Interactive Computing at the VSC

Kris Davie

VIB-CBD

Bioinformatics Expertise Unit Leader

06/12/2019

http://bit.ly/vsc_bioinf_sc_pdf

http://bit.ly/vsc_bioinf_sc

VIB-KU LEUVEN CENTER FOR BRAIN & DISEASE RESEARCH

KU LEUVEN

SCIENCE MEETS LIFE

Biomed Resources

- Resources Directory
  - Path:- /staging/leuven/res_00001/
  - Based on Illumina iGenomes
    - Come with uniform chromosome names etc.
      - Within annotation, not between
    - Versioned and reproducible, standard file structure
    - Small script to link/move files to our structure
  - Standard indexes => Easily comparable data
  - Extra annotations can be added by myself, everything must be reproducible and documented
  - Now hosting cisTarget databases
  - Note: Not all iGenomes are made the same (i.e. MM10 UCSC has no MT genes)

Biomed Containers

- Singularity Containers
- /staging/leuven/res_00001/software
  - CellRanger
  - STAR(solo)
  - Scanpy (with ipython)
  - cisTopic
  - pySCENIC
- `singularity run -B /staging/:/staging/,/ddn1/:/ddn1/ /staging/leuven/res_00001/software/STAR/2.7.0f/STAR_2.7.0f.sif --soloType Droplet –readFilesCommand ......`
  - `-B bind directories for use within the container`
- All containers built by myself have an entrypoint as that command
- Completed (tested, documented and reliable) pipelines will eventually be here too

Other Biomed Software

- We keep some software specific to Biomed applications in the following directories on the VSC:
- ThinKing: `/data/leuven/software/biomed/haswell_centos7`
- Genius: `/data/leuven/software/biomed/skylake_centos7`
- You can enable these modules using

- `module use /data/leuven/software/biomed/${VSC_ARCH_LOCAL}_centos7`

Working in Python or R interactively

- I will only be supporting and usage through JupyterLab and the Command line
- For now, everyone requires their own install – we will do this together (slides will also be available) using conda
- JupyterLab can be set up to look and work similar Rstudio if you want

Log in to these servers

- Everyone - Connect to `login5-tier2.hpc.kuleuven.be`
- Row by row

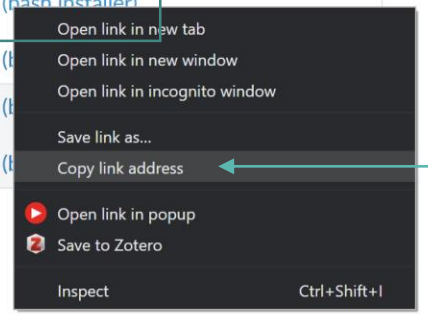
```
1. ssh r04i02n09
2. ssh r04i02n10
3. ssh r04i02n11
4. ssh r04i02n12
```

Install Miniconda

```
$ cd $VSC_DATA

# Copy link from https://docs.conda.io/en/latest/miniconda.html
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh

$ bash Miniconda3-latest-Linux-x86_64.sh -p $(pwd)/jupyter
# Press enter, type yes, check path is correct, yes to init

$ exec bash
```

Miniconda

	Windows	Mac OS X	Linux
Python 3.7	64-bit (exe installer) 32-bit (exe installer)	64-bit (bash installer) 64-bit (.pkg installer)	64-bit (bash installer) 32-bit (
Python 2.7	64-bit (exe installer) 32-bit (exe installer)	64-bit (bash installer) 64-bit (.pkg installer)	64-bit (32-bit (



Installation instructions

Other resources:

- Miniconda with Python 3.7 for Power8 & Power9
- Miniconda with Python 2.7 for Power8 & Power9
- Miniconda Docker images
- Miniconda AWS images
- Archive and MD5 sums for the installers
- conda change log

These Miniconda installers contain the conda package manager and Python. Once Miniconda is installed, you can use the conda command to install any other packages and create environments, etc. For example:

```
$ conda install numpy
...
$ conda create -n py3k anaconda python=3
...
```

Install JupyterLab

```
# JupyterLab
$ conda create -n jupyter python=3.7.3 nb_conda_kernels jupyterlab nodejs

# conda config --set ssl_verify false
# R
$ conda create -n R_Seurat -c bioconda -c conda-forge r-Seurat r-irkernel
python=3.7.3

# Python and Bash
$ conda create -n scanpy python=3.7.3 ipykernel -c bioconda scanpy
$ conda activate scanpy
$ pip install bash_kernel
$ conda activate jupyter
$ $VSC_DATA/jupyter/envs/scanpy/bin/python -m bash_kernel.install
```

Connect to a server

- `qsub -I -l walltime=01:00:00 -A lp_bioinfworkshop6dec`
  - `qsub` # Submit a job to the cluster
  - `-I` # Request an interactive job
  - `-l walltime=01:00:00` # For 1 hour
  - `-A lp_bioinfworkshop6dec` # Using credits from the `lp_bioinfworkshop6dec` account
- Do this from a tmux/screen session, from a login-node!
- Nodes can be shared using a group_list
  - `-W group_list=lp_big_tier2` # Allow users from the group `lp_big_tier2` to also log in via ssh
  - Cannot exceed the resources requested.
  - This uses credits from the original submission!

Start JupyterLab

```
$ hostname  # We will use this to connect later
$ tmux  # This will let your jupyter session stay alive when you disconnect

$ conda activate jupyter

$ jupyter lab --port <vsc username number i.e. 30922>
              --no-browser --ip=0.0.0.0

# Press control-B (enters command mode), then press D (detach)
# To reconnect to tmux, tmux attach
```

```
 _____                 _
|  __ \               (_)
| |  \/  ___  _ __    _  _   _  ___
| | __  / _ \| '_ \ | || | | |/ __|
| |_\ \|  __/| | | || || |_| |\__ \
 \____/ \___||_| |_||_| \__,_||___/


Informatie over deze server (FQDN): r23i27n23.genius.hpc.kuleuven.be
 * cluster: genius
 * role: compute
 * hardware: ProLiant XL230k Gen10 (x86_64)
 * os: CentOS 7.6.1810
 * kernel: 3.10.0-957.10.1.el7.x86_64

 vsc30922@r23i27n23 [ ~ ]     conda activate jupyter
(jupyter)  vsc30922@r23i27n23 [ ~ ]     jupyter lab --port 30922 --no-browser --ip=0.0.0.0
[I 20:48:11.295 LabApp] [nb_conda_kernels] enabled, 2 kernels found
[I 20:48:12.135 LabApp] JupyterLab extension loaded from /data/leuven/309/vsc30922/jupyter/envs/jupyter/lib/python3.7/site-packages/jupyterlab
[I 20:48:12.135 LabApp] JupyterLab application directory is /data/leuven/309/vsc30922/jupyter/envs/jupyter/share/jupyter/lab
[I 20:48:12.137 LabApp] Serving notebooks from local directory: /vsc-hard-mounts/leuven-user/309/vsc30922
[I 20:48:12.137 LabApp] The Jupyter Notebook is running at:
[I 20:48:12.137 LabApp] http://r23i27n23:30922/?token=0b03426406170415dede05211ca2a094cec9586bc76e2ca9
[I 20:48:12.137 LabApp]  or http://127.0.0.1:30922/?token=0b03426406170415dede05211ca2a094cec9586bc76e2ca9
[I 20:48:12.137 LabApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 20:48:12.141 LabApp]

    To access the notebook, open this file in a browser:
        file:///run/user/2530922/jupyter/nbserver-22080-open.html
    Or copy and paste one of these URLs:
        http://r23i27n23:30922/?token=0b03426406170415dede05211ca2a094cec9586bc76e2ca9
     or http://127.0.0.1:30922/?token=0b03426406170415dede05211ca2a094cec9586bc76e2ca9
```

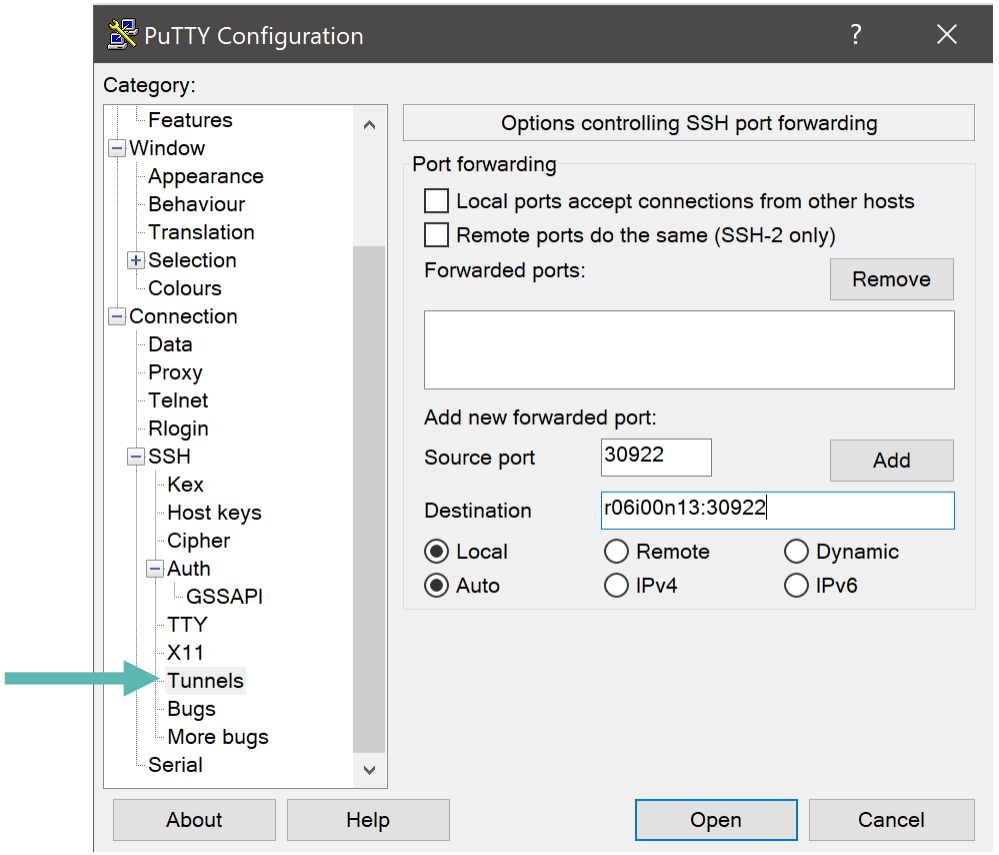
VIB-KU LEUVEN CENTER FOR BRAIN & DISEASE RESEARCH

KU LEUVEN

SCIENCE MEETS LIFE

PuTTY (Windows)

```
1.ssh r04i02n09
2.ssh r04i02n10
3.ssh r04i02n11
4.ssh r04i02n12
```



