BIOINFORMATICS PIPELINE FOR SCRNA-SEQ: FROM RAW DATA TO INSIGHTS

Jerry Li Ph.D.

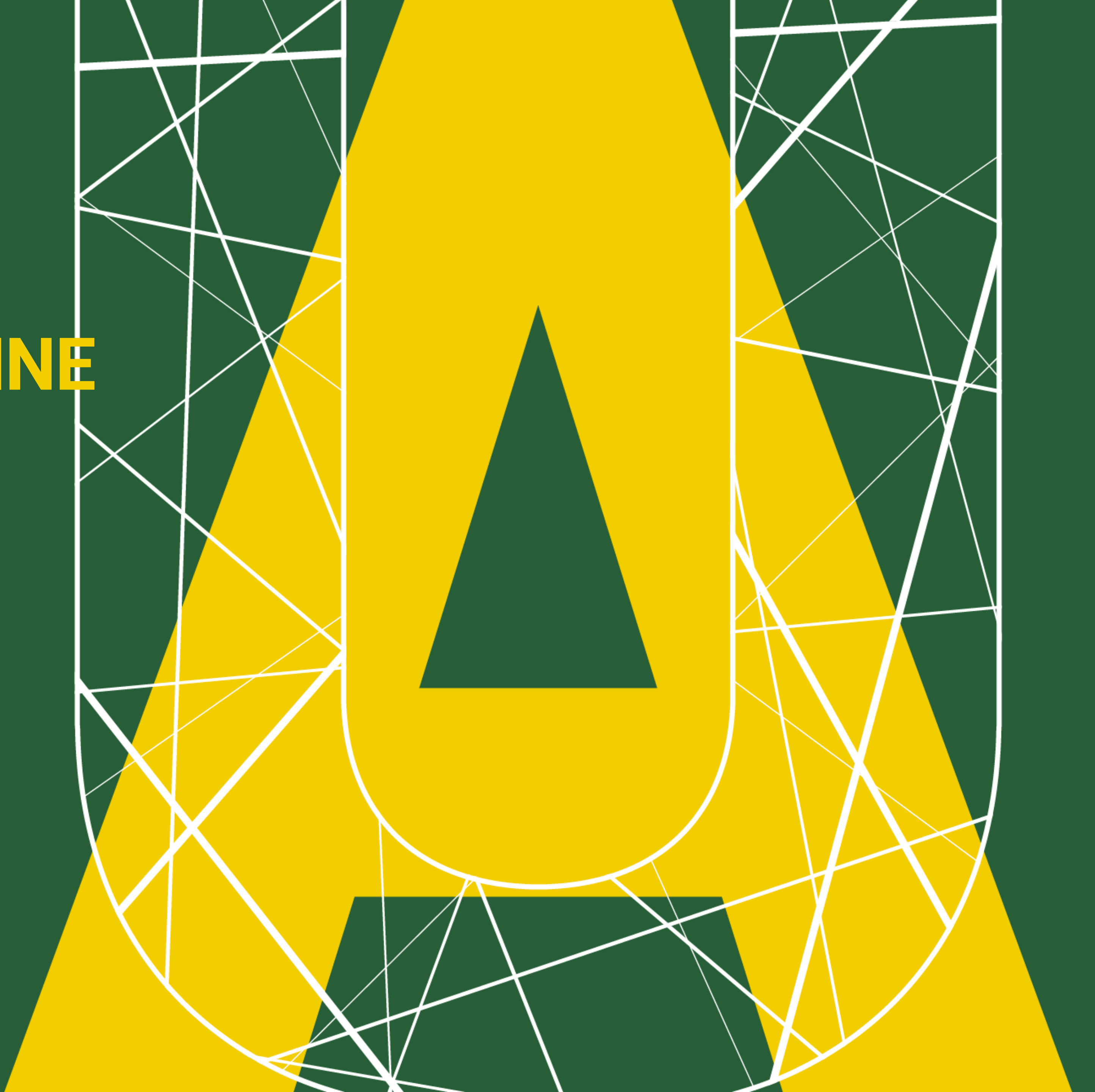
Research Support Analyst

Digital Research Services, IST

jiarui.li@ualberta.ca

Oct 9, 2025

UNIVERSITY
OF ALBERTA



Outline

- Introduction
- Input Data preprocessing
- QC
- Normalized expression
- Clustering
- Marker genes and cell-type annotation

