Single cell RNA sequencing data analysis
Practical exercises
4-6 February 2019

Åsa Björklund

asa.bjorklund@scilifelab.se

NBIS NATIONAL BIOINFORMATICS INFRASTRUCTURE SWEDEN

SciLifeLab

Practical on zoom

- Work in groups of 5, the idea is that you can help each other out and discuss results.
- One zoom breakout room per group, you should have already been assigned to a group in slack. Please stay online in the breakout room for your group.
- Come to main room for question or tell us on slack an we can come to your breakout room.
- We have some extra rooms where we can sit with a student if there are specific questions.

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)		Scanpy_ST (.ipynb)
Trajectory inference	Slingshot_ti (.Rmd)	Slingshot_ti	PAGA_ti

NBIS NATIONAL BIOINFORMATICS INFRASTRUCTURE SWEDEN

SciLifeLab

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
- Scran:
    - 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.
    - Cons: Requires quite some python knowledge. Does not yet have all the functionality of the R based tools

Seurat object

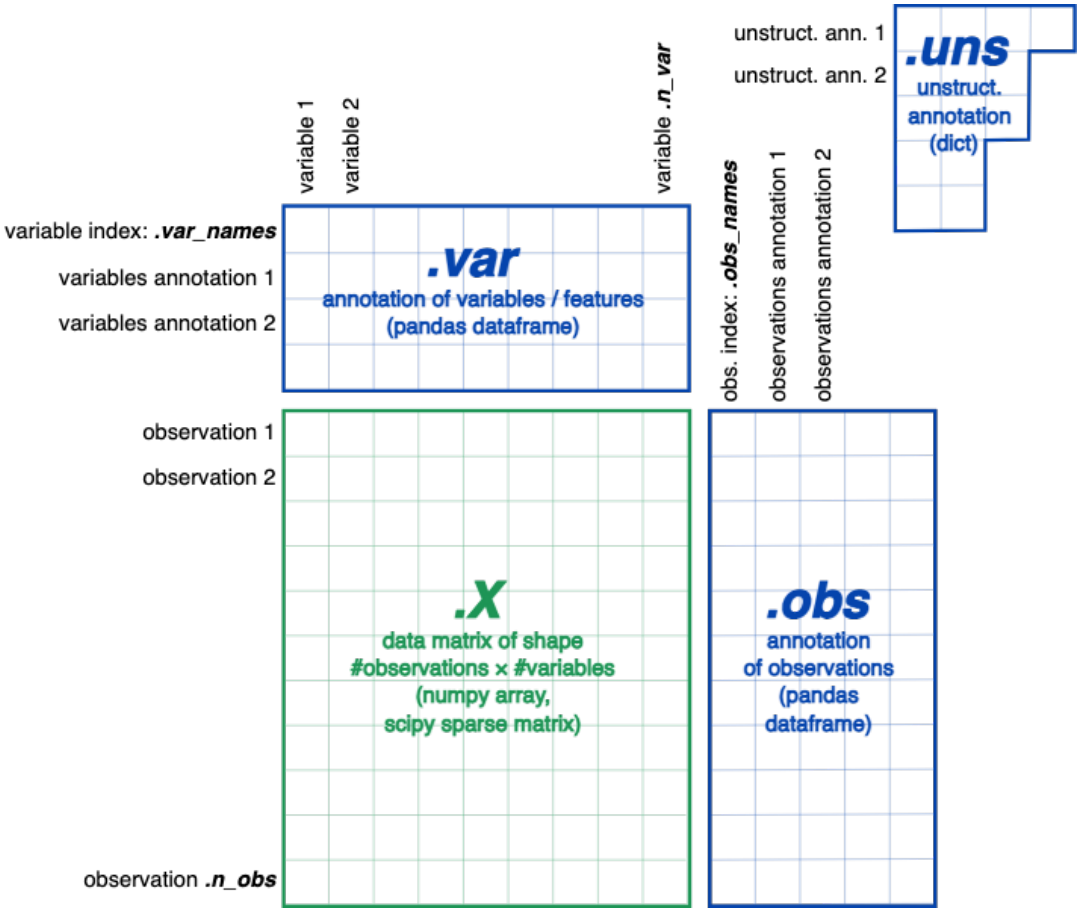
Slot	Function
`assays`	A list of assays within this object
`meta.data`	Cell-level meta data
`active.assay`	Name of active, or default, assay
`active.ident`	Identity classes for the current object
`graphs`	A list of nearest neighbor graphs
`reductions`	A list of DimReduc objects
`project.name`	User-defined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

NBIS NATIONAL BIOINFORMATICS INFRASTRUCTURE SWEDEN

SciLifeLab

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

- 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**

- We may find bugs and update the code – in that case, update your git repo with command "`git pull`"

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.

Memory issues in R

- scRNAseq datasets are often large, think about how you code. Avoid duplicating objects!
- Remove unused matrices and clear memory with gc().
- Most packages store scRNAseq as sparse matrices, will require package Matrix or similar for many standard matrix operations.
- If you still have issues with memory in R, test setting e.g. R_MAX_VSIZE=70Gb in the .Renviron file. Default is 16Gb.

Troubleshooting

- Slack channel - **#exercises**
- 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.

Rmarkdown demonstration