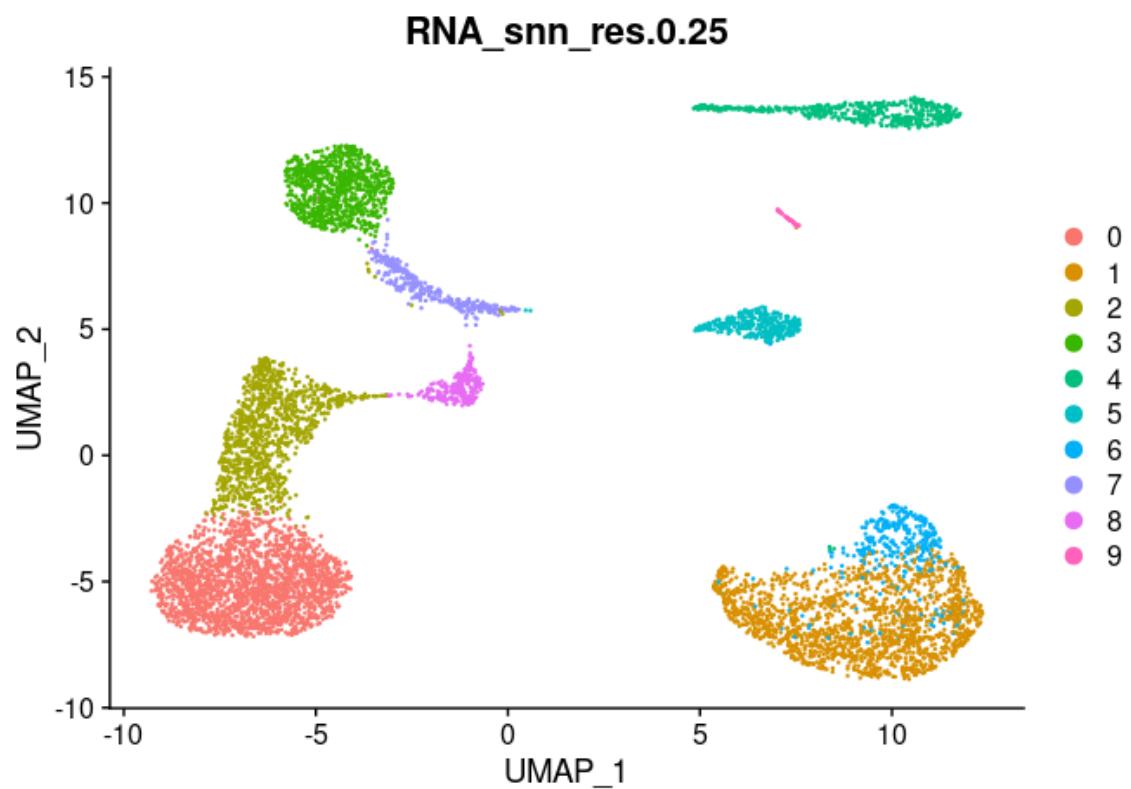


Note the number of clusters

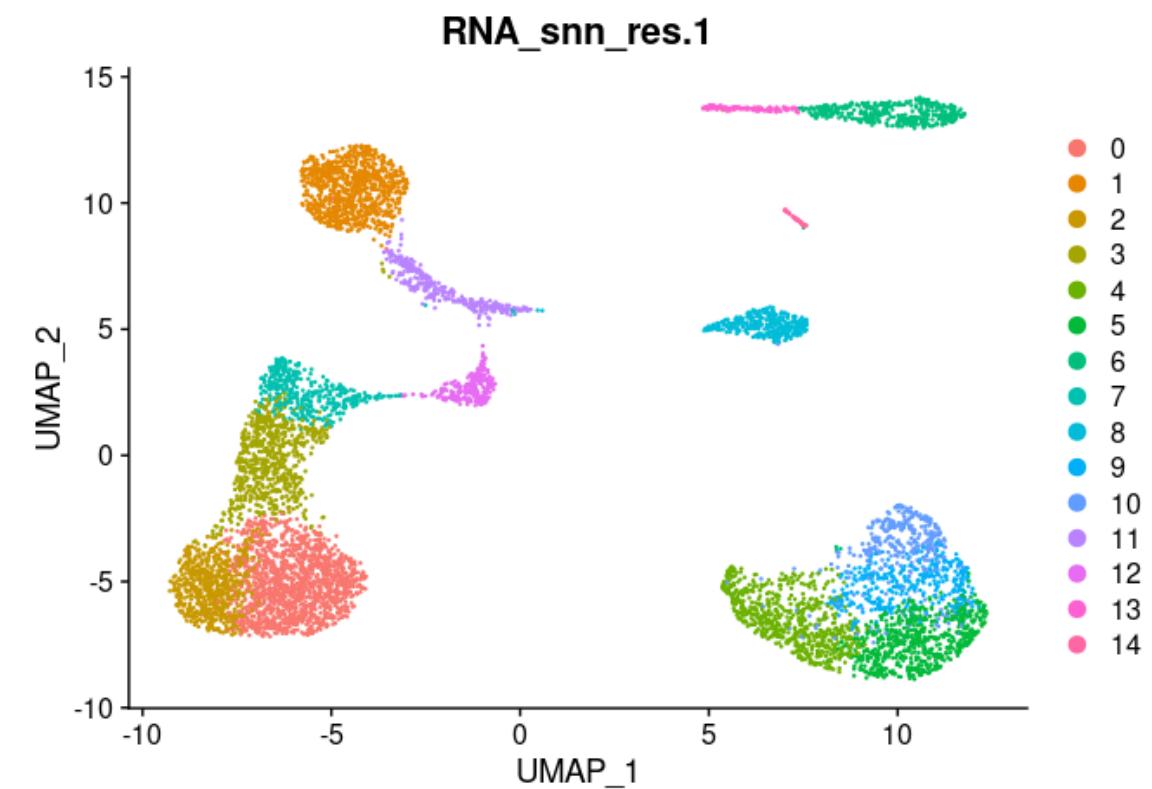
Unsupervised clustering: resolutions



Low resolution



High resolution



Note the number of clusters



UNSW
SYDNEY



Visualisation: tSNE

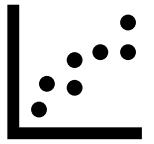
The screenshot shows the RStudio interface within the posit Cloud environment. The left sidebar lists 'Spaces' (Your Workspace, single cell workshop), 'Learn' (Guide, What's New, Primers, Cheat Sheets), and a workspace titled 'Single Cell Workshop / single_cell_pipeline' by Sara Ballouz. The main area displays an R script named 'pbmc_tutorial.Rmd' with the following content:

```
299
300 `r tsne, fig.height=5, fig.width=7}
301 pbmc <- RunTSNE(pbmc, dims = 1:10)
302 `r
303
304 `r tsneplot, fig.height=5, fig.width=7}
305 DimPlot(pbmc, reduction = 'tsne')
306 `r
```

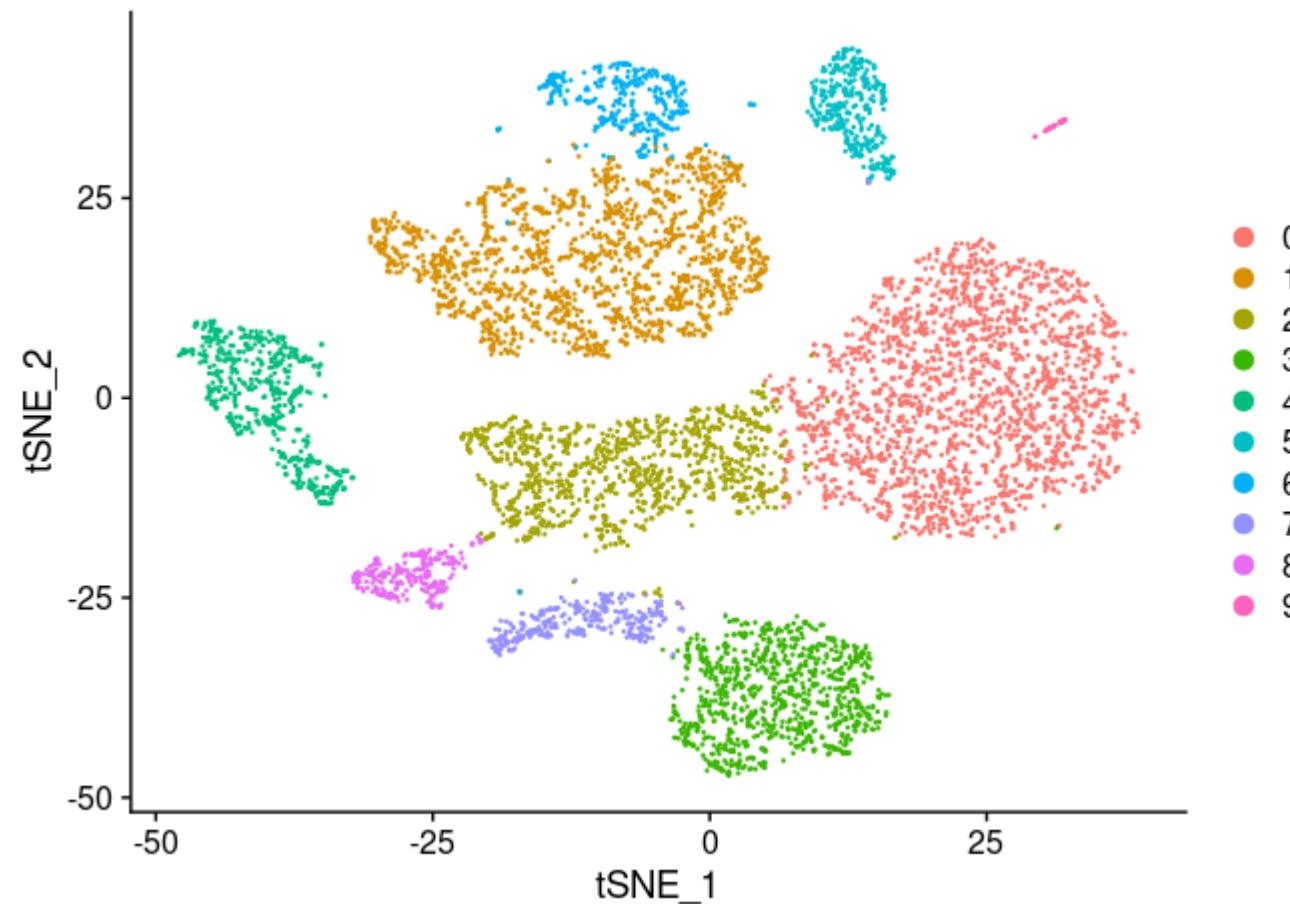
A purple circle with the number '3' is overlaid on the 'Run' button in the toolbar. Below the code, there is a UMAP plot showing two distinct clusters of points (red and yellow) on a coordinate system with axes labeled 'UMAP_1' and 'UMAP_2'. The plot area includes a legend with 'ABC' and a magnifying glass icon.

*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.

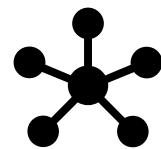


Visualisation: tSNE





Cell annotation pipeline



1

Clustering



2

Marker genes



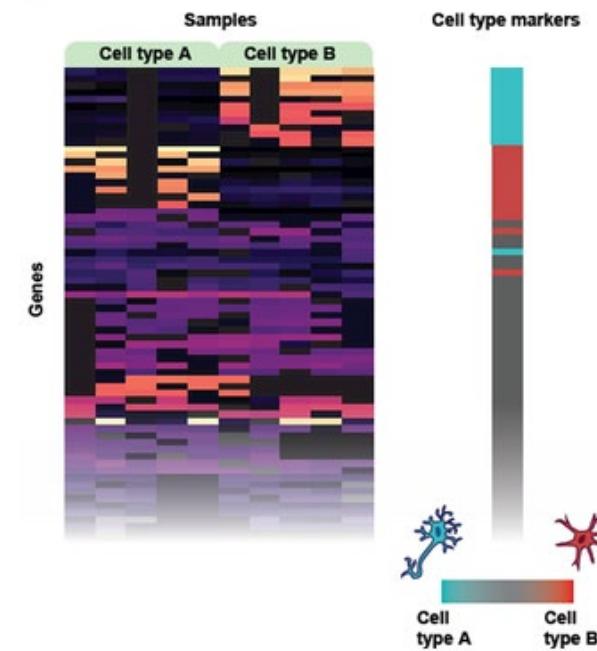
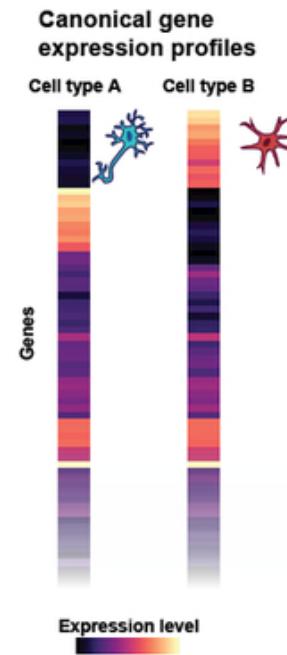
3

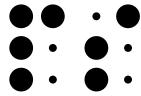
Visualisation



Marker genes: differential expression

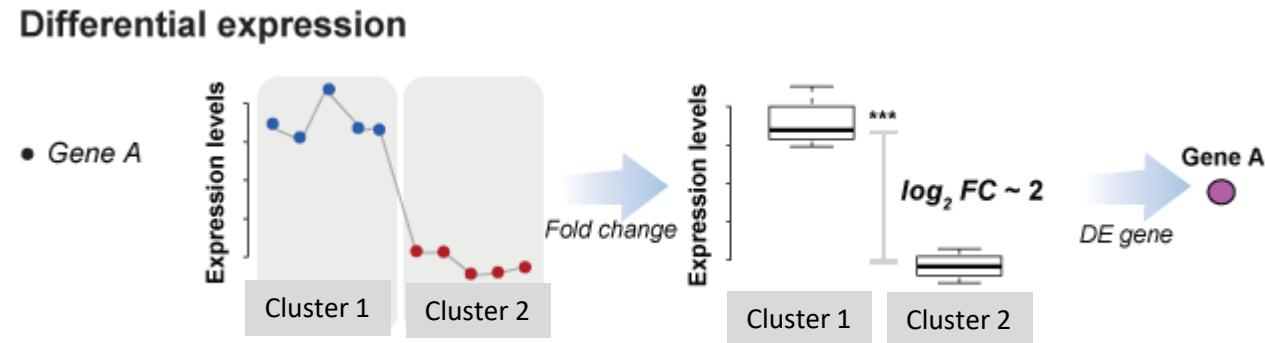
- To find marker genes, we need to find the genes which are highly expressed in one cluster but not in others





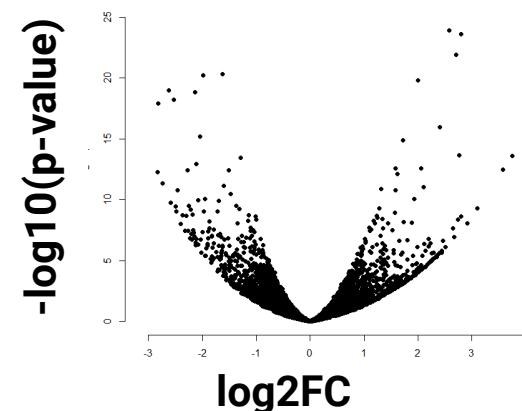
Differential expression: overview

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

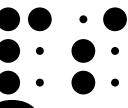


$$\log_{2}FC = \log_2\left(\frac{\text{Average expression cluster 1}}{\text{Average expression cluster 2}}\right)$$

Volcano plot



Differential gene expression analysis



itCloud ×

Single Cell Workshop / single_cell_pipeline Sara Ballouz

File Edit Code View Plots Session Build Debug Profile Tools Help

pbmc_tutorial.Rmd x

Go to file/function Addins

Source Visual

tSNE_1

```
307 ~`{r markers1}
308 cluster5.markers <- FindMarkers(pbmc, ident.1 = 5, min.pct = 0.25)
309 head(cluster5.markers, n = 5)
310 ~`{r markers2}
311 pbmc.markers <- FindAllMarkers(pbmc, only.pos = TRUE, min.pct = 0.25,
312 logfc.threshold = 0.25)
313 pbmc.markers %>% group_by(cluster) %>% slice_max(n = 2, order_by =
314 avg_log2FC)
```

4

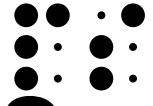
The screenshot shows the RStudio interface with a project titled "Single Cell Workshop / single_cell_pipeline" by "Sara Ballouz". The code editor displays R code for finding markers. Line 312 is highlighted with a green vertical bar. A purple circle containing the number 4 is positioned over the code area.

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



The screenshot shows an RStudio interface with three tabs: Console, Terminal (selected), and Background Jobs. The Terminal tab displays R code and its output. The output shows the first few rows of a data frame named 'pbmc.markers'. A purple circle with the number '5' is overlaid on the terminal window.