Objectives

- Understand the principles and workflow of single-cell RNA sequencing (scRNA-seq)

- Learn the importance of quality control in scRNA-seq and the rationale behind it

- Gain hands-on experience running one of the most widely used analysis pipelines

Note

- The slides can be found in Github:
  https://github.com/ualberta-rcg/scRNA-seq

- Apptainer container and sample FASTQ:
  https://drive.google.com/drive/folders/18vXOcOPEPUGW85fM6ZbCxKQJQuHnanQD

- Workshop Cluster
  https://tinyurl.com/UofA-scRNA-seq

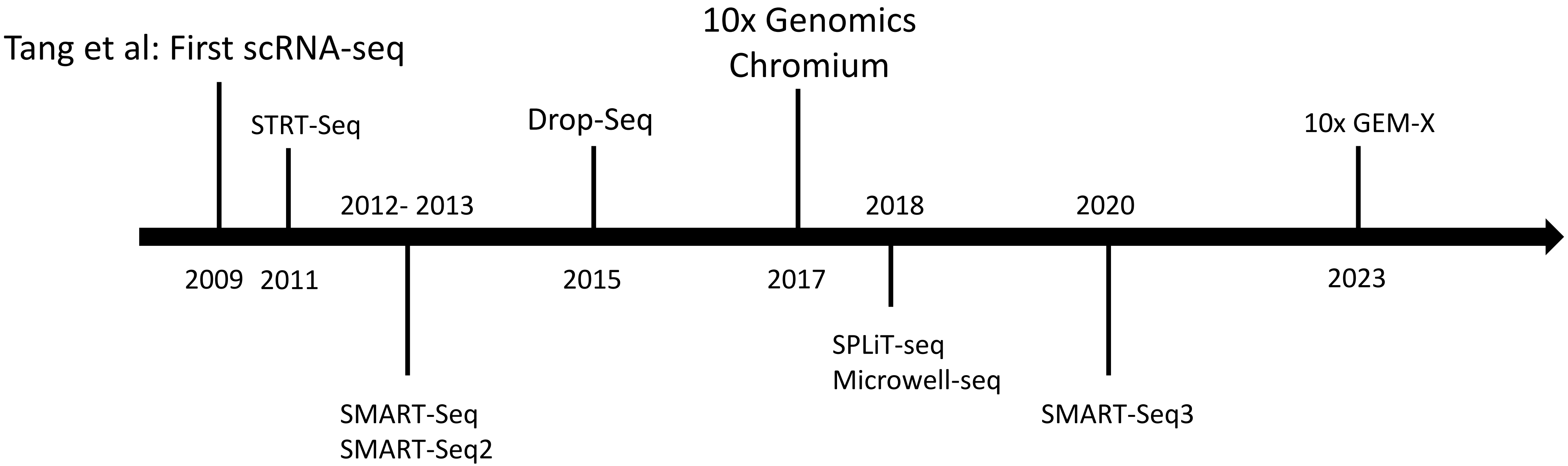
Login by ssh

- Our workshop cluster

```
ssh user000@fall2025-uofa.c3.ca
```

Introduction

The Milestone of scRNA-seq



