

# Day1 Single cell RNA-seq

NIEHS Workshop--Analyzing NGS data  
[https://github.com/zhu16/NIEHS\\_NGS\\_Workshop](https://github.com/zhu16/NIEHS_NGS_Workshop)

July 13, 2021

9am-2pm

Yunhua Zhu, PhD  
zhanghua@gmail.com  
Computational Genomics Specialist, Transcriptomics  
Medical Science & Computing

# Instructor's background

- Bachelor @ National University of Singapore(NUS) in Biochemistry 2000-03
- Ph.D @ NUS in stem cell biology | 2006 -10
- Postdoc @ Hopkins with wet lab & dry lab | 2014 - 2019
  - Neurogenesis w/t single-cell RNA-seq
  - Neurodegeneration w/t single-nucleus RNA-seq
  - Learning R, Bash script statistics through Google and Youtube
- Computational Genomics Specialist @ BCBB | 2019
  - Single-cell RNA-seq, CITE-seq
    - SMARTseq2 pipeline with FASTQC, RSEM, SENIC
    - 10X genomics pipeline with Cellranger, Seurat, Scanpy and RNA Velocity.
  - Bulk RNA-seq
    - Deconvolution Bulk RNA-seq using ABIS and CybersortX.
    - Functional annotation with clusterProfiler.
  - Epigenetics
    - ATAC-seq and Chip-seq



# Aims of the workshop

- Target audience: NIEHS researchers who want to learn NGS data analysis.
- Aims: to provide an overview of NGS analysis workflow for scRNA-seq, bulk RNA-seq and ChIP-seq.
- Prerequisite: comfortable with shell scripts, R and python codes.
- We will use biowulf HPC as the computation environment, and use Jupyter lab as an interface
- The github repository is for illustration only. Data files are included in Biowulf folders for use in the class.
- The README.md contains information to set up the workplace.
- This is 2-day (4hr each day) workshop on NGS data analysis.
  - On the 1st day, we will focus on single cell RNA-seq. I will start from NGS\_Workshop\_Day1.pdf from the resources folder, do a ~30mins presentation.
  - This will help setting up the working environment and get background knowledge on single-cell RNA-seq.
  - The presentation will be followed by practical exercises in the scRNA-seq/notebooks folder.
  - On the 2nd day, we will discuss Bulk RNA-seq and ChIP-seq. I will start from the NGS\_Workshop\_Day2.pdf file to do a presentation, which will be followed by exercises in the bulk-RNA and ChIP-seq folder.
- **We will use jupyter lab interface on Biowulf (an NIH HPC system), Please refer to [Getting\\_started.md](#) to setup the working environment.**

# Keep in mind

- It is very limited time for a broad topic
- I will not try to cover many details.
- It is designed to provide just an overview on computation and a concrete set of starting materials.
- Most of these are selected online materials (with references) that is tailored to run on Biowulf.
- Questions and feedback are welcome in the workshop and in the google sheet.
  - [Questions & Feedback](#)

Me when people ask me how I learned programming:



# Agenda -- day 1, single cell RNA-seq

- 9-10AM, setting up the system
  - Get to know each other
  - Set up pseudolink (NIEHS\_NGS) to the /data/studentxx folder
  - Copy workshop files to NIEHS\_NGS
  - Initiate jupyter lab
- Questions and break (10 minutes)
- 10:30-11AM, Introducing single cell RNA-seq 30 minutes.
- 11-12AM, From bcl, fastq to expression matrix
  - Convert bcl to fastq files using the tiny-bcl folder 15 minutes
  - Using cellranger count to get expression matrix
- 12-12:30PM, Questions and break
- 12PM-1PM, Run basic pipelines
  - Using Seurat pipeline to import alignment result do QC, and normalization.
  - (optional) Using Scanpy pipeline to import alignment results to do QC, and normalization.
- 1PM-1:30PM, use RNA velocity to infer trajectory.
  - Run Velocyto CLI to get the reads aligned to non-spliced region.
  - Import into scVelo (python) to do trajectory inference.
- 1:30PM—2PM, Prepare for day2
  - Download and install IGV, IGB

# Agenda -- day 2, bulk RNA-seq and ChIP-seq

## Part II, Bulk RNA-seq

- 9:00-9:15AM, connect to biowulf
- 9:15-9:30AM, introduction to bulk RNA-seq
- 9:30-10AM, From Fastq files, alignment with RSEM, Tophat
- 10AM-10:30AM, Visualize the output using IGV.
- 10:30-11:00AM, Normalization and differential analysis with DESeq2
- Questions and break

## Part III, Chip-seq

- 11:30AM-12PM, Introduction to ChIP-seq analysis
- 12:00-12:15AM, Alignment to genome using Bowtie2
- 12:15-12:25AM, Use MACS2 to call peaks, and use Homer to call differential peaks
- 12:25-12:45AM, Use IGB to visualize peaks
- 12:45-1:15AM, Use CEAS to annotate peaks and summarize statistics.
- 1:15am -1:45AM, Use Homer to study TF binding site analysis. And use GREAT for gene ontology.

## Review & Summary

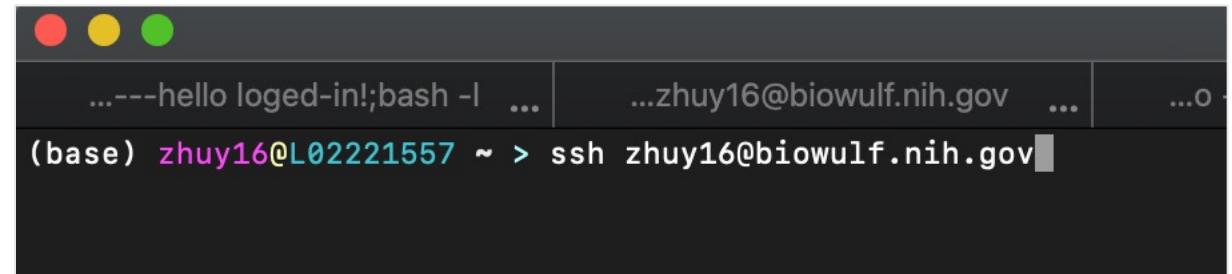
(code can be copied from Getting\_started.md)

# Log in to biowulf and go to the folder

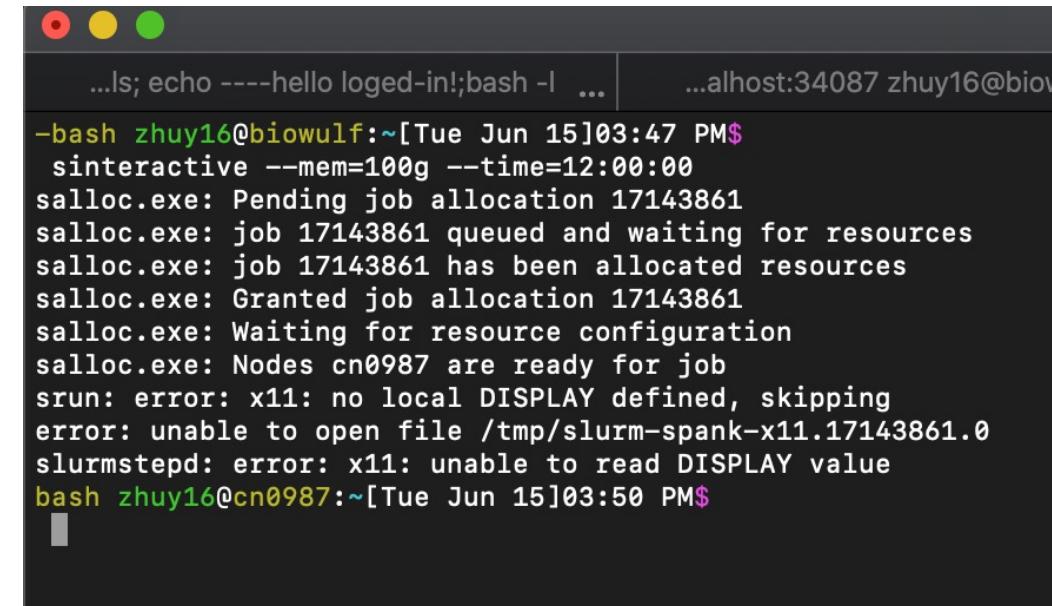
- ssh [user\\_id@biowulf.nih.gov](mailto:user_id@biowulf.nih.gov)
- # set up sudo links to your storage folder:
- module load tmux
- tmux new # get a detachable terminal
- sinteractive # get a basic compute node
- cd
- ln -s /data/user\_id NIEHS\_NGS

# copy the workshop material to your scratch folder:

- cp -r /spin1/users/classes/NIEHS\_NGS/workshop ~/NIEHS\_NGS/



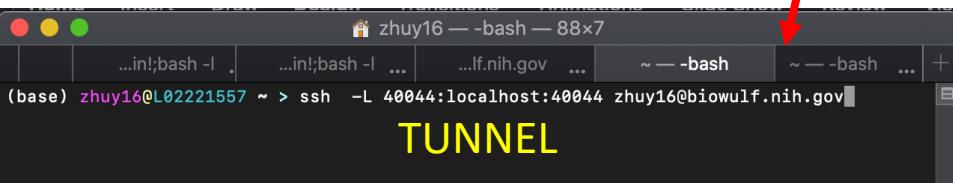
```
...---hello loged-in!;bash -l ... zhuy16@biowulf.nih.gov ...
(base) zhuy16@L02221557 ~ > ssh zhuy16@biowulf.nih.gov
```



```
...ls; echo ----hello loged-in!;bash -l ... alhost:34087 zhuy16@biowulf.nih.gov
-bash zhuy16@biowulf:~[Tue Jun 15]03:47 PM$ sinteractive --mem=100g --time=12:00:00
salloc.exe: Pending job allocation 17143861
salloc.exe: job 17143861 queued and waiting for resources
salloc.exe: job 17143861 has been allocated resources
salloc.exe: Granted job allocation 17143861
salloc.exe: Waiting for resource configuration
salloc.exe: Nodes cn0987 are ready for job
srun: error: x11: no local DISPLAY defined, skipping
error: unable to open file /tmp/slurm-spank-x11.17143861.0
slurmstepd: error: x11: unable to read DISPLAY value
bash zhuy16@cn0987:~[Tue Jun 15]03:50 PM$
```

# Start up the jupyter lab, --our working interface

- # Open another terminal
- ssh [studentxx@biowulf.nih.gov](mailto:studentxx@biowulf.nih.gov)
- Enter password
- module load tmux
- module load tmux; tmux new -ct 'sinteractive --mem=50g --time=12:00:00 --tunnel'
- # Copy the tunnel script to another (3<sup>rd</sup>) terminal, execute and enter password, to establish a ssh tunnel between local computer and the work node.
- module load jupyter R/4.0.5 && jupyter lab --ip localhost --port \$PORT1 --no-browser
- # wait until an URL link appears, copy it to your web browser, to get connected to biowulf through the jupyter lab interface.



Log in screen

\*\*\*WARNING\*\*\*

You are accessing a U.S. Government information system, which includes (1) this computer, (2) this computer network, (3) all computers connected to this network, and (4) all devices and storage media attached to this network or to a computer on this network. This information system is provided for U.S. Government-authorized use only.

Unauthorized or improper use of this system may result in disciplinary action, as well as civil and criminal penalties.

By using this information system, you understand and consent to the following:

\* You have no reasonable expectation of privacy regarding any communications or data transiting or stored on this information system. At any time, and for any lawful Government purpose, the government may monitor, intercept, record, and search and seize any communication or data transiting or stored on this information system.

\* Any communication or data transiting or stored on this information system may be disclosed or used for any lawful Government purpose.

--

Notice to users: This system is rebooted for patches and maintenance on the first Monday of every month at 7:15AM unless Monday is a holiday, in which case it is rebooted the following Tuesday. Running cluster jobs are not affected by the monthly reboot.

```
bcell    CSIMicrobes deprived_210611.bashrc  ini_conda_base.sh  ncbi-outdir  R      scratch  staging_for_deletion  zaya
condo.sh  data      igv      ini_miniconda_base.sh  NIEHS_NGS  r_site-library
----hello loged-in!
salash zhuy16@biowulf:~[Tue Jun 15]02:33 PM$ module load tmux; tmux new -ct 'sinteractive --mem=200g --gres=lscratch:10 --time=12:00:00 --tunnel'
salloc.exe: job 1/133459 has been allocated resources
salloc.exe: Granted job allocation 17133459
salloc.exe: Waiting for resource configuration
salloc.exe: Nodes cn2006 are ready for job
srn: error: x11: no local DISPLAY defined, skipping
error: unable to open file /tmp/slurm-spank-x11.17133459.0
slurmstepd: error: x11: unable to read DISPLAY value

Created 1 generic SSH tunnel(s) from this compute node to
biowulf for your use at port numbers defined
in the $PORTIn ($PORT1, ...) environment variables.

Please create a SSH tunnel from your workstation to these ports on biowulf.
On Linux/MacOS, open a terminal and run:

  ssh -L 40044:localhost:40044 zhuy16@biowulf.nih.gov

For Windows instructions, see https://hpc.soe.ucsc.edu/tutorials/jupyterlab.html

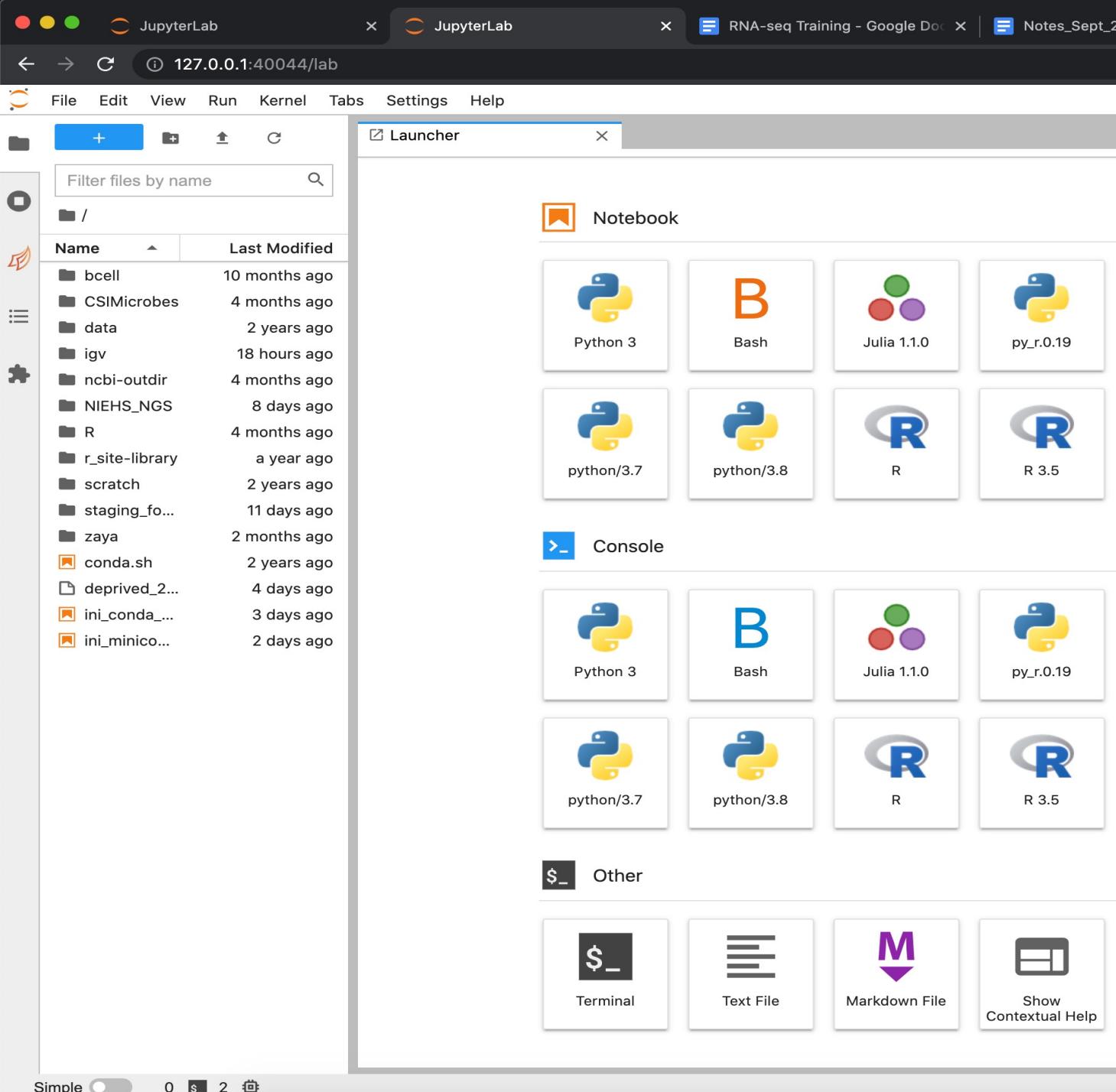
bash zhuy16@cn2006:~[Tue Jun 15]02:30 PM$ module load jupyter R/4.0.5 && jupyter 1
[3] 0:srun*
```