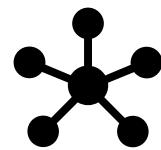
```
R 4.3.1 . /cloud/project/
TMEM123      TMEM123
RPL37A       RPL37A
EIF3E        EIF3E
PABPC1       PABPC1
RPL12        RPL12
[ reached 'max' / getOption("max.print") -- omitted 5368 rows ]
> head(pbmc.markers)
  p_val avg_log2FC pct.1 pct.2 p_val_adj cluster  gene
TCF7    0  1.4011959  0.975  0.430      0     0 TCF7
LEF1    0  1.3606533  0.922  0.337      0     0 LEF1
CCR7    0  1.2487564  0.911  0.377      0     0 CCR7
CD3E    0  0.9917280  0.991  0.482      0     0 CD3E
LDHB    0  0.9811911  0.989  0.671      0     0 LDHB
SARAF   0  0.9422573  0.995  0.814      0     0 SARAF
```

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



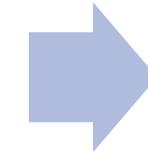
1

Clustering



2

Marker genes

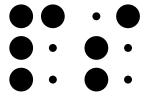


3

Visualisation

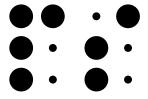


UNSW
SYDNEY

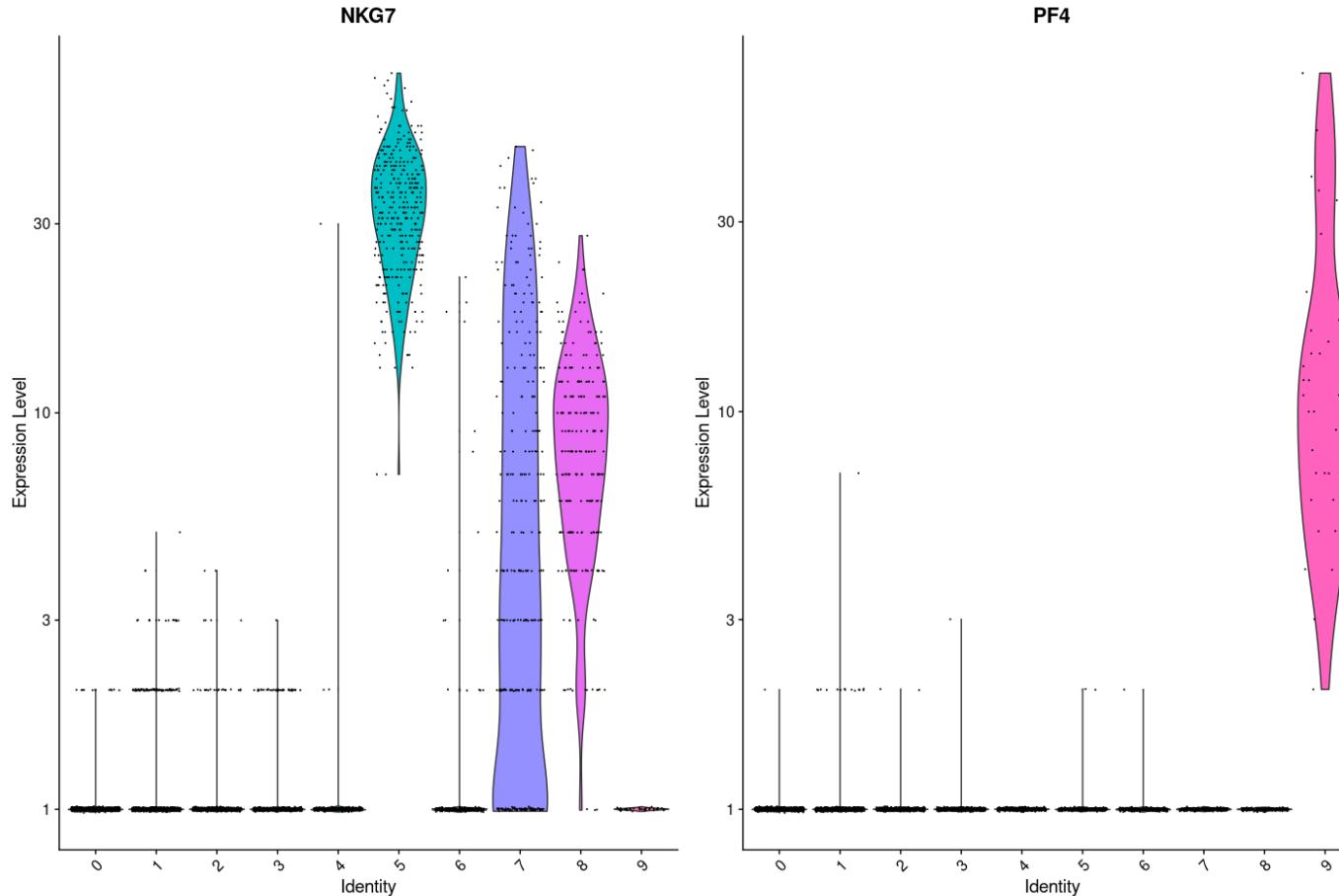


Visualisation

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

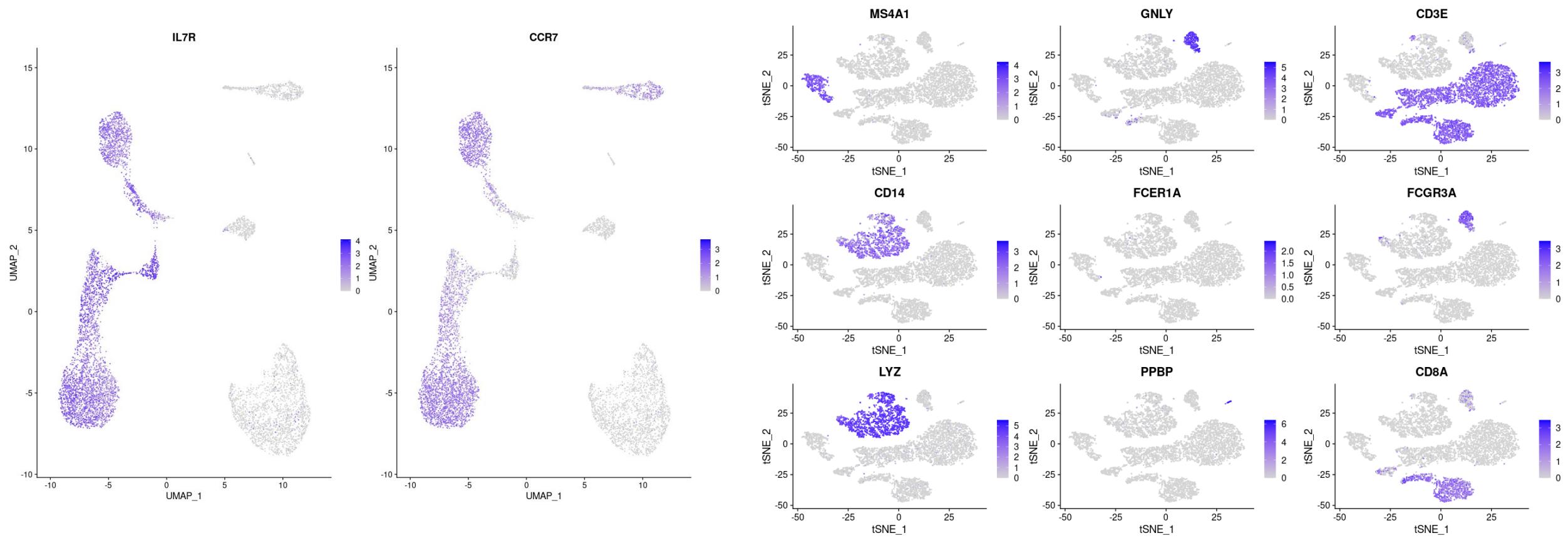
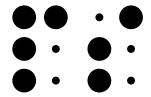


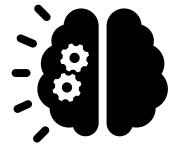
Visualisation: Violin plots



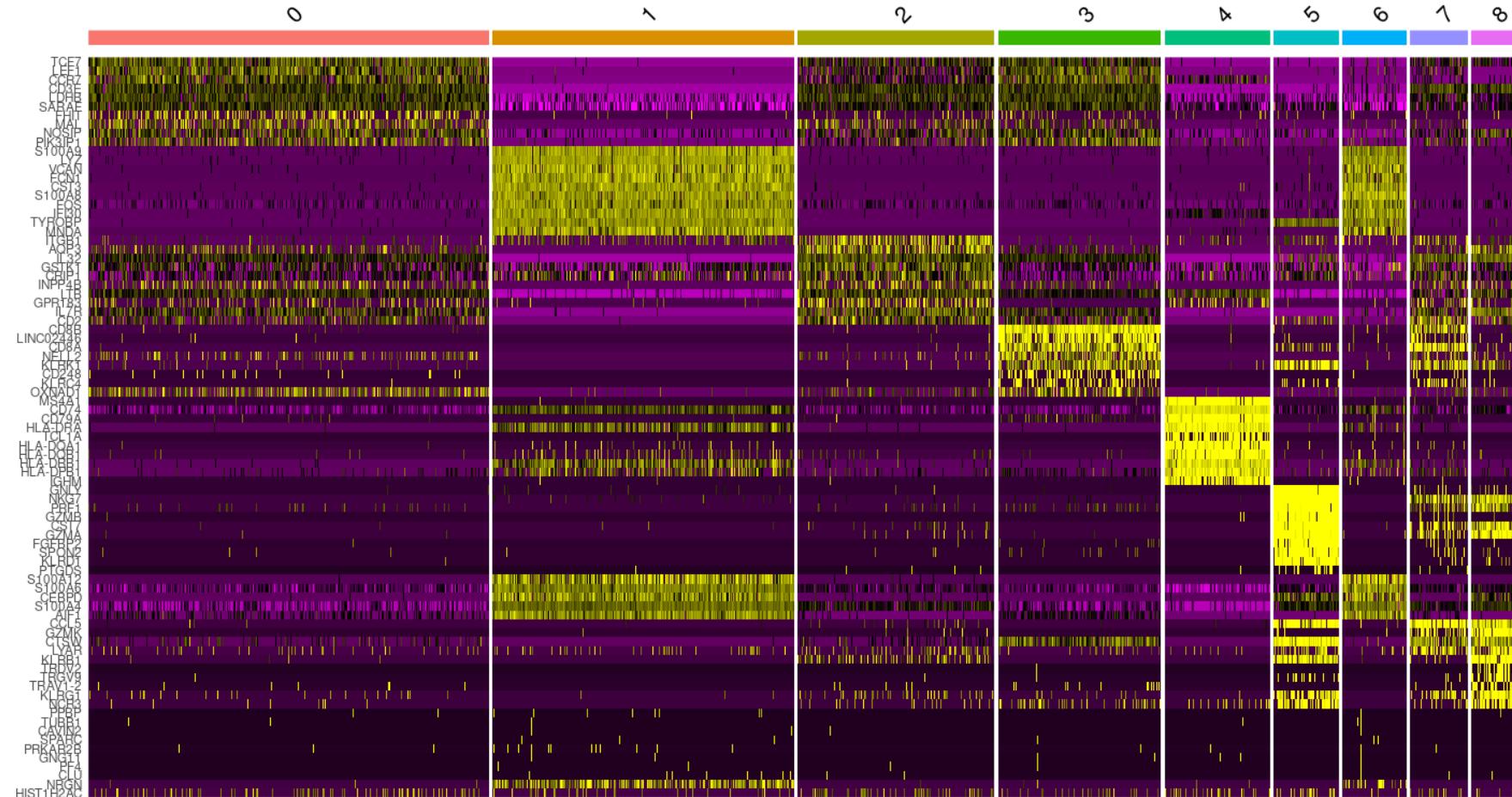
- Each dot is a cell
- X-axis shows clusters
- Y-axis shows gene expression level

Visualisation: UMAP/tSNE





Visualisation: Heatmaps



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

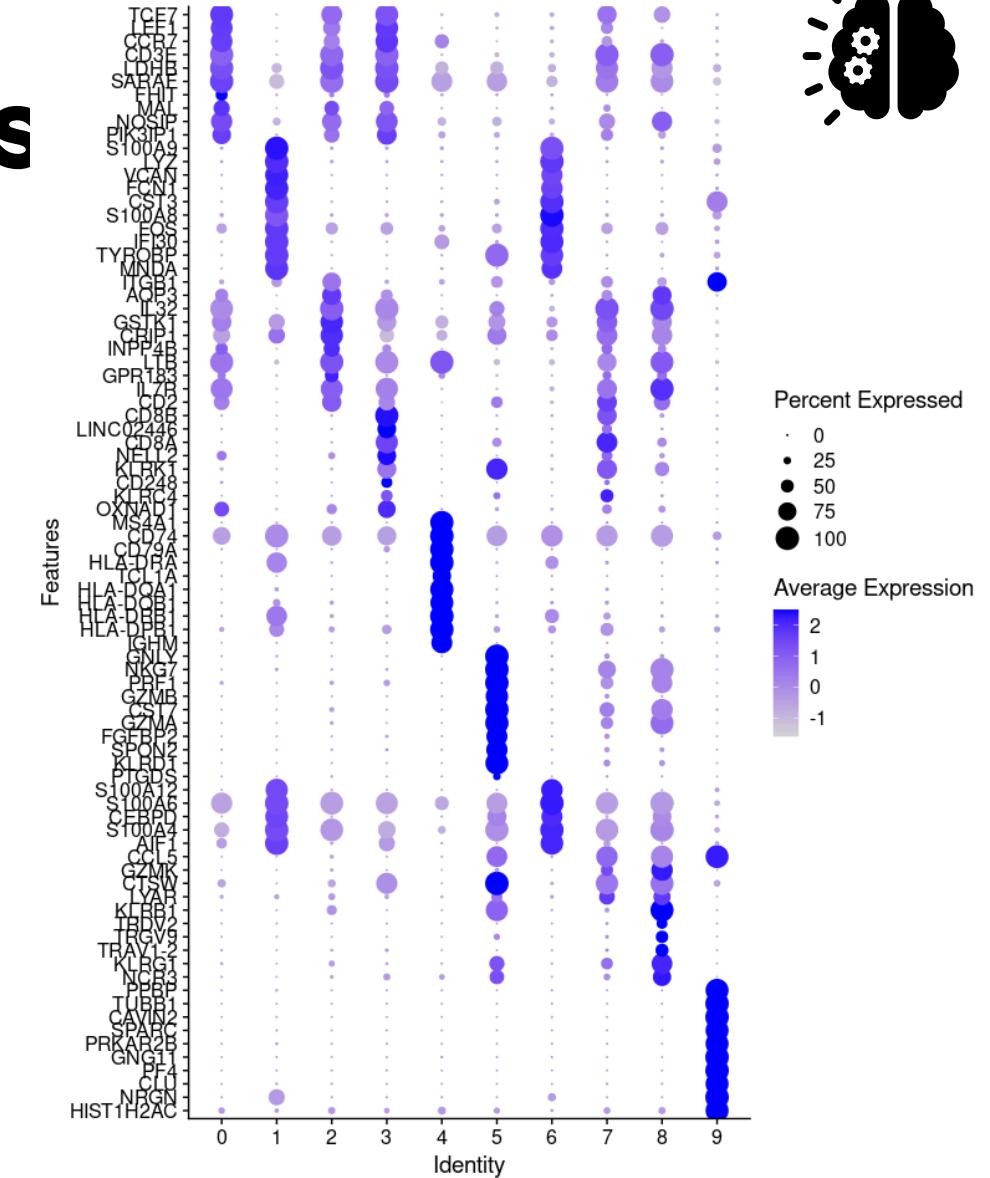


UNSW
SYDNEY

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

Learn

Guide

What's New

Primers

Community

New Space

Learn

New Space

```
315  ````{r markerplots, fig.height=10, fig.width=15}
316  VlnPlot(pbmc, features = c("MS4A1", "CD79A"))
317  # you can plot raw counts as well
318  VlnPlot(pbmc, features = c("NKG7", "PF4"), slot = 'counts', log =
319  TRUE)
320  FeaturePlot(pbmc, features = c("MS4A1", "GNLY", "CD3E", "CD14",
321  "FCER1A", "FCGR3A", "LYZ", "PPBP", "CD8A"), reduction = "tsne")
322  FeaturePlot(pbmc, features = c("IL7R", "CCR7"), reduction = "umap")
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)
327  -> top10
328  DoHeatmap(pbmc, features = top10$gene) + NoLegend()
329  ````
```

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.

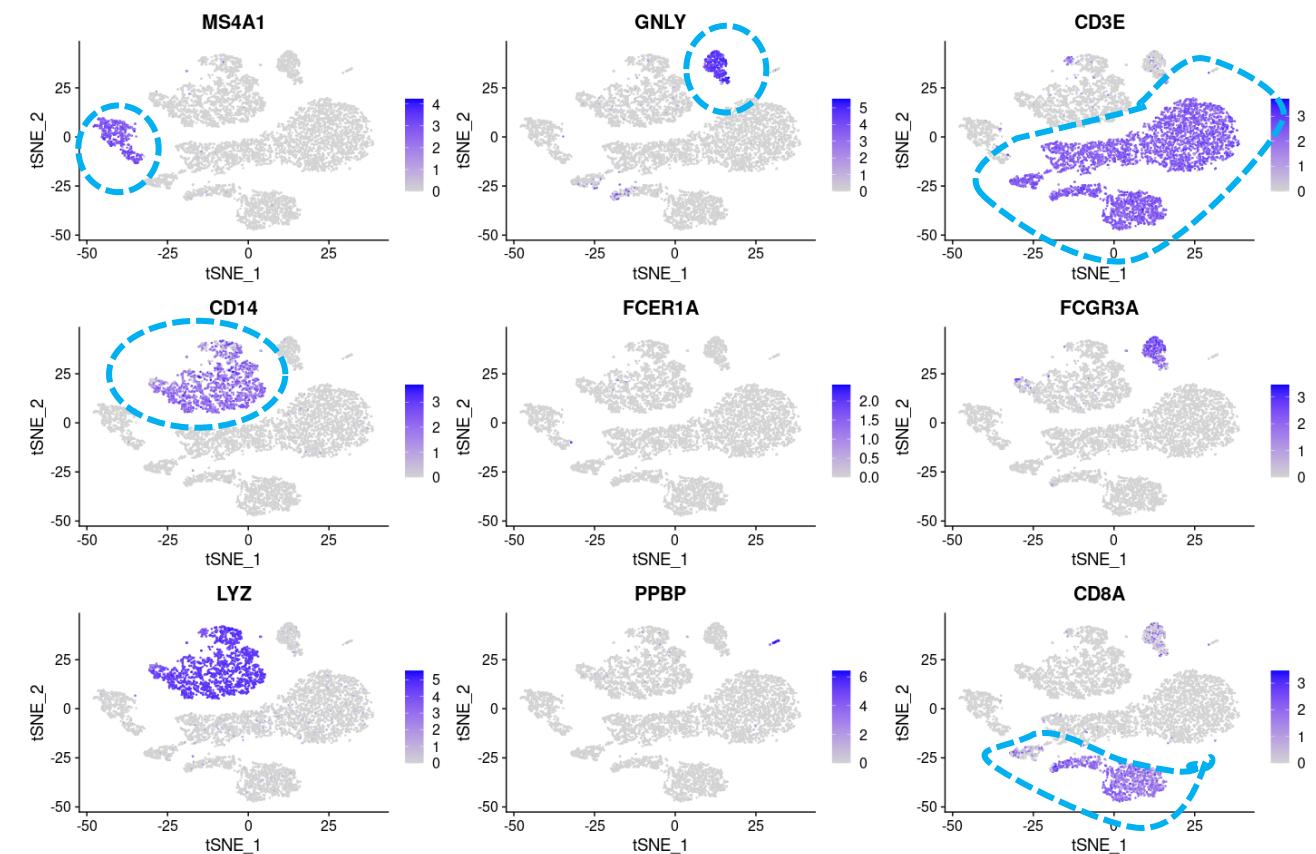
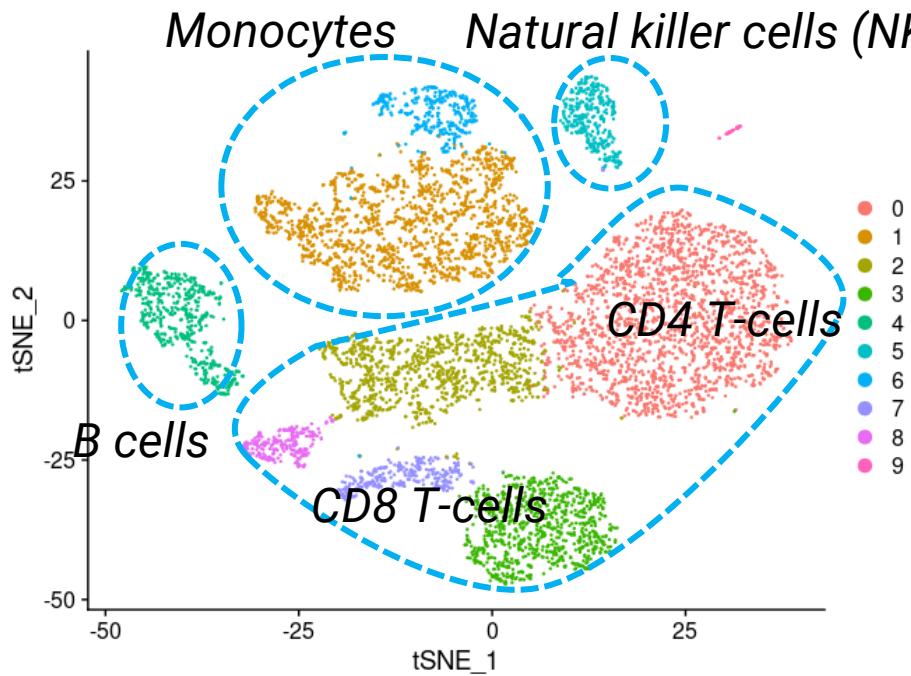


UNSW
SYDNEY



Cell annotation – manual

- Use known marker genes to identify clusters.