2009 Tang et al: First scRNA-seq

2011 STRT-Seq

2012- 2013 SMART-Seq SMART-Seq2

2015 Drop-Seq

2017 10x Genomics Chromium

2018 SPLiT-seq Microwell-seq

2020 SMART-Seq3

2023 10x GEM-X

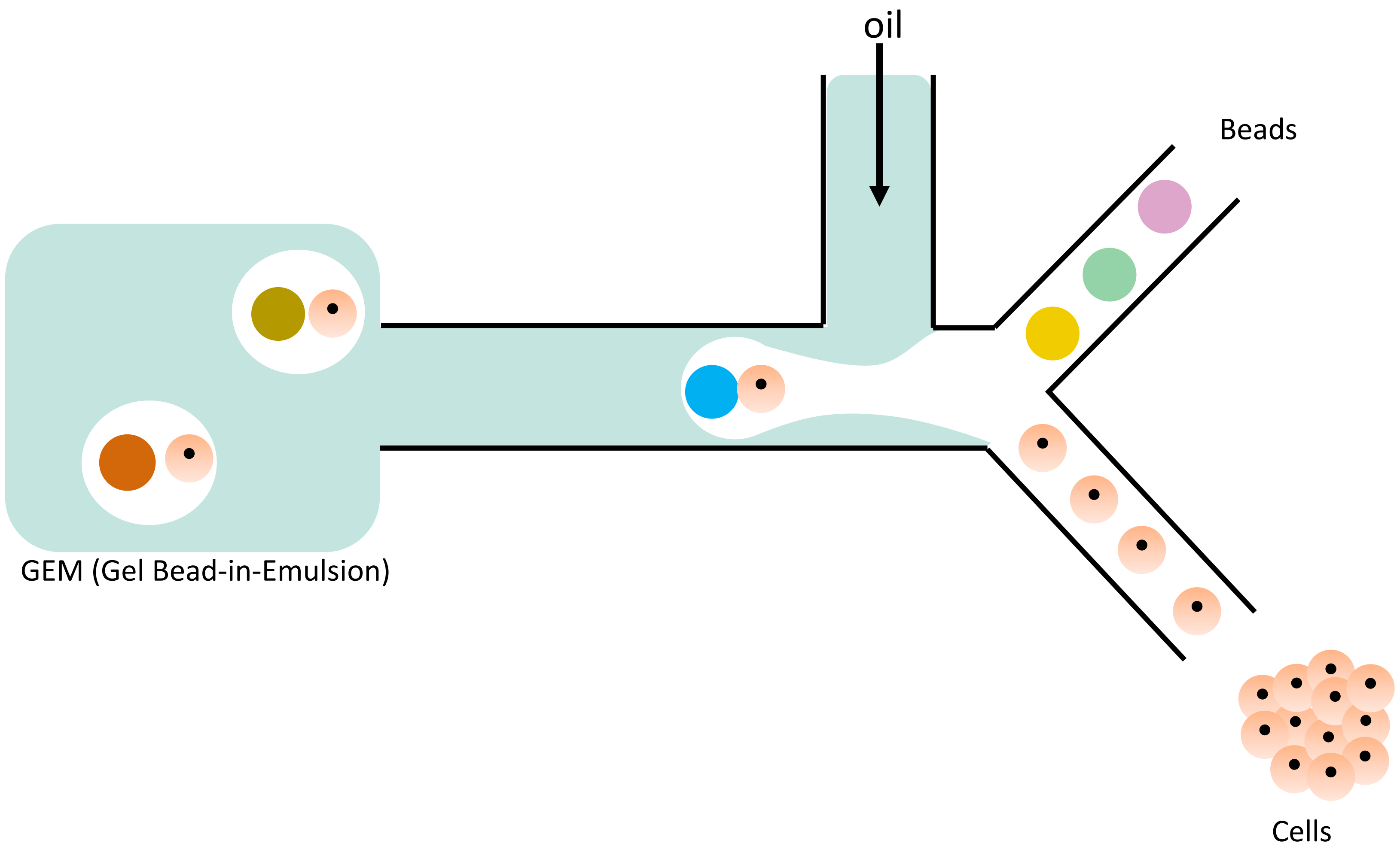
scRNA-seq Wet Lab Pipeline

Protocol	Type	Transcript Coverage	UMI Support	Throughput	Cost per Cell	Special Features / Use Cases
10x Chromium	Droplet-based	3’ end or 5’ end	Yes	Very High (> 1M cells)	$0.10–$0.50	Standardized, reliable and reproducible
SPLiT-seq	Plate-based	3’ end	YES	Very High	~$0.01	Cost-friendly
Microwell-seq	Microwell-based	3’ end	Yes	Medium	$0.01-$0.05	Used in Mouse Cell Atlas; optimized for bulk processing