VIB-KU LEUVEN CENTER FOR BRAIN & DISEASE RESEARCH

KU LEUVEN

SCIENCE MEETS LIFE

MobaSSH (Windows)

```
1. ssh r04i02n09
2. ssh r04i02n10
3. ssh r04i02n11
4. ssh r04i02n12
```

- Session -> SSH
  - Remote Host: `r04i02n09` <- Change to server you want you use
  - Specify Username: vsc username (i.e. vsc30922)
  - Network settings
    - Tick connect through gateway
    - Gateway SSH server: login5-tier2.hpc.kuleuven.be
    - Username: vsc username
- Tunneling
  - New tunnel
    - Forwarded port: vsc username number (i.e. 30922)
    - ssh server: login5-tier2.hpc.kuleuven.be
    - Remote server: `r04i02n09` <- Change to server you want you use
    - Remote port: vsc username number
  - Save and name Jupyter

MacOS or Ubuntu

```
1. ssh r04i02n09
2. ssh r04i02n10
3. ssh r04i02n11
4. ssh r04i02n12
```

- Use built in terminal for SSH

- `ssh vsc30922@ login5-tier2.hpc.kuleuven.be -L30922:r04i02n09:30922`

Session settings

SSH | Telnet | Rsh | Xdmcp | RDP | VNC | FTP | SFTP | Serial | File | Shell | Browser | Mosh | Aws S3 | WSL

Basic SSH settings

Remote host * r23i27n23 | Specify username vsc30922 | Port 22

Advanced SSH settings | Terminal settings | Network settings | Bookmark settings

Connect through SSH gateway (jump host)

Gateway SSH server login-thinking.hpc.kule | Port 22 | User vsc30922

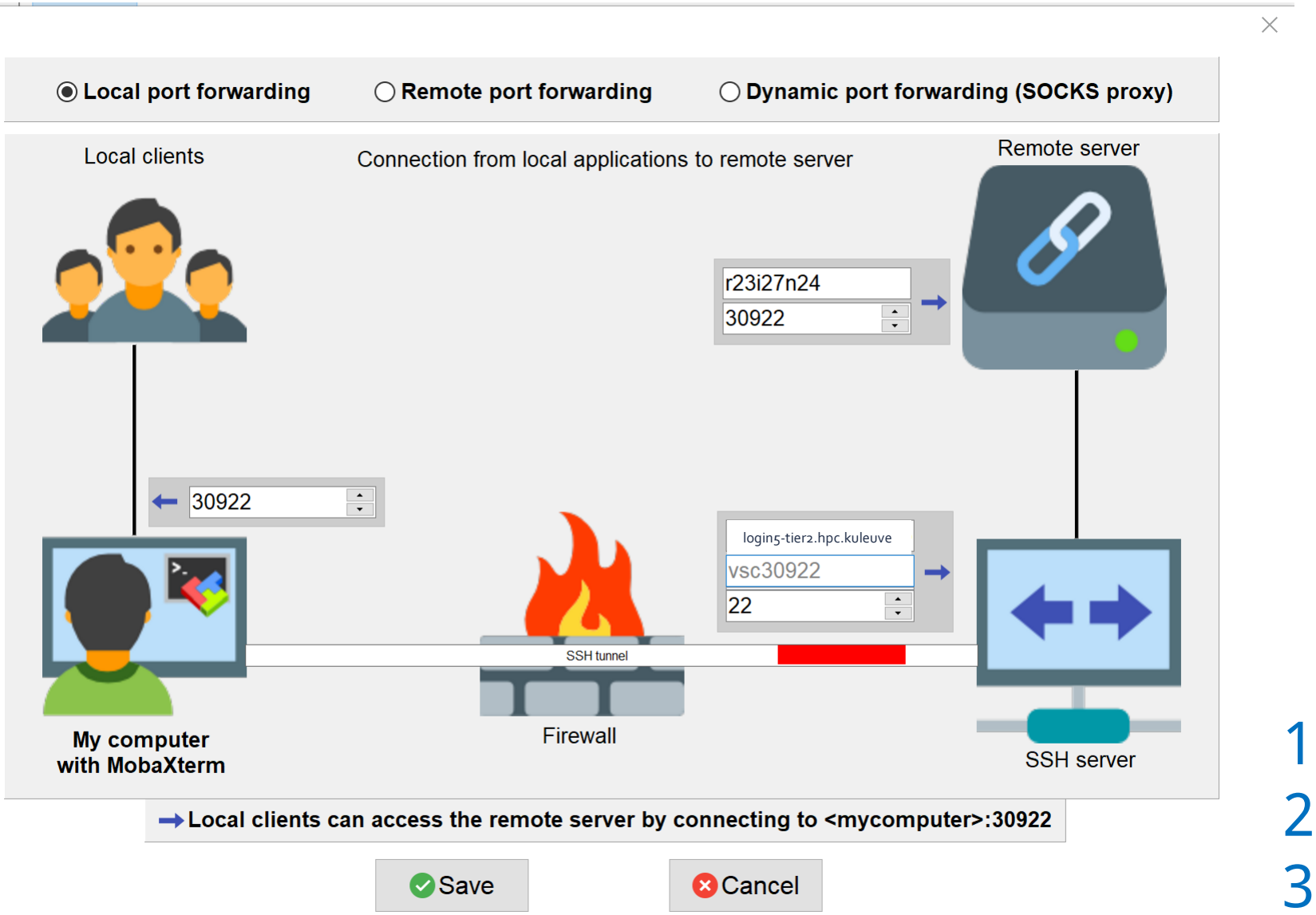
Use private key

Proxy settings (experimental)

Proxy type: None | Host: | Login: | Port: 1080

OK | Cancel

```
1.ssh r04i02n09
2.ssh r04i02n10
3.ssh r04i02n11
4.ssh r04i02n12
```




```
1.ssh r04i02n09
2.ssh r04i02n10
3.ssh r04i02n11
4.ssh r04i02n12
```

VIB-KU LEUVEN CENTER FOR BRAIN & DISEASE RESEARCH

KU LEUVEN

SCIENCE MEETS LIFE

MobaSSHTunnel

Welcome to MobaSSHTunnel - Graphical port forwarding tool

Name	Type	Start/stop	Forward port	Destination server	SSH server	Settings
Jupyter	Local		30922	r23i27n24:30922	vsc30922@ login-thinking.hpc.kuleuven.be:22	

```
1.ssh r04i02n09
2.ssh r04i02n10
3.ssh r04i02n11
4.ssh r04i02n12
```

New SSH tunnel | Start all tunnels | Stop all tunnels | Exit

```
 _____                 _
|  __ \               (_)
| |  \/  ___  _ __    _  _   _  ___
| | __  / _ \| '_ \ | || | | |/ __|
| |_\ \|  __/| | | || || |_| |\__ \
 \____/ \___||_| |_||_| \__,_||___/


Informatie over deze server (FQDN): r23i27n23.genius.hpc.kuleuven.be
 * cluster: genius
 * role: compute
 * hardware: ProLiant XL230k Gen10 (x86_64)
 * os: CentOS 7.6.1810
 * kernel: 3.10.0-957.10.1.el7.x86_64

 vsc30922@r23i27n23 [ ~ ]     conda activate jupyter
(jupyter)  vsc30922@r23i27n23 [ ~ ]     jupyter lab --port 30922 --no-browser --ip=0.0.0.0
[I 20:48:11.295 LabApp] [nb_conda_kernels] enabled, 2 kernels found
[I 20:48:12.135 LabApp] JupyterLab extension loaded from /data/leuven/309/vsc30922/jupyter/envs/jupyter/lib/python3.7/site-packages/jupyterlab
[I 20:48:12.135 LabApp] JupyterLab application directory is /data/leuven/309/vsc30922/jupyter/envs/jupyter/share/jupyter/lab
[I 20:48:12.137 LabApp] Serving notebooks from local directory: /vsc-hard-mounts/leuven-user/309/vsc30922
[I 20:48:12.137 LabApp] The Jupyter Notebook is running at:
[I 20:48:12.137 LabApp] http://r23i27n23:30922/?token=0b03426406170415dede05211ca2a094cec9586bc76e2ca9
[I 20:48:12.137 LabApp]  or http://127.0.0.1:30922/?token=0b03426406170415dede05211ca2a094cec9586bc76e2ca9
[I 20:48:12.137 LabApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 20:48:12.141 LabApp]

    To access the notebook, open this file in a browser:
        file:///run/user/2530922/jupyter/nbserver-22080-open.html
    Or copy and paste one of these URLs:
        http://r23i27n23:30922/?token=0b03426406170415dede05211ca2a094cec9586bc76e2ca9
     or http://127.0.0.1:30922/?token=0b03426406170415dede05211ca2a094cec9586bc76e2ca9
```

VIB-KU LEUVEN CENTER FOR BRAIN & DISEASE RESEARCH

KU LEUVEN

SCIENCE MEETS LIFE

File Edit View Run Kernel Tabs Settings Help

/

Name	Last Modified
01.CLEAN_FASTQ	a month ago
10.MAPPING	a month ago
20.COUNTS	a month ago
cache	3 months ago
data	3 years ago
FlyBrain_Data	2 years ago
git	15 hours ago
kea2	4 years ago
Music	5 years ago
Old	7 months ago
perl5	9 months ago
Pictures	5 years ago
Public	5 years ago
R	3 months ago
scratch	4 years ago
seaborn-data	24 days ago
Software	a year ago
Solo.out	6 months ago
stg_00002	a year ago
stg_00003	5 months ago
tmp	16 days ago
Volumes	2 years ago
acl.sh	3 months ago
AddLevelsToLoom.py	a year ago
celeryconfig.py	2 years ago
core.8856	2 months ago
HCA_Pancreas_2515cells.loom	2 months ago
inbox	2 days ago
inode.pbs.e50086847	a month ago
inode.pbs.e50087359	a month ago
inode.pbs.e50087888	a month ago

Launcher

Notebook

Python 3 | Bash | Python [conda env:jupyter] * | Python [conda env:python] | R [conda env:r_seurat]

Console

Python 3 | Bash | Python [conda env:jupyter] * | Python [conda env:python] | R [conda env:r_seurat]

Other

Terminal | Text File | Markdown File | Contextual Help

0 1 Launcher

VIB-KU LEUVEN CENTER FOR BRAIN & DISEASE RESEARCH

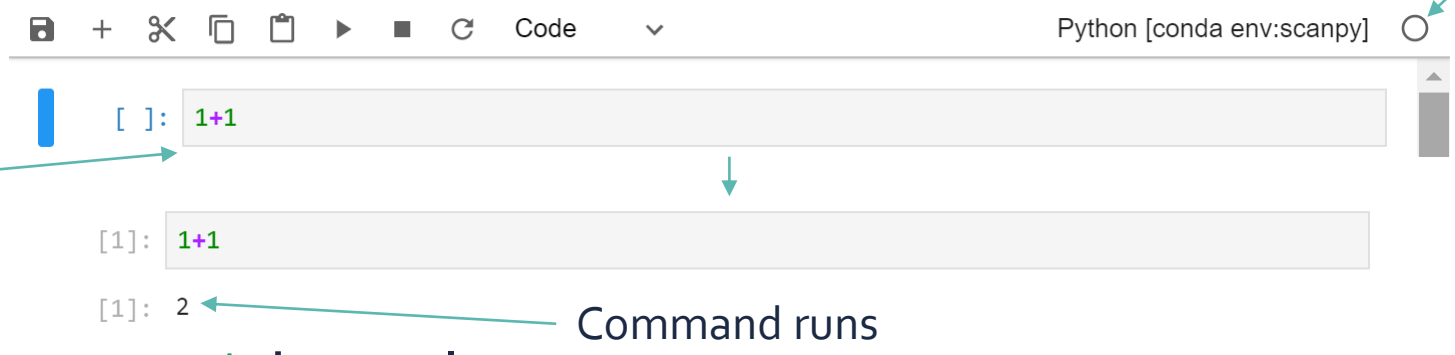
KU LEUVEN

SCIENCE MEETS LIFE

Make sure it works!

- Select the `scanpy` kernel

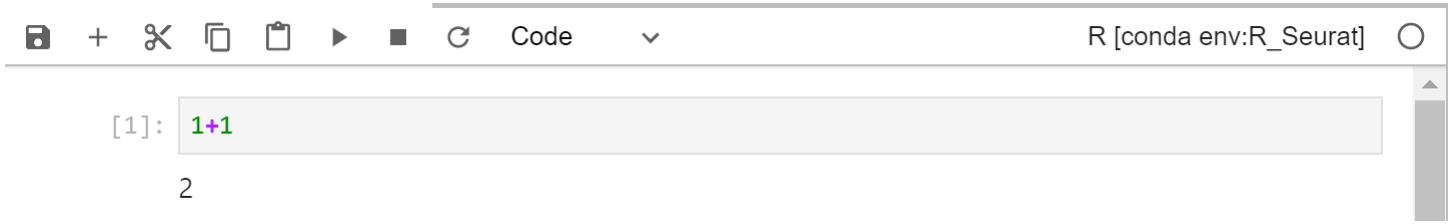
Kernel is ready and idle



Syntax highlighting works

Command runs

- Select the `seurat` kernel



JupyterLab Tips and Tricks

- Many extensions available. Make sure you're in your jupyter conda environment
  - https://github.com/lckr/jupyterlab-variableInspector (Currently doesn't install)
  - https://github.com/jupyterlab/jupyterlab-git
  - https://github.com/jupyter-widgets/ipywidgets
- Right click -> Jupyter menu
  - Contextual help -> Shows help for current function
  - New console for notebook -> Gives a console, like Rstudio for working within same environment as notebook
  - Shift + right click for system menu -> Copy, paste etc.

Basic Single Cell Analysis

Kris Davie

VIB-CBD

Bioinformatics Expertise Unit Leader

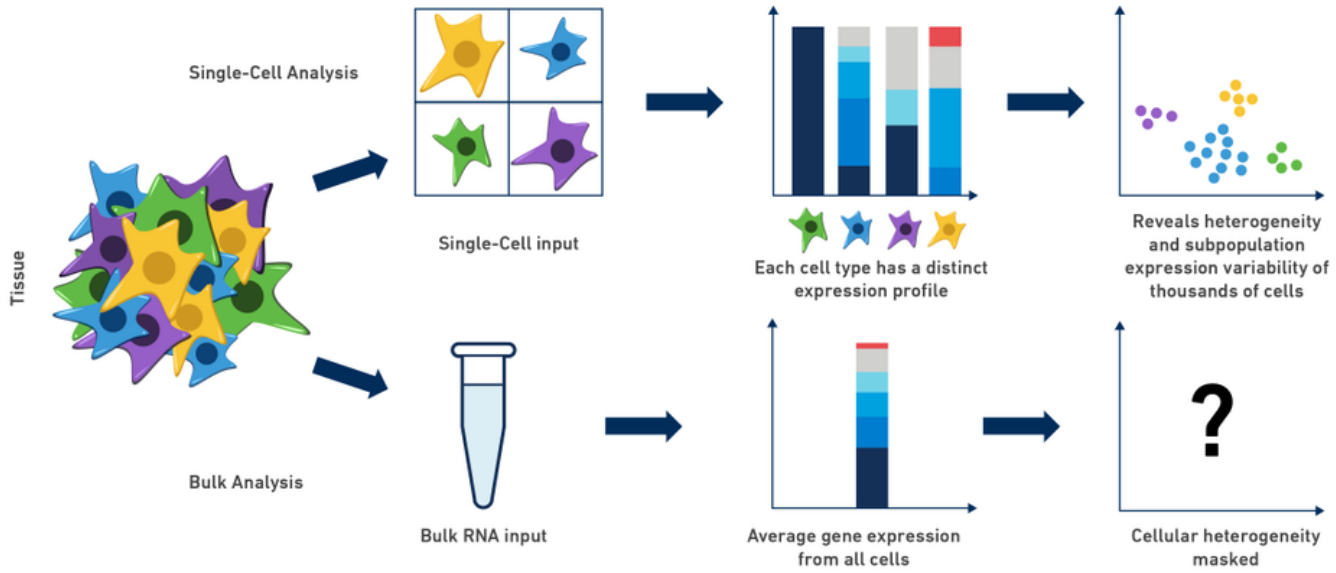
06/12/2019

VIB-KU LEUVEN CENTER FOR BRAIN & DISEASE RESEARCH

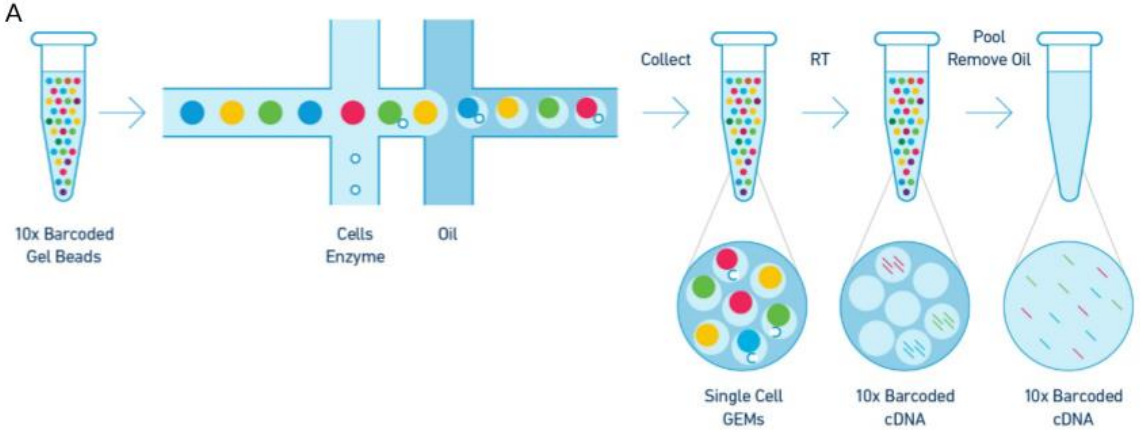
KU LEUVEN

SCIENCE MEETS LIFE

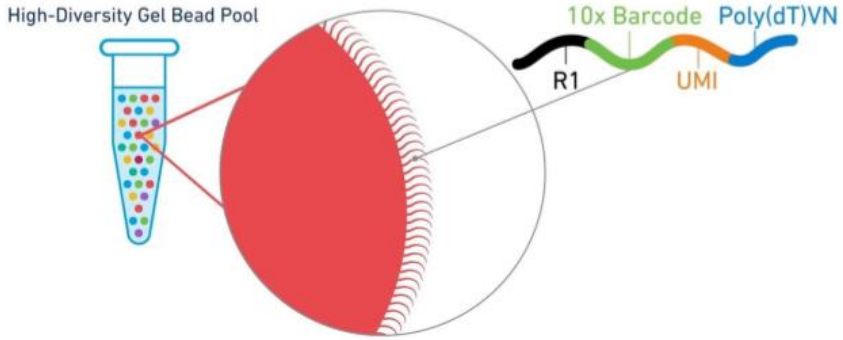
Why single cell?

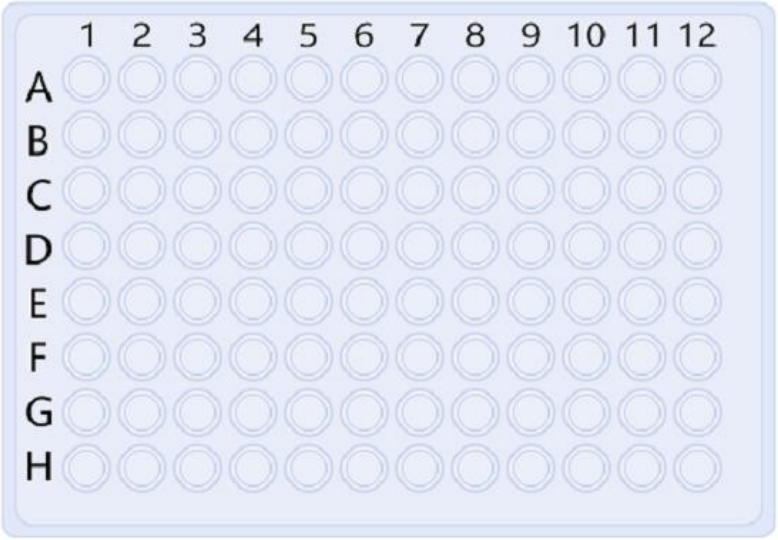


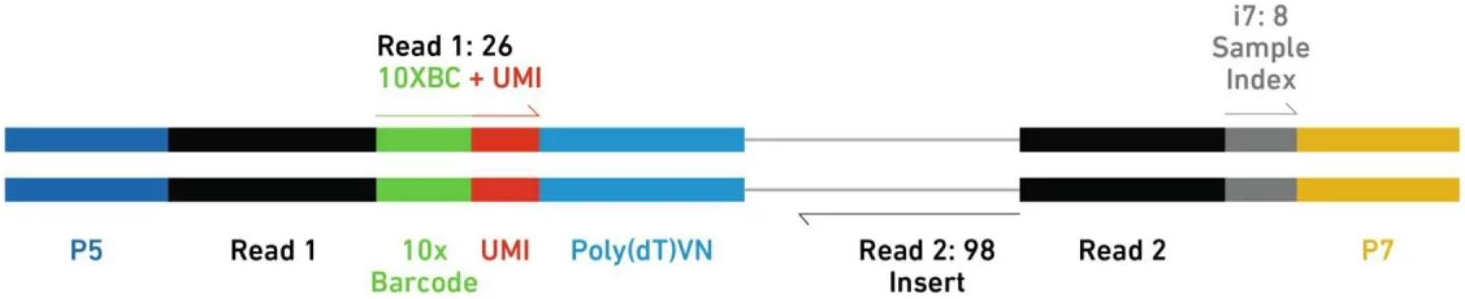
A



B

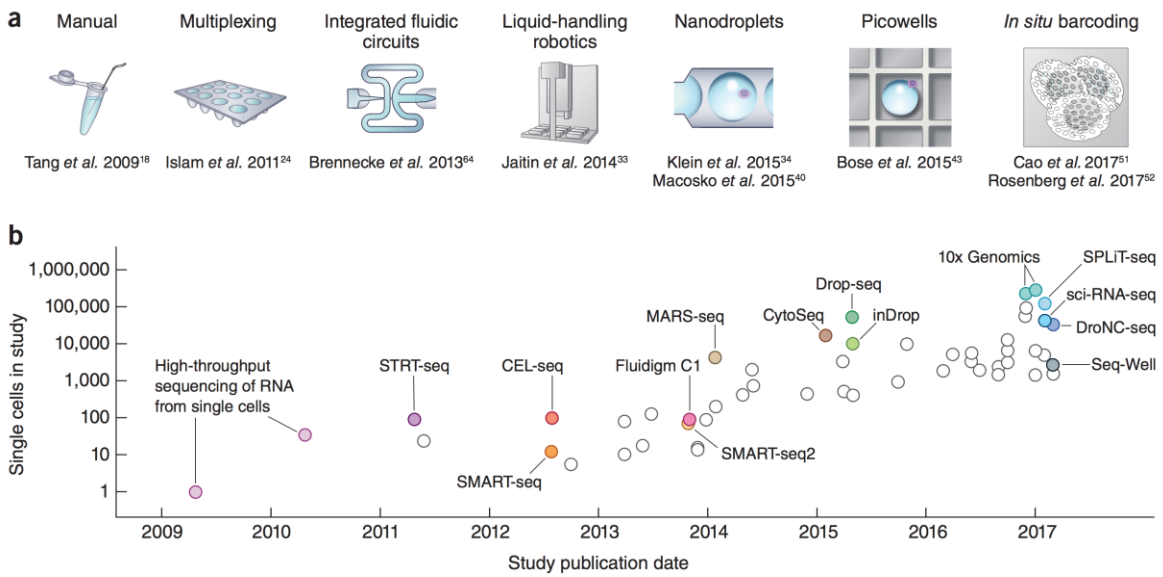




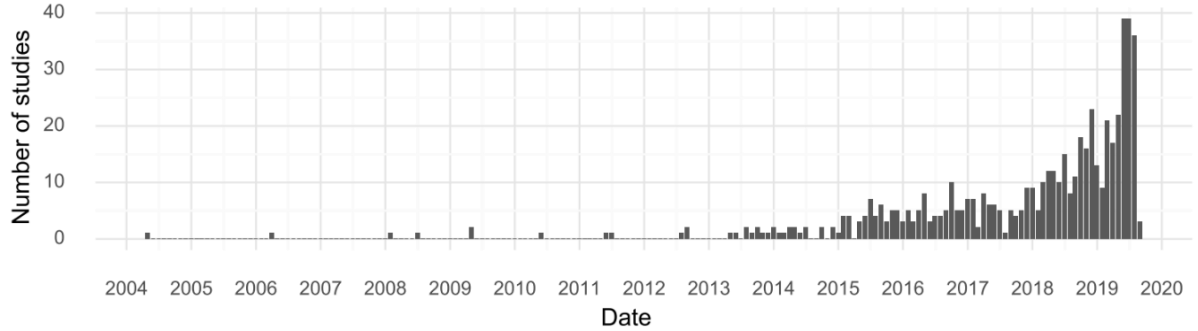


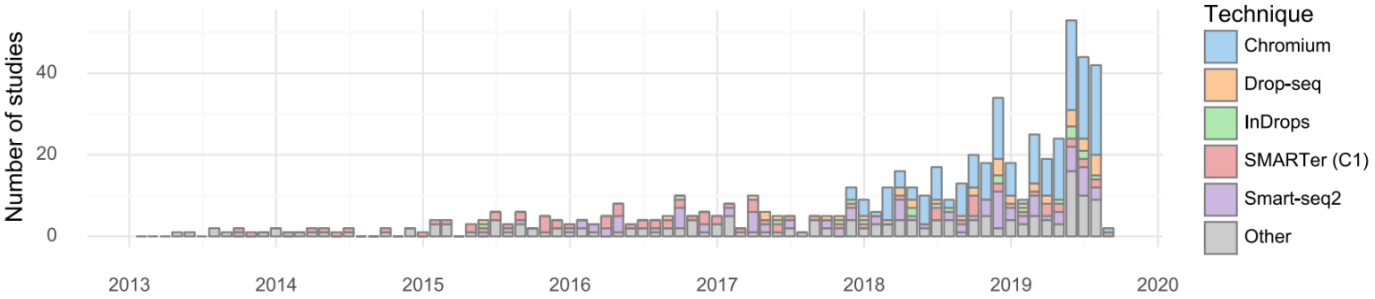
SCIENCE MEETS LIFE

1)



2)





Month	Studies	Median cells
Jan 2019	9	3,368
Feb 2019	21	11,175
Mar 2019	16	11,452
Apr 2019	21	17,725
May 2019	39	14,585
Jun 2019	39	15,000
Jul 2019	36	13,966

Tissue	Studies
Brain	64
Culture	47
Blood	16
Heart	16
Pancreas	16
Embryo	14
Lung	12

Journal	Studies
bioRxiv	63
Nature	50
Cell	49
Cell Reports	35
Science	34
Nature Communications	29
Genome Biology	19

VIB-KU LEUVEN CENTER FOR BRAIN & DISEASE RESEARCH

KU LEUVEN

SCIENCE MEETS LIFE

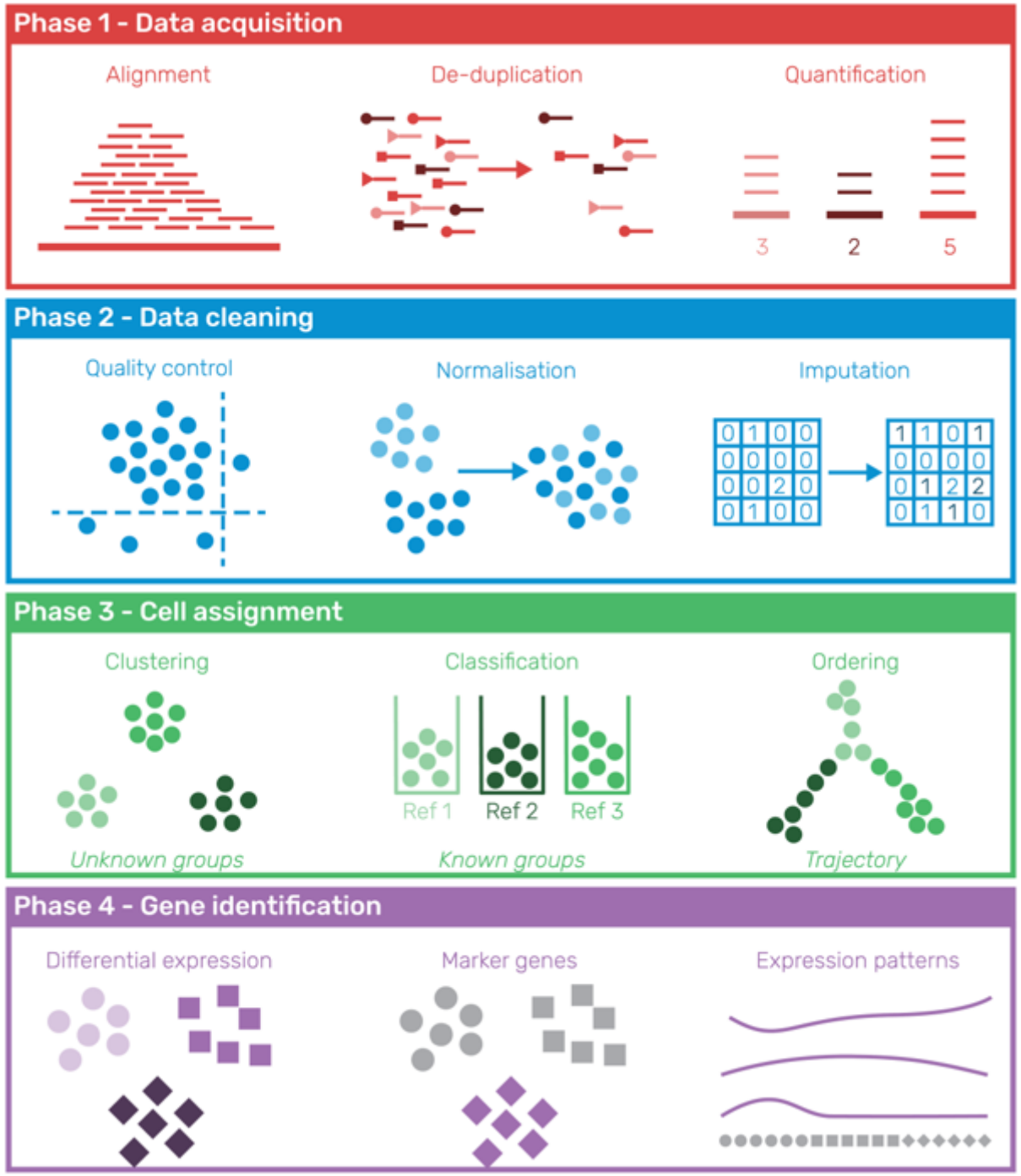
1. Exponential scaling of single-cell RNA-seq in the past decade. Svensson, Vento-Tormo & Teichmann, **Nature Protocols, Mar 2018**
2. A curated database reveals trends in single cell transcriptomics. Svensson & de Veiga Beltrame, **bioRxiv, Aug 2019**

STAR/STARsolo[B]
Kallisto[B]
HISAT2[B]
CellRanger[B/P]

scran[R]
Scater[R]
scTransform[R]
scPrep[P]

louvain[R/P]
Garnett[R]
scMatch[P]
Dynverse[R/P]

SCDE[R]
DESeq2[R]
MAST[R]



CellRanger

Scanpy[P]
Seurat[R]

VIB-KU LEUVEN CENTER FOR BRAIN & DISEASE RESEARCH

KU LEUVEN

Zappia, Phipson & Oshlack. **PLOS Computational Biology 2018**

Phase 1 – Mapping/Quantification

- Very computationally intensive, not for today, but multiple options
- 10X genomics – CellRanger
  - `module load CellRanger/3.1.0 or`
  - `singularity exec -B /staging/:/staging/,/ddn1/:/ddn1/ /staging/leuven/res_00001/software/CellRanger/3.1.0/CellRanger_3.1.0.sif cellranger`

-
```
cellranger mkfastq \
        --id=PROJECT_NAME \
        --run=/path/to/SEQUENCER_OUTPUT \
        --csv =/path/to/SAMPLESHEET \
        --jobmode=local \
        --localcores=NUM_CORES \
        --localmem=MEMORY
```

-
```
cellranger count \
  --id=PROJECT_NAME \
  --sample=SAMPLE_NAME \
  --fastqs=/path/to/MKFASTQ_OUTPUT \
  --transcriptome =/path/to/REFERENCE \
  --jobmode=local \
  --localcores=NUM_CORES \
  --localmem=MEMORY
```

Phase 1 – Mapping/Quanitification

- Very computationally intensive, not for today, but multiple options

- Other scRNA – STAR(solo)
  - `module load STAR/2.7.1a-foss-2018a or`
  - `singularity exec -B /staging/:/staging/,/ddn1/:/ddn1/ /staging/leuven/res_00001/software/STAR/2.7.1a/STAR_2.7.1a.sif STAR`

- STAR supports 10X-like libraries and others e.g. SMART-seq2/CEL-seq2 etc.

SMART-seq like (1 file = 1 cell)

```
singularity run -B /staging/:/staging/,/ddn1/:/ddn1/
/staging/leuven/res_00001/software/STAR/2.7.1a/STAR_2.7.1a.sif \
        --outSAMtype BAM SortedByCoordinate \
        --runThreadN NUM_CORES \
        --genomeDir /path/to/REFERENCE \
        --genomeLoad LoadAndKeep \
        --limitBAMsortRAM 50000000000 \
        --readFilesIn /path/to/FASTQ_FILE.gz \
        --readFilesCommand zcat \
        --outFileNamePrefix /path/to/OUTPUT_ \
        --quantMode GeneCounts \
        --outReadsUnmapped Fastx &> /dev/null
```

10X like (1 file = many barcoded cells)

```
singularity run -B /staging/:/staging/,/ddn1/:/ddn1/
/staging/leuven/res_00001/software/STAR/2.7.1a/STAR_2.7.1a.sif \
        --soloType CB_UMI_Simple \
        --soloCBwhitelist /path/to/WHITELIST \
        --soloCBstart 1 \
        --soloCBlen 16 \
        --soloUMIstart 17 \
        --soloUMIlen 10 \
        --outSAMtype BAM SortedByCoordinate \
        --runThreadN NUM_CORES \
        --genomeDir /path/to/REFERENCE \
        --genomeLoad LoadAndKeep \
        --limitBAMsortRAM 50000000000 \
        --readFilesIn /path/to/FASTQ_FILE_1.gz /path/to/FASTQ_FILE_2.gz \
        --readFilesCommand zcat \
        --outFileNamePrefix /path/to/OUTPUT_ \
        --quantMode GeneCounts \
        --outReadsUnmapped Fastx &> /dev/null
```

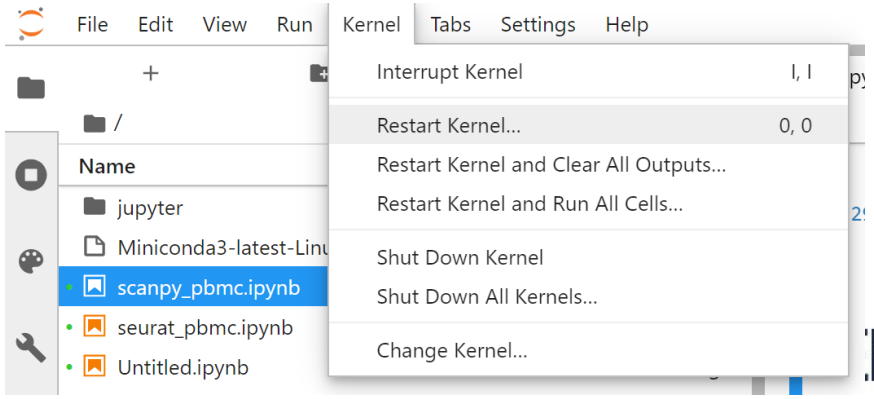
Phase 2-4 - Prep environment

Scanpy

Seurat

```
library(reticulate)
use_condaenv(condaenv="scanpy")

reticulate::py_install(packages =
'umap-learn')
```

RESTART KERNEL



Phase 2-4 - Prep environment

Scanpy

```
import numpy as np
import pandas as pd
import scanpy as sc


sc.settings.verbosity = 3
# verbosity: errors (0), warnings
(1), info (2), hints (3)
sc.logging.print_versions()


sc.settings.set_figure_params(dpi=80)
```

Seurat

```
library(dplyr)
library(Seurat)
library(reticulate)
use_condaenv(condaenv="scanpy")


reticulate::py_install(packages =
'umap-learn')
sessionInfo()
options(repr.plot.width=7,
repr.plot.height=7)
```

Phase 2-4 – Load Data

Scanpy

```
adata = sc.read_10x_mtx(
'/staging/leuven/res_00001/datasets/10X/pbmc_1k_v3/filtered_feature_bc_matrix/',
var_names='gene_symbols',
cache=False)

adata
```

```
[4]: View of AnnData object with n_obs × n_vars = 1222 × 33538
         var: 'gene_ids', 'feature_types'
```

Seurat

```
pbmc.data <- Read10X(data.dir =
"/staging/leuven/res_00001/datasets/10X/pbmc_1k_v3/filtered_feature_bc_matrix/")

pbmc <- CreateSeuratObject(counts =
pbmc.data, project = "pbmc1k")

pbmc
```

```
An object of class Seurat
33538 features across 1222 samples within 1 assay
Active assay: RNA (33538 features)
```

Phase 2-4 – Pre-filter

Scanpy

```
sc.pp.filter_cells(adata, min_genes=200)
sc.pp.filter_genes(adata, min_cells=3)

adata
```

```
[7]: AnnData object with n_obs × n_vars = 1176 × 15246
        obs: 'n_genes'
        var: 'gene_ids', 'feature_types', 'n_cells'
```

Seurat

```
pbmc <- CreateSeuratObject(
counts = pbmc.data,
project = "pbmc1k",
min.cells = 3,
min.features = 200)
```

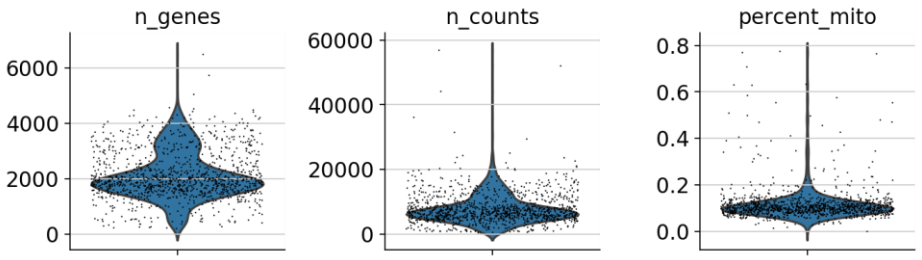
```
An object of class Seurat
15246 features across 1176 samples within 1 assay
Active assay: RNA (15246 features)
```

Phase 2-4 – Basic QC stats

Scanpy

```
mito_genes =
adata.var_names.str.startswith('MT-')
adata.obs['percent_mito'] =
np.sum(adata[:, mito_genes].X,
axis=1).A1 / np.sum(adata.X, axis=1).A1
adata.obs['n_counts'] =
adata.X.sum(axis=1).A1
```

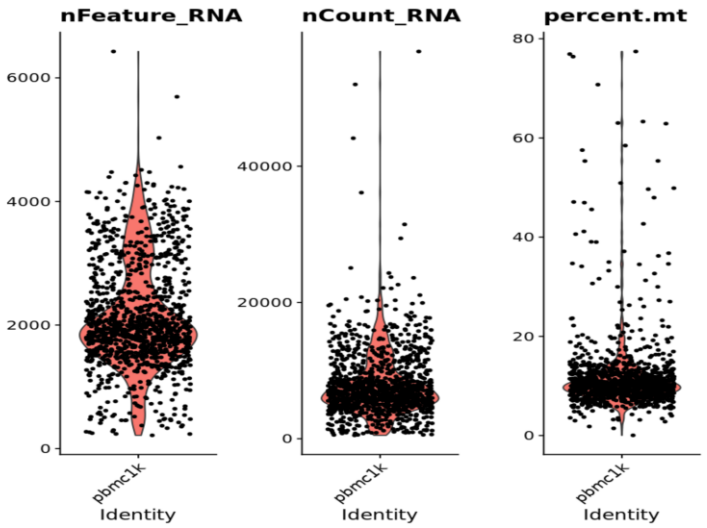
```
sc.pl.violin(adata, ['n_genes',
'n_counts', 'percent_mito'], jitter=0.4,
multi_panel=True)
```



Seurat

```
pbmc[["percent.mt"]] <-
PercentageFeatureSet(pbmc, pattern="^MT-")
```

```
VlnPlot(pbmc, features = c("nFeature_RNA",
"nCount_RNA", "percent.mt"), ncol = 3)
```

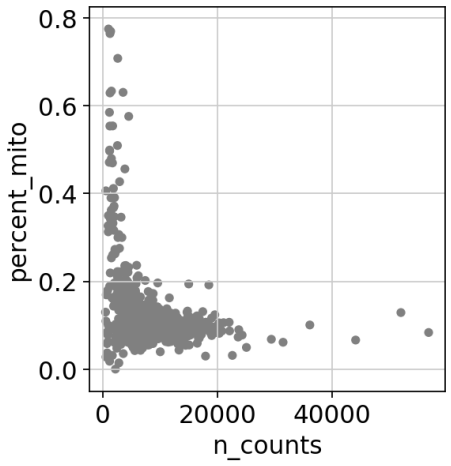


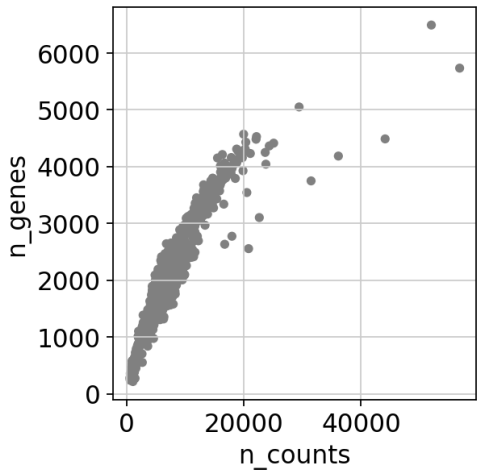
SCIENCE MEETS LIFE

Phase 2-4 – Basic QC stats

Scanpy

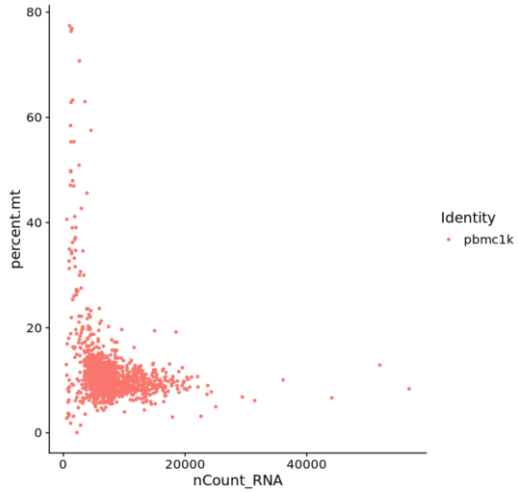
```
sc.pl.scatter(adata, x='n_counts', y='percent_mito')
sc.pl.scatter(adata, x='n_counts', y='n_genes')
```

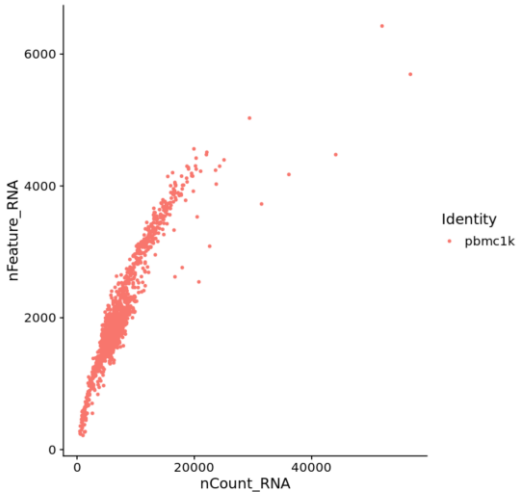




Seurat

```
FeatureScatter(pbmc, feature1 = "nCount_RNA", feature2 = "percent.mt")
FeatureScatter(pbmc, feature1 = "nCount_RNA", feature2 = "nFeature_RNA")
```



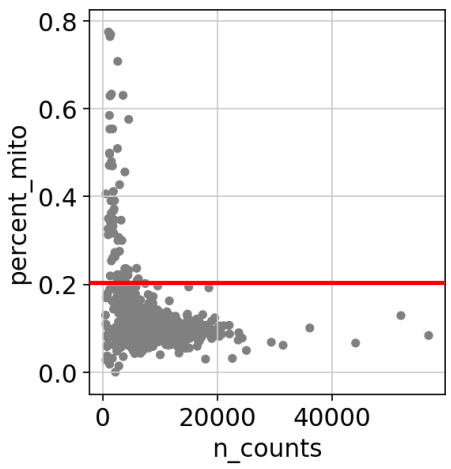


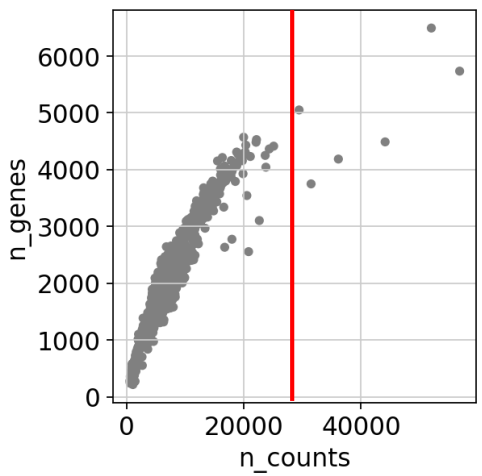
SCIENCE MEETS LIFE

Phase 2-4 – Basic QC Filter

Scanpy

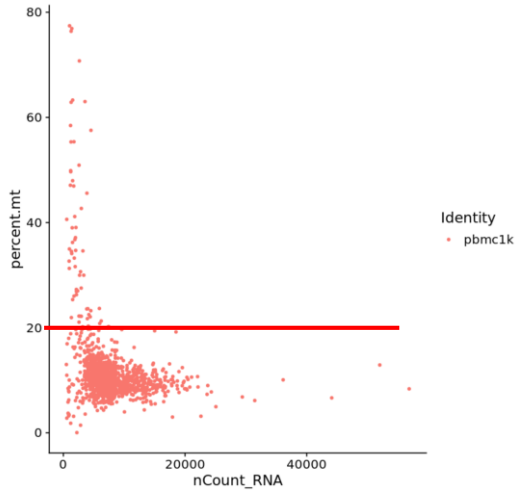
```
adata = adata[adata.obs['n_counts'] < 25000, :]
adata = adata[adata.obs['percent_mito'] < 0.20, :]
```

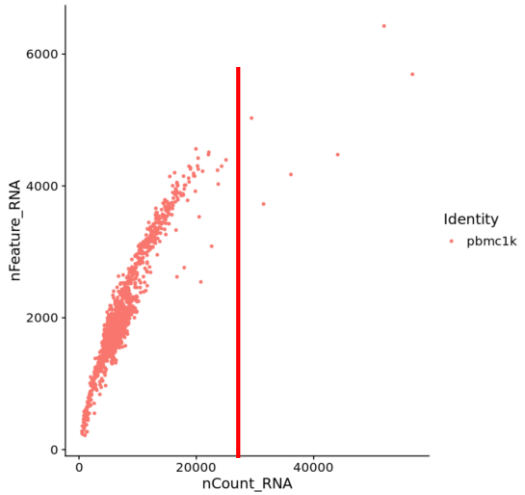




Seurat

```
pbmc <- subset(pbmc,
subset = nFeature_RNA > 200 & nCount_RNA < 25000 & percent.mt < 20)
```





SCIENCE MEETS LIFE

Phase 2-4 – Store Raw Data

Scanpy

```
adata.raw = adata
```

Seurat

```
Automatically done
```

- Why?
- Some functions (differential expression) work more reliably on raw data
- Others (dimensionality reduction) work better on normalized data

Phase 2-4 – Scaling/Normalizing/HVGs

Scanpy

```
sc.pp.normalize_per_cell(adata,
counts_per_cell_after=1e4)

sc.pp.log1p(adata)

sc.pp.highly_variable_genes(adata,
min_mean=0.0125,
max_mean=3,
min_disp=0.5)

adata
```

```
[20]: AnnData object with n_obs × n_vars = 1106 × 15246
          obs: 'n_genes', 'percent_mito', 'n_counts'
          var: 'gene_ids', 'feature_types', 'n_cells', 'highly_variable', 'means', 'dispe
      rsions', 'dispersions_norm'
```

Seurat

```
pbmc <- NormalizeData(pbmc,
normalization.method = "LogNormalize",
scale.factor = 10000)
pbmc <- FindVariableFeatures(pbmc,
selection.method = "vst",
nfeatures = 2000)

pbmc
```

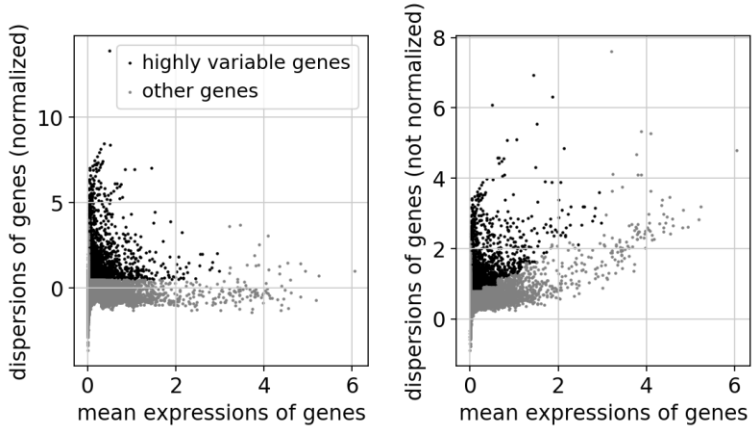
```
An object of class Seurat
15246 features across 1106 samples within 1 assay
Active assay: RNA (15246 features)
```

SCIENCE MEETS LIFE

Phase 2-4 – Scaling/Normalizing/HVGs

Scanpy

```
sc.pl.highly_variable_genes(adata)
```

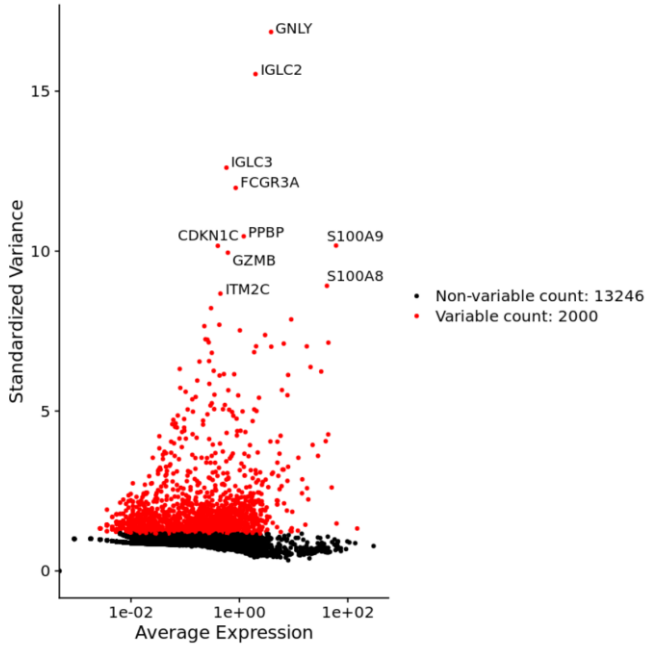



```
adata = adata[:,
adata.var['highly_variable']]

adata
```

```
[22]: View of AnnData object with n_obs × n_vars = 1106 × 2316
      obs: 'n_genes', 'percent_mito', 'n_counts'
      var: 'gene_ids', 'feature_types', 'n_cells', 'highly_variable', 'means', 'dispe
rsions', 'dispersions_norm'
```

Seurat

```
top10 <- head(VariableFeatures(pbmc), 10)

plot1 <- VariableFeaturePlot(pbmc)
LabelPoints(plot = plot1, points = top10,
repel = TRUE)
```

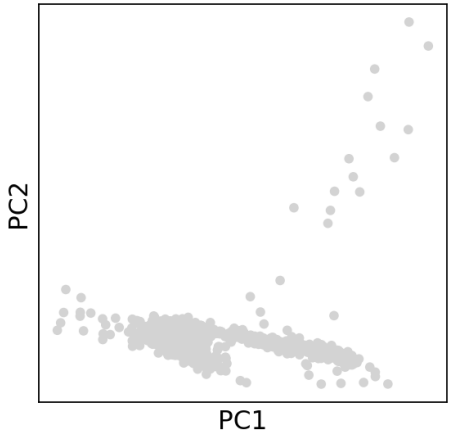


SCIENCE MEETS LIFE

Phase 2-4 – Scaling and PCA

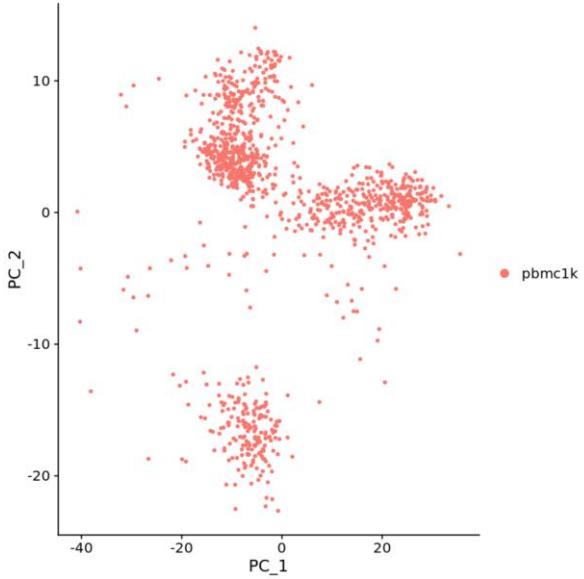
Scanpy

```
sc.pp.regress_out(adata, ['n_counts', 'percent_mito'])
sc.pp.scale(adata, max_value=10)

sc.tl.pca(adata, svd_solver='arpack')
sc.pl.pca(adata)
```



Seurat

```
pbmc <- ScaleData(pbmc,
vars.to.regress = c("percent.mt",
"nCount_RNA"))
pbmc <- RunPCA(pbmc, features =
VariableFeatures(object = pbmc))
DimPlot(pbmc, reduction = "pca")
```



VIB-KU LEUVEN CENTER FOR BRAIN & DISEASE RESEARCH

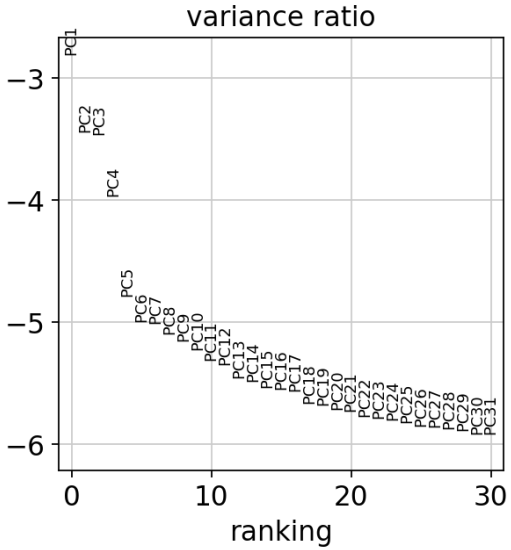
KU LEUVEN

SCIENCE MEETS LIFE

Phase 2-4 – Scaling and PCA

Scanpy

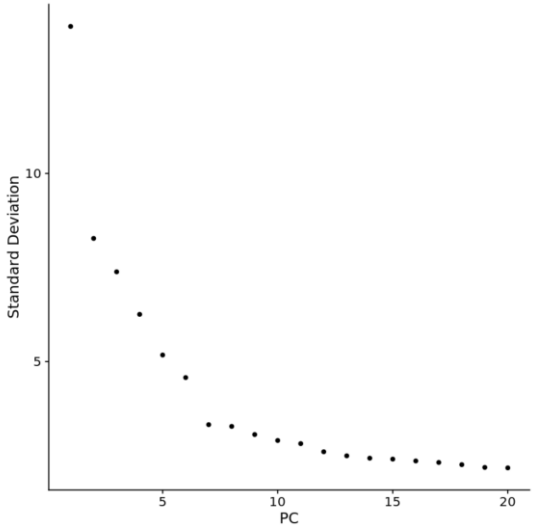
```
sc.pl.pca_variance_ratio(adata, log=True)
```




```
sc.pp.neighbors(adata, n_neighbors=10, n_pcs=12)
```

Seurat

```
ElbowPlot(pbmc)
```




```
pbmc <- FindNeighbors(pbmc, k.param=10, dims = 1:11)
```

VIB-KU LEUVEN CENTER FOR BRAIN & DISEASE RESEARCH

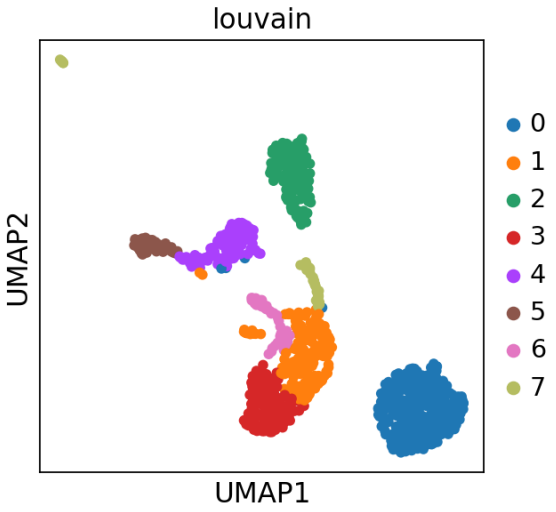
KU LEUVEN

SCIENCE MEETS LIFE

Phase 2-4 – Clusters and UMAP/tSNE

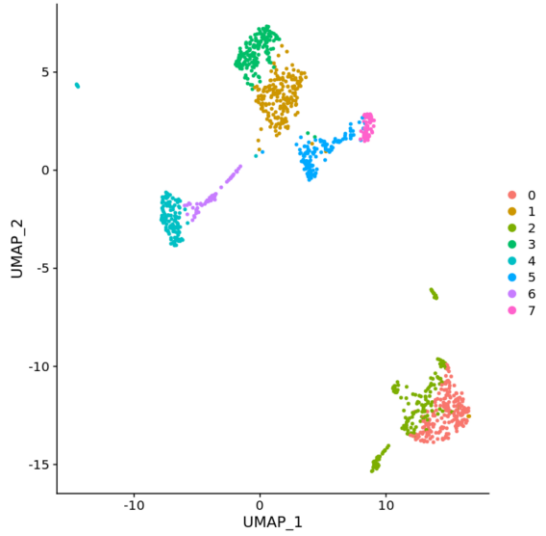
Scanpy

```
sc.tl.louvain(adata, resolution=0.5)
sc.tl.umap(adata)

sc.pl.umap(adata, color=['louvain'])
```



Seurat

```
pbmc <- FindClusters(pbmc, resolution = 0.5)
pbmc <- RunUMAP(pbmc, dims = 1:11)

DimPlot(pbmc, reduction = "umap")
```



VIB-KU LEUVEN CENTER FOR BRAIN & DISEASE RESEARCH

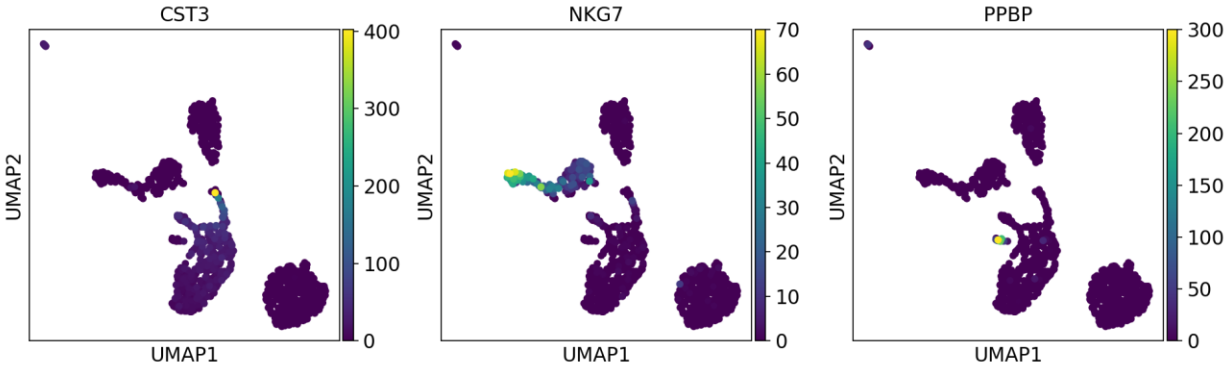
KU LEUVEN

SCIENCE MEETS LIFE

Phase 2-4 – Plotting Genes

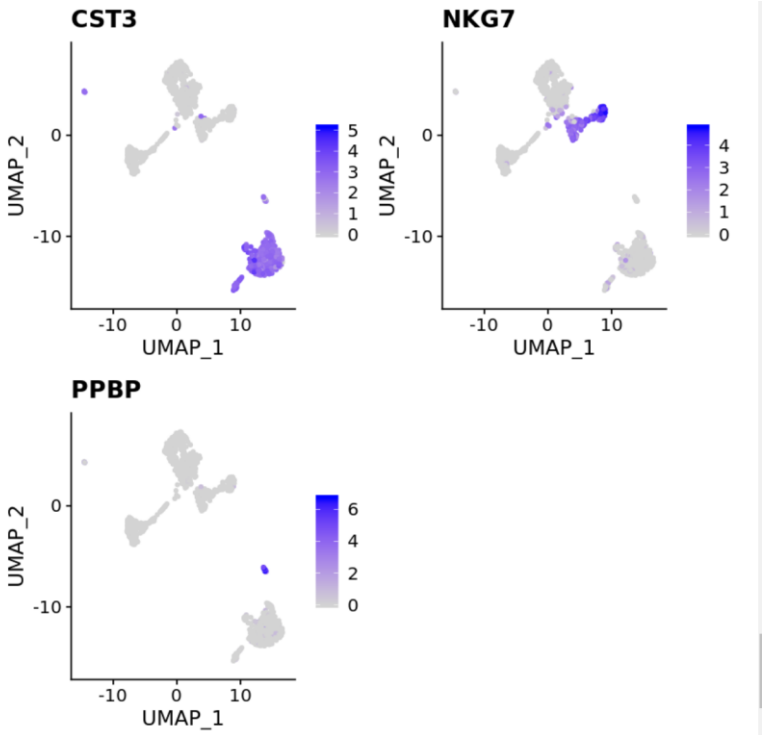
Scanpy

```
sc.pl.umap(adata, color=['CST3', 'NKG7', 'PPBP'])
```



Seurat

```
FeaturePlot(pbmc, features = c('CST3', 'NKG7', 'PPBP'))
```



VIB-KU LEUVEN CENTER FOR BRAIN & DISEASE RESEARCH

KU LEUVEN

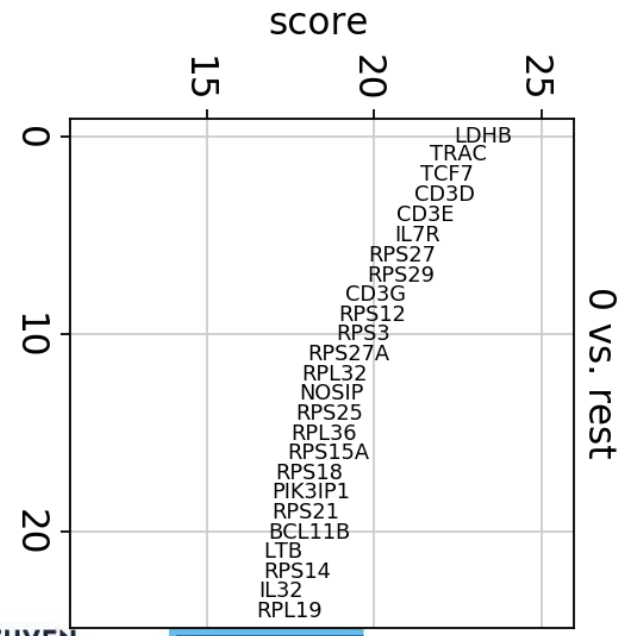
SCIENCE MEETS LIFE

Phase 2-4 – Marker Genes

Scanpy

```
sc.tl.rank_genes_groups(adata, 'louvain', method='wilcoxon')
```

```
sc.pl.rank_genes_groups(adata, n_genes=25, sharey=False)
```



Seurat

```
cluster1.markers <- FindMarkers(pbmc, ident.1 = 1, min.pct = 0.25)
```

```
head(cluster1.markers, n = 5)
```

	p_val	avg_logFC	pct.1	pct.2	p_val_adj
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
IL7R	1.833519e-74	1.2522889	0.939	0.275	2.795383e-70
IL32	1.702048e-66	1.0978864	0.970	0.308	2.594942e-62
TRAC	8.655482e-66	1.0224241	0.975	0.357	1.319615e-61
LDHB	1.952684e-59	0.8850129	0.990	0.704	2.977062e-55
CD3D	3.168628e-58	0.8645340	0.960	0.309	4.830891e-54

```
pbmc.markers <- FindAllMarkers(pbmc, only.pos = TRUE, min.pct = 0.25, logfc.threshold = 0.25)
```

Further Reading

- https://scanpy-tutorials.readthedocs.io/en/latest/pbmc3k.html
- https://satijalab.org/seurat/v3.1/pbmc3k_tutorial.html

- **Current best practices in single-cell RNA-seq analysis: a tutorial**
  - Luecken and Theis, MSB 2019

- Single cell viewer
- http://scope.aertslab.org/
- https://github.com/aertslab/SCopeLoomR