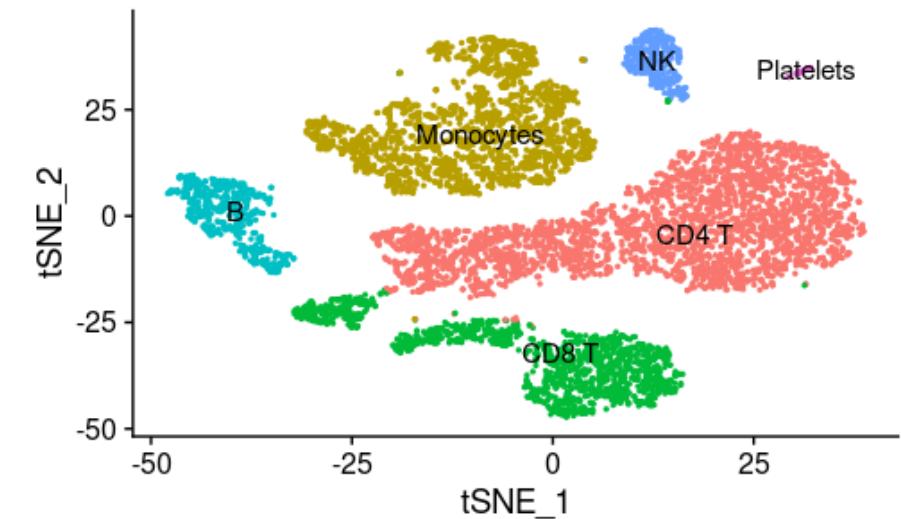
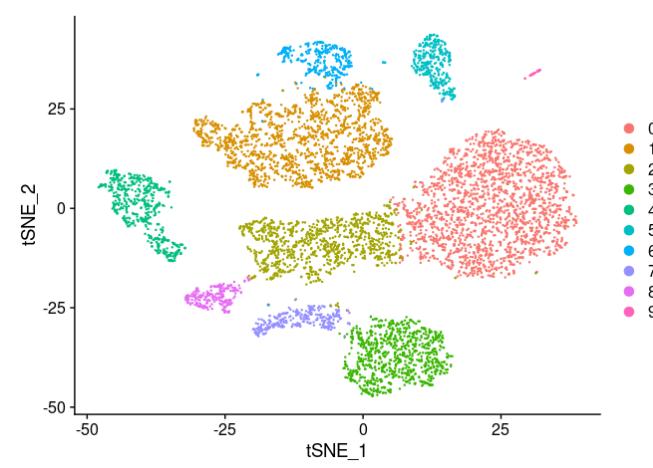


Cell annotation – manual

```
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 <- RenameIdentents(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.

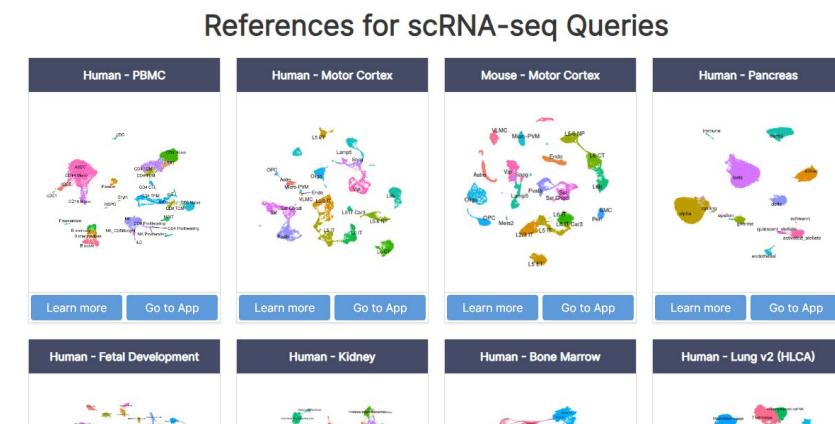
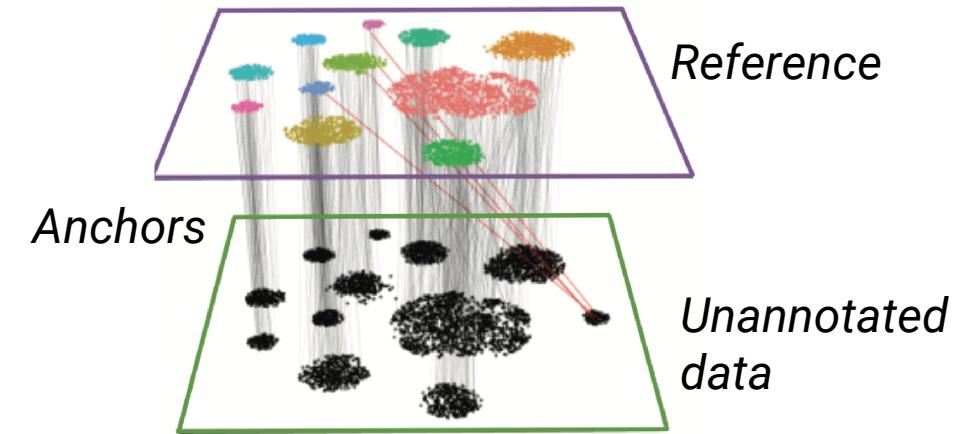


UNSW
SYDNEY



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!

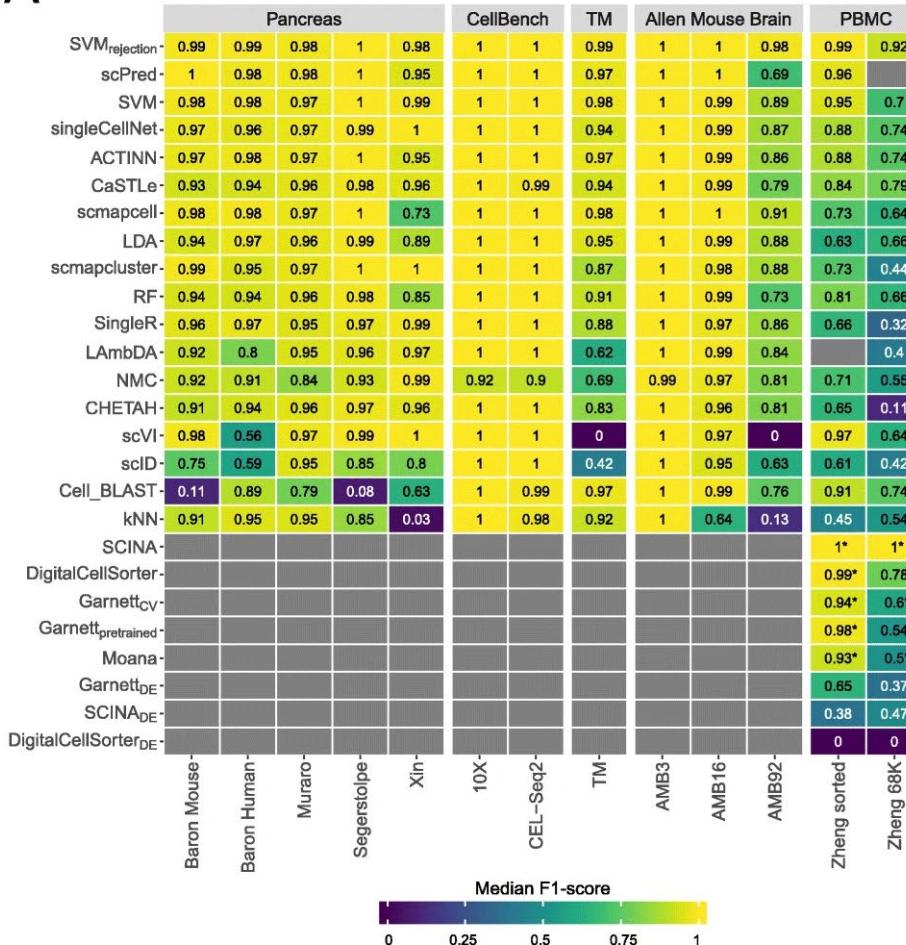


<https://azimuth.hubmapconsortium.org/>

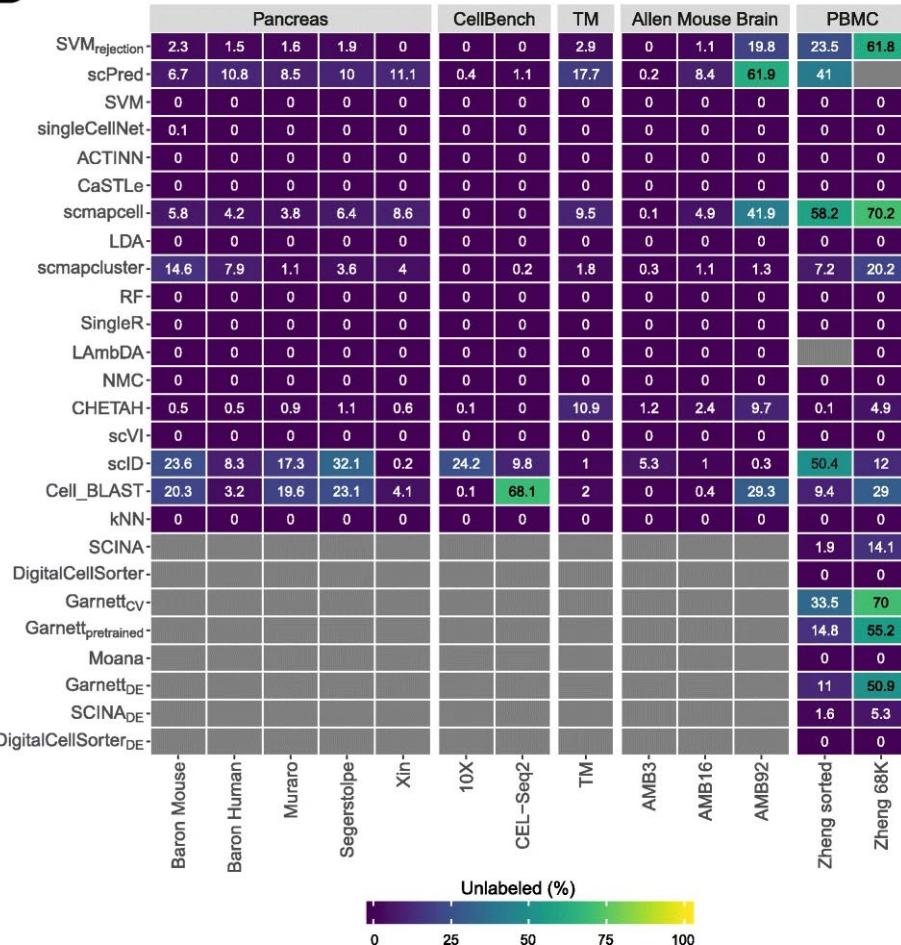


Annotation tools

A



B





Cell annotation – Azimuth demo

The screenshot shows the posit Cloud interface. On the left, there's a sidebar with 'Spaces' (Your Workspace, single cell workshop), 'Learn' (Guide, What's New), 'Primers', and 'Cheat Sheets'. The main area is a code editor for 'pbmc_tutorial.Rmd*'. The code is as follows:

```
362
363  ```{r cell_annotation - azimuth online}
364  imputed.assay <- readRDS('data/pbmcs10k/azimuth/azimuth_impADT.Rds')
365  pbmc <- pbmc[, Cells(imputed.assay)]
366  pbmc[['impADT']] <- imputed.assay
367
368
369  projected.umap <- readRDS('data/pbmcs10k/azimuth/azimuth_umap.Rds')
370  pbmc <- pbmc[, Cells(projected.umap)]
371  pbmc[['umap.proj']] <- projected.umap
372
373  predictions <- read.delim('data/pbmcs10k/azimuth/azimuth_pred.tsv',
374  row.names = 1)
374  pbmc <- AddMetaData(
375    object = pbmc,
376    metadata = predictions)
377
378  DimPlot(pbmc, reduction = 'tsne', group.by="predicted.celltype.I2")
379
380  ````
```

A purple circle with the number '8' is positioned over the line 'DimPlot(pbmc, reduction = 'tsne', group.by="predicted.celltype.I2")'.

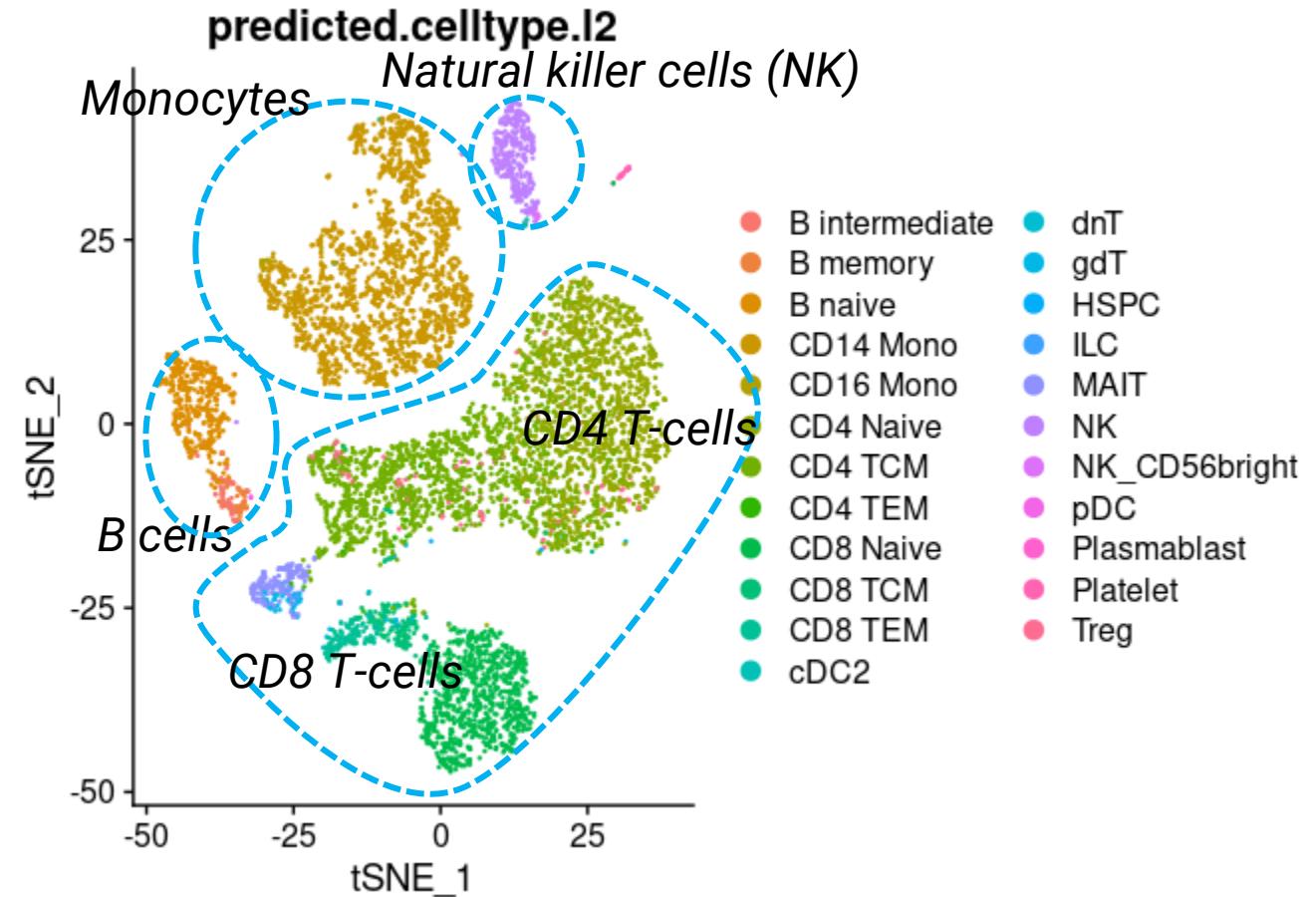
Upload and store the predictions, then plot "predicted.celltype.I2"





Manual versus automatic

Pretty good match!
More granular resolution
can be selected with “I3”
And broader cell-types with
“I1”



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

single_cell_integration_pipeline

ASSIGNMENT

RStudio Project

Sara Ballouz

Space members

Created Aug 29, 2023 4:00 PM

Derived from: single_cell_pip

single_cell_pipeline

ASSIGNMENT

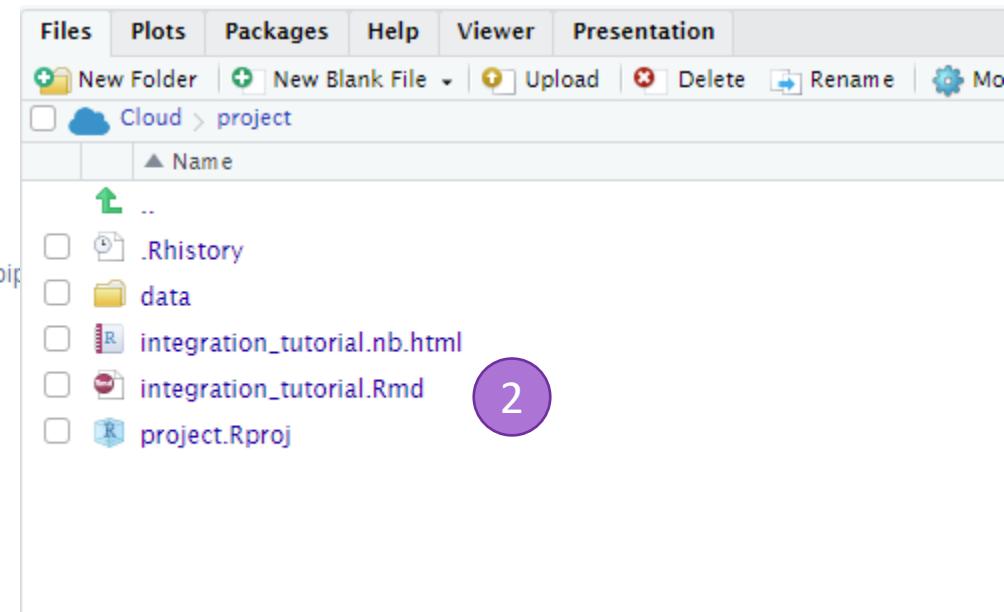
RStudio Project

Sara Ballouz

Space members

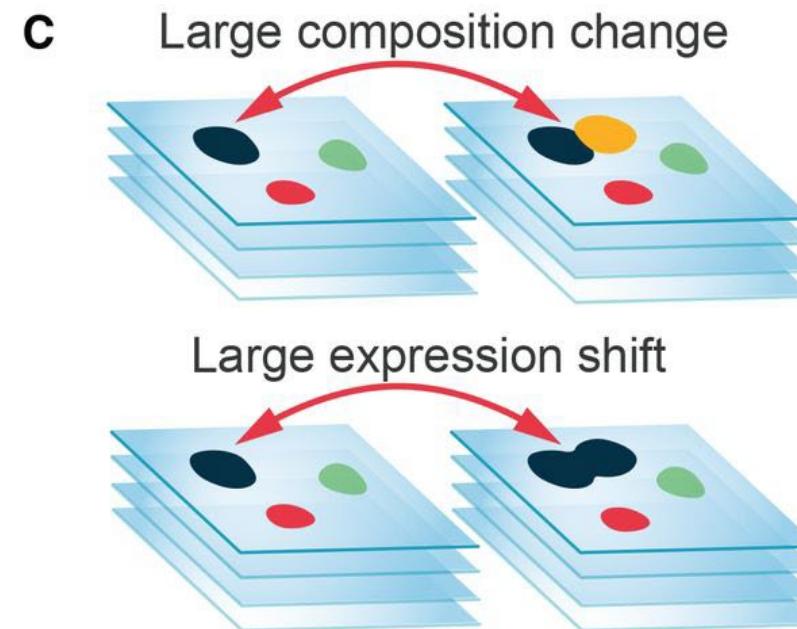
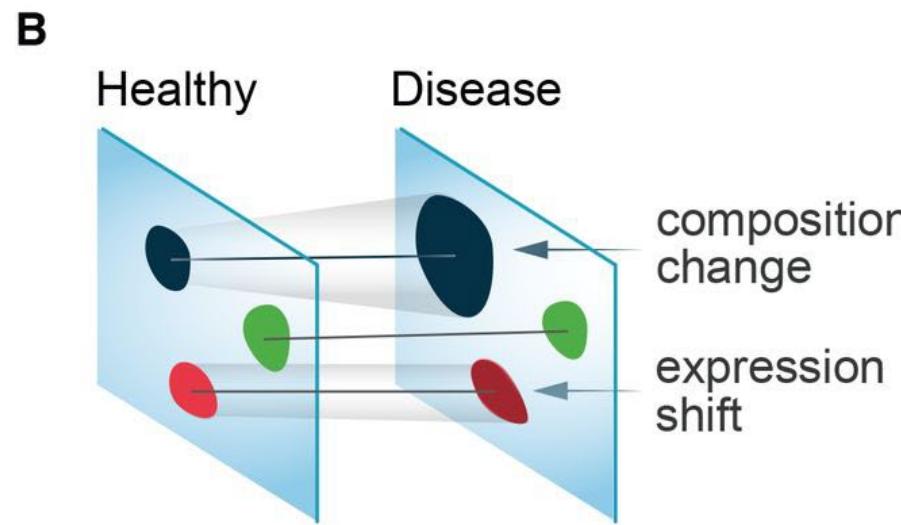
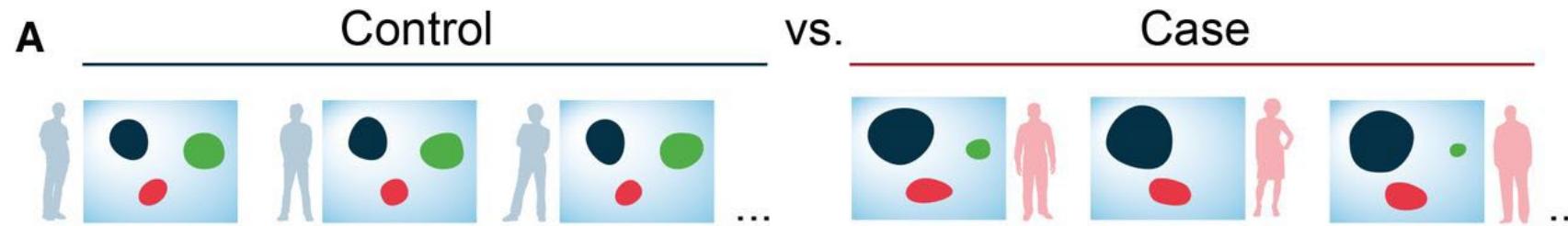
Created Aug 17, 2023 2:37 PM

1 derived project

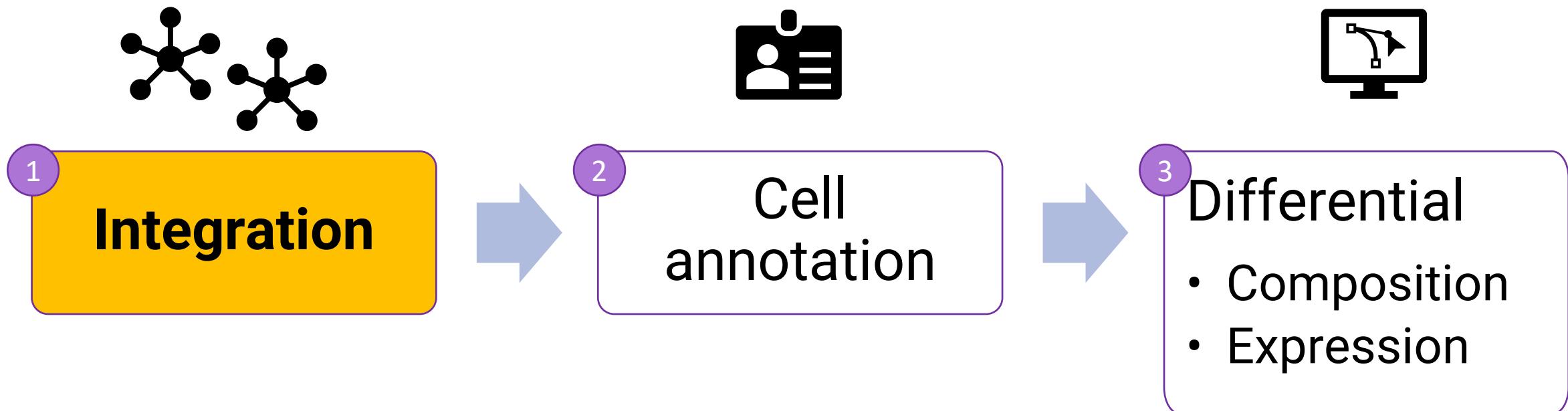


UNSW
SYDNEY

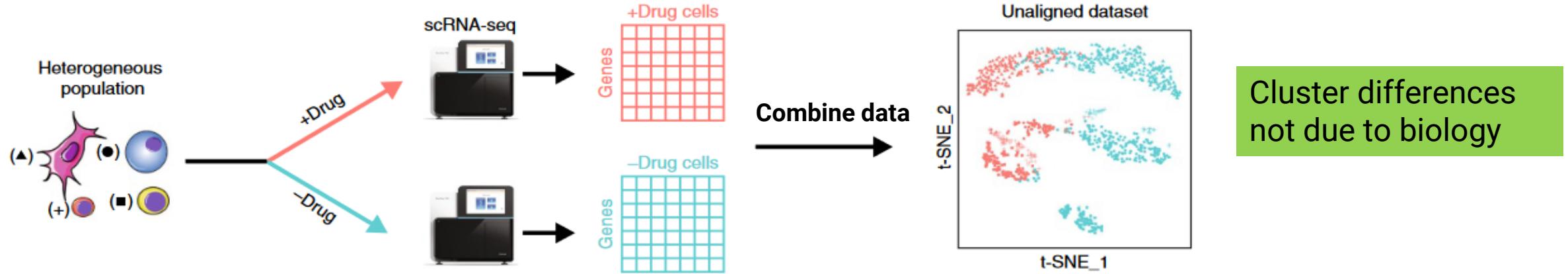
Analysis case versus control



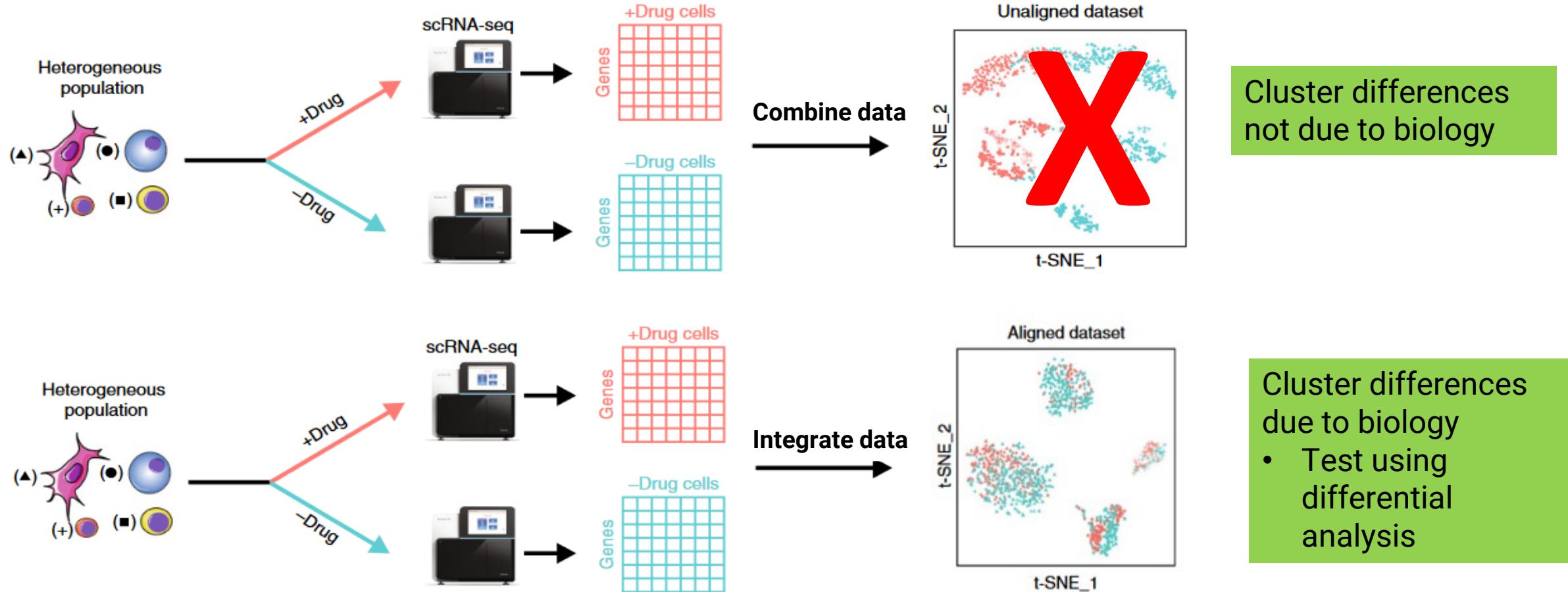
Case versus control pipeline

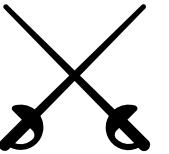


Two different samples may have batch effects

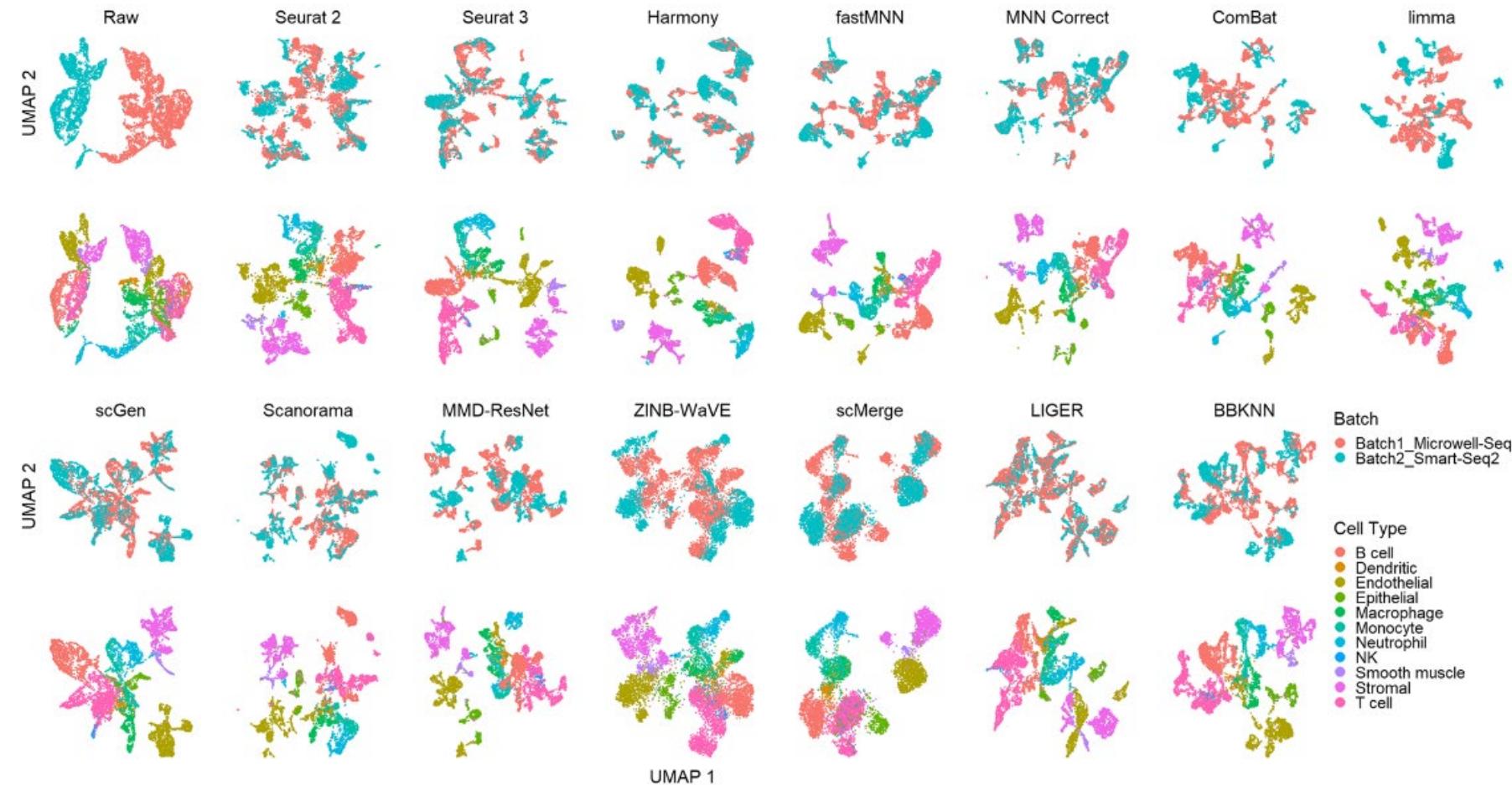


Two different samples may have batch effects





Many batch integration methods





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

```
## Setup the Seurat Objects
``{r init}
# Load the patient datasets
pbmc.data <- Read10X(data.dir = "data/patient6/baseline/")
pbmc_base_p6 <- CreateSeuratObject(counts = pbmc.data, project = "baseline-p6", min.cells = 3, min.features = 200)
pbmc.data <- Read10X(data.dir = "data/patient6/D7/")
pbmc_d7_p6 <- CreateSeuratObject(counts = pbmc.data, project = "D7-p6", min.cells = 3, min.features = 200)
pbmc.data <- Read10X(data.dir = "data/patient8/baseline/")
pbmc_base_p8 <- CreateSeuratObject(counts = pbmc.data, project = "baseline-p8", min.cells = 3, min.features = 200)
pbmc.data <- Read10X(data.dir = "data/patient8/D7/")
pbmc_d7_p8 <- CreateSeuratObject(counts = pbmc.data, project = "D7-p8", min.cells = 3, min.features = 200)
pbmc.data <- Read10X(data.dir = "data/patient9/baseline/")
pbmc_base_p9 <- CreateSeuratObject(counts = pbmc.data, project = "baseline-p9", min.cells = 3, min.features = 200)
pbmc.data <- Read10X(data.dir = "data/patient9/D7/")
pbmc_d7_p9 <- CreateSeuratObject(counts = pbmc.data, project = "D7-p9", min.cells = 3, min.features = 200)
```

Chunk 7: prepare for integration ±

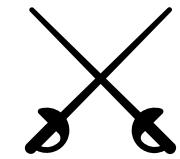
R Markdown ±

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

```
82  
83 ```{r mito, fig.height=7, fig.width=13}  
84 pbmc.list = list(pbmc_base_p6, pbmc_d7_p6, pbmc_base_p8, pbmc_d7_p8,  
pbmc_base_p9, pbmc_d7_p9)  
85 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.