A red arrow points from the 'TUNNEL' text in the bottom right corner of the slide to the 'ssh -L 40044:localhost:40044 zhuy16@biowulf.nih.gov' command in the terminal window.

```
[base] zhuy16@L02221557 ~ > ssh -L 40044:localhost:40044 zhuy16@biowulf.nih.gov
To access the server, open this file in a browser:
File:///Applications/sshuttle.app/Contents/Resources/local/share/jupyter/runtime/jpservr-26881-open.html
Or connect directly to this URL:
http://localhost:40844/lab?token=a4864789e14833dc54987687847c63c566c665f58fb20f
http://127.0.0.1:40844/lab?token=a4864789e14833dc54987687847c63c566c665f58fb20f
Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
URL
```

# Jupyter lab

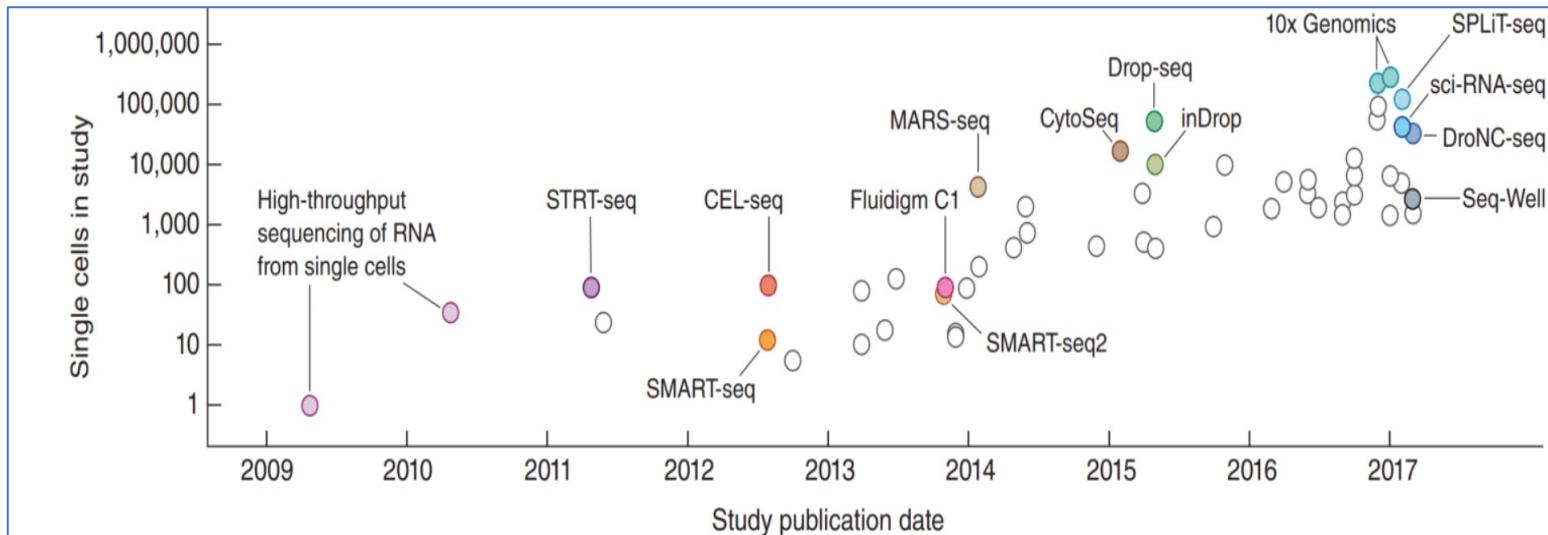
- Navigate files
- Bash terminal
- Control kernels
- Jupyter notebooks for
  - Kernels
    - Bash
    - R
    - Python
    - Julia
  - Document analysis
  - Git version control
  - Report results



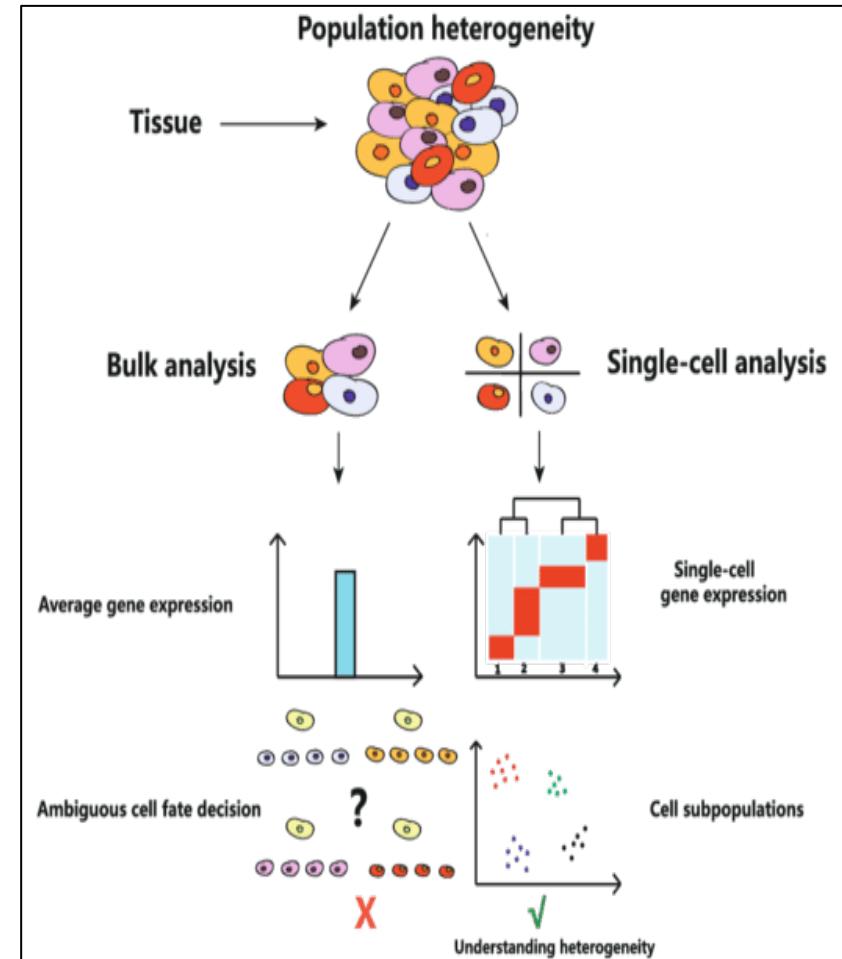
# Single cell RNA-seq

- Evolution of single cell technologies
- Introduction to 10X genomics
- Bioinformatic workflow

# Evolution of single cell technologies

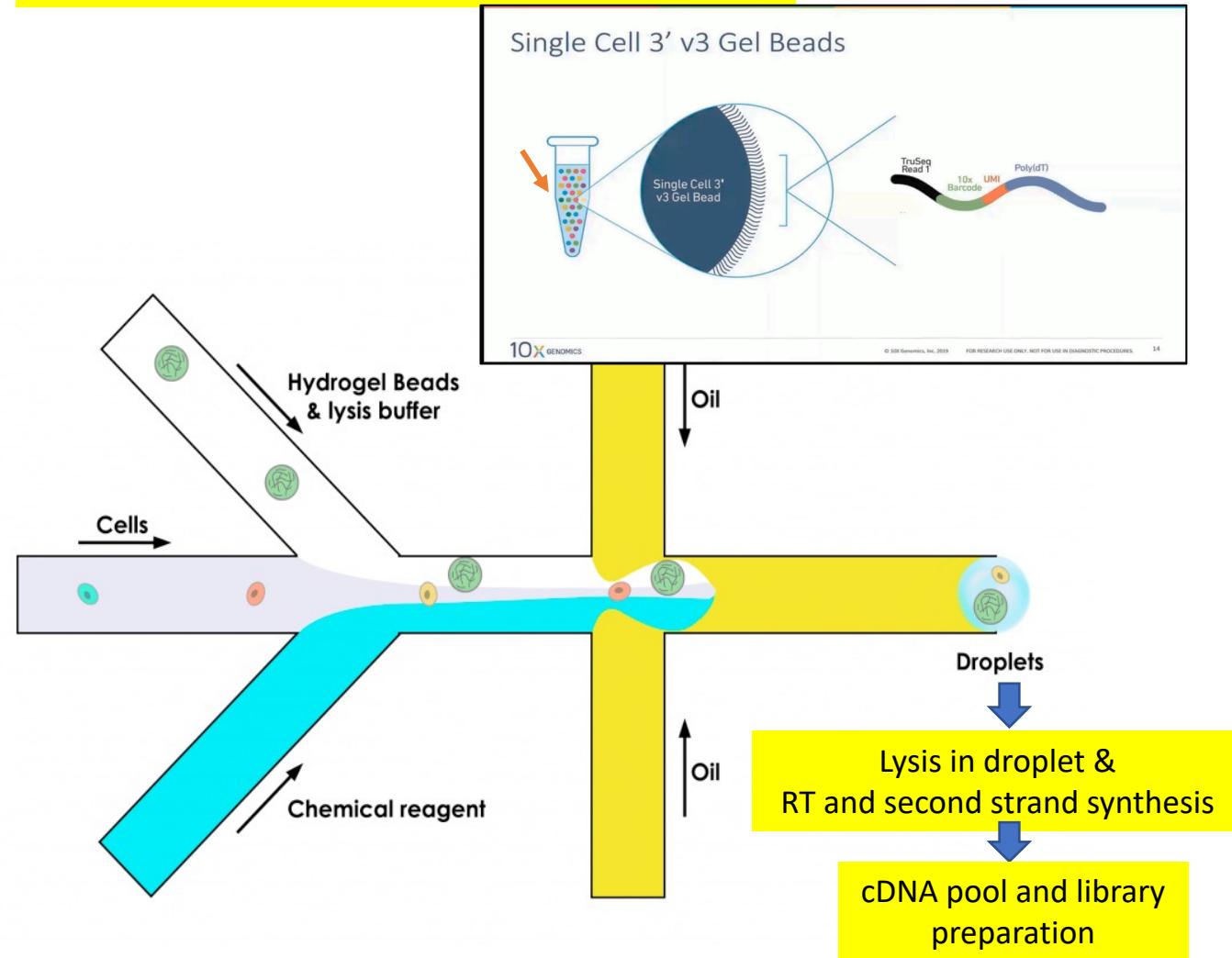


Exponential scaling of single-cell RNA-seq in the past decade.  
Nature protocols 13;4;599-604)