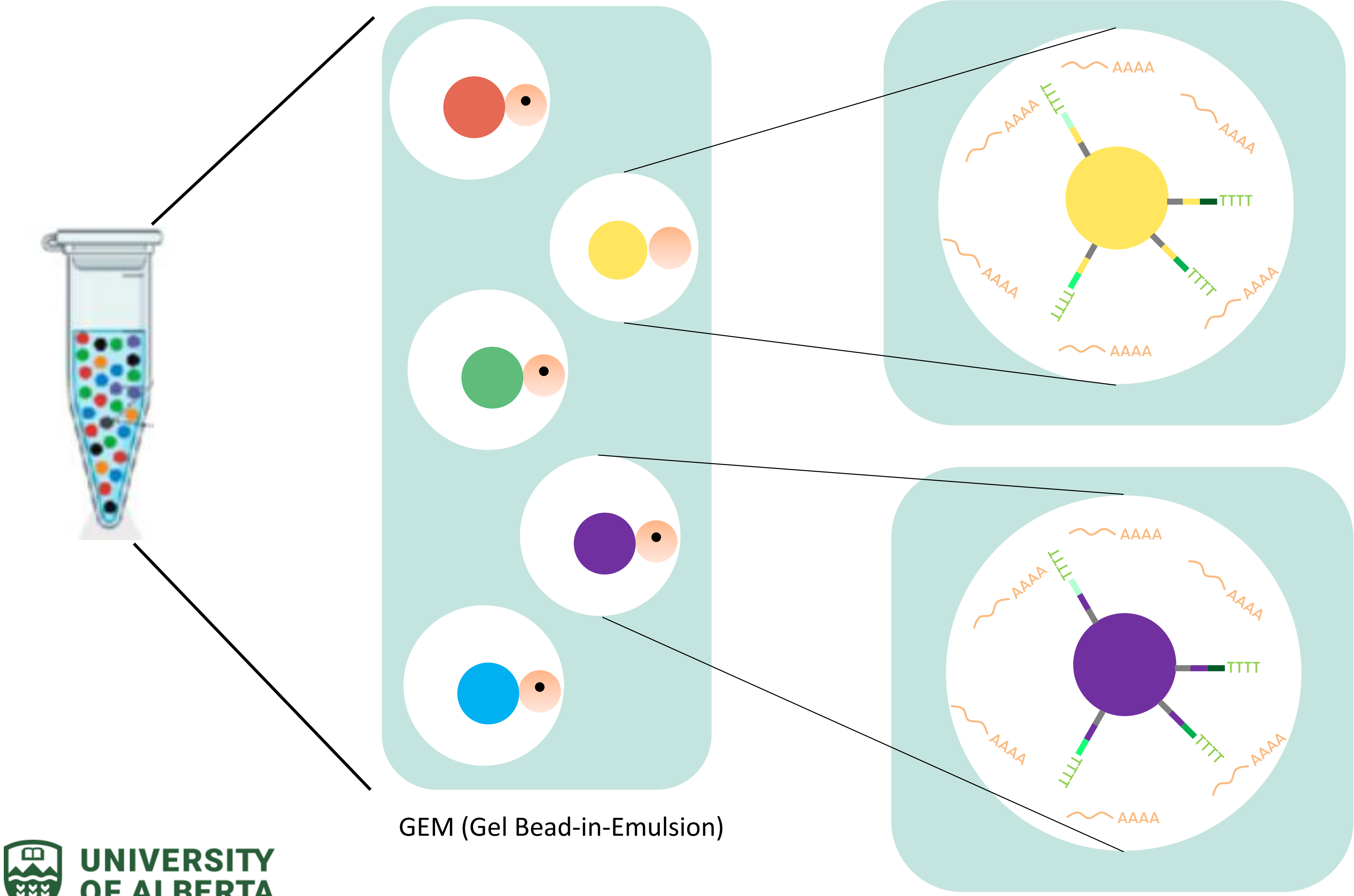
Why 10x Genomics is preferred over others?