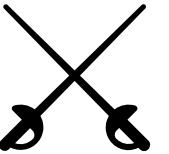


We “loop” through the list to perform all the regular steps – this is the MT-ratio calculation.

```
86 for( i in 1:length(pbmc.list)){  
87   pbmc.list[[i]][["percent.mt"]] <- PercentageFeatureSet(pbmc.list[[i]],  
pattern = "^\$MT-\$")  
88 }  
89 ...  
90 ```{r }  
91 #Visualize QC metrics as a violin plot  
92 VlnPlot( pbmc.list[[1]], features = c("nFeature_RNA", "nCount_RNA",  
"percent.mt"), ncol = 3)  
93 VlnPlot( pbmc.list[["Baseline-p9"]], features = c("nFeature_RNA",  
"nCount_RNA", "percent.mt"), ncol = 3)  
94 ```
```



UNSW
SYDNEY



```
95  
96  ````{r subsetting}  
97  for( i in 1:length(pbmc.list)){ [No Title]  
98    pbmc.list[[i]] <- subset(pbmc.list[[i]], subset = nFeature_RNA > 200 &  
99      nFeature_RNA < 2500 & percent.mt < 7)  
100  }  
101  ````
```

3

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

```
102  
103  ````{r prepare for integration}  
104  pbmc.list <- lapply(X = pbmc.list, FUN = SCTransform)  
105  features <- SelectIntegrationFeatures(object.list = pbmc.list, nfeatures =  
106    3000)  
107  pbmc.list <- PrepSCTIntegration(object.list = pbmc.list, anchor.features =  
108    features)  
109  ````
```

4

5

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

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

UNSW
SYDNEY

```
108 ````{r anchors}
109 pbmc.anchors <- FindIntegrationAnchors(object.list = pbmc.list, normalization.method = "SCT",
110   anchor.features = features)
110 ````
```

6

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



```
111
112 ````{r integrate data}
113 pbmc.combined.sct <- IntegrateData(anchorset = pbmc.anchors, normalization.method = "SCT")
114 ````
```

7

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

```
116 ````{r clustering etc}
117 pbmc.combined.sct <- RunPCA(pbmc.combined.sct, verbose = FALSE)
118 pbmc.combined.sct <- RunUMAP(pbmc.combined.sct, reduction = "pca", dims = 1:30)
119 #pbmc.combined.sct <- RunTSNE(pbmc.combined.sct, reduction = "pca", dims = 1:30)
120 pbmc.combined.sct <- FindNeighbors(pbmc.combined.sct, dims = 1:10)
121 pbmc.combined.sct <- FindClusters(pbmc.combined.sct, resolution = 0.5)
121 ````
```

8

The usual steps including clustering and dimension reduction.

```
123 ````{r plots and metadata}
124 DimPlot(pbmc.combined.sct, reduction = "umap")
125 DimPlot(pbmc.combined.sct, reduction = "umap", group.by = "orig.ident")
126
127
128 pbmc.combined.sct[["patients"]] = factor(gsub("baseline-[D7-", "", pbmc.combined.sct$orig.ident ))
129 pbmc.combined.sct[["type"]] = factor(gsub("-p\\d", "", pbmc.control$orig.ident ))
130
131
132 DimPlot(object = pbmc.combined.sct, reduction = "umap", group.by = "patients")
133 DimPlot(object = pbmc.combined.sct, reduction = "umap", group.by = "type")
134 ````
```

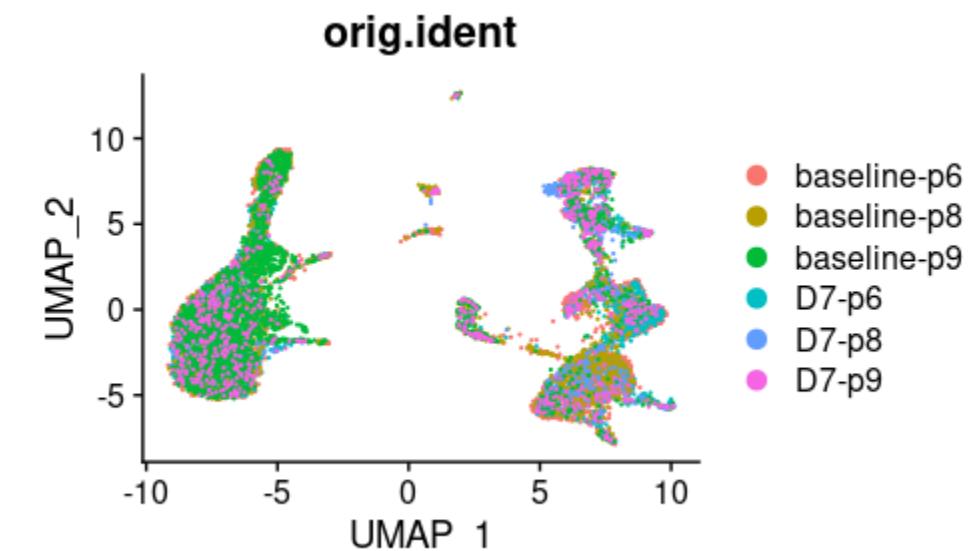
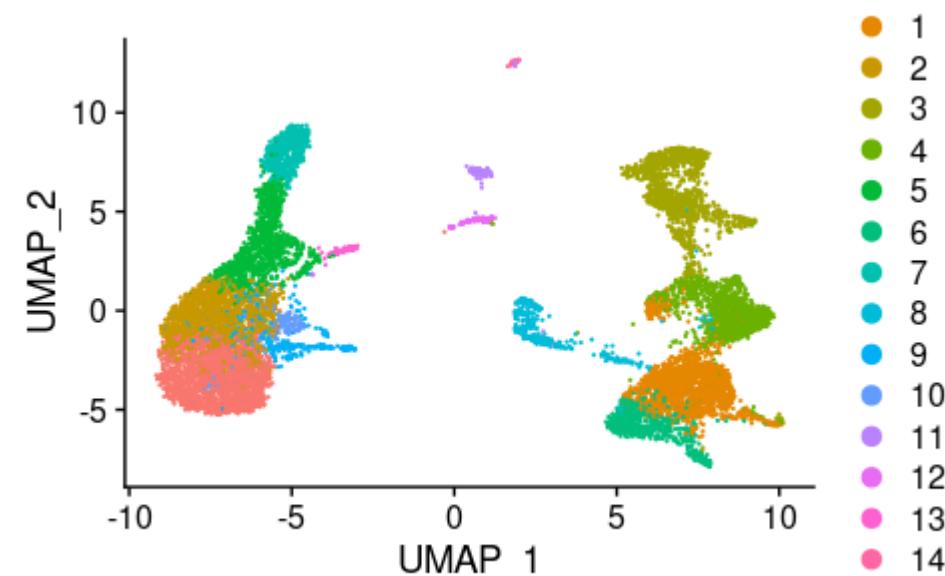
9

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

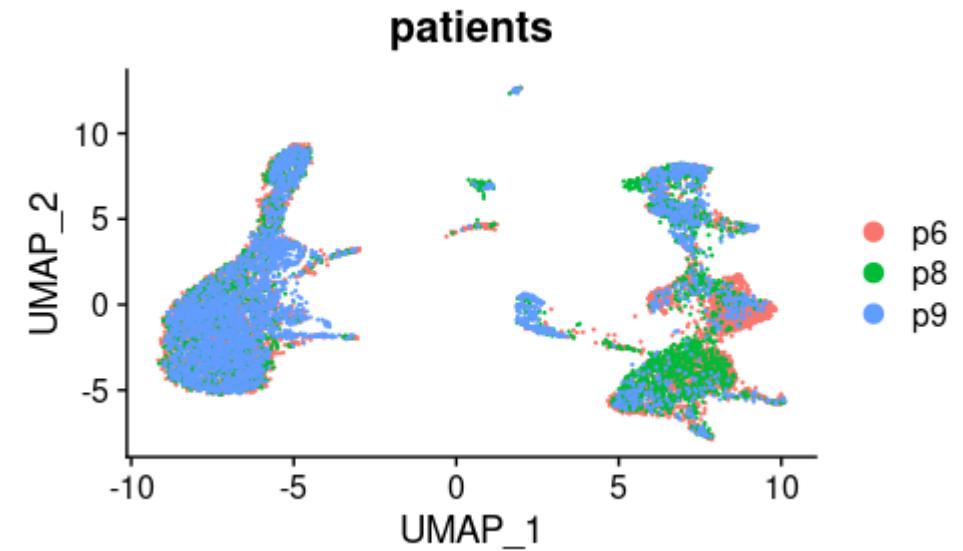
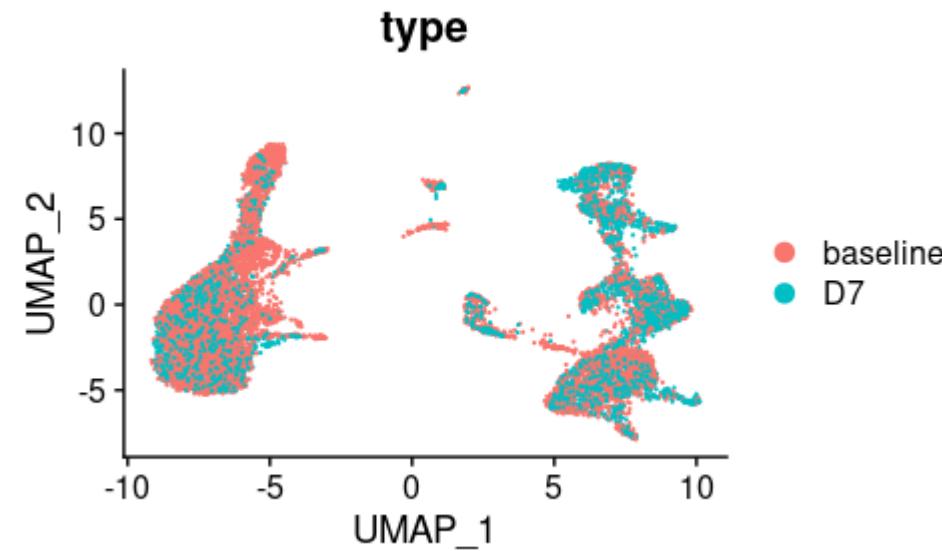
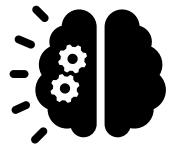


UNSW
SYDNEY

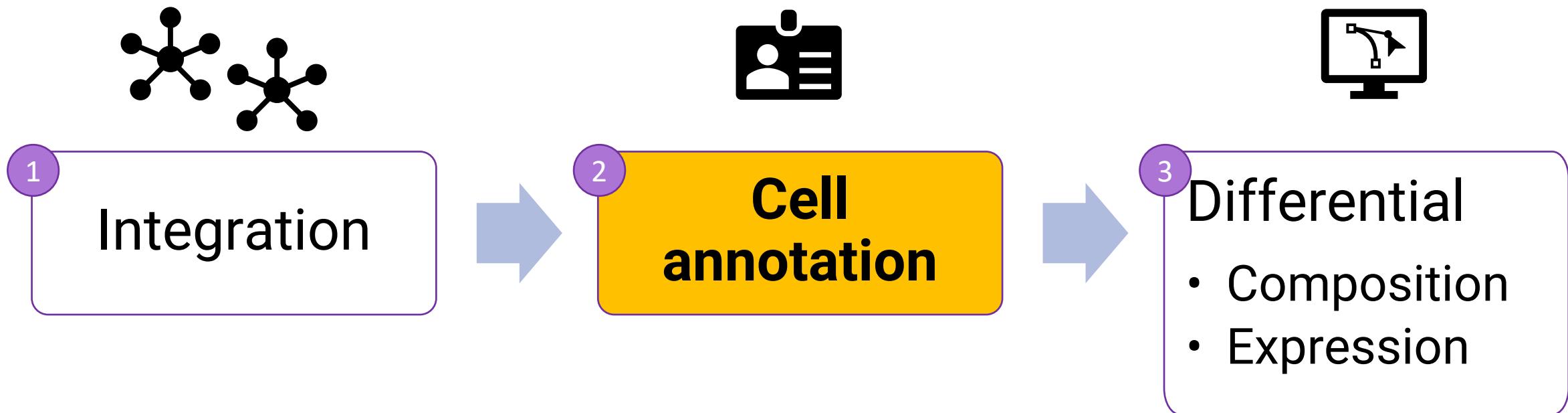
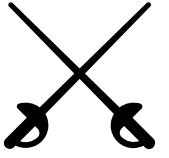
Sanity checks



Sanity checks



Case versus control pipeline



Cell annotation – automatic



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

The screenshot shows a RStudio interface with the following details:

- Title Bar:** Single Cell Workshop / single_cell_integration_pipeline
- File Menu:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help
- Toolbar:** Includes icons for New File, Open, Save, Print, Go to file/function, Addins, and Run.
- Code Editor:** The script is named "integrationTutorial.Rmd". It contains R code for cell annotation. A purple circle highlights line 10: `predictions <- read.delim('data/patient9/azimuth/azimuth_pred.tsv', row.names = 1)`.
- Code Preview:** Shows the preview pane with the R code and some output.
- Help:** A purple circle highlights the number "10" near the bottom left of the slide, likely indicating the step number.

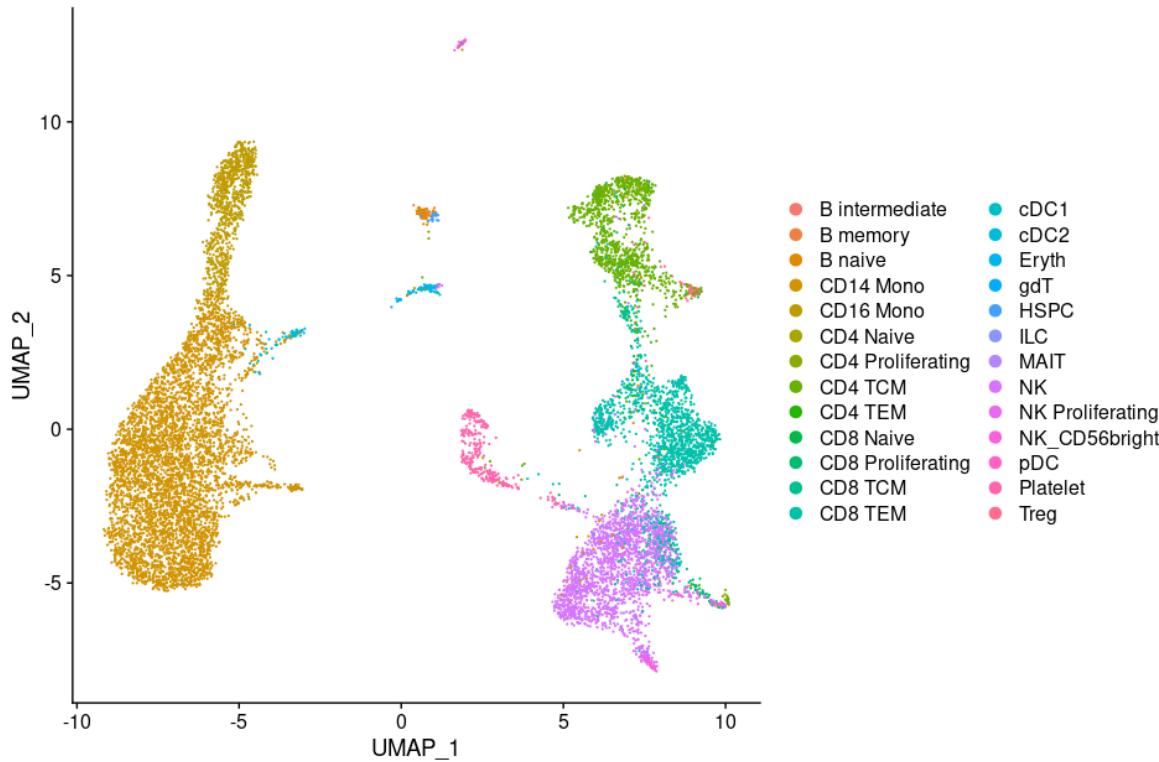
```
131
132 DimPlot(object = pbmc.combined.sct, reduction = "umap", group.by = "patients")
133 DimPlot(object = pbmc.combined.sct, reduction = "umap", group.by = "type")
134
135 ...
136
137
138 ``{r cell annotation auto}
139 #imputed.assay <- readRDS('data/patient9/azimuth/azimuth_impADT.Rds')
140 #pbmc.combined.sct <- pbmc.combined.sct[, Cells(imputed.assay)]
141 #pbmc.combined.sct[['impADT']] <- imputed.assay
142
143
144 predictions <- read.delim('data/patient9/azimuth/azimuth_pred.tsv', row.names = 1)
145 pbmc.combined.sct <- AddMetaData(object = pbmc.combined.sct, metadata = predictions)
146
147
148 #projected.umap <- readRDS('data/patient9/azimuth/azimuth_umap.Rds')
149 #pbmc.combined.sct <- pbmc.combined.sct[, Cells(projected.umap)]
150 #pbmc.combined.sct[['umap.proj']] <- projected.umap
151 ...
```



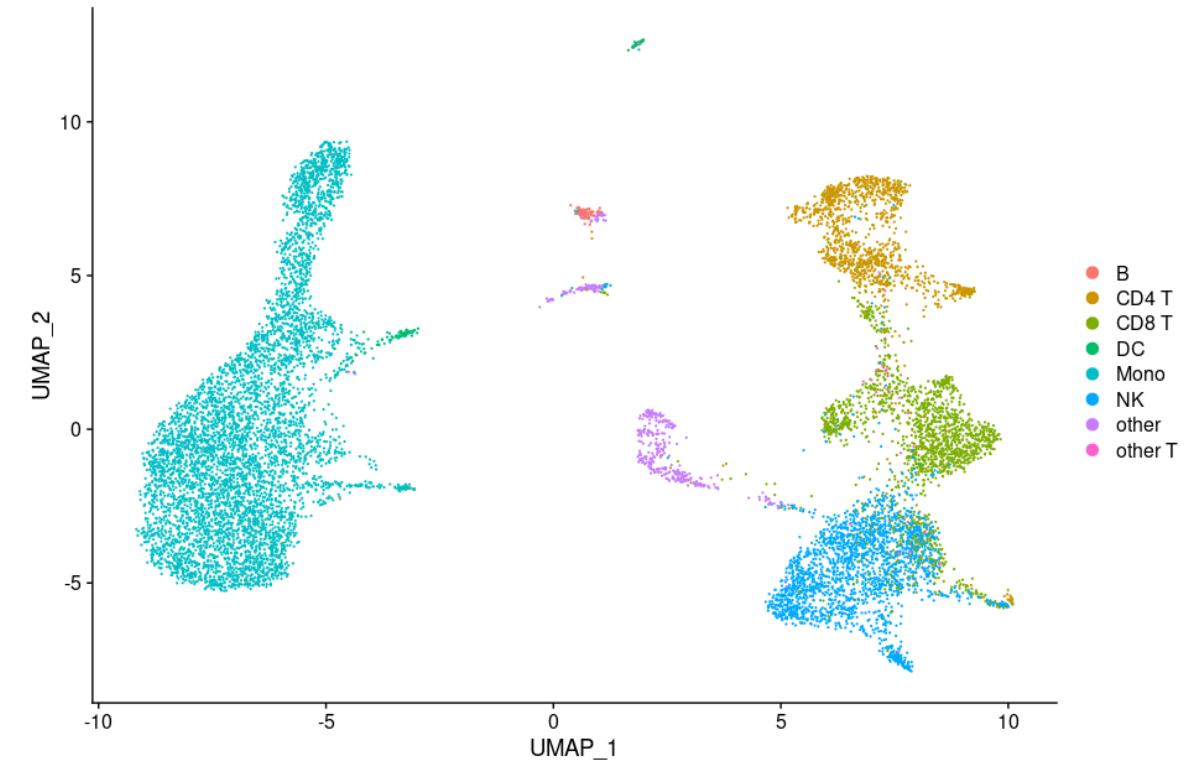
```
153
154 ````{r fig.width=10, fig.height=7}
155 DimPlot(pbmc.combined.sct, reduction = "umap", group.by = "predicted.celltype.l2")
156 DimPlot(pbmc.combined.sct, reduction = "umap", group.by = "predicted.celltype.l1")
157 ````
```



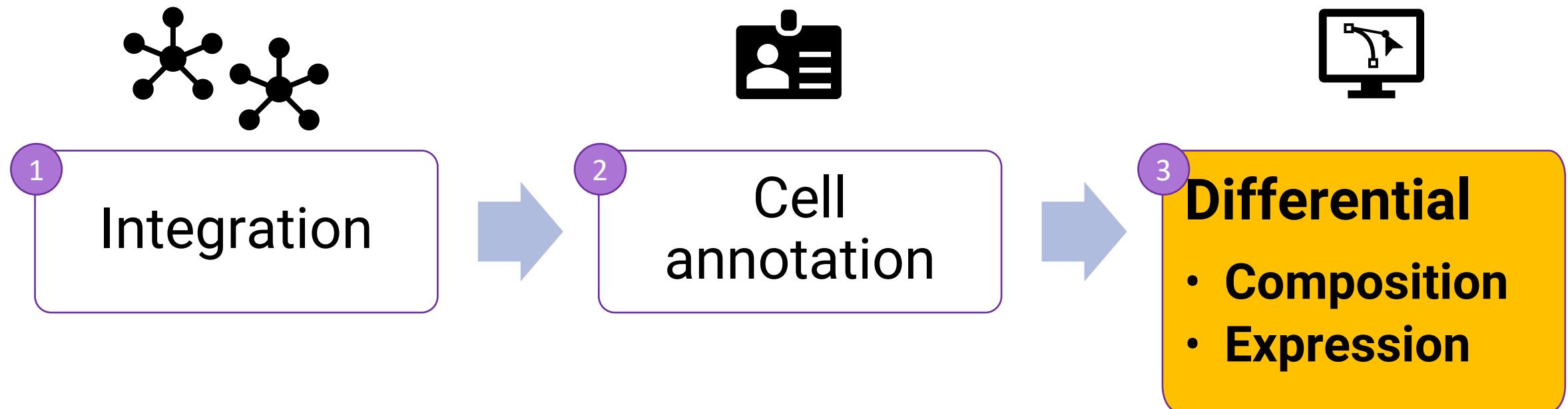
predicted.celltype.l2



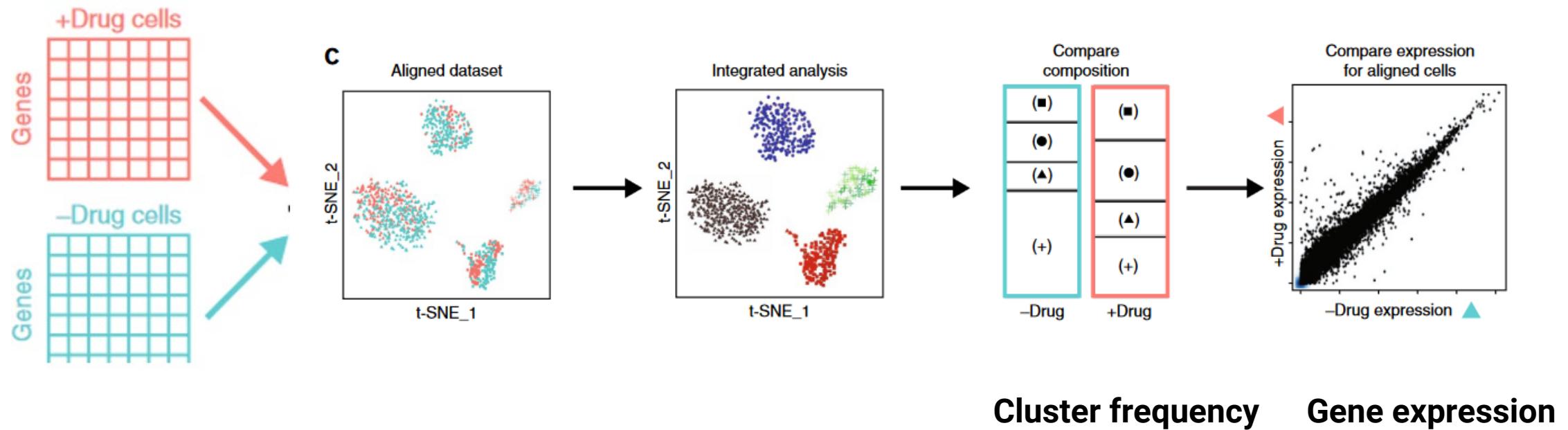
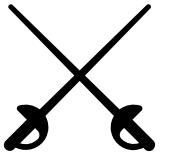
predicted.celltype.l1



Case versus control pipeline



Two types of comparison



Comparing cluster proportions



```
169 170 ~~~{r proportions per sample}
171 freq = plyr::count(cbind(as.character(pbmc.combined.sct$predicted.celltype.l1),
172 as.character(pbmc.combined.sct$orig.ident) ))
173 freq_table <- tidyverse::spread(freq, key=1, value=3)
174 freq_table[is.na(freq_table)] = 0
175 frac_table = t((freq_table[, -1])/rowSums(freq_table[, -1] , na.rm=T)) * 100
176 colnames(frac_table) = freq_table[,1]
177 my_color_palette <- scales::hue_pal()(8)
178 barplot( frac_table , col=my_color_palette )
179
180 ~~~
181 ~~~
182
```

11

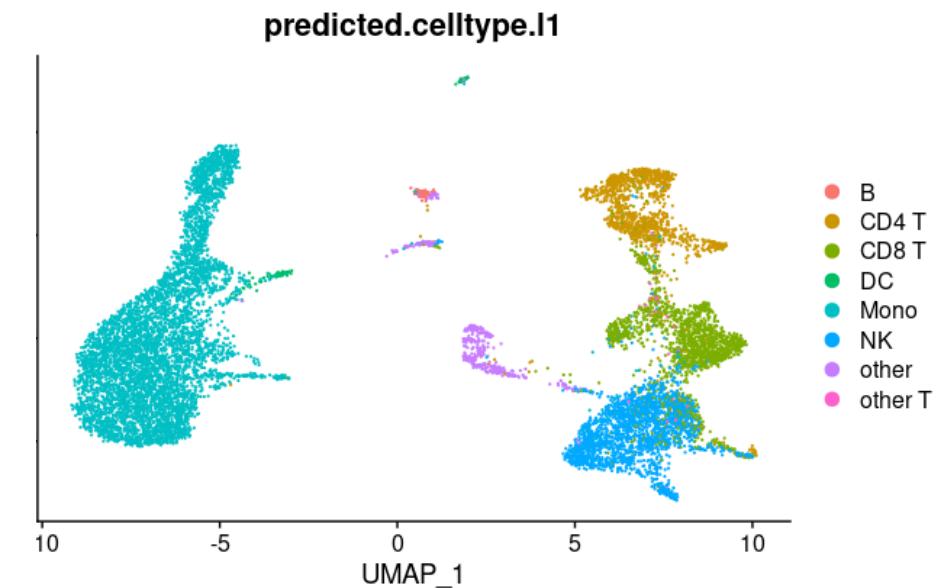
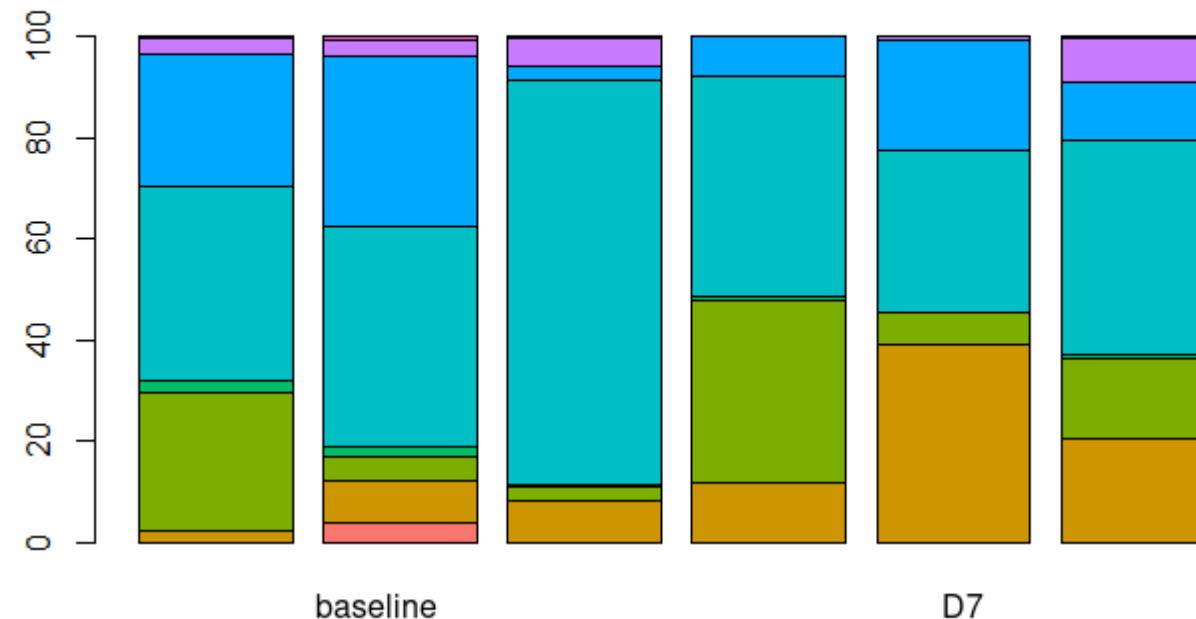
Calculate the frequency of each cell type per condition per sample.

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



UNSW
SYDNEY

Comparing cluster proportions





Testing for significance

```
182  
183 ## Run either t-test or ANOVA to calculate significance  
184 ``{r test for significance}  
185 pbmc.combined.sct$predicted.celltype.11 <- factor(pbmc.combined.sct$predicted.celltype.11)  
186 props = propeller(clusters=pbmc.combined.sct$predicted.celltype.11,  
sample=pbmc.combined.sct$orig.ident, group=pbmc.combined.sct$type)  
187 props  
188 ````
```

12

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.



UNSW
SYDNEY



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



```
189 ## Prepare data
190 Since we ran SCTtransform, 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.
191 ```{r find markers/DE, fig.height=8, fig.width=15}
192 pbmc.combined.sct = PrepSCTFindMarkers(pbmc.combined.sct)
193 filt = pbmc.combined.sct@meta.data$predicted.celltype.11 == "CD8 T"
194
195 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)
196
197 ````
```

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”.

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",
201 ylab="-log10(p-value)", pch=19, main = "baseline vs D7")
202 abline(h=-log10(0.05/dim(b_v_d7.markers)[1]), lwd=2, col=4, lty=2)
203 abline(v=c(1,-1), lty=3, lwd=2, col=4)
204 abline(v=c(0.5,-0.5), lty=2, col=4)
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.

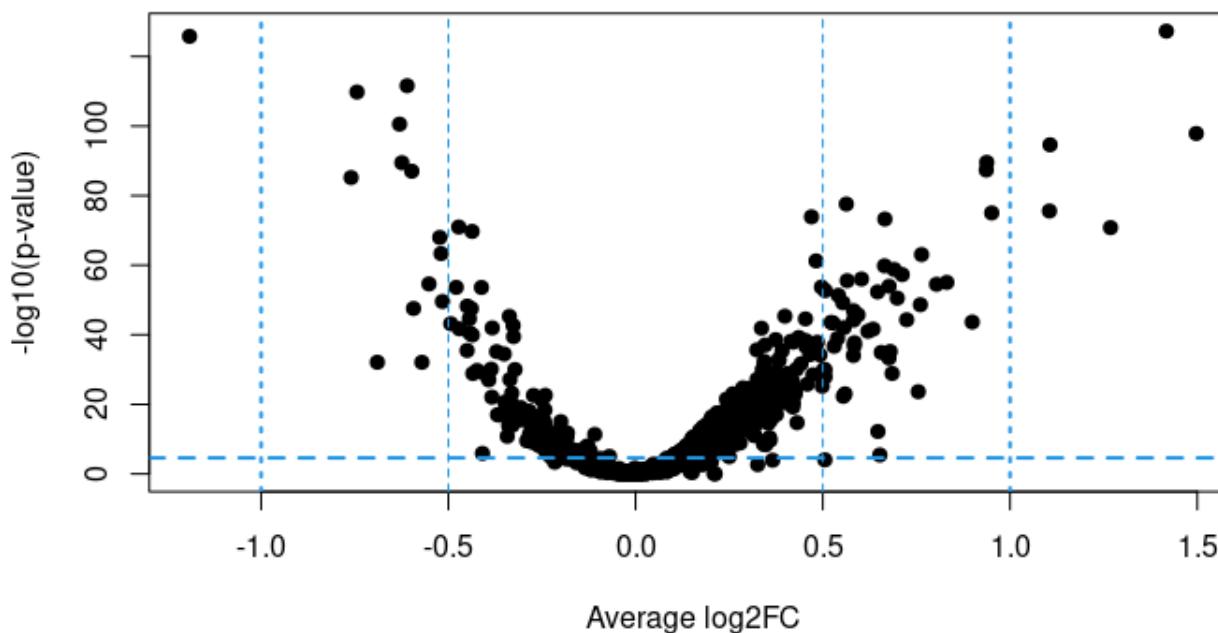


UNSW
SYDNEY

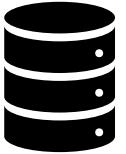
Gene biomarkers (volcano plot)



baseline vs D7



	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



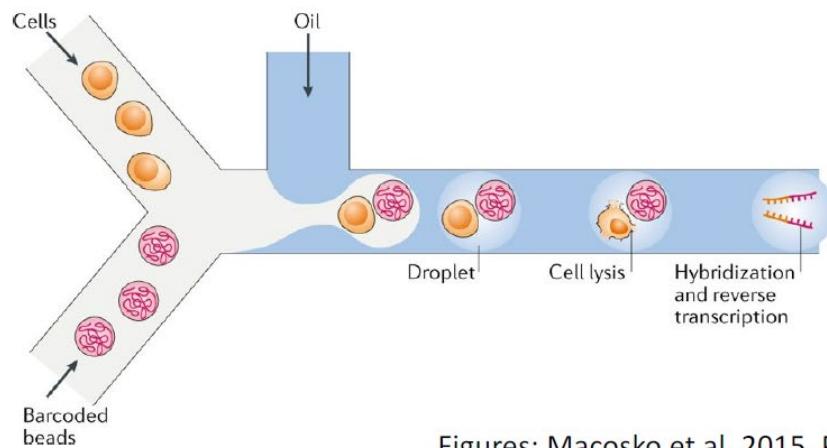
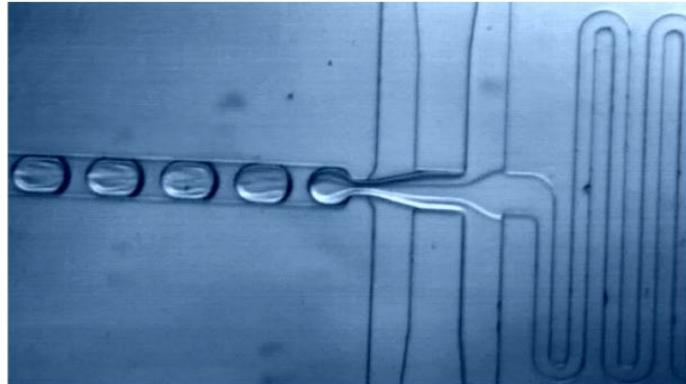
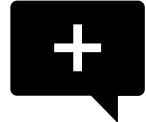
R, RStudio and Seurat installation

- 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



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

- **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

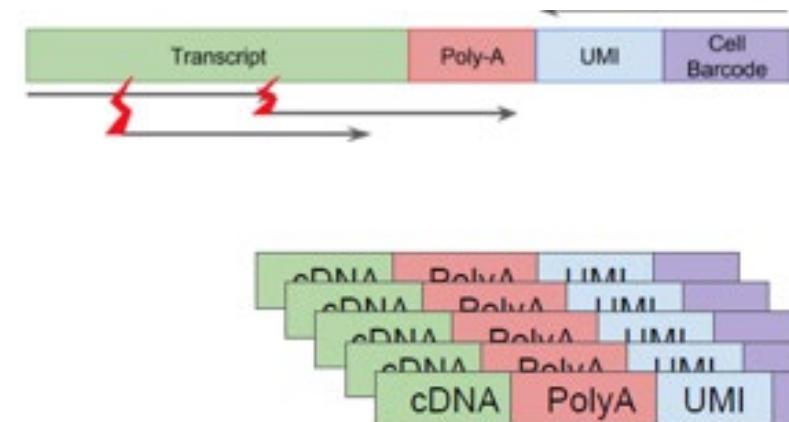
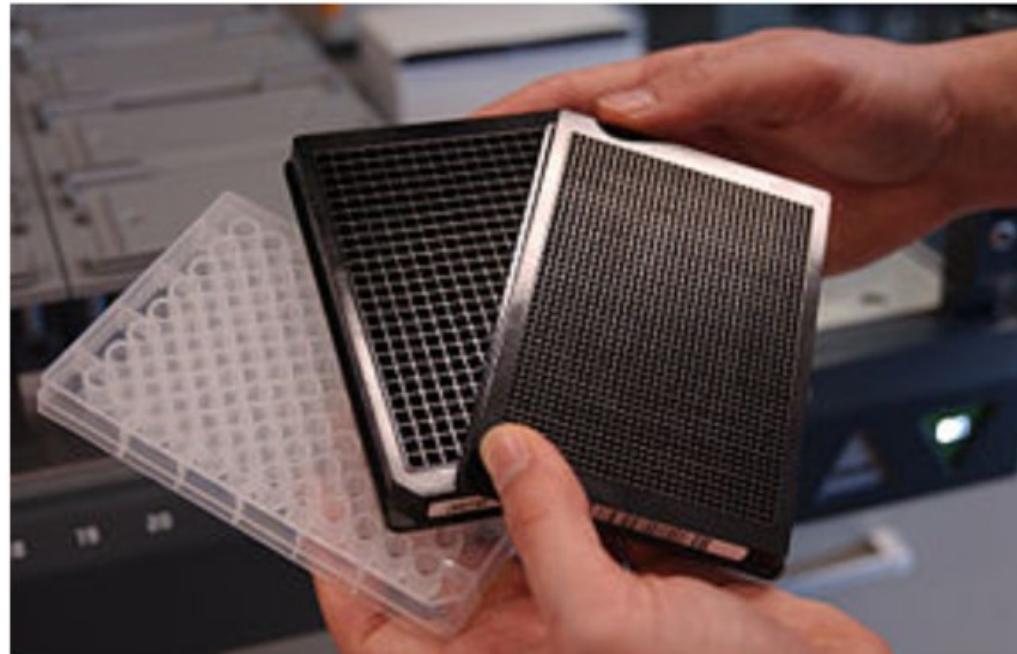
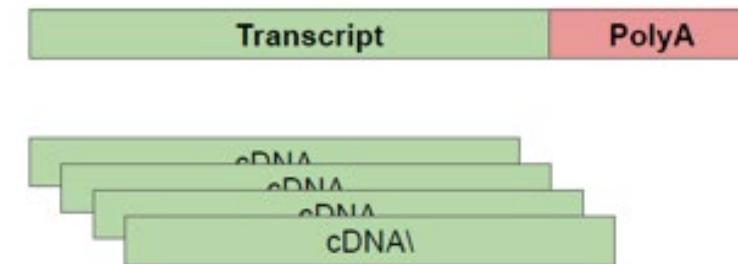




Plate-based



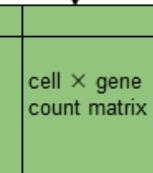
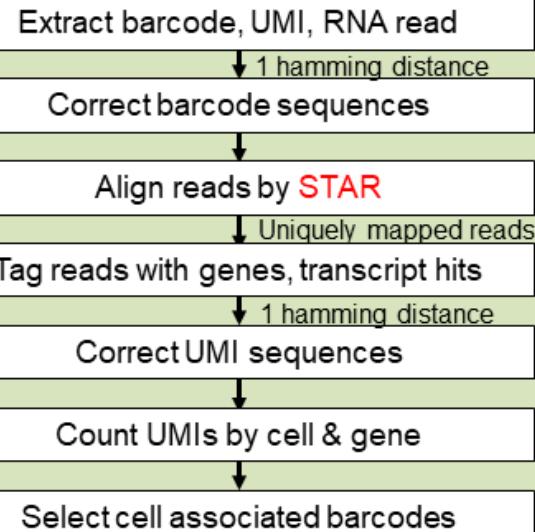
- **Pro:** Captures entire transcript
- **Con:** Low number of cells
- Useful for questions involving isoforms, mutations, SNVs



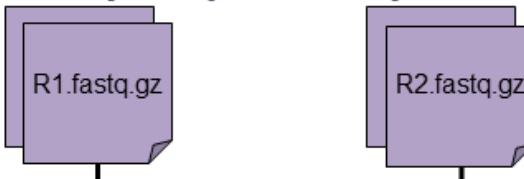
➤ 10X



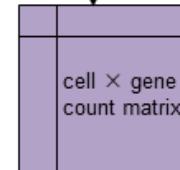
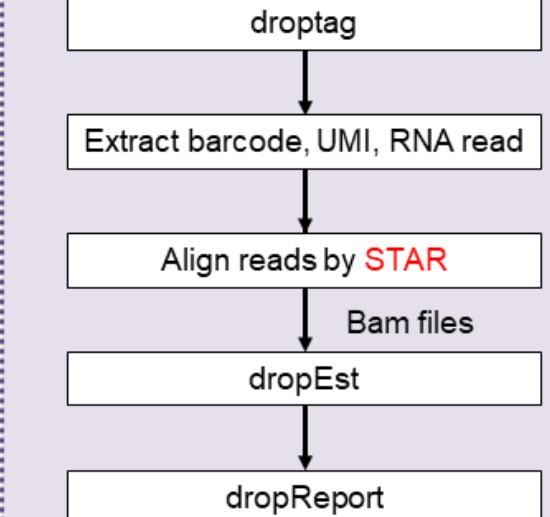
Cell Ranger count



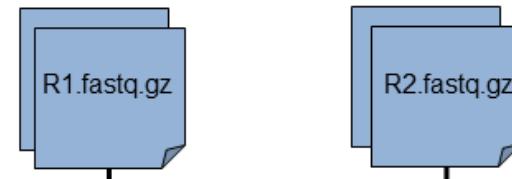
➤ Drop-seq & inDrop



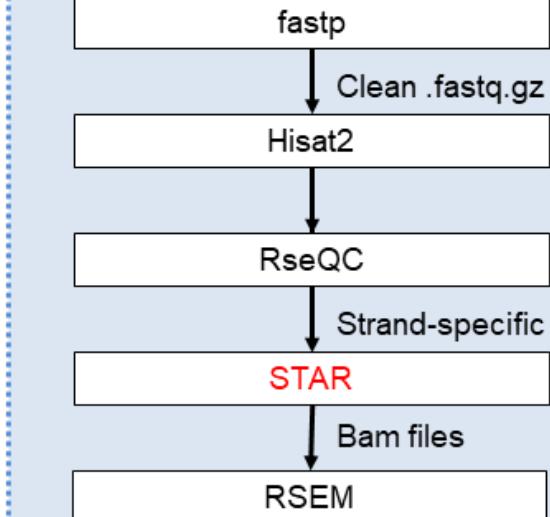
dropEst



➤ Bulk /SMART-based



STAR+RSEM





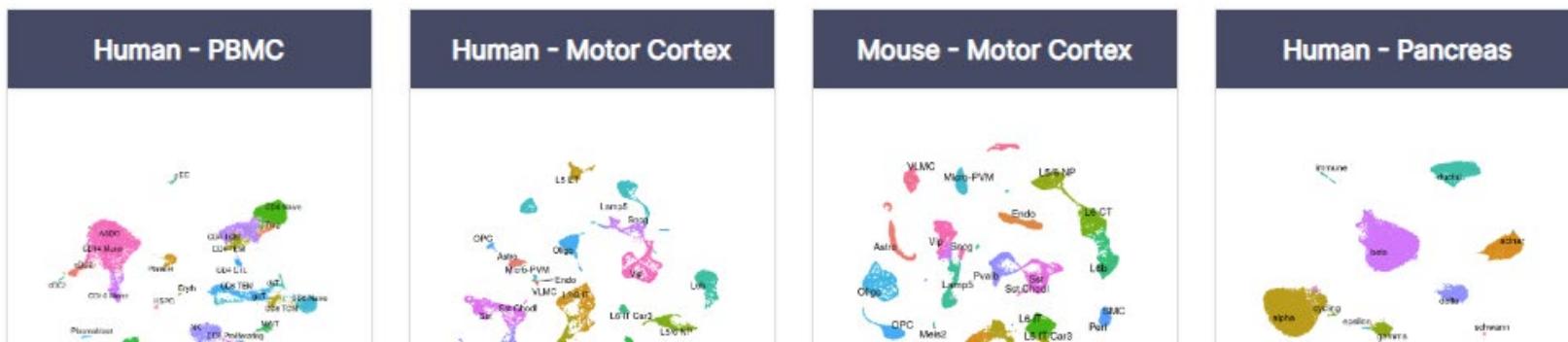
Azimuth

App for reference-based single-cell analysis

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



Azimuth demo



References for scRNA-seq Queries



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

Human
reference



UNSW
SYDNEY

File Upload ?

[Browse...](#)

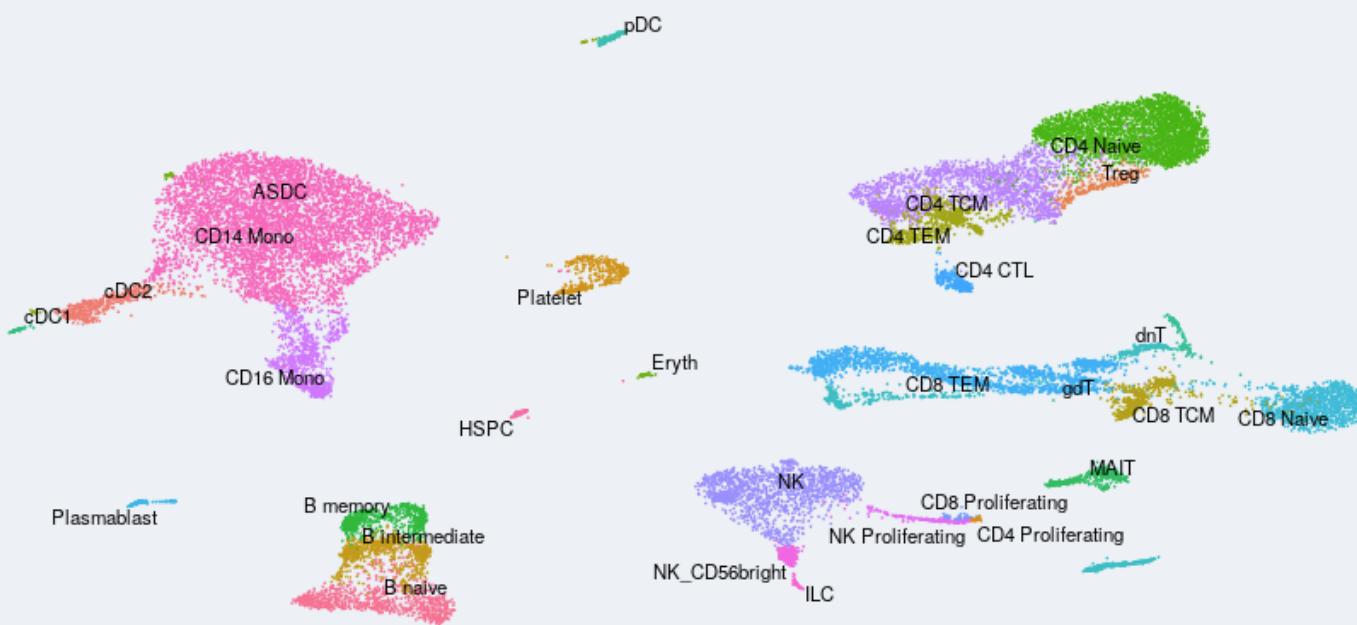
No file selected

[Load demo dataset](#)[Upload a file](#)[Welcome](#)[Feedback](#)

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, bioRxiv 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.



3

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

Data upload



File Upload



Browse... pbmc10k.rds

pbmc10k.rds

Load demo dataset

Upload a file

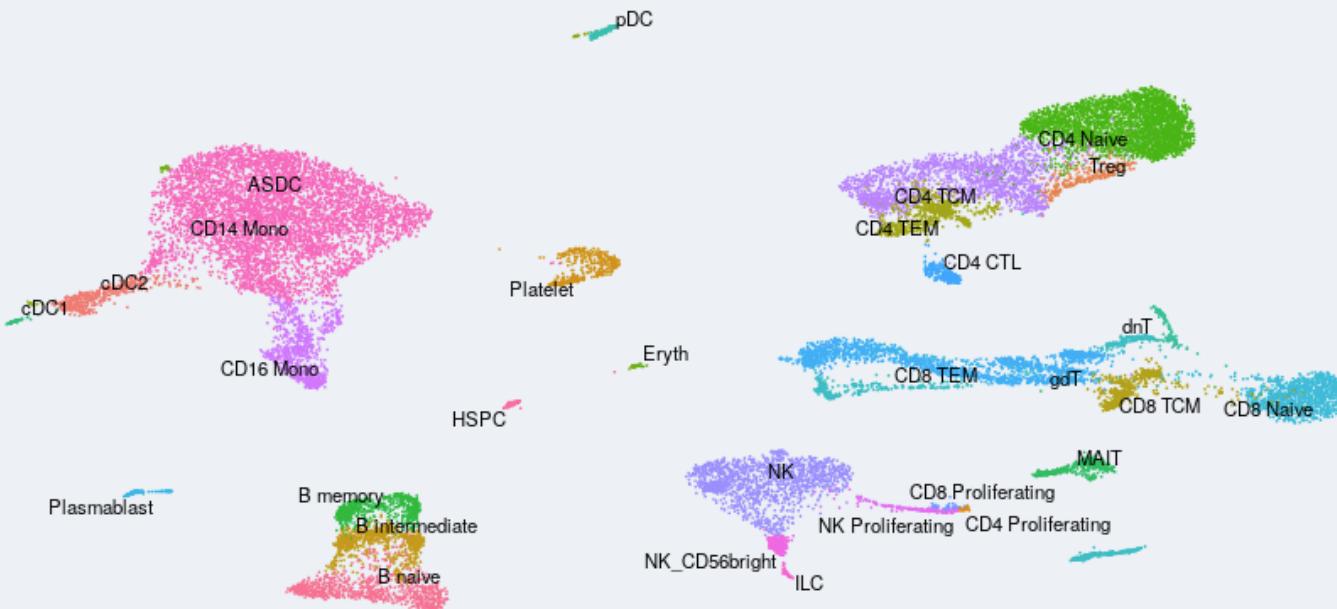
Welcome

Feedback

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, bioRxiv 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

debug ID: 886e5ed99398

Azimuth version: 0.3.2

Seurat version: 4.0.0

Reference version: 1.0.0



File Upload

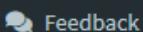
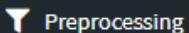
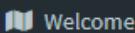


Browse... pbmc10k.rds

Upload complete

Load demo dataset

7499 cells uploaded



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

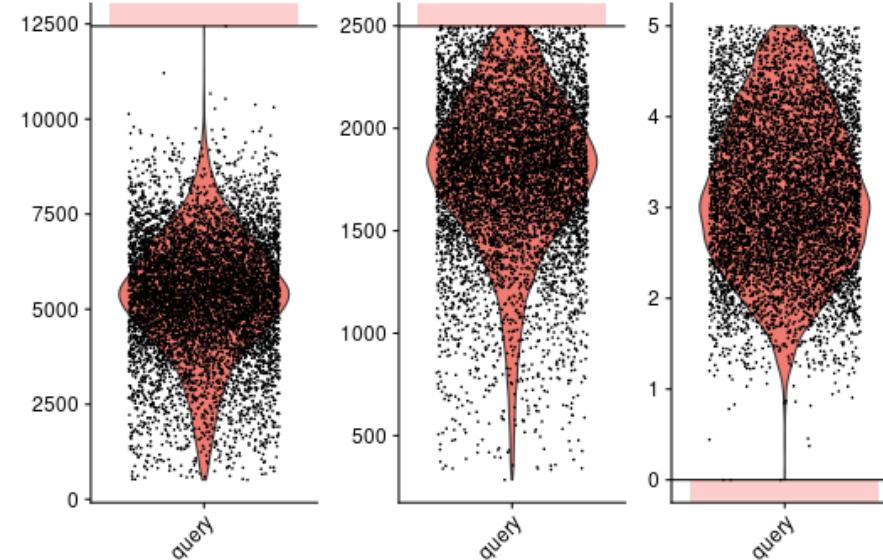
cells uploaded

 Log-scale Y-axis Hide points

nCount_RNA

nFeature_RNA

percent.mt



	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

File Upload ?

Browse... pbmc10k.rds
Upload complete

Load demo dataset

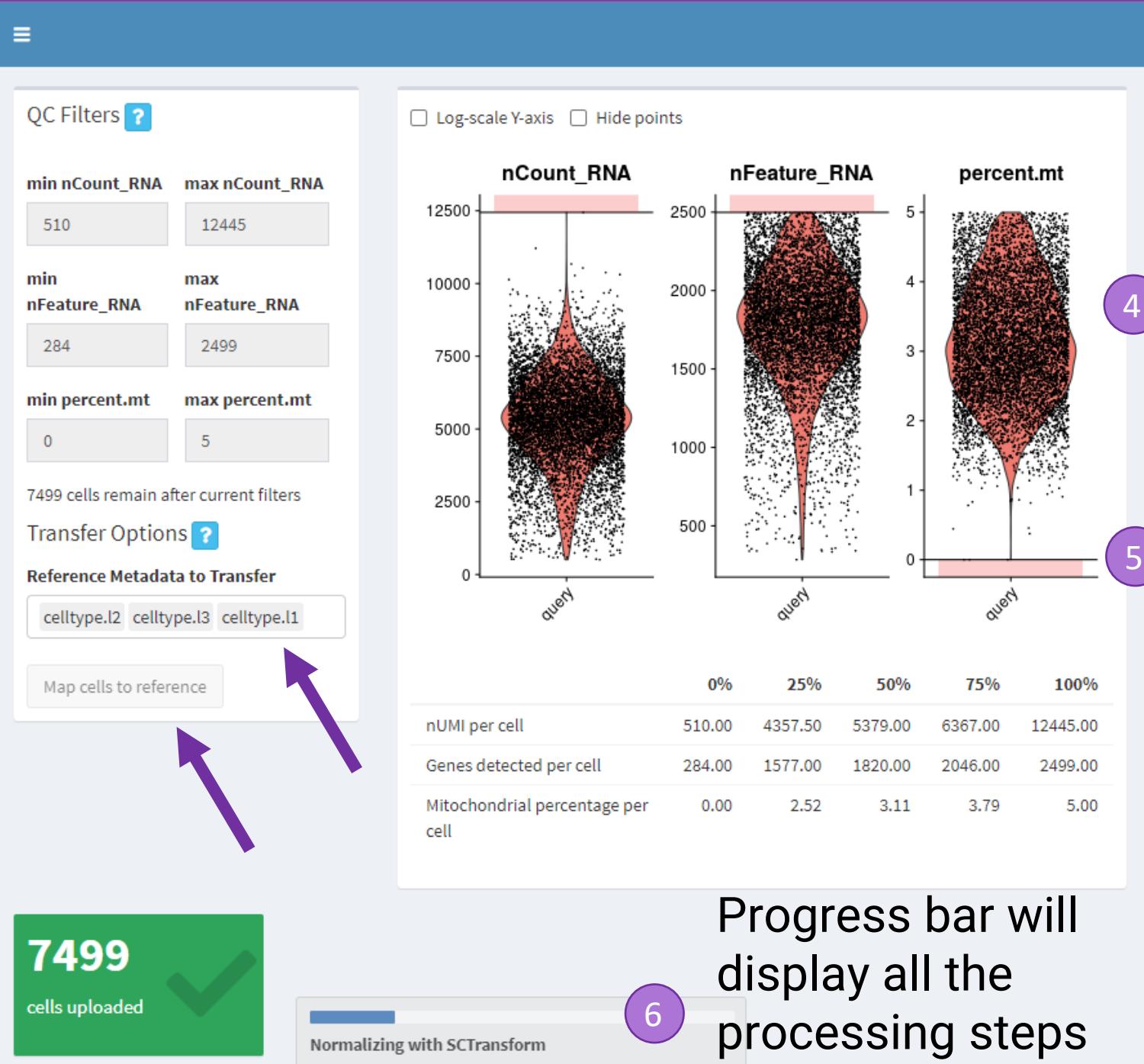
7499 cells uploaded

Welcome

Preprocessing

Feedback

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



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

Select metadata

Progress bar will display all the processing steps





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

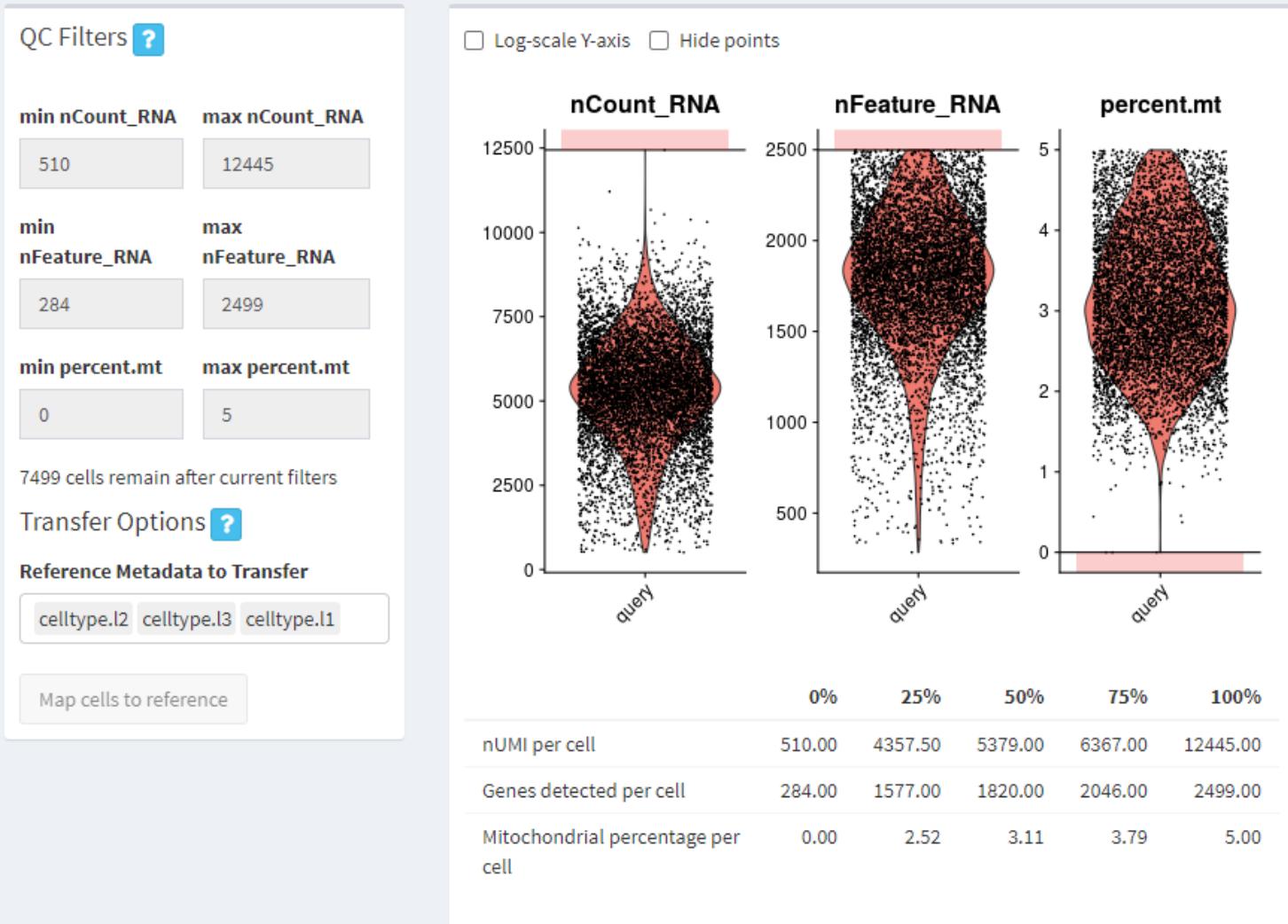
Feature Plots

Download Results

Feedback

7

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



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



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

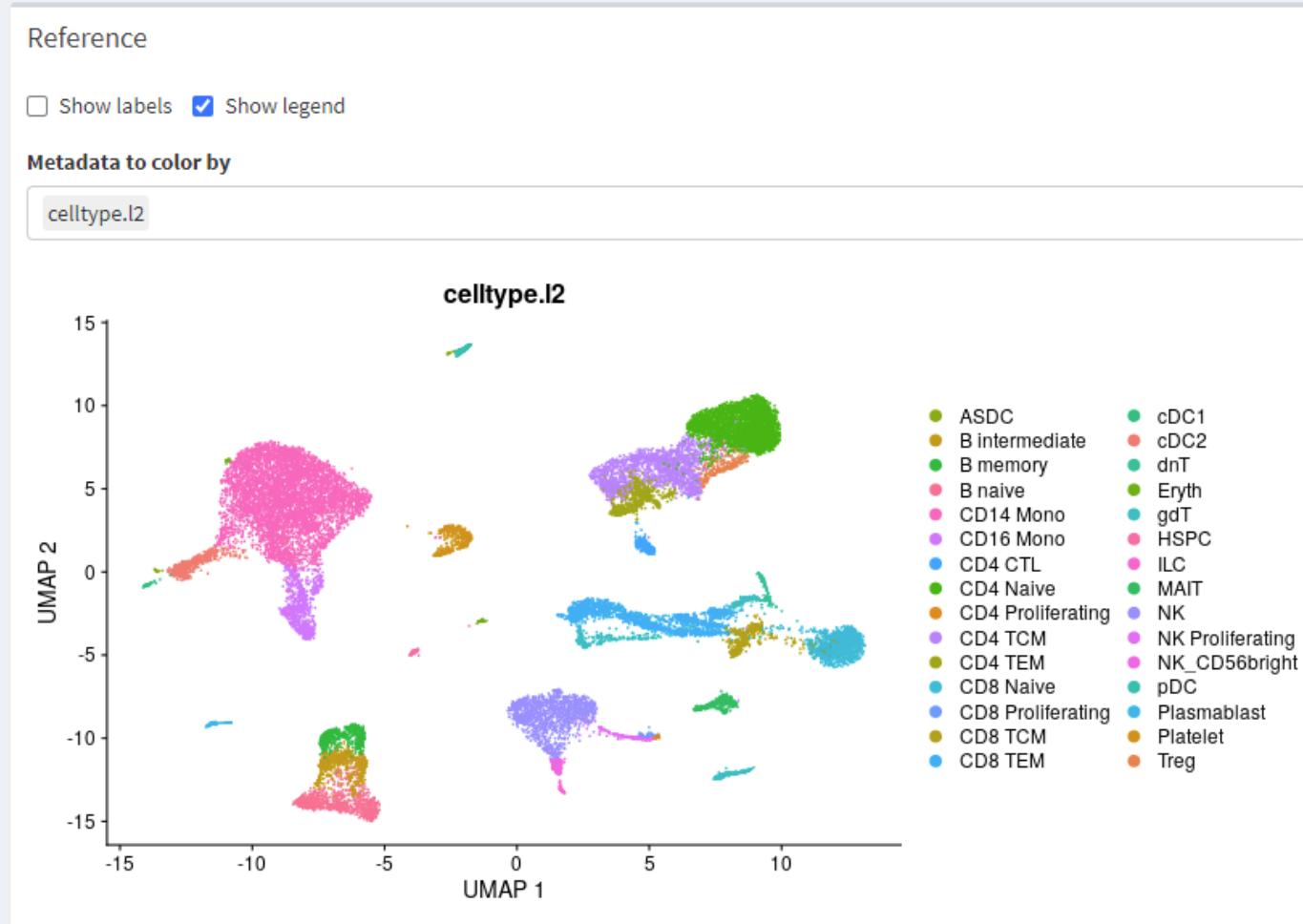
Cell Plots 8

Feature Plots

Download Results

Feedback

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



UMAPs of both reference and query (user) data.

Cell plots



UNSW
SYDNEY



Query

Metadata to color by

predicted.celltype.l2

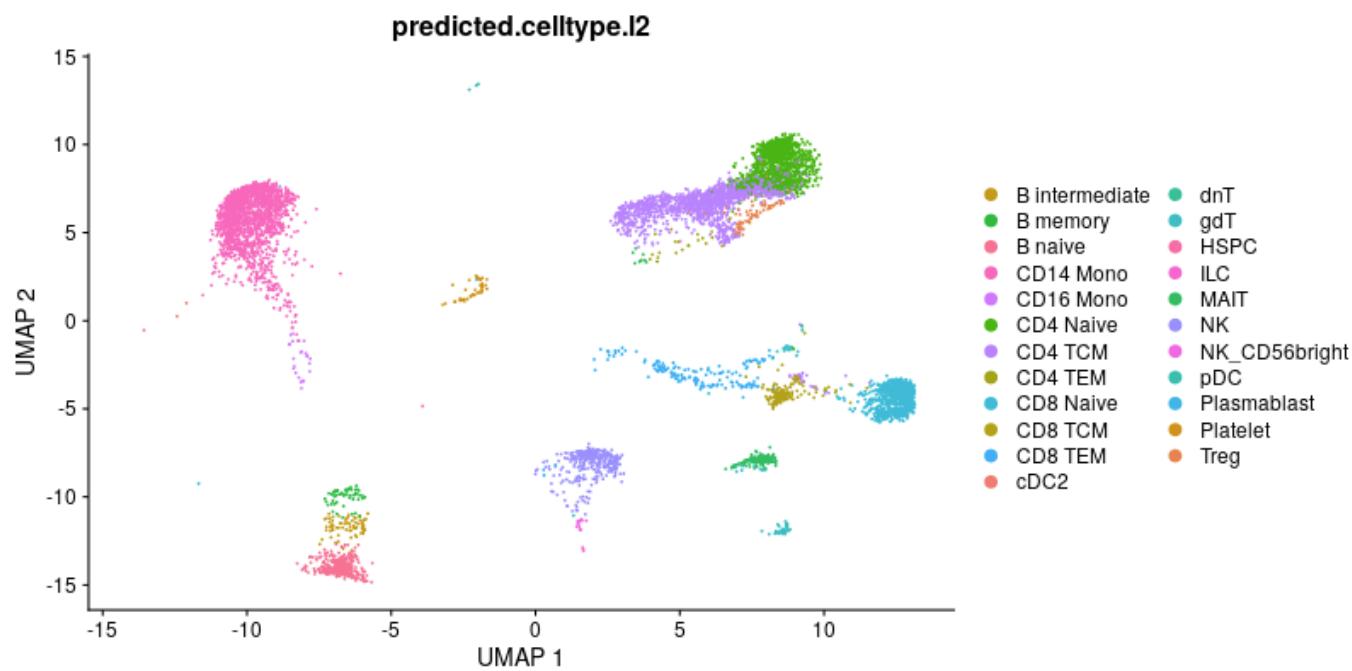
Metadata table [?](#)

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

UNSW
SYDNEY

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](#)

9

[Download Results](#)[Feedback](#)

debug ID: 886e5ed99398

Azimuth version: 0.3.2

Seurat version: 4.0.0

Reference version: 1.0.0

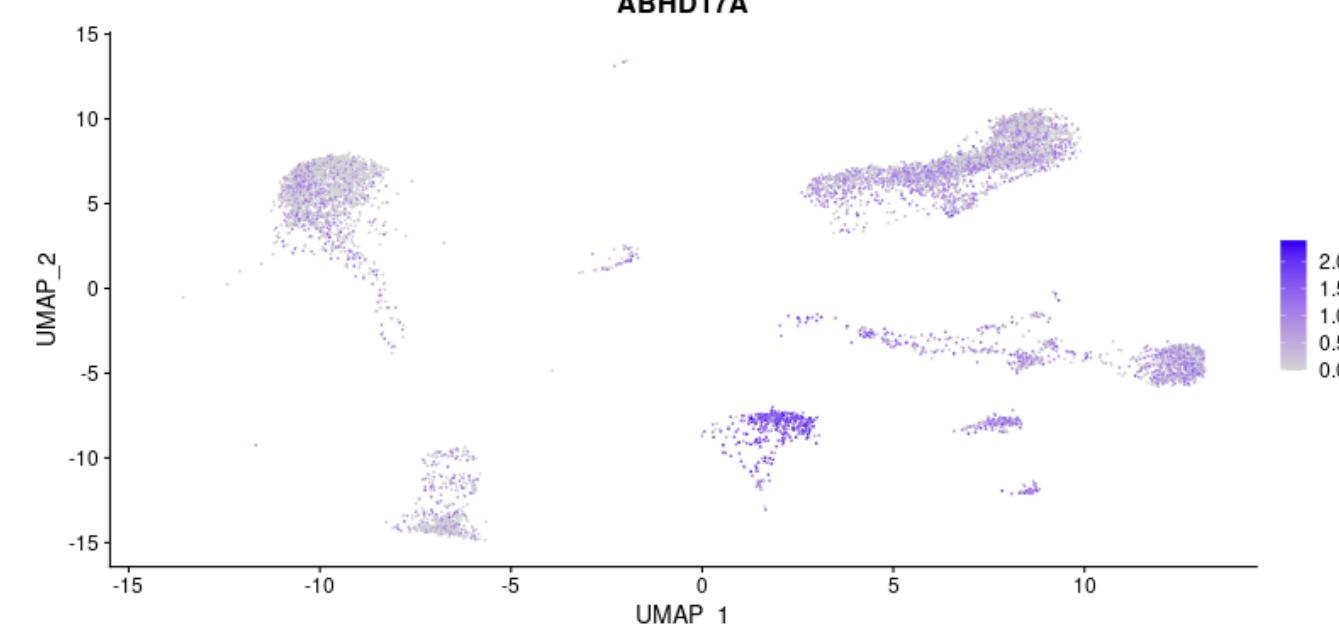
Feature Plots

Feature

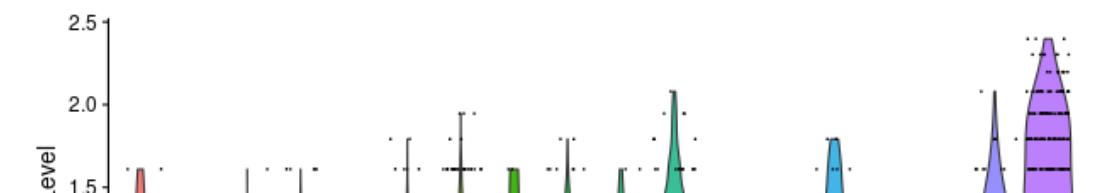
ABHD17A

Imputed protein

Prediction Scores and Metadata

 Hide points

ABHD17A



UMAPs and
violin plots of
features.

Feature plots

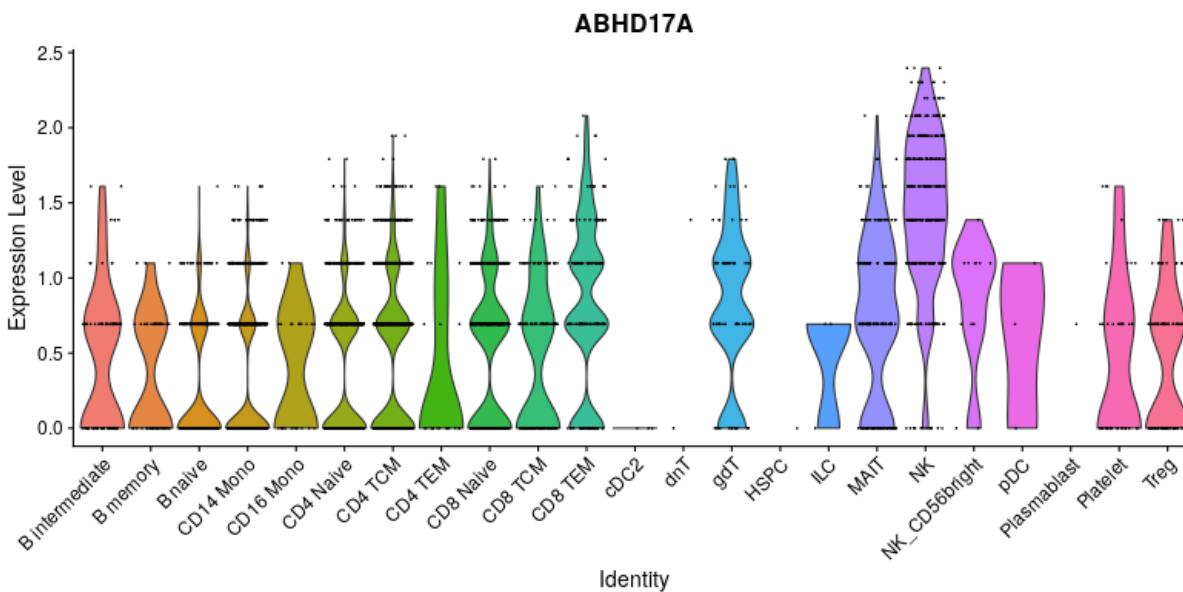
UNSW
SYDNEY



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



UNSW
SYDNEY



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

10

Feedback

debug ID: 886e5ed99398

Azimuth version: 0.3.2

Seurat version: 4.0.0

Reference version: 1.0.0

Analysis script template



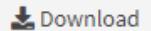
UMAP (Seurat Reduction RDS)

```
projected.umap <- readRDS('azimuth_umap.Rds')
object <- object[, Cells(projected.umap)]
object[['umap.proj']] <- projected.umap
```



Imputed protein (Seurat Assay RDS)

```
imputed.assay <- readRDS('azimuth_impADT.Rds')
object <- object[, Cells(imputed.assay)]
object[['impADT']] <- imputed.assay
```



Predicted cell types and scores (TSV)

```
predictions <- read.delim('azimuth_pred.tsv', row.names = 1)
object <- AddMetaData(
  object = object,
  metadata = predictions)
```



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

Download!

UNSW
SYDNEY

Merged (not integrated)

