



Single cell RNA sequencing data analysis

Practical exercises












Åsa Björklund

asa.bjorklund@scilifelab.se

Practicalities

- Work alone or in pairs as you chose yourself.
- TAs will be around to answer questions about the exercises.
- If you finish before hand, please try different settings in the algorithms we are using. Or try another pipeline.
- If you do not finish on time. Just execute all the code in the notebook so that you can continue with the next step and go back later.

<https://nbisweden.github.io/workshop-scRNAseq/exercises>

Tutorial	 Seurat	 Scater/Scran	 Scanpy
 Quality Control	Seurat_qc (.Rmd)	Scater_qc (.Rmd)	ScanPY_qc (.ipynb)
 Dimensionality reduction	Seurat_dr (.Rmd)	Scater_dr (.Rmd)	Scanpy_dr (.ipynb)
 Data integration	Seurat_integr (.Rmd)	Scater_integr (.Rmd)	Scanpy_integr (.ipynb)
 Clustering	Seurat_clust (.Rmd)	Scater_clust (.Rmd)	Scanpy_clust (.ipynb)
 Differential expression	Seurat_dge (.Rmd)	Scater_dge (.Rmd)	Scanpy_dge (.ipynb)
 Celltype prediction	Seurat_ct (.Rmd)	Scater_ct (.Rmd)	Scanpy_ct (.ipynb)
 Spatial transcriptomics	Seurat_ST (.Rmd)	Scater_ST (.Rmd)	Scanpy_ST (.ipynb)
 Trajectory inference	Slingshot_ti (.Rmd)	Slingshot_ti	PAGA_ti

Three main pipelines for analysing single cell data:

- Seurat:
 - R based, centered around Seurat objects.
 - Mainly developed for droplet based data
 - Easy to use, recommended for R beginners
 - Cons: uses a LOT of memory
- Scrn:
 - R based, centered around SingleCellExperiment objects
 - Has more different statistical methods
 - Can handle spike-ins
 - Cons: More complicated than Seurat to run.
- Scanpy:
 - Python based
 - Handles large datasets better. More and more development here.
 - Cons: Requires quite some python knowledge. Does not have all the functionality of the R based tools.

Seurat object

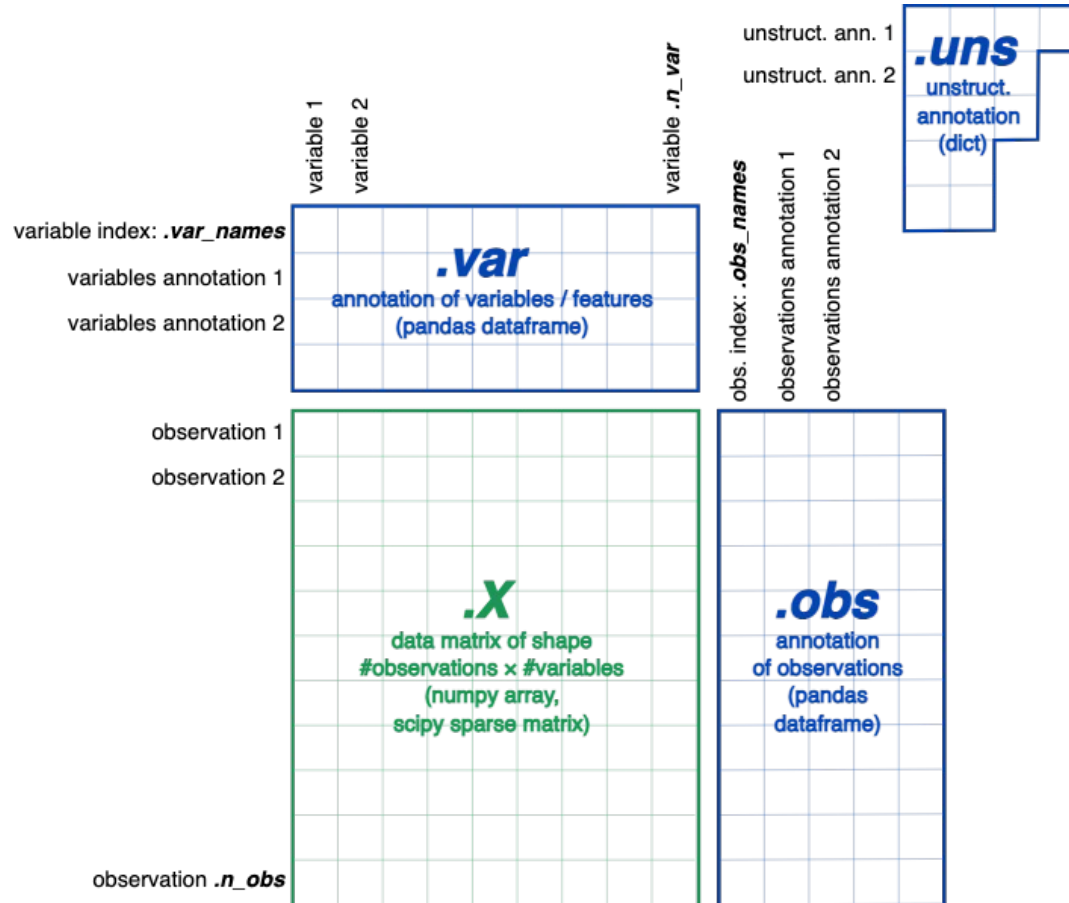
Slot	Function
<code>assays</code>	A list of assays within this object
<code>meta.data</code>	Cell-level meta data
<code>active.assay</code>	Name of active, or default, assay
<code>active.ident</code>	Identity classes for the current object
<code>graphs</code>	A list of nearest neighbor graphs
<code>reductions</code>	A list of DimReduc objects
<code>project.name</code>	User-defined project name (optional)
<code>tools</code>	Empty list. Tool developers can store any internal data from their methods here
<code>misc</code>	Empty slot. User can store additional information here
<code>version</code>	Seurat version used when creating the object

SingleCellExperiment (SCE) objects

```
## class: SingleCellExperiment
## dim: 611 379
## metadata(2): SuppInfo which_qc
## assays(3): tophat_counts logcounts counts
## rownames(611): 0610007P14Rik 0610009B22Rik ... 9930111J21Rik1
##   9930111J21Rik2
## rowData names(0):
## colnames(379): SRR2140028 SRR2140022 ... SRR2139341 SRR2139336
## colData names(22): NREADS NALIGNED ... Animal.ID passes_qc_checks_s
## reducedDimNames(2): PCA TSNE
## altExpNames(3): ERCC RIKEN original
```

<https://bioconductor.org/packages/release/bioc/vignettes/SingleCellExperiment/inst/doc/intro.html>

AnnData (Scanpy) objets

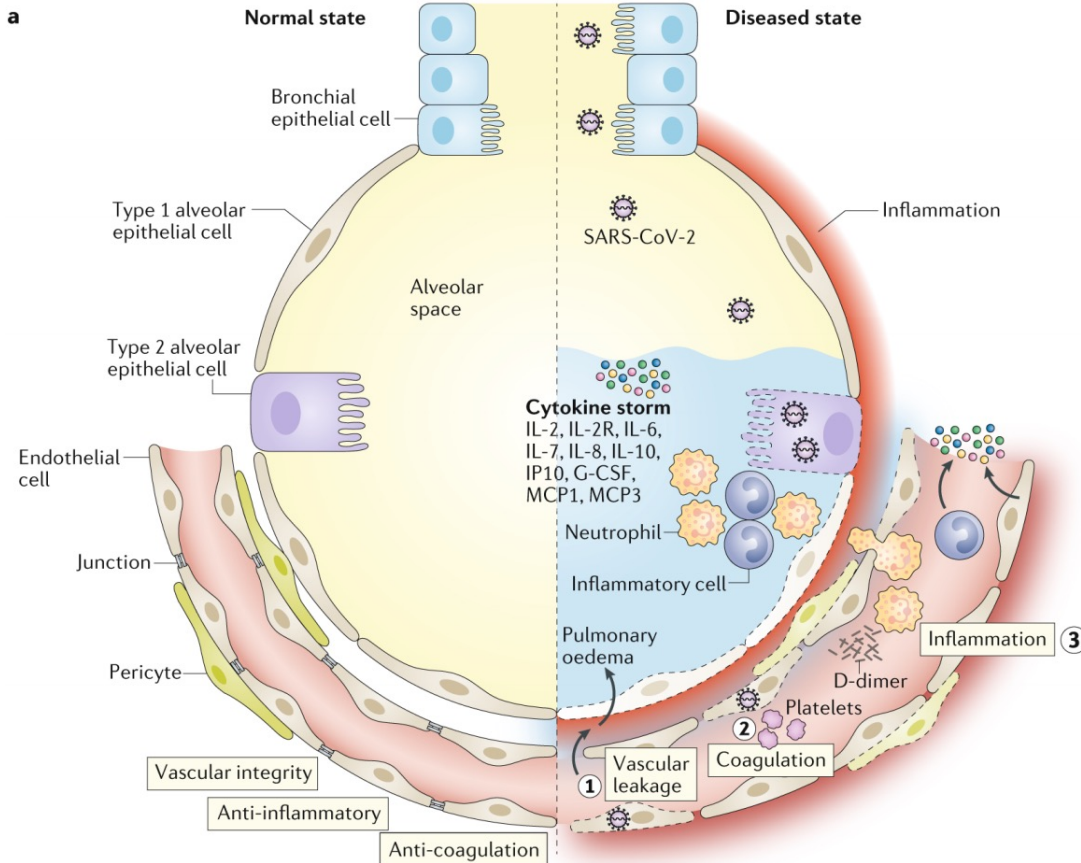


https://anndata.readthedocs.io/en/latest/ann_data.AnnData.html

What to chose?

- It is recommended that you go through all the steps with one pipeline as each exercise depends on saved objects from the previous step.
- Everyone works in very different pace. Focus on one of the pipelines first. If you have time left over, you can also try out the other ones.

The datasets – Covid-19 PBMCs



Teuwen et al (2020) *Nat reviews Immunology*

Elderly patients usually develop severe lung inflammation and lung dysfunction.

Many cell types orchestrate the immune response to the virus.

Their relative contribution at the single-cell resolution is still unclear

The datasets – Covid-19 PBMCs

- Data from paper: "Immunophenotyping of COVID-19 and influenza highlights the role of type I interferons in development of severe COVID-19" Lee et al. Sci Immuno
- We have selected 3 controls and 3 severe covid samples and subsampled to 1500 cells per subject for computational speed/memory.
- ST and trajectory lab will be with other datasets.

Installation of all packages

- We have created a conda environment for the course that should contain all packages you need for the exercises
- However, for slingshot trajectory inference lab, there is an additional conda environment that needs to be installed.
- If you chose to instead work with standard R installations, you can use the list of required packages in the environment file and install them on your own.

Why conda?

- Often easier installations compared to traditional R installation for packages with C-compilation etc.
- Good way to manage different versions of packages in different projects.
- There are other ways of managing packages. E.g packrat for R, pyenv for python etc.

The code:

- All code for the exercises is available as R-markdown documents, or jupyter notebooks, in the folder:
workshop-scRNAseq/labs/compiled/
- Please report to us if you find any errors in the code!
 - Slack channel **#exercises**
 - An Issue on the github page.
- We may find bugs and update the code – in that case, update your git repo with command “git

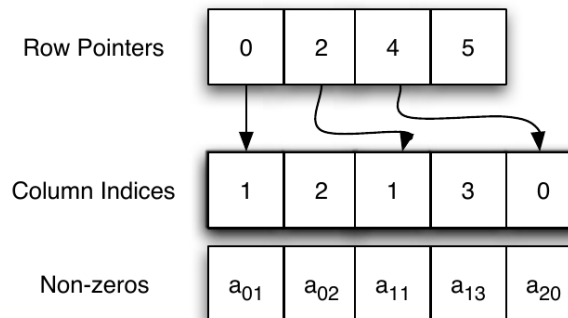
Reproducible coding

- You should always be able to find and recreate the results.
 - Scripts should be able to run from input files to create the output.
 - Never work with saved R sessions!
- Name your scripts with relevant names so you can find them 2 years later 😊
- Always backup code – good idea to use github that also gives you version control.

Sparse vs dense matrices

- scRNAseq data is large matrices with many zeros -> perfect for sparse matrices.
- Only has representation of non-zero value and its positions.
- In R – need package Matrix for any matrix operations. Seurat uses dgCMatrix format.
- In python - scipy.sparse, normally csr_matrix

0	a_{01}	a_{02}	0
0	a_{11}	0	a_{13}
a_{20}	0	0	0

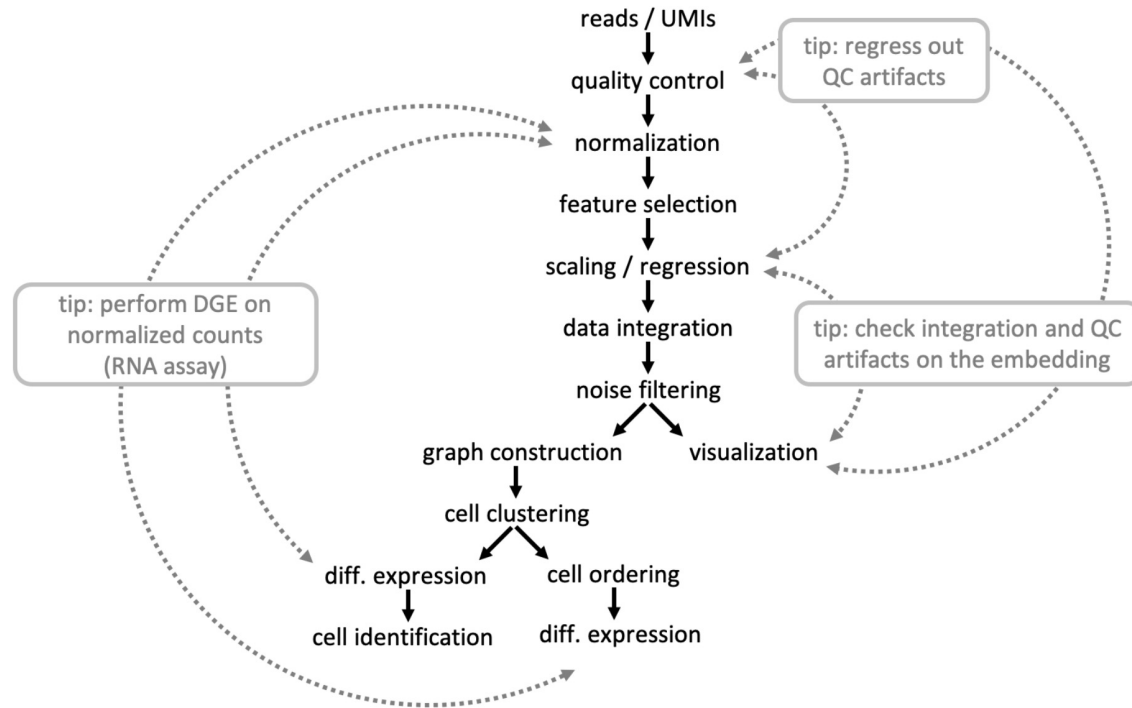


Memory issues

- scRNAseq datasets are often large, think about how you code. Avoid duplicating objects!
- Remove unused matrices and clear memory with `gc()`.
- Try to keep your matrices sparse!
- If you still have issues with memory in R, test setting e.g. `R_MAX_VSIZE=70Gb` in the `.Renv` file. Default is 16Gb. (check FAQ section)
- In Seurat – can use `DietSeurat` function to remove assays, data slots etc.

Troubleshooting

- Slack channel - **#exercises** or just raise your hand
- It is important that you learn how to troubleshoot yourselves.
 - Look at your error messages, perhaps the answer is there?
 - If not – Google is your best friend! Forums like Seqanswers, Stackexchange, Bioconductor support forum, specific forums (or github issues) for each package may have the answer.
- TAs are there to answer any questions and give suggestions, but we may not always have the answer.



Downloading data

► Running bash code in RStudio

Seurat Objects

Rmarkdown (.Rmd)

- Complete reports with both text, code and plots.
- 3 main parts:
 - **Yaml header** – specify output formats and config.
 - **Code chunks** – all code, define output styles for plots and code evaluation
 - **Markdown text** – follows markdown syntax to produce headers and text.

```
---  
title: "Untitled"  
author: "Anonymous"  
output: html_document  
---
```

This is the start of my report. The above is metadata saved in a YAML header.

```
Here's some code  
{r}  
dim(iris)
```

End a line with two spaces to start a new paragraph.
italics and `_italics_`
bold and `__bold__`
superscript²
~~strikethrough~~
[\[link\] \(www.rstudio.com\)](#)

```
# Header 1
```

```
## Header 2
```

<https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf>

Rmarkdown demonstration