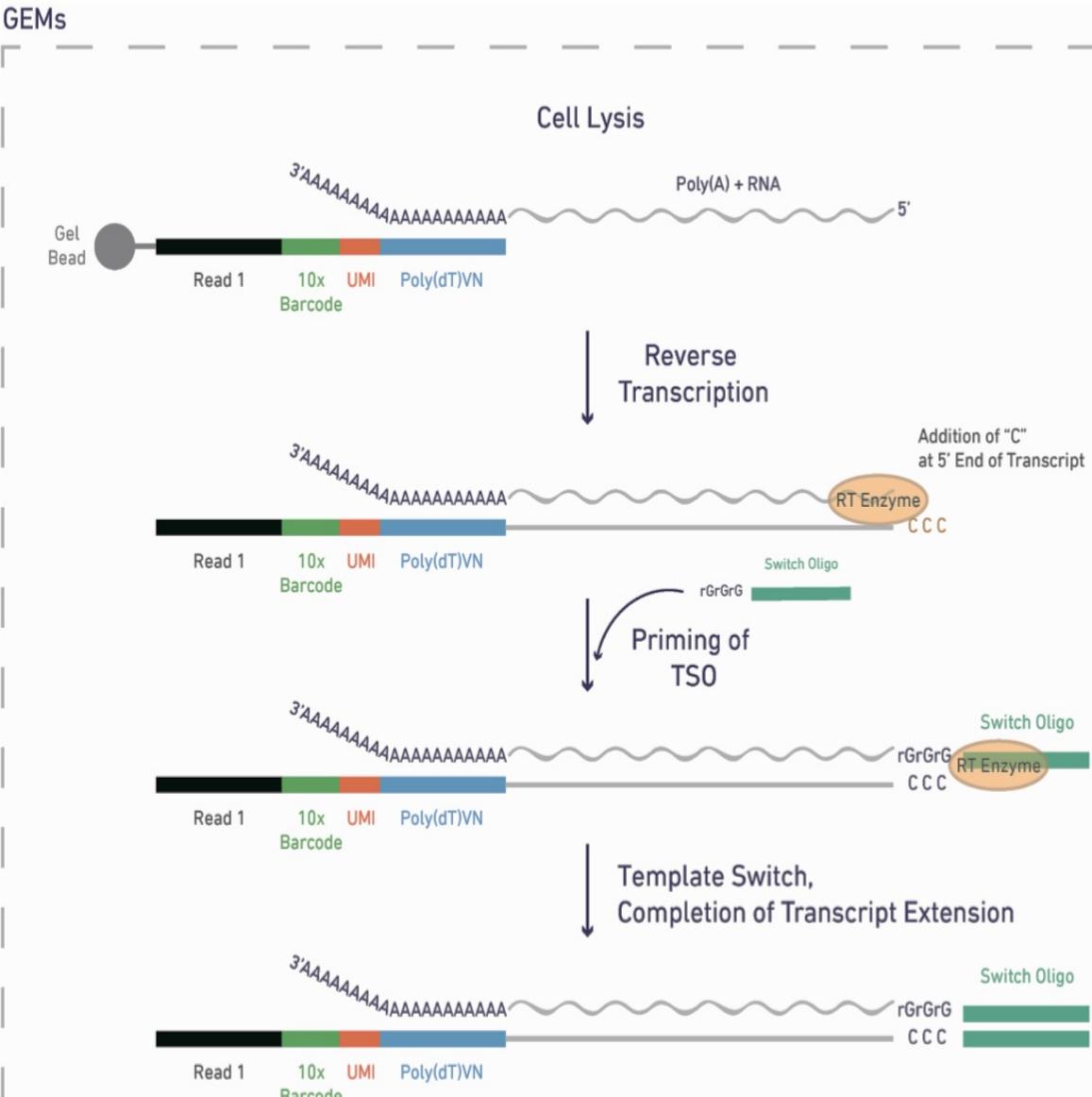


# 10X Genomics – commercial solution that facilitates automatic generation of Gel Bead-in-Emulsion (GEM)

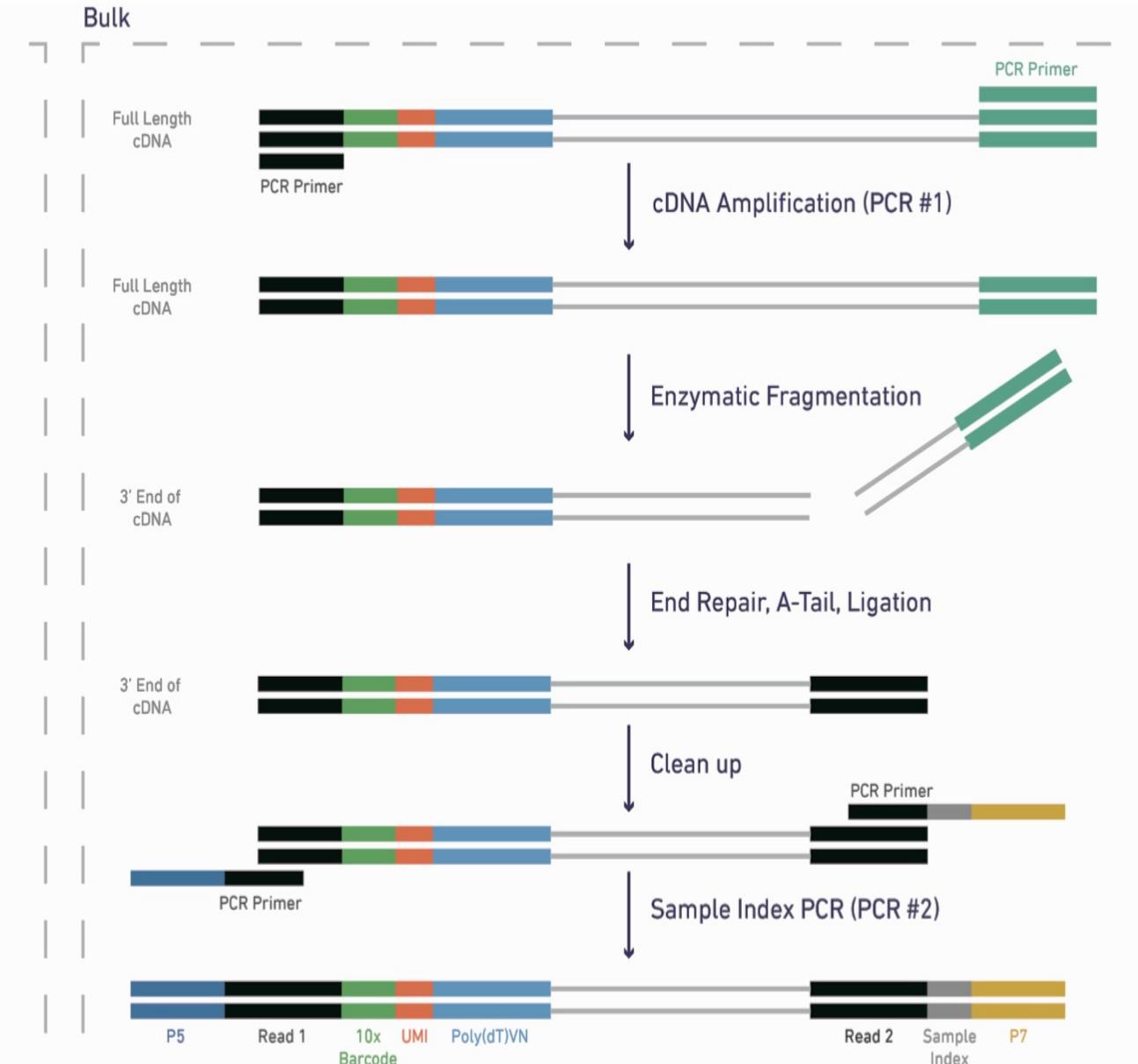
- In a GEM droplet, one hydrogel bead and one cell were captured
- One **hydrogel bead** is attached with **millions of poly-T primers** with an identical unique barcode.
- cDNA and the second-strand synthesis in the droplet
- Droplets (~1nL) are disrupted to collect all the barcoded samples for highly multiplexed library preparation and sequencing
- **Standardized automation** and reagent has made sequencing library preparation very efficient



# Inside individual GEMs (Gel Bead-in-Emulsion)

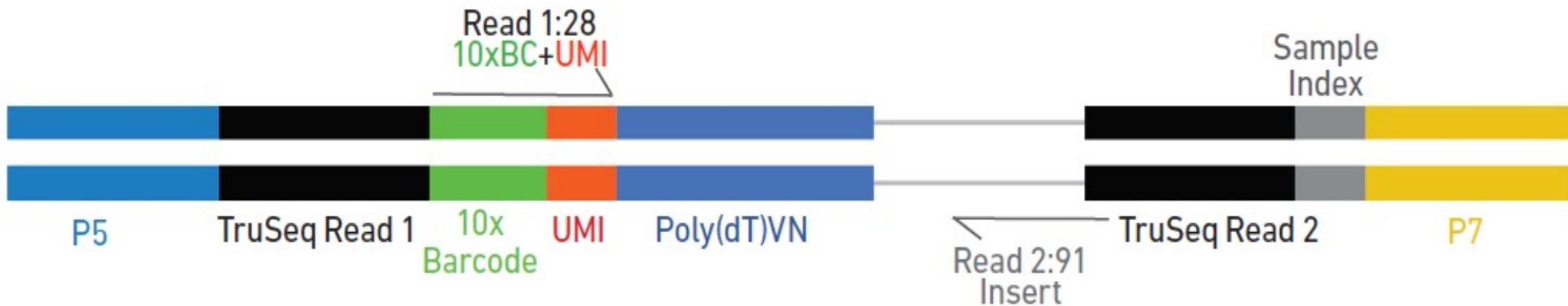


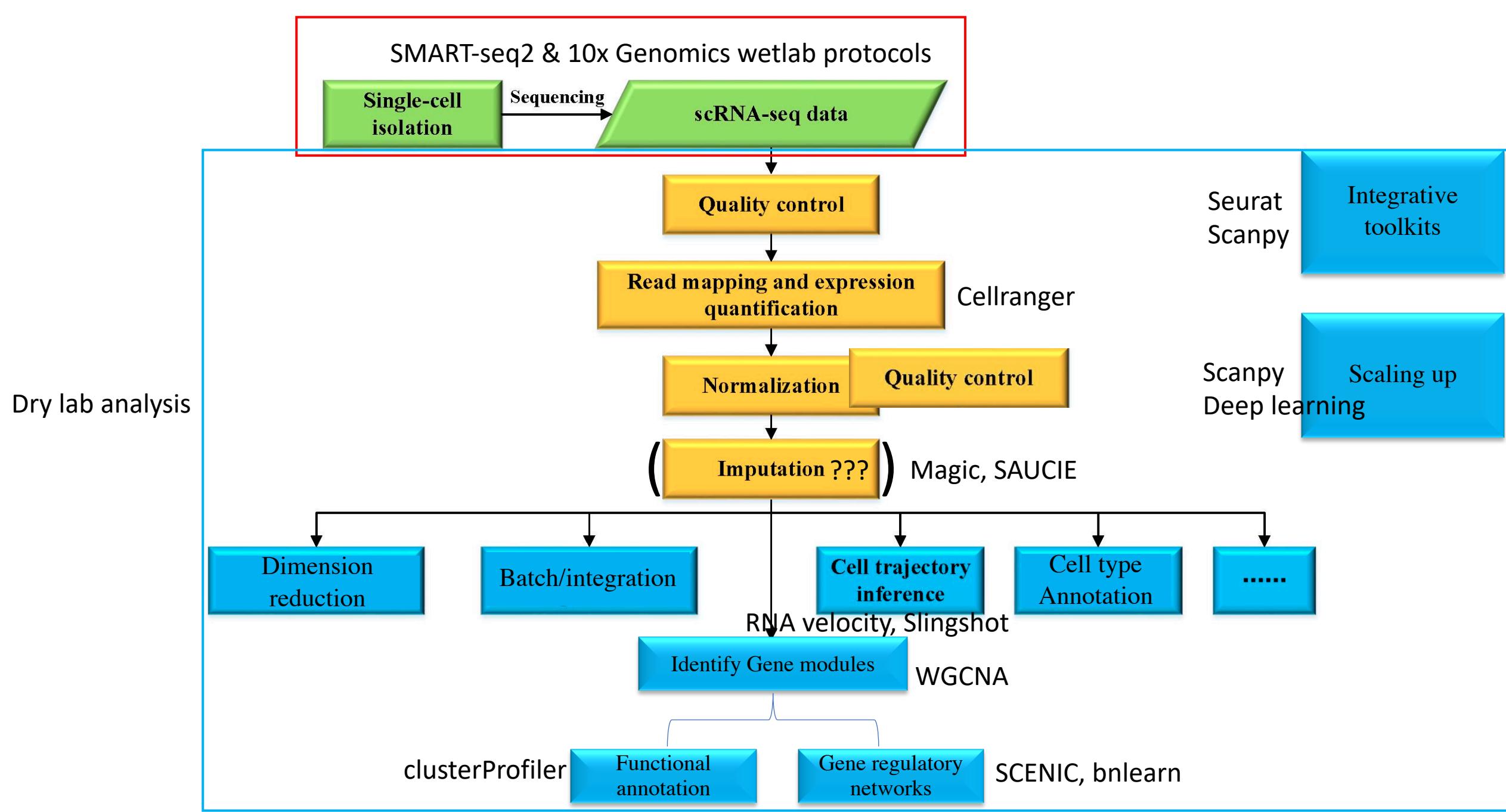
# Pooled cDNA processed in bulk

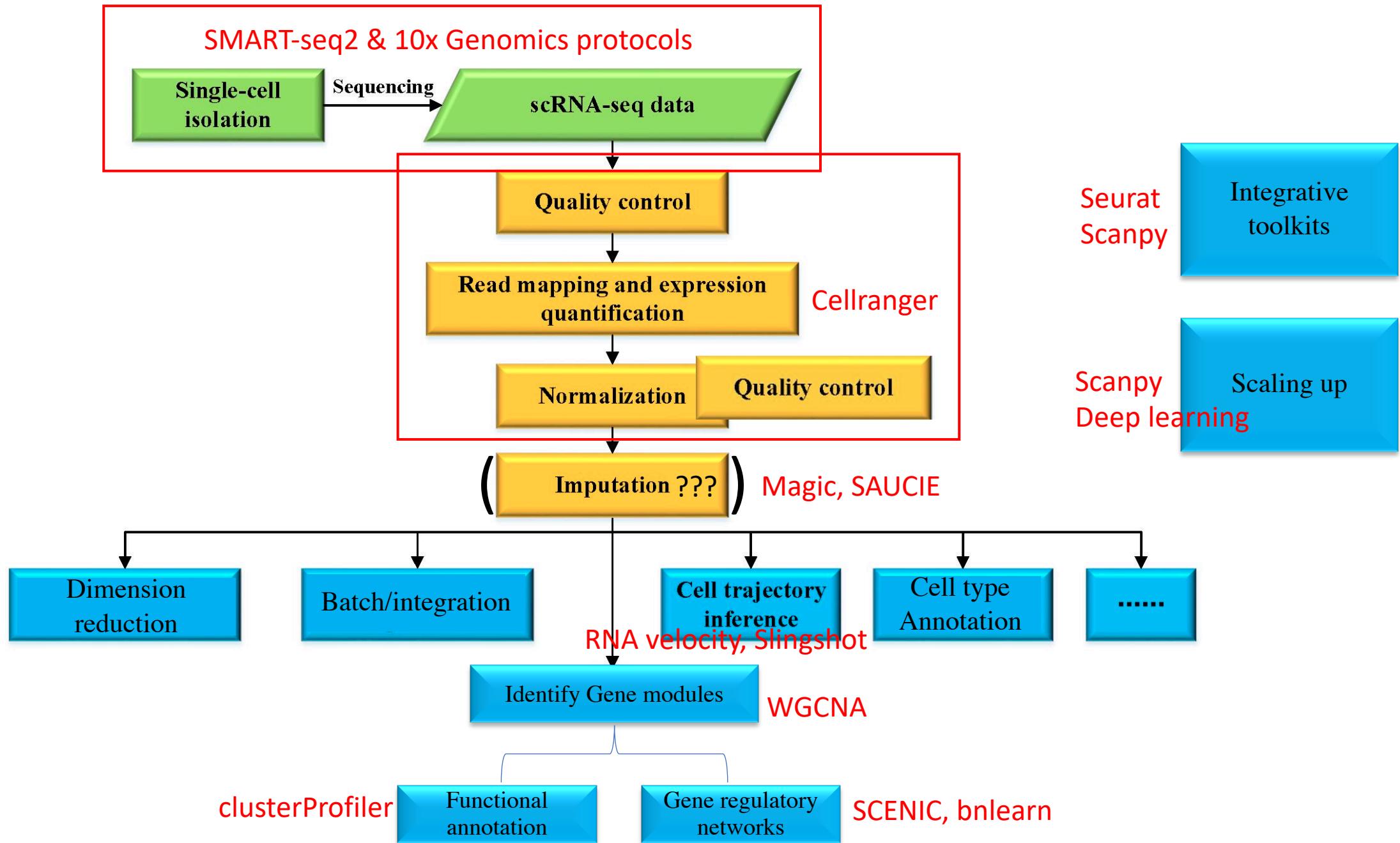


# Next-seq reading the paired ends

- 10XBC: 16 bp barcodes  $2^{16}=65536$  possible unique cells
- UMI,  $2^{12}=4096$  unique copies of mRNA can be distinguished for each gene
- Read2 will read into cDNA to identify the identity of the gene
- Sample barcode, identify the batch of your library sample
- All information will be summarized by the Cellranger software







JupyterLab

JupyterLab

RNA-seq Training - Google Doc

Notes\_Sept\_2019-2021 - Google Doc

STAT1\_6h\_IFNa\_peaks.bed

127.0.0.1:40044/lab/tree/NIEHS\_NGS/NIEHS\_NGS\_Workshop/scRNA-seq/notebooks/001\_convert\_bcl\_to\_fastq.ipynb

File Edit View Run Kernel Tabs Settings Help

+ ☰ Filter files by name

Name / ... / scRNA-seq / notebooks /

Name	Last Modified
001_convert_bcl_to_fastq.ipynb	a day ago
002_decompress_pbmc_fastq.ipynb	a day ago
003_cellranger_count_pbmc.ipynb	21 hours ago
004_R_pbmc3k_tutorial.ipynb	a day ago
004.1_R_convertSeuratToLoom.ipynb	a day ago
005_Python_pbmc3k.ipynb	a day ago
006.0_bash_prepareVelocity.ipynb	7 hours ago
006.1_VelocityBasics.ipynb	a day ago

Let's go to Jupyter lab...  
There notebook files are in github as well  
Step 001, make fastq from bcl files

Launcher 001\_convert\_bcl\_to\_fastq.ipynb

ref:  
cellranger documentation on biowulf, <https://hpc.nih.gov/apps/cellranger.html>  
davetang's blog on cellranger, <https://davetang.org/muse/2018/08/09/getting-started-with-cell-ranger/>

## 1. Demo on making fastq from data generated from a flowcell, a 'bcl' files

```
[22]: module load cellranger  
[-] Unloading cellranger 6.0.1  
[+] Loading cellranger 6.0.1  
These files have been downloaded on biowulf
```

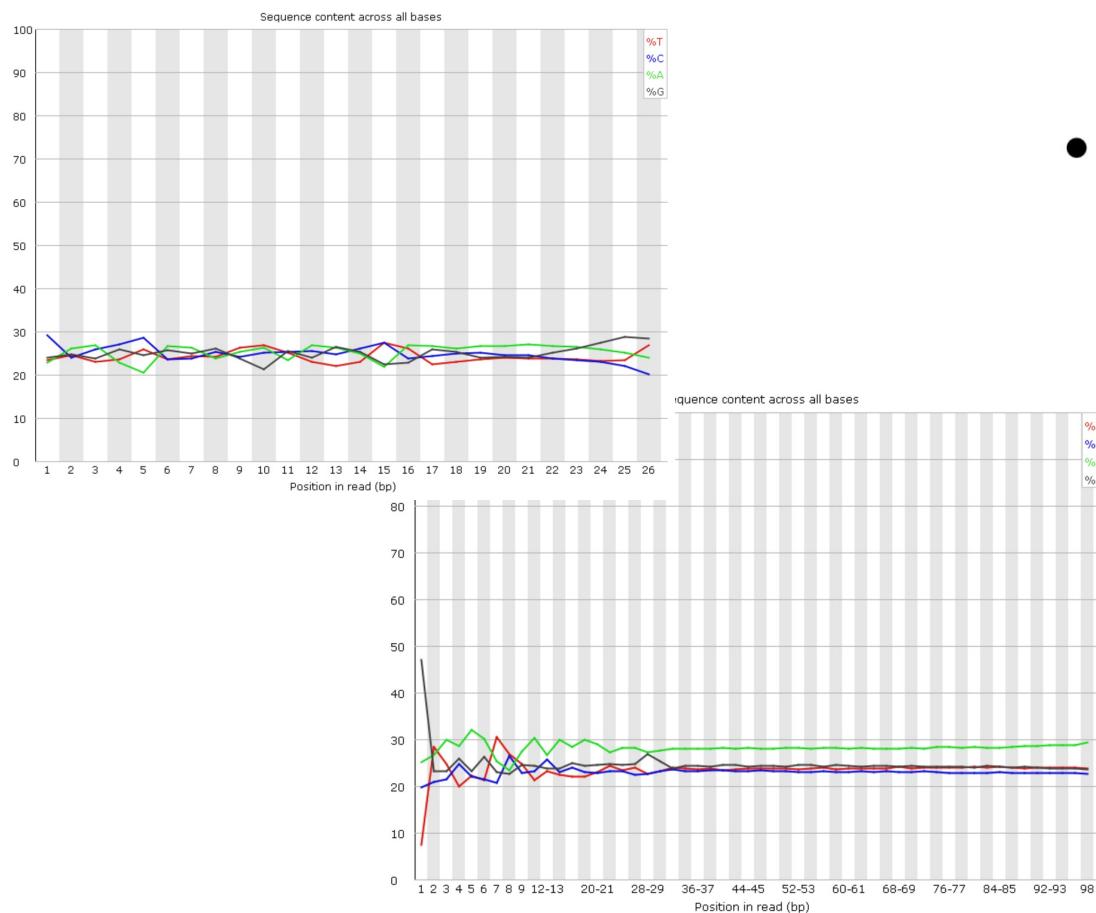
```
[23]: # wget -c -N http://s3-us-west-2.amazonaws.com/10x.files/samples/cell-exp/2.1.0/pbmc8k/pbmc8k_f  
# wget -O cellranger-2.2.0.tar.gz "http://cf.10xgenomics.com/releases/cell-exp/cellranger-2.2.0  
# wget http://cf.10xgenomics.com/supp/cell-exp/refdata-cellranger-GRCh38-1.2.0.tar.gz
```

```
[24]: cp ${CELLRANGER_TEST_DATA:-none}/cellranger-tiny-bcl-1.2.0.tar.gz ..data/  
cp ${CELLRANGER_TEST_DATA:-none}/cellranger-tiny-bcl-samplesheet-1.2.0.csv ..data/  
cp: cannot create regular file '../data/': Not a directory  
cp: cannot create regular file '../data/': Not a directory  
: 1
```

```
[4]: cat ..data/cellranger-tiny-bcl-samplesheet-1.2.0.csv  
[Header],,,,,,,  
IEMFileVersion,4,,,...  
Investigator Name,rir,,,...
```