- Fully automated
- Consistent and well support
- Can be integrated to multi-modal data such as Assay for Transposase-Accessible Chromatin (ATAC)
- Ease to use and no need custom setup

10x Chromium 3’ scRNA-seq – GEM Formation



10x Chromium 3’ scRNA-seq



GEM (Gel Bead-in-Emulsion)

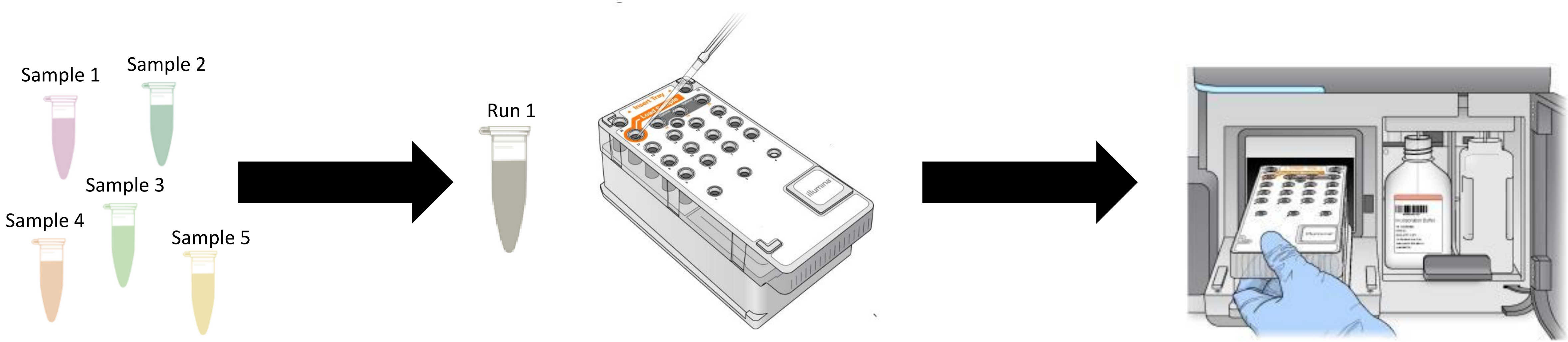
Reverse Transcription

ds-cDNA synthesis

PCR with sample-specific primers

UNIVERSITY OF ALBERTA

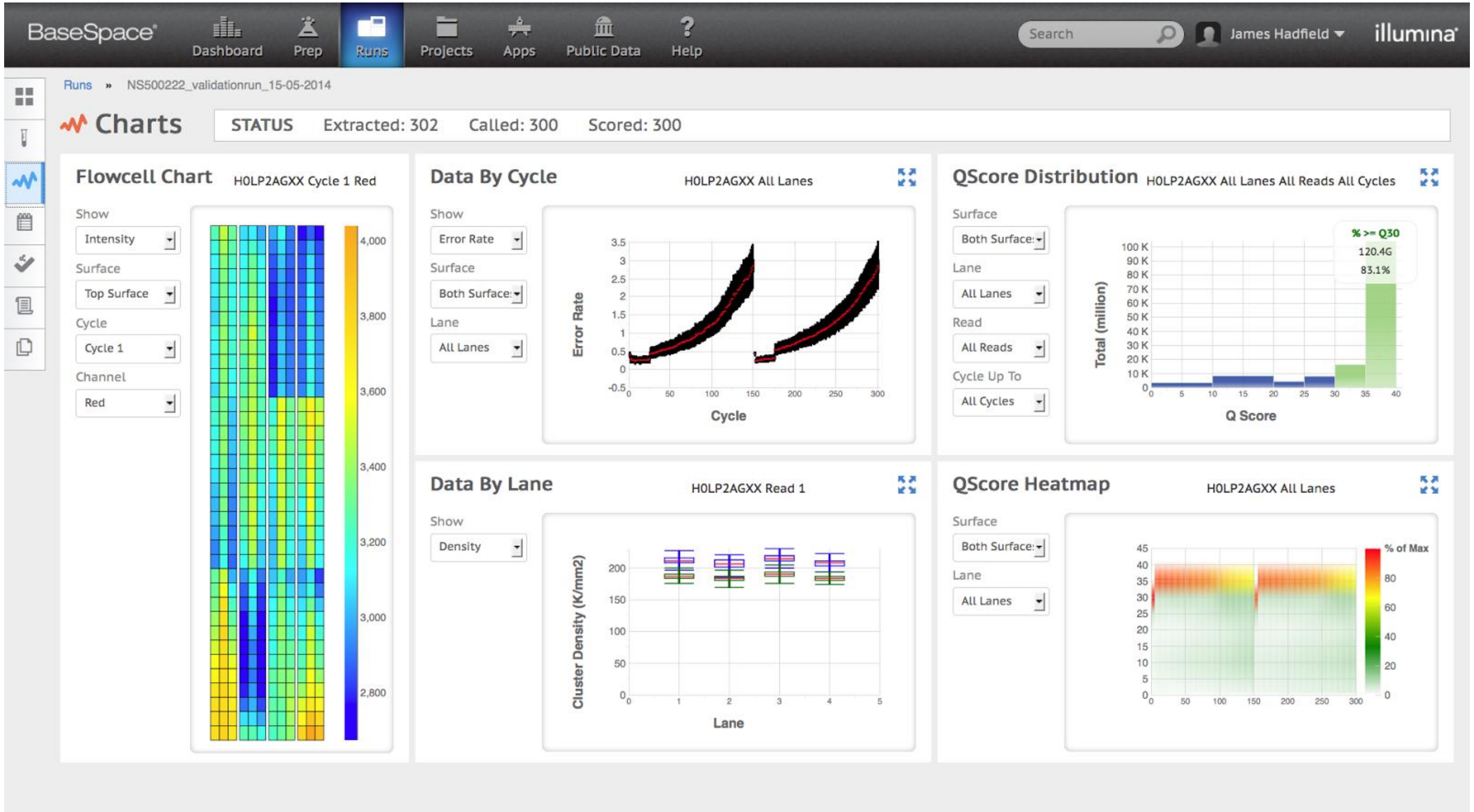
10x Chromium 3’ scRNA-seq



UNIVERSITY OF ALBERTA

illumina

10x Chromium 3’ scRNA-seq – Sequencing Report



Runs » NS500222_validationrun_15-05-2014

Charts

STATUS Extracted: 302 Called: 300 Scored: 300

Flowcell Chart H0LP2AGXX Cycle 1 Red

Show: Intensity
Surface: Top Surface
Cycle: Cycle 1
Channel: Red

Data By Cycle H0LP2AGXX All Lanes

Show: Error Rate
Surface: Both Surfaces
Lane: All Lanes

QScore Distribution H0LP2AGXX All Lanes All Reads All Cycles

Surface: Both Surfaces
Lane: All Lanes
Read: All Reads
Cycle Up To: All Cycles

Data By Lane H0LP2AGXX Read 1

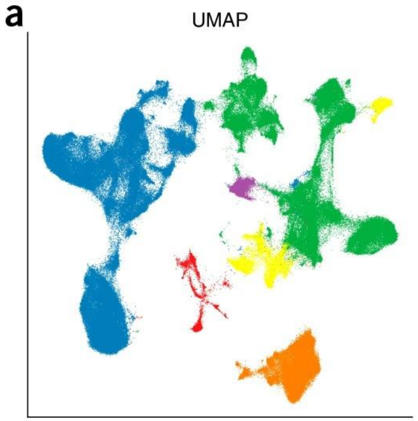
Show: Density

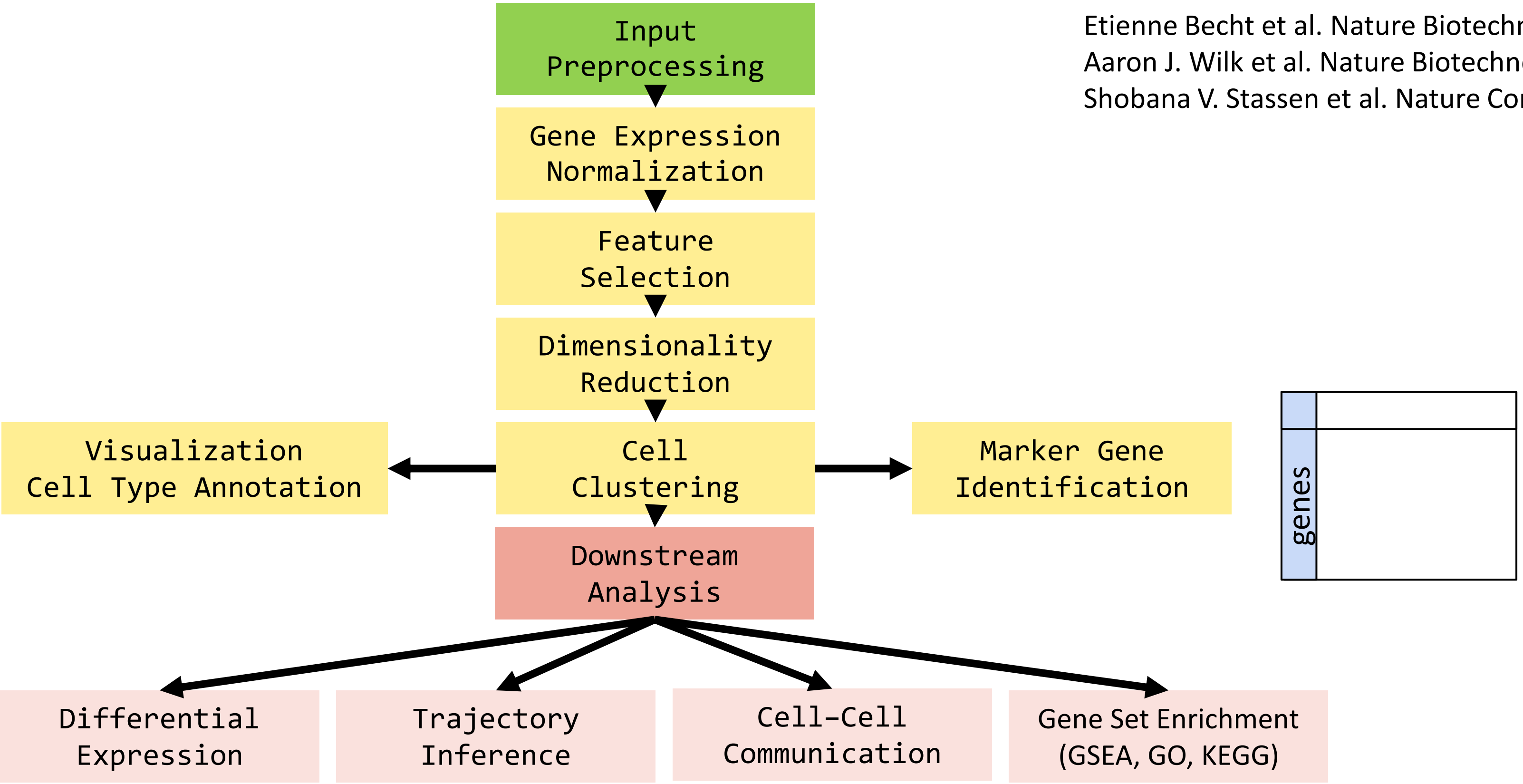
QScore Heatmap H0LP2AGXX All Lanes

Surface: Both Surfaces
Lane: All Lanes

The Pipeline of scRNA-seq Analysis

Etienne Becht et al. Nature Biotechnology. 37: 38–44 (2019)
Aaron J. Wilk et al. Nature Biotechnology. 42: 470–483 (2024)
Shobana V. Stassen et al. Nature Communications. 12: 5528 (2021)




Input Preprocessing

Gene Expression Normalization

Feature Selection

Dimensionality Reduction

Cell Clustering

Visualization Cell Type Annotation

Marker Gene Identification

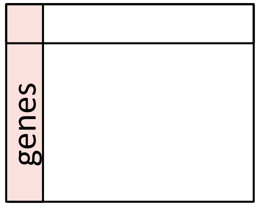
Downstream Analysis