# QC on sequencing results

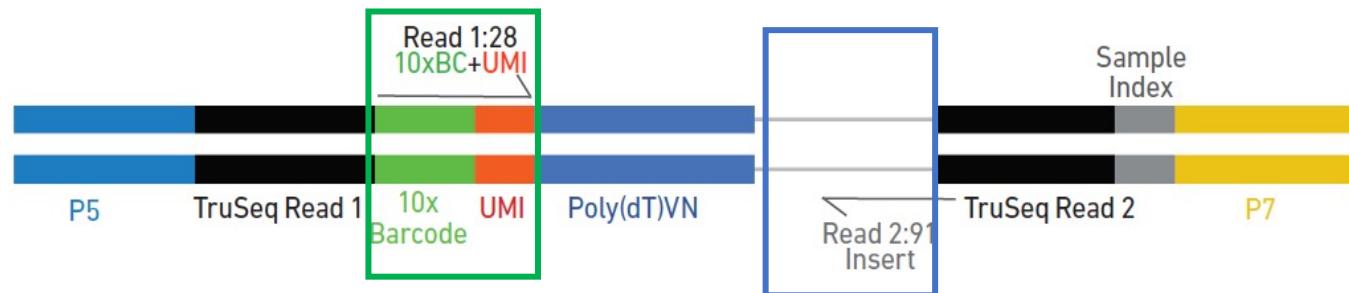
scrALI001\_S1\_L001\_I1\_001.fastq.gz  
scrALI001\_S1\_L001\_R1\_001.fastq.gz  
scrALI001\_S1\_L001\_R2\_001.fastq.gz



- I1
  - Index file. All identical (or one of 4) at Babraham

- R1
  - Barcode reads
    - 16bp cell level barcode
    - 10bp UMI

- R2
- 3' RNA-seq read



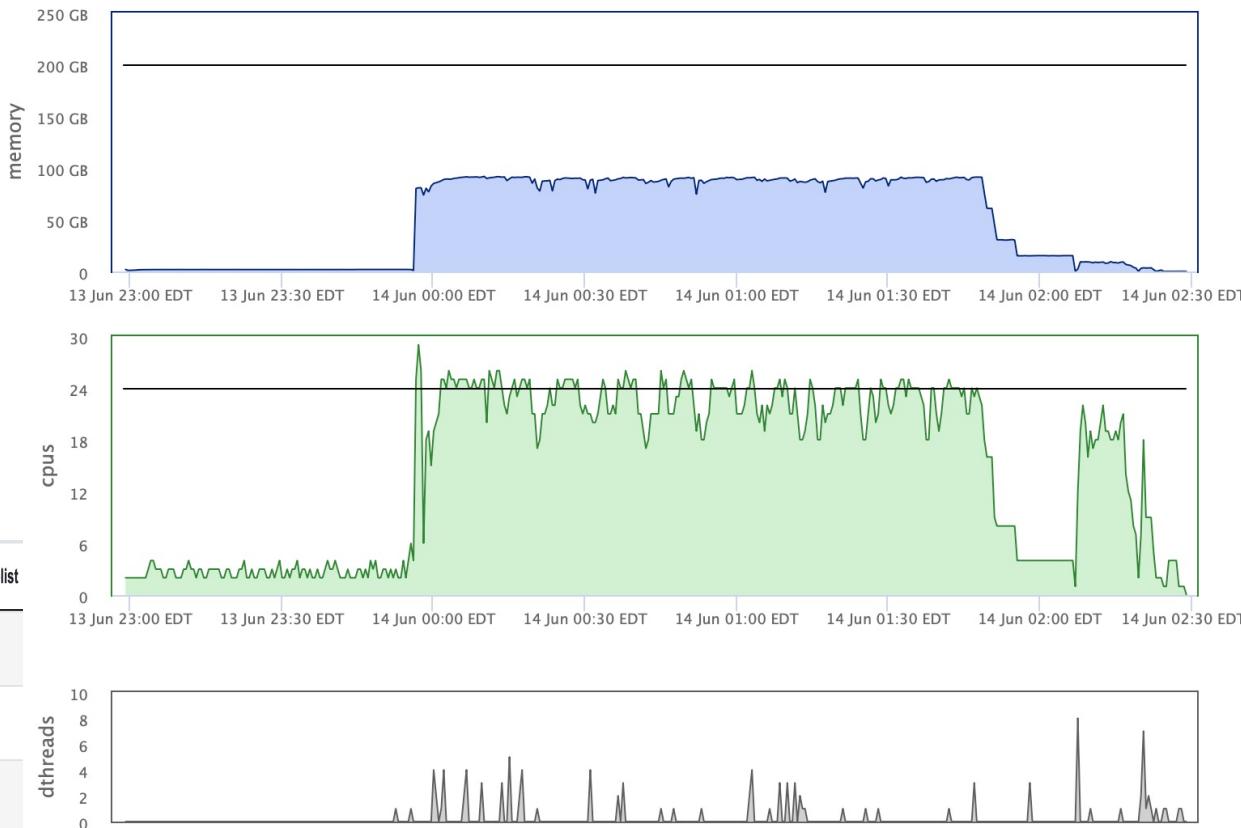
# Alignment results and Quality Controls

- Cellranger
  - cellranger mkfastq
  - cellranger count
  - cellranger aggr for multiple batches
- Quality controls
  - FASTQC on fastq files
  - Number of cells per experiment
  - Number of UMI per cell
  - Number of genes per cell
  - Percentage of mitochondrial reads
  - Removal of doublets/aggregates

# Monitoring jobs status on biowulf

Login to the user dashboard to monitor  
<https://hpc.nih.gov/dashboard/>

jobid	jobname	state	statetime	nodelist
17006700	cellranger_pbmc3.sh	RUNNING	2021-06-13 22:58:42 EDT	cn0925
17006693	cellranger_pbmc.sh	FAILED	2021-06-13 22:57:30 EDT	cn0925
17006574	sinteractive	RUNNING	2021-06-13 22:45:00 EDT	cn1057



Sbatch jobs can be listed on terminal using  
**sjobs -u user\_id**

[Export to PNG](#)

```
sjobs -u zhuy16
User   JobId   JobName   Part   St Reason Runtime   Walltime   Nodes   CPUs   Memory   Dependency   Nodelist
=====
zhuy16 17006003 cellranger norm    R     1:39:28  2:00:00   1     16  200 GB   cn3422
zhuy16 17006574 sinteracti interactive R    14:30    8:00:00   1     2  100 GB   cn1057
zhuy16 17006700 cellranger norm    R     0:48     12:00:00  1     24  200 GB   cn0925
```

jobname	cellranger_pbmc3.sh
user	zhuy16
submitted	2021-06-13 22:58:41 EDT
state	COMPLETED
submission script	/data/classes/NIEHS_NGS/RNA-workshop-2021/scRNA-seq/data/cellranger_pbmc3.sh
work directory	/data/classes/NIEHS_NGS/RNA-workshop-2021/scRNA-seq/data

# Cellranger count report

FATSTQ

I1.FASTQ  
R1.FASTQ  
R2.FASTQ

Cellranger →

report summary ...

## Estimated Number of Cells

15,894

Mean Reads per Cell

11,380

Median Genes per Cell

2,174

## Sequencing

Number of Reads

180,878,636

Valid Barcodes

98.1%

Sequencing Saturation

10.3%

Q30 Bases in Barcode

98.4%

Q30 Bases in RNA Read

82.7%

Q30 Bases in UMI

98.7%

## Mapping

Reads Mapped to Genome

95.4%

Reads Mapped Confidently to Genome

90.2%

Reads Mapped Confidently to Intergenic Regions

3.0%

Reads Mapped Confidently to Intronic Regions

12.8%

Reads Mapped Confidently to Exonic Regions

74.4%

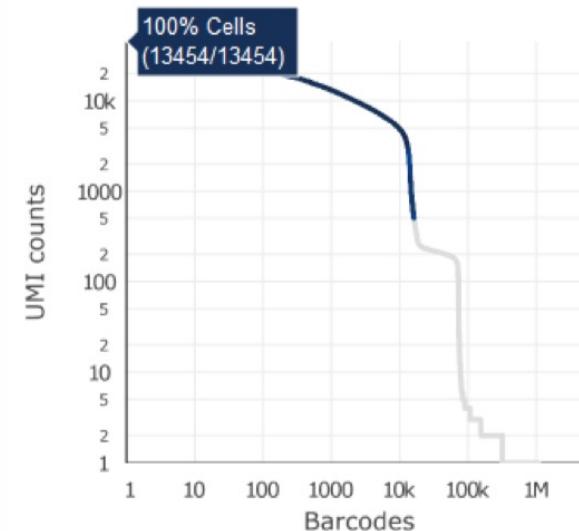
Reads Mapped Confidently to Transcriptome

71.9%

Reads Mapped Antisense to Gene

0.9%

## Cells



Estimated Number of Cells

15,894

Fraction Reads in Cells

88.1%

Mean Reads per Cell

11,380

Median Genes per Cell

2,174

Total Genes Detected

20,185

Median UMI Counts per Cell

5,742

## Sample

Name

embryoid\_d4

Description

Transcriptome

mm10

Chemistry

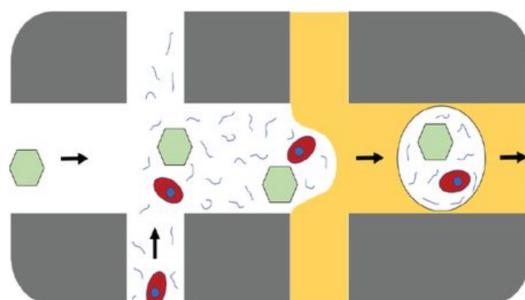
Single Cell 3' v3

Cell Ranger Version

3.0.2

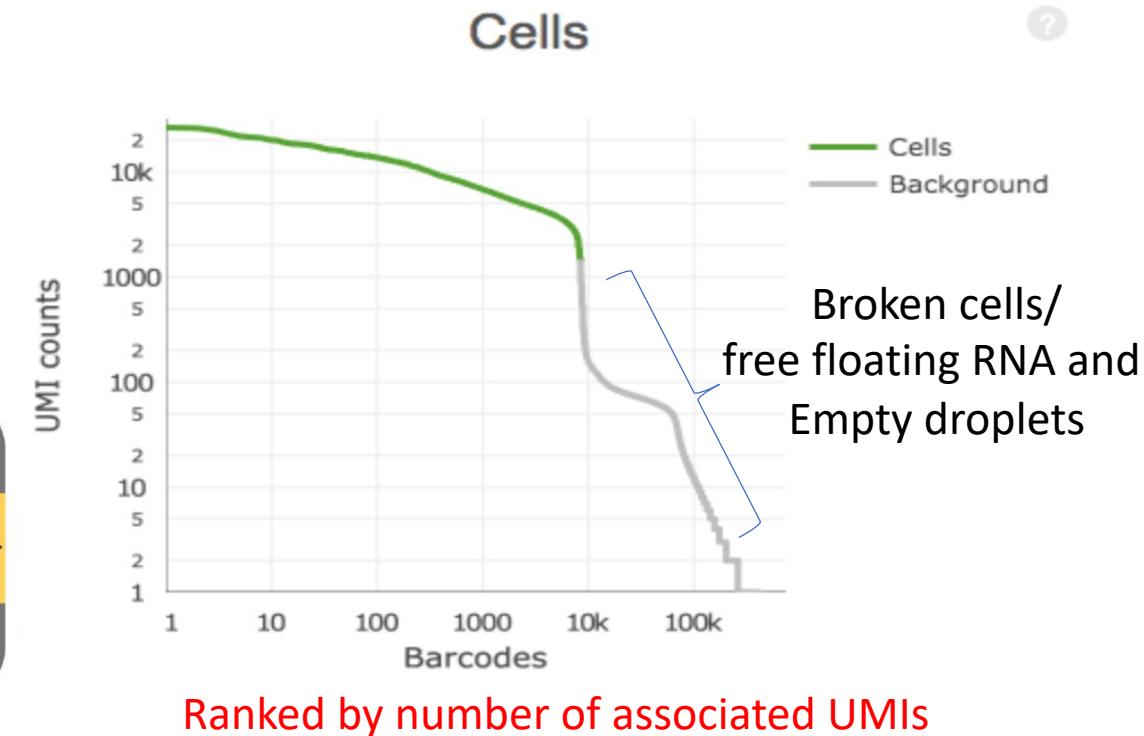
# cellranger count output

- Reads level
  - Cellranger
    - cellranger mkfastq
      - Generate fastq files from image “.bcl” files
    - **cellranger count → sparse matrix**
      - umi, unique molecule identifier
    - cellranger aggr
      - Combine count data from multiple batches
  - (For CITE-seq and HASH-tag)
    - Cite-seq-count



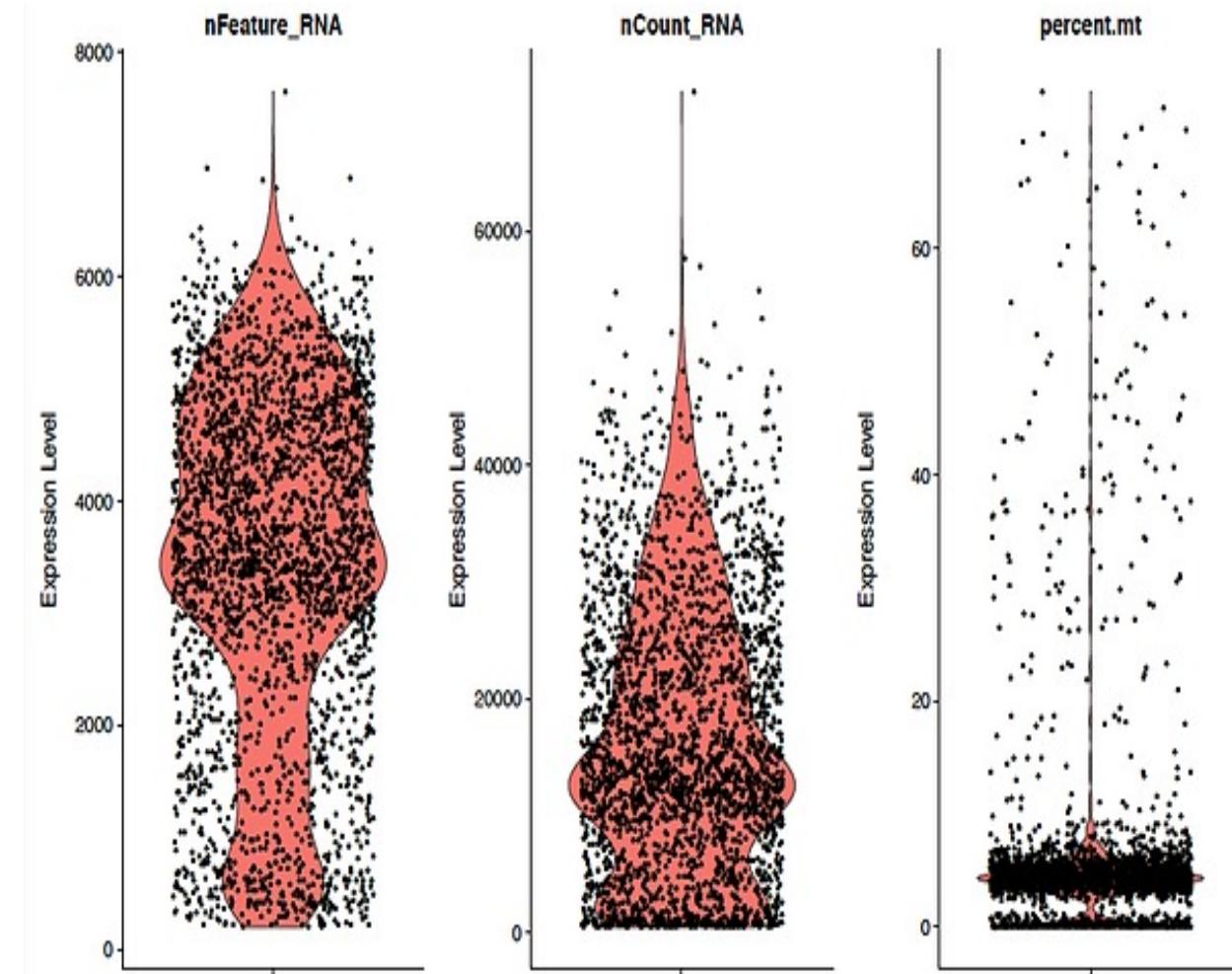
Output:

```
$ cd /home/jdoe/runs/sample345/outs  
$ tree filtered_feature_bc_matrix  
filtered_feature_bc_matrix  
└── barcodes.tsv.gz    --cells  
└── features.tsv.gz   --genes  
└── matrix.mtx.gz     --sparse matrix  
0 directories, 3 files
```



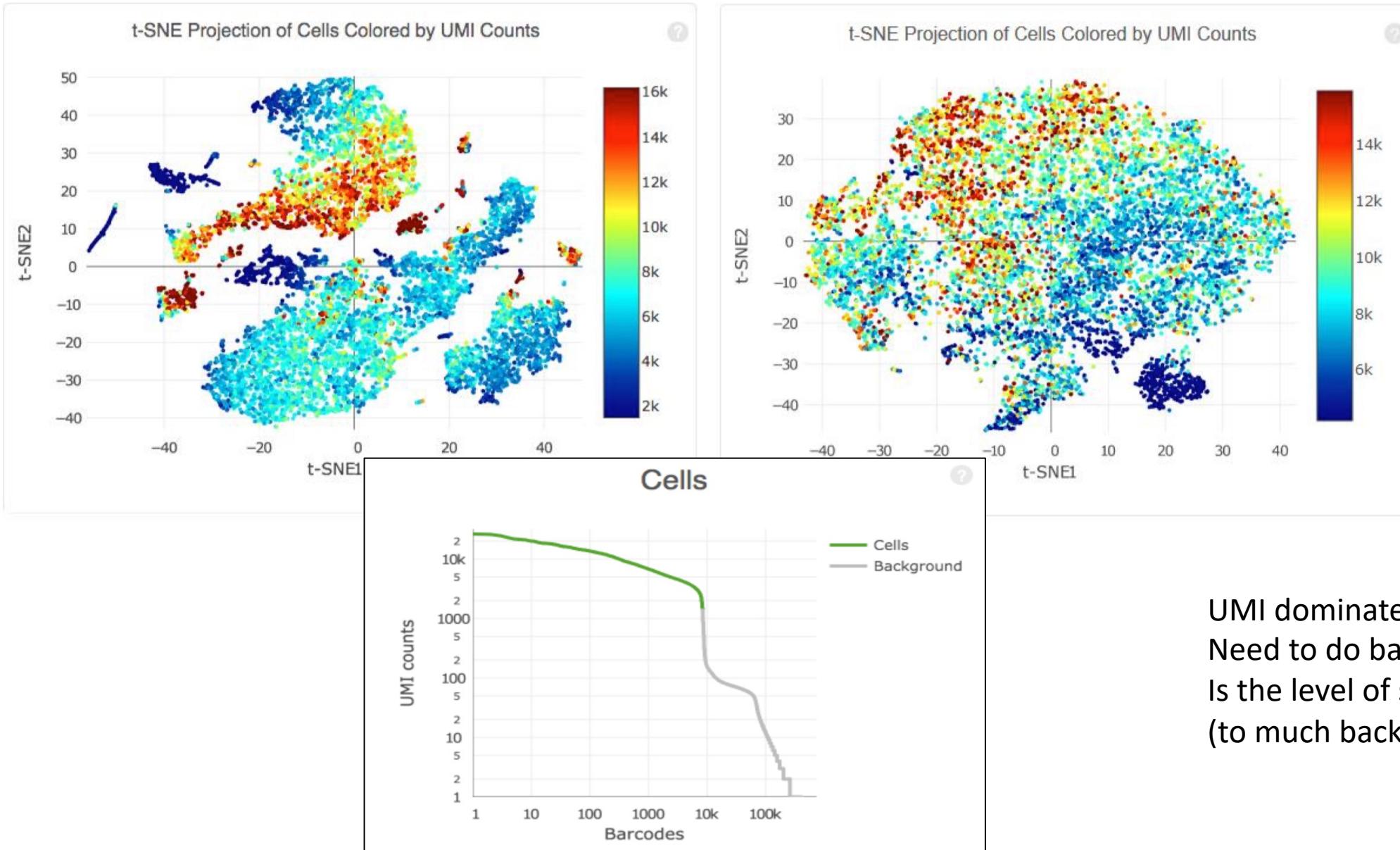
# Seurat -- Quality Controls

- Quality of reads – FASTQC
- Percentage of mitochondrial reads
  - Too many mitochondria reads may indicate that cells are dying/dead/broken
  - Case-by-case, the range may vary with method of library prep methods/cell type
- How many cells are you capturing?
  - Typically few thousands in each 10X run
- The sequencing depth
  - Are they acceptable in the field (minimal 2,000 reads /cell?) determined by the Cellranger
- Alignment to the genome and exons
  - Should be 90-100% to the genome
  - A reasonably narrow range 70-80% to the exons
  - (Could be 30% to exons if you use nucleus, which contain lots of introns)
- Expected markers expressed?
  - Highly expressed genes, cell type markers, automatic detection such as scMCA etc
- Be prepared to see differences between RNA (because of the depth and dropouts) and proteins.
- Confounding factors?
  - Batch effect?
  - Is your dimension reduction capturing biological or technical variations?
  - Can be evaluated by WGCNA and visualization in PCA or tSNE



<https://www.biostars.org/p/377422/>

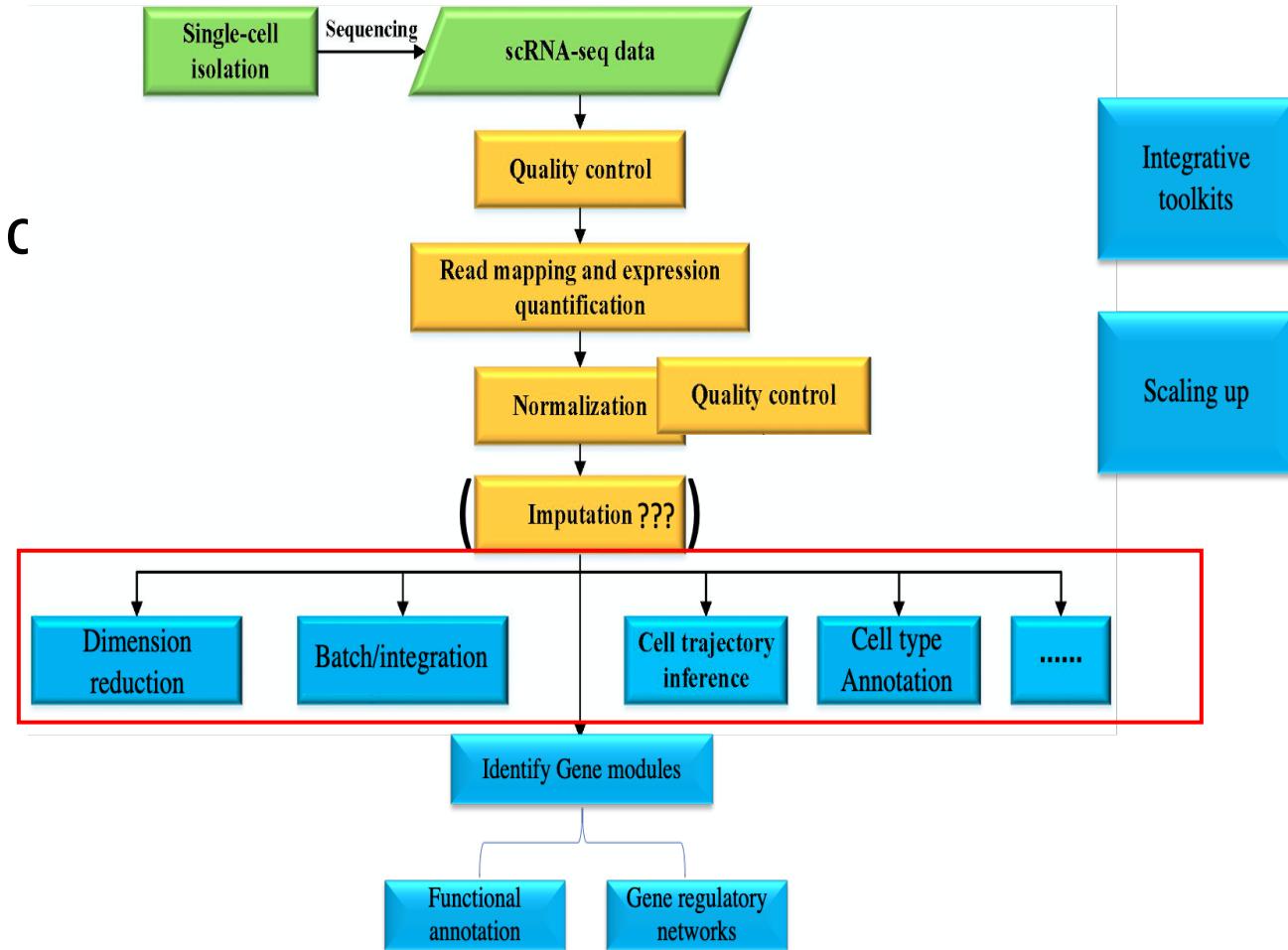
# Is coverage variation affecting your data?



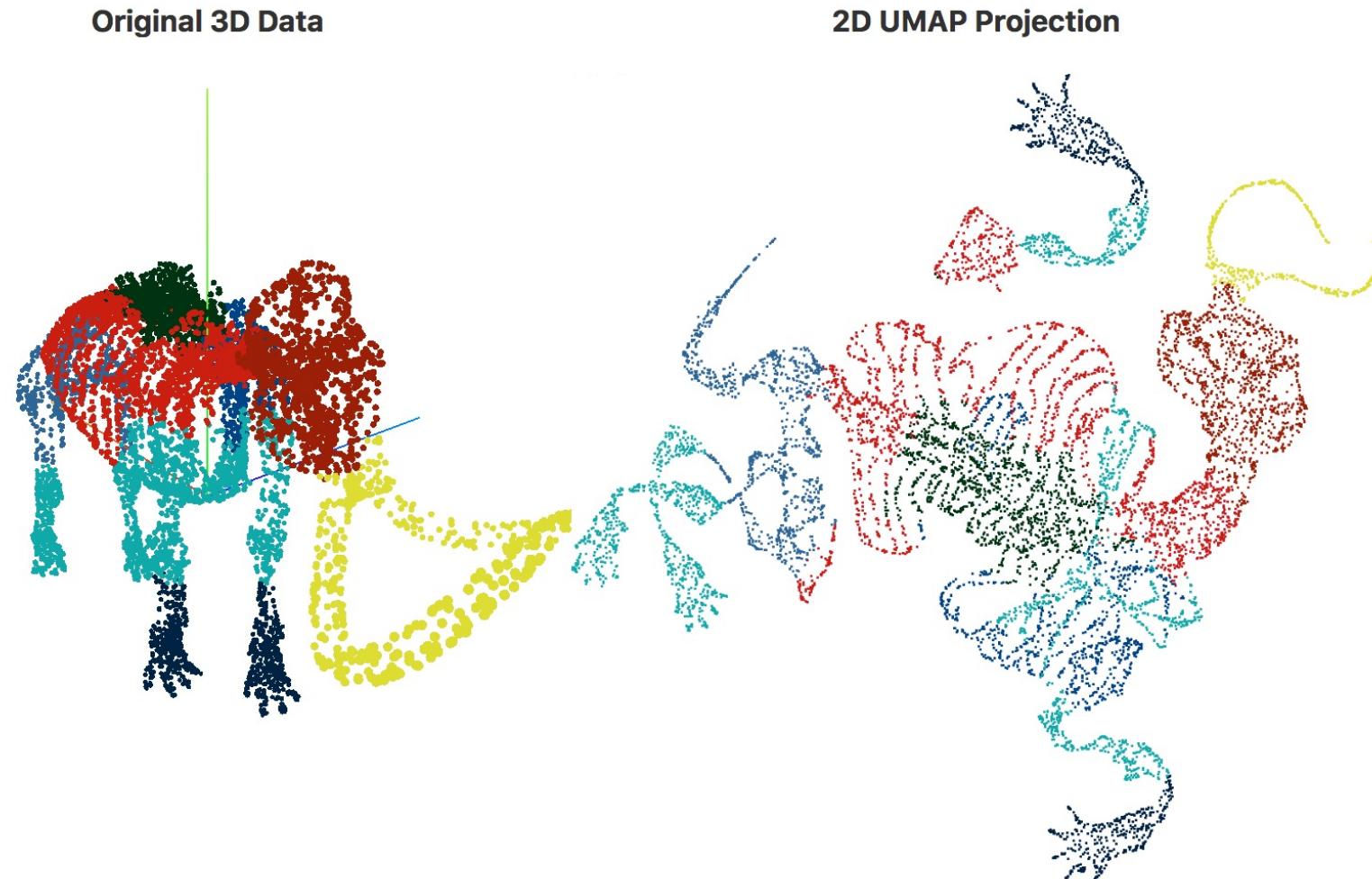
UMI dominate the variance?  
Need to do batch correction?  
Is the level of separation enough/expected  
(to much background RNA?)

# Cell-based analysis

- Clustering and annotate the biologic identities of clusters
- Inference of trajectory
- Batch correction/data integration.
- comprehensive workflow/toolkits



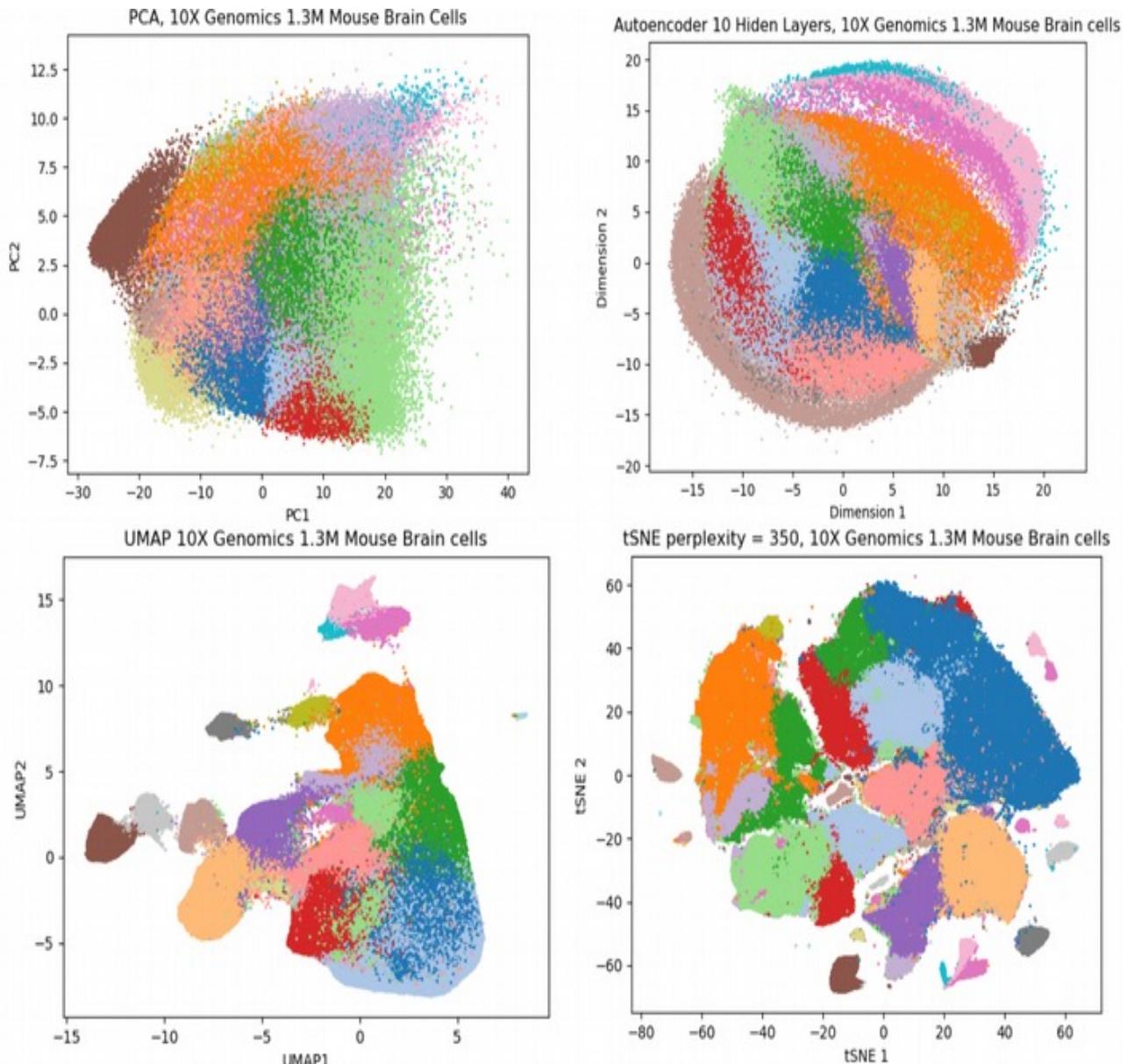
# Dimension reduction--an intuitive illustration



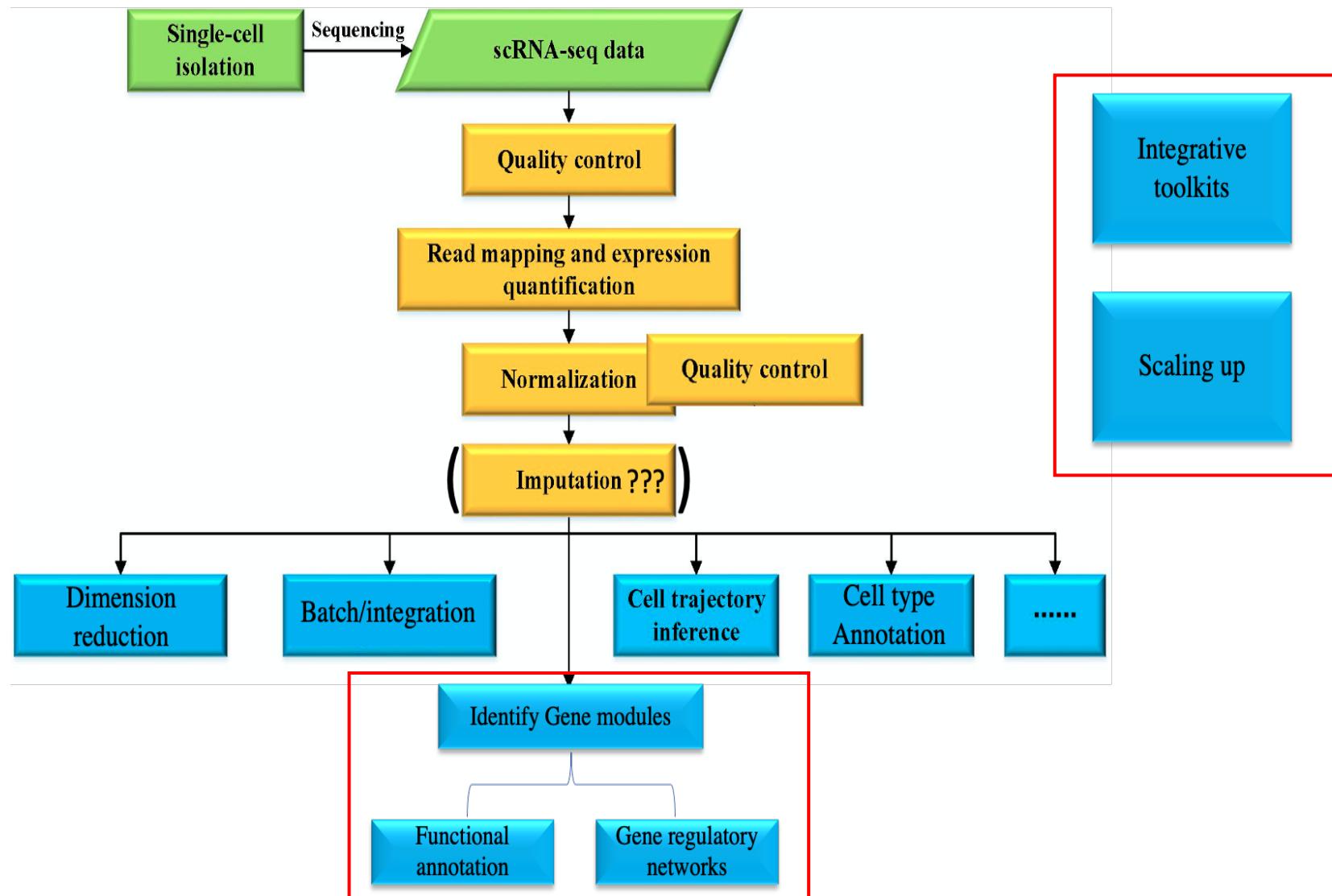
Screenshot by Mariam from  
<https://pair-code.github.io/understanding-umap/>

# Dimension reduction

- Dimension reduction
  - PCA
    - Linear reduction
    - Based on distances
    - 2D structure in PCA depends on certain observed dominant variations
    - Often not sufficient for large number of cells
  - tSNE
    - Non-linear reduction
    - Attention to **local similarity**
    - Global shape is less meaningful
    - Add new data changes the whole pattern
  - UMAP
    - Consider both global and local structure
    - **Learnt embeddings** can be saved for new batch of data
  - AutoEncoder
    - Fast algorithm to handle up to millions of cells



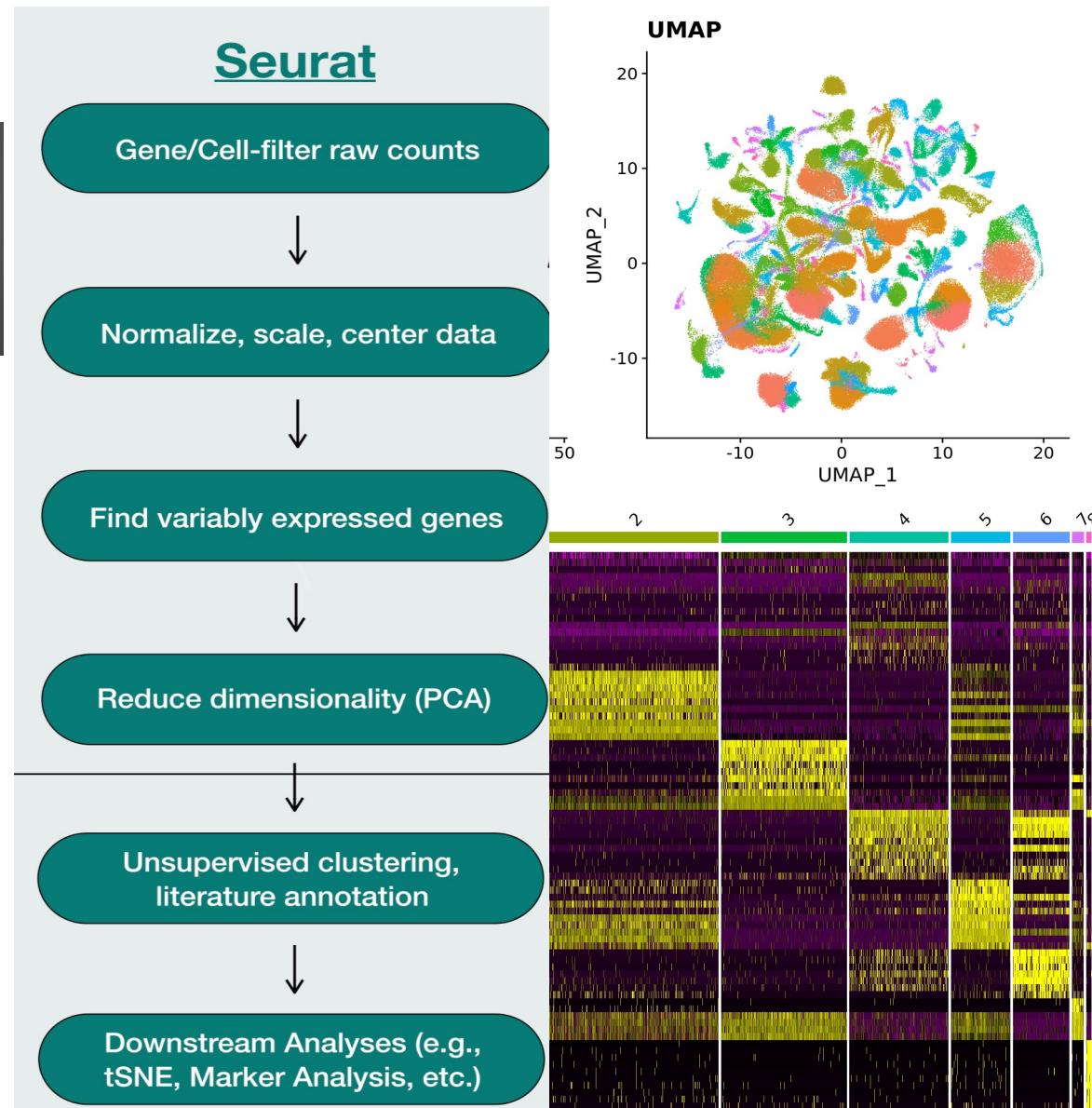
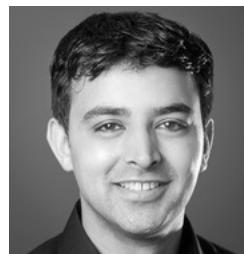
# Comprehensive toolkits



# Comprehensive pipeline tools for explorative analysis -- Seurat

- Seurat pipeline

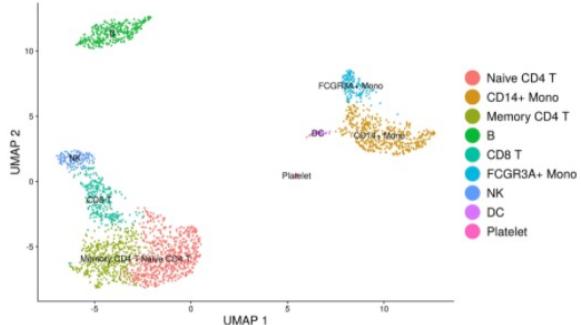
- General QC assessment
- Cell type annotation
- Batch correction and meta analysis
- Multimodal analysis (for CITE-seq, Hash-tagging, ATAC-seq)
- Comparative analysis across different conditions



# Multiple vignettes for different tasks

Basic pipeline: QC,  
Dimension reduction  
Clustering  
Marker identifications

## Guided tutorial – 2,700 PBMCs



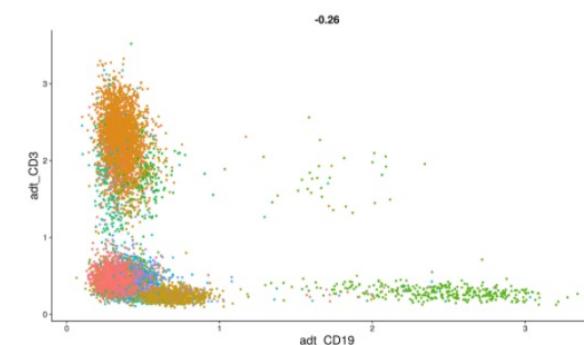
A basic overview of Seurat that includes an introduction to common analytical workflows.

[Start from here](#)

GO

## Integrating with CITE-seq, HASH-seq

## Multimodal analysis



An introduction to working with multimodal datasets in Seurat.

GO

### Introduction to scRNA-seq integration

A dot plot showing gene expression levels across various samples. The y-axis is labeled 'Sample' and the x-axis is 'Features'. A legend on the left lists sample names. A color scale indicates 'Percent Expressed' from 0 to 100. A 'GO' button is present at the bottom.

### Mapping and annotating query datasets

A 3D tSNE plot showing the mapping of a query dataset onto a reference dataset. The plot features two distinct clusters of points (ctrl and stim) connected by lines, indicating shared cell types. A 'GO' button is present at the bottom.

### Fast integration using reciprocal PCA (rPCA)

A UMAP plot showing the results of reciprocal PCA (rPCA) integration. It displays two distinct clusters labeled 'ctrl' (red) and 'stim' (blue) on the x-axis (UMAP 1) and y-axis (UMAP 2). A 'GO' button is present at the bottom.

Identify anchors using the reciprocal PCA (rPCA) workflow, which performs a faster and more conservative integration

### Tips for integrating large datasets

A UMAP plot showing the integration of a large dataset. The plot is labeled 'orig.ident' and shows multiple distinct clusters. A legend on the right lists MantorBM1 through MantorBM8. A 'GO' button is present at the bottom.

Tips and examples for integrating very large scRNA-seq datasets (including >200,000 cells)

### Integrating scRNA-seq and scATAC-seq data

Two UMAP plots side-by-side. The left plot is labeled 'RNA' and the right is 'ATAC'. Both plots show distinct clusters for 'ctrl' (red) and 'stim' (blue) samples. A 'GO' button is present at the bottom.

Annotate, visualize, and interpret an scATAC-seq experiment using scRNA-seq data from the same biological system

### Multimodal Reference Mapping

A UMAP plot showing multimodal reference mapping. The plot is labeled 'CD16 Mono' and shows a cluster of points with a color gradient from white to red. A color scale on the right indicates 'Prediction Score' from 0.00 to 1.00. A 'GO' button is present at the bottom.

Analyze query data in the context of multimodal reference atlases.

GO

GO

GO

# Download vignette/tutorial to use on your data

- Linked to github for you to download the code.
- Follow through the tutorial using sample data included in the package.
- Change the input file to your our data to use the analytic pipelines.

The screenshot shows the Seurat 4.0.0 website. The top navigation bar includes links for Install, Get started, Vignettes (with a dropdown arrow), Extensions, FAQ, News, Reference, and Archive. A home icon is also present. The main content area features a title 'Seurat - Guided Clustering Tutorial'. Below the title, a red oval highlights the text 'Compiled: February 08, 2021' and 'Source: vignettes/pbmc3k\_tutorial.Rmd'. To the right, a large text block states '.Rmd can be converted to ipynb To be handled in Jupyter Notebooks'. Further down, a section titled 'Setup the Seurat Object' provides a brief overview of the dataset and the process of reading in data using the `Read10X()` function. A code block at the bottom shows R code for initializing a Seurat object from a PBMC dataset.

Seurat 4.0.0    Install    Get started    Vignettes ▾    Extensions    FAQ    News    Reference    Archive   

## Seurat - Guided Clustering Tutorial

Compiled: February 08, 2021  
Source: vignettes/pbmc3k\_tutorial.Rmd

(.Rmd can be converted to ipynb  
To be handled in Jupyter Notebooks)

### Setup the Seurat Object

For this tutorial, we will be analyzing the a dataset of Peripheral Blood Mononuclear Cells (PBMC) freely available from 10X Genomics. There are 2,700 single cells that were sequenced on the Illumina NextSeq 500. The raw data can be found [here](#).

We start by reading in the data. The `Read10X()` function reads in the output of the `cellranger` pipeline from 10X, returning a unique molecular identified (UMI) count matrix. The values in this matrix represent the number of molecules for each feature (i.e. gene; row) that are detected in each cell (column).

We next use the count matrix to create a `Seurat` object. The object serves as a container that contains both data (like the count matrix) and analysis (like PCA, or clustering results) for a single-cell dataset. For a technical discussion of the `Seurat` object structure, check out our [GitHub Wiki](#). For example, the count matrix is stored in `pbmc[["RNA"]].@counts`.

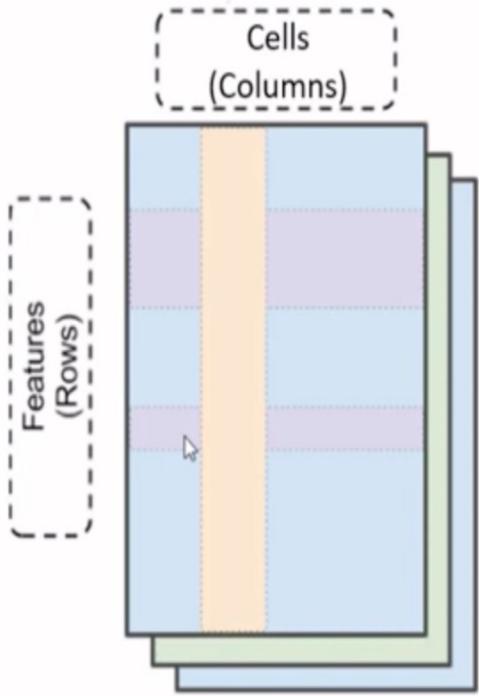
```
library(dplyr)
library(Seurat)
library(patchwork)

# Load the PBMC dataset
pbmc.data <- Read10X(data.dir = ".../data/pbmc3k/filtered_gene_bc_matrices/hg19/")
# Initialize the Seurat object with the raw (non-normalized data).
pbmc <- CreateSeuratObject(counts = pbmc.data, project = "pbmc3k", min.cells = 3, min.features = 20
pbmc
```

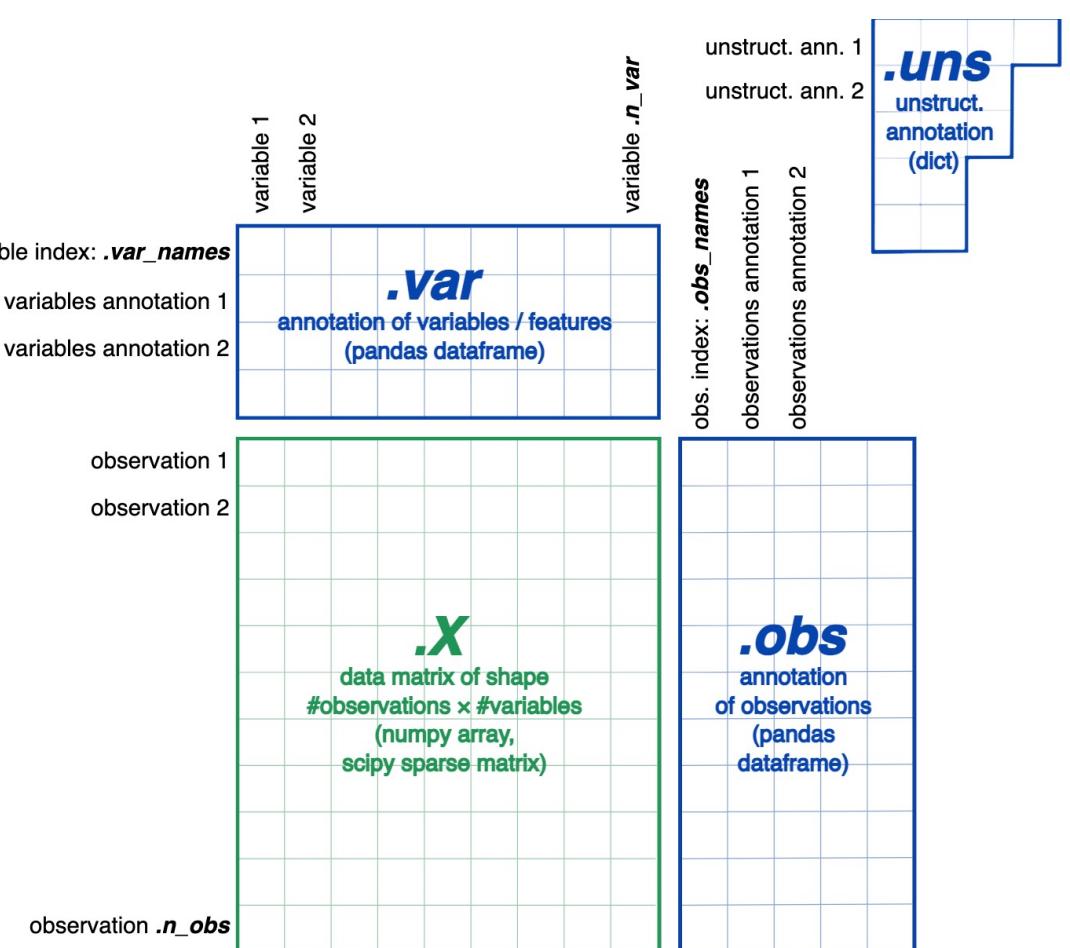
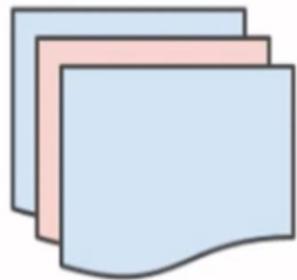
### Contents

- Setup the Seurat Object
- Standard pre-processing workflow
- Normalizing the data
- Identification of highly variable features (feature selection)
- Scaling the data
- Perform linear dimensional reduction
- Determine the 'dimensionality' of the dataset
- Cluster the cells
- Run non-linear dimensional reduction (UMAP/tSNE)
- Finding differentially expressed features (cluster biomarkers)
- Assigning cell type identity to clusters

# R vs Python, Object-Oriented Programming (OOP)



Seurat Object in Seurat



Anndata Object in Scanpy

# Scale up with python implementations

- Python packages/toolkits are increasingly popular
  - scanpy pipeline
  - scVelo pipeline
- Some has a R rapper.



- Use python in R through Reticulate
- Use R in python through rpy2



## Scanpy vs. Seurat

Satija et al., Nat. Biotechn. (2015)

Scanpy is benchmarked with Seurat.

- preprocessing: <1 s vs. 14 s
- regressing out unwanted sources of variation: 6 s vs. 129 s
- PCA: <1 s vs. 45 s
- clustering: 1.3 s vs. 65 s
- tSNE: 6 s vs. 96 s
- marker genes (approximation): 0.8 s vs. 96 s



stable

Search docs

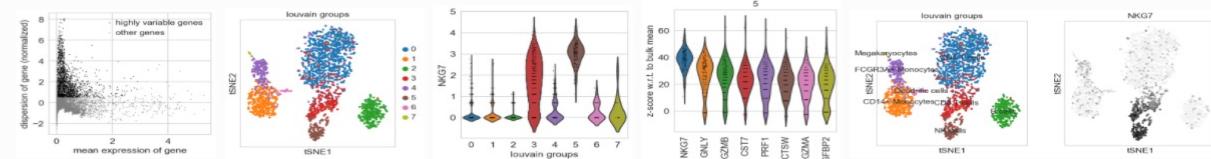
Tutorials

- Clustering
- Visualization
- Trajectory inference
- Integrating datasets
- Spatial data
- Further Tutorials
- Usage Principles
- Installation
- API
- External API
- Ecosystem
- Release notes
- News
- Contributing
- Contributors
- References

## Tutorials

### Clustering

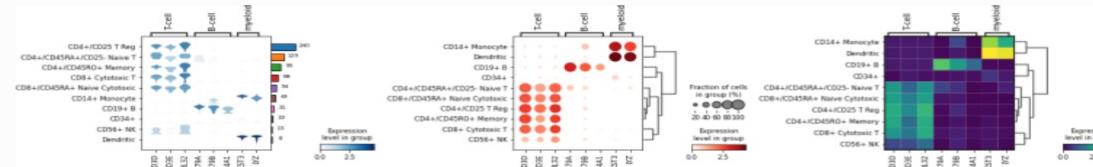
For getting started, we recommend Scanpy's reimplementation [→ tutorial: pbmc3k](#) of Seurat's [\[Satija15\]](#) clustering tutorial for 3k PBMCs containing preprocessing, clustering and the identification of cell types via known marker genes.



(.ipynb format can be converted to .RMD to be run in Rstudio)

### Visualization

This tutorial shows how to visually explore genes using scanpy. [→ tutorial: plotting/core](#)



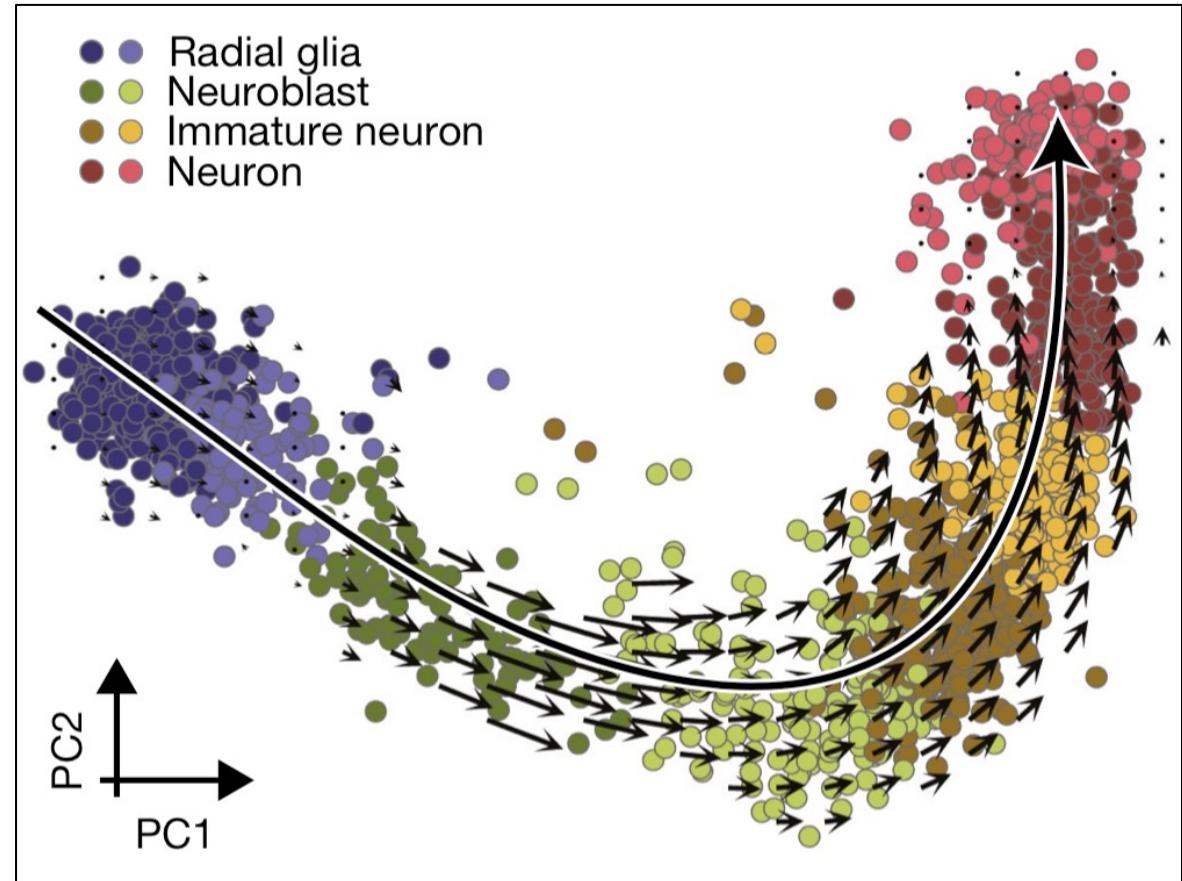
### Trajectory inference

Get started with the following example for hematopoiesis for data of [\[Paul15\]](#): [→ tutorial: paga-paul15](#)



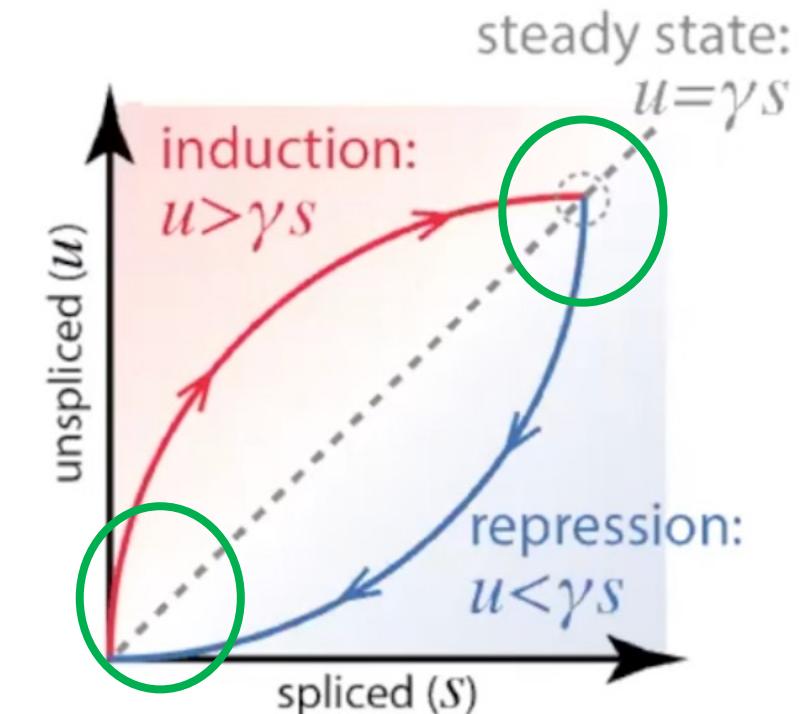
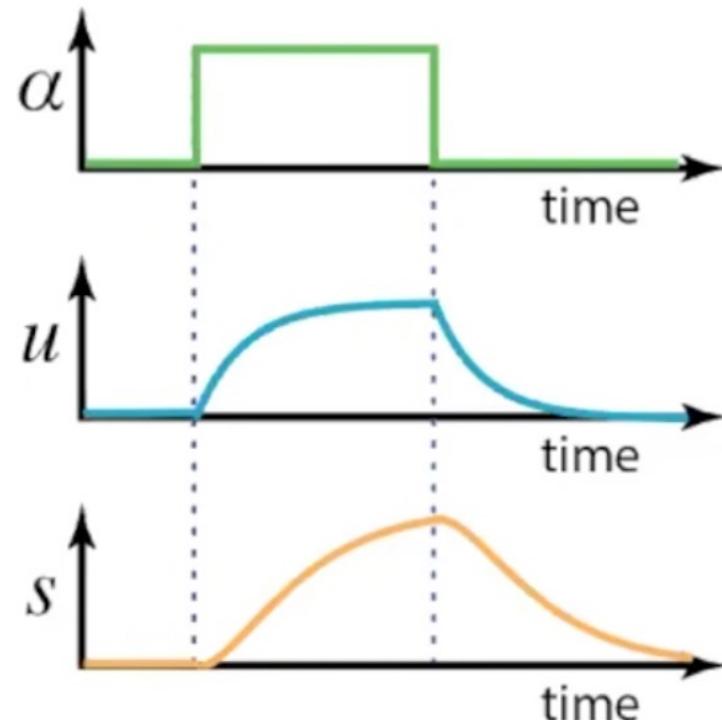
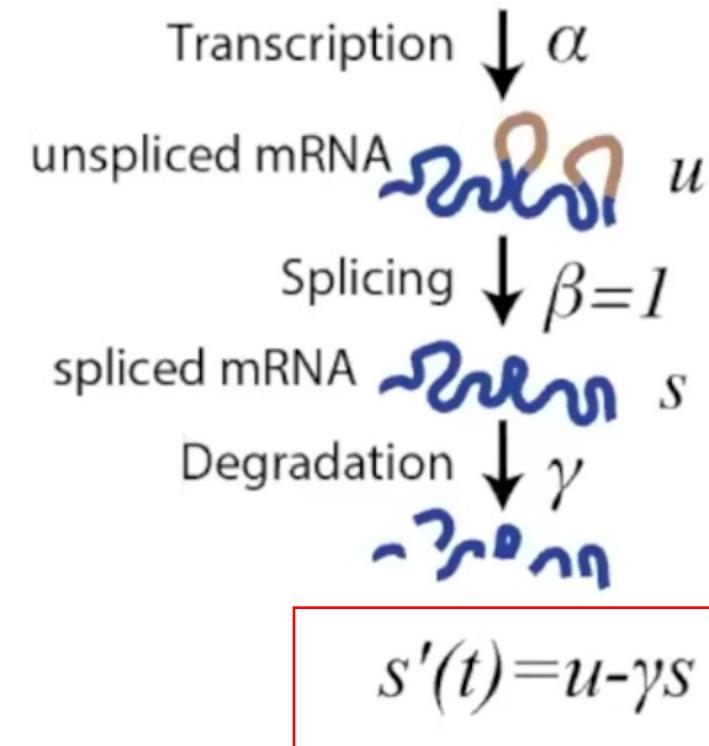
# Trajectory analysis by RNA velocity

- When cells differentiate, **new genes** will start to be expressed
- Transcripts have introns and will be spliced off given time
- Through assessing the present percentage of reads in introns, increase or decrease of expression can be modeled



<https://liorpachter.wordpress.com/tag/velocyto/>  
<http://pklab.med.harvard.edu/software.html>

# Modeling RNA dynamics



**Concept of RNA velocity**

$$\frac{du(t)}{dt} = \alpha - \beta u(t), \quad \frac{ds(t)}{dt} = \beta u(t) - \gamma s(t)$$

**Steady-state model (velocyto)**

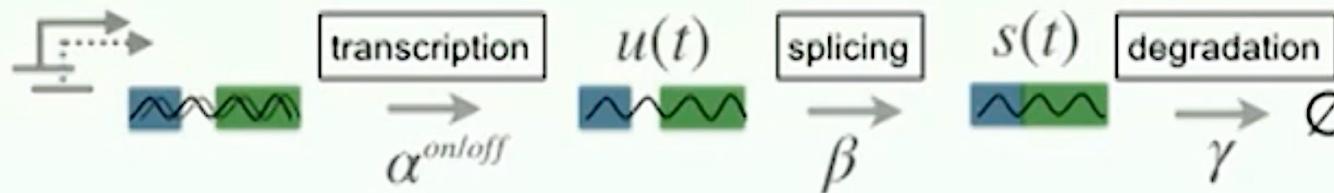
- Fit lin.reg. on extreme quantile cells (steady states)
- Estimate velocities as deviation from steady state

$$u_\infty \approx \gamma' s_\infty \quad (\beta = 1)$$

$$v_i = u_i - \gamma' s_i$$

2 assumptions:  
steady states has been observed  
a constant splicing rate  $\beta$  across all RNA

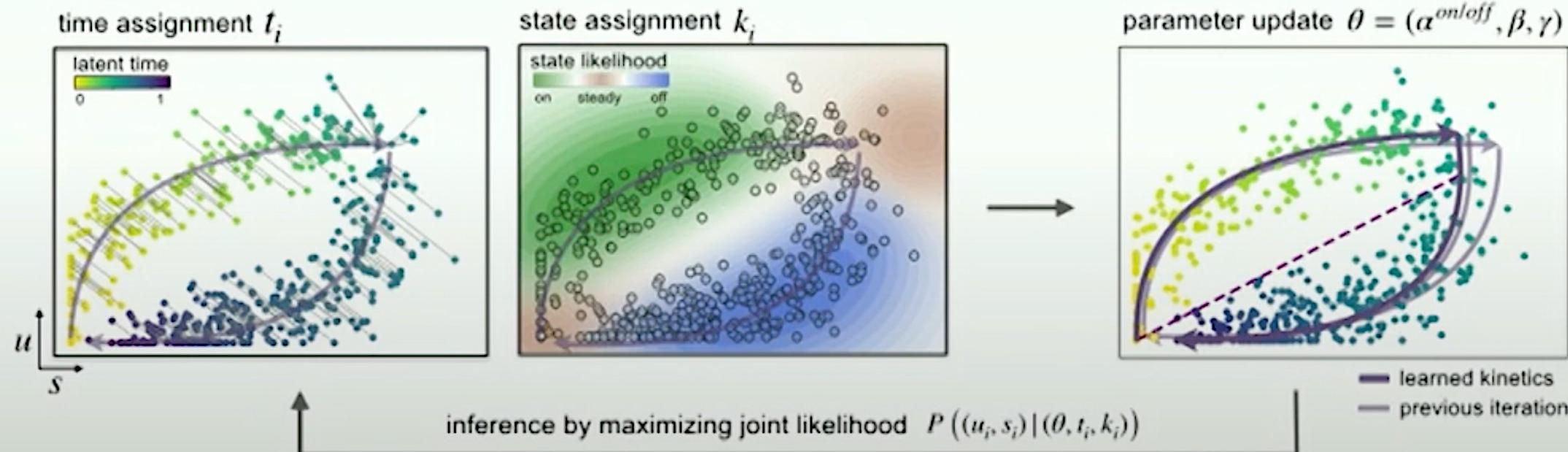
# Generalizing RNA velocity to dynamical populations



$$u(t) = u_0 e^{-\beta t} + \frac{\alpha}{\beta} (1 - e^{-\beta t})$$

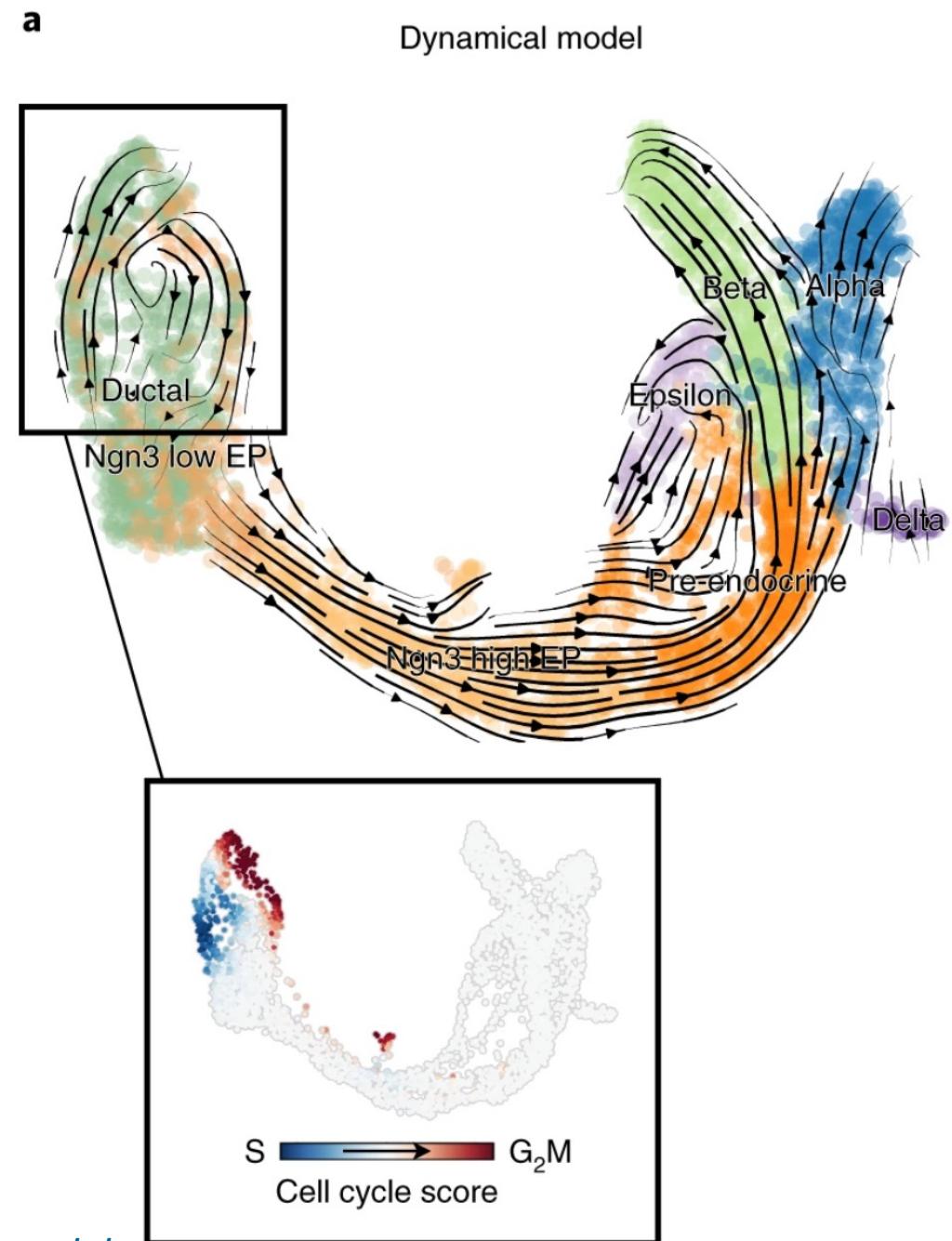
$$s(t) = s_0 e^{-\gamma t} + \frac{\alpha}{\gamma} (1 - e^{-\gamma t}) + \frac{\alpha - \beta u_0}{\gamma - \beta} (e^{-\gamma t} - e^{-\beta t}) \quad \tau = t - t_0$$

parameters of **reaction rates**  $\theta = (\alpha^{off}, \alpha^{on}, \beta, \gamma)$   
 cell-specific **latent variables**  $\eta_i = (t_0^{(i)}, t_i, k_i)$   
 (switch, time, state)



# Steps of RNA-velocity

- From cellranger produced bam files
  - Sort bam files
  - Using Velocyto CLI to identify exon/intron reads as loom files.
  - Use scVelo to model the data and recover RNA dynamics
  - Through Markov process to predict which neighbor is the most probable destiny for the cell.
  - Backtracking and forward tracking to get the destiny and ancestor of the cell.



[Nature, 2018. invented the concept](#)

[Nature Biotechnology, 2020. a generalized model](#)

# Get information about spliced/unspliced transcripts from bam files

#Build pipe line and shared scripts and instruction to researchers

[https://github.com/zhuy16/single-cell-RNA-seq/tree/master/scvel\\_notebooks](https://github.com/zhuy16/single-cell-RNA-seq/tree/master/scvel_notebooks)

Step 1, finding reads mapped to un-spliced regions.

#Sort BAM files with Barcodes

```
samtools sort -t CB -O BAM -o cellsorted_possorted_genome_bam.bam possorted_genome_bam.bam
```

#Analyze reads from spliced/non-spliced transcripts,

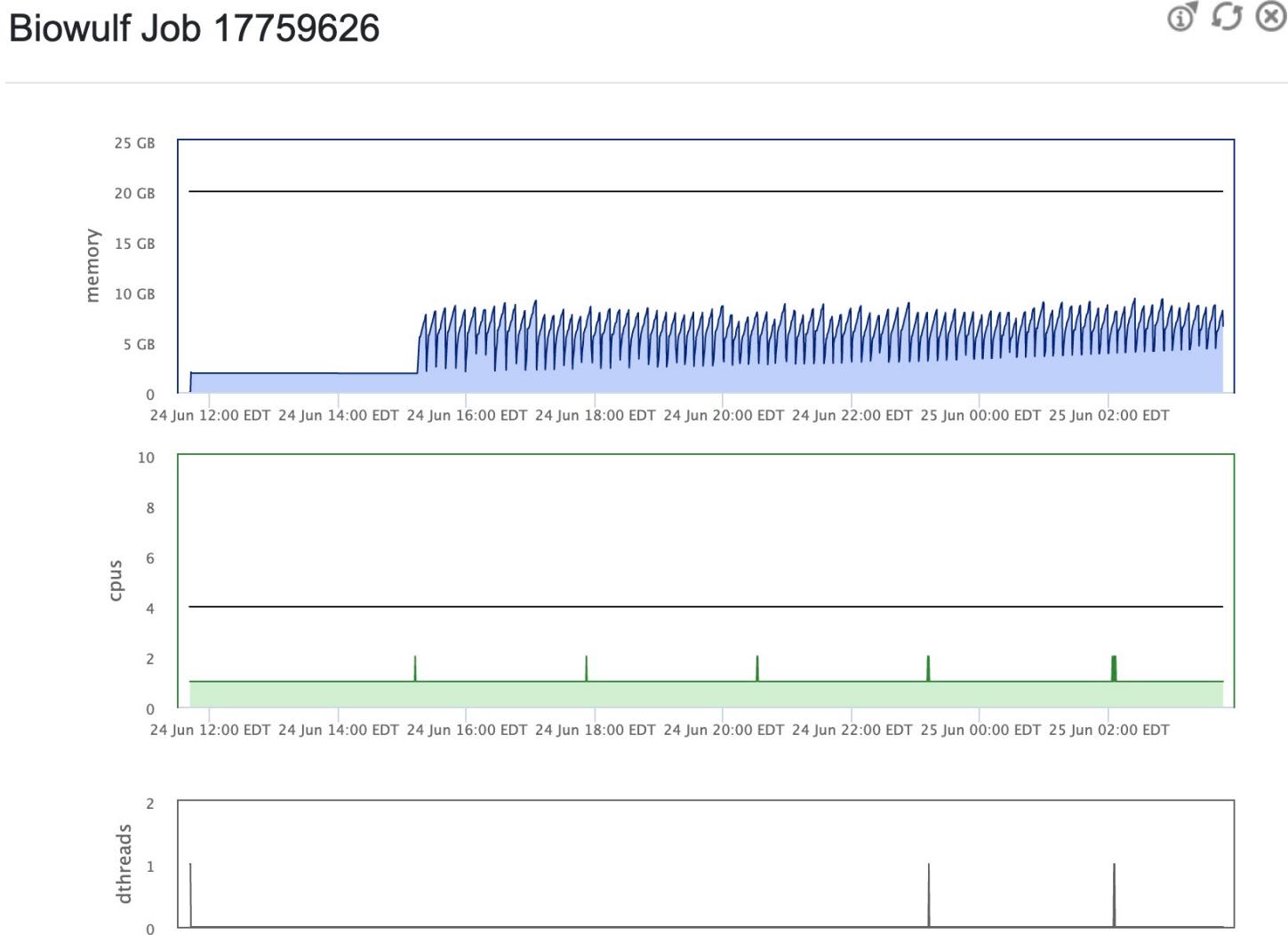
module load python/3.7.3-foss-2016b

velocyto run10x cellranger\_output\_folder genes.gtf

```
cellsorrted_possorted_genome_bam.bam.tmp.0000.bam  
cellsorrted_possorted_genome_bam.bam.tmp.0001.bam  
cellsorrted_possorted_genome_bam.bam.tmp.0002.bam  
cellsorrted_possorted_genome_bam.bam.tmp.0003.bam  
cellsorrted_possorted_genome_bam.bam.tmp.0004.bam  
cellsorrted_possorted_genome_bam.bam.tmp.0005.bam  
cellsorrted_possorted_genome_bam.bam.tmp.0006.bam  
cellsorrted_possorted_genome_bam.bam.tmp.0007.bam  
cellsorrted_possorted_genome_bam.bam.tmp.0008.bam  
cellsorrted_possorted_genome_bam.bam.tmp.0009.bam  
cellsorrted_possorted_genome_bam.bam.tmp.0010.bam  
cellsorrted_possorted_genome_bam.bam.tmp.0011.bam  
cellsorrted_possorted_genome_bam.bam.tmp.0012.bam
```

```
cellsorrted_possorted_genome_bam.bam.tmp.0509.bam  
cellsorrted_possorted_genome_bam.bam.tmp.0510.bam  
cellsorrted_possorted_genome_bam.bam.tmp.0511.bam  
cellsorrted_possorted_genome_bam.bam.tmp.0512.bam  
cellsorrted_possorted_genome_bam.bam.tmp.0513.bam  
cellsorrted_possorted_genome_bam.bam.tmp.0514.bam  
cellsorrted_possorted_genome_bam.bam.tmp.0515.bam
```

# A example for velocityo CLI job – half a day



# Demo on RNA velocity

# Summary of day 1, scRNA-seq

- Set up the working environment
  - Login and pseudolinks to data
  - Copy files to scratch drive
  - Evoke jupyter lab IDE
- Introduction to scRNA-seq
- Make fastq from bcl files
- Use cellranger count to do alignment
- Use Seurat/scanpy pipeline to do QC and dimension reduction
- Use RNA velocity to infer trajectory
- For tomorrow: please install IGV and IGB for visualization of alignment, on your local machine.

# Reference

- Make fastq and cellranger count
  - <https://davetang.org/muse/2018/08/09/getting-started-with-cell-ranger/>
- Seurat pipeline on pbmc
  - [https://satijalab.org/seurat/articles/pbmc3k\\_tutorial.html](https://satijalab.org/seurat/articles/pbmc3k_tutorial.html)
- Scanpy pipeline
  - <https://scanpy-tutorials.readthedocs.io/en/latest/pbmc3k.html>
- RNA velocity
  - velocyto CLI
    - <https://velocyto.org/velocyto.py/tutorial/cli.html>
  - scVelo
    - [Scvelo: https://scvelo.readthedocs.io/VelocityBasics/](https://scvelo.readthedocs.io/VelocityBasics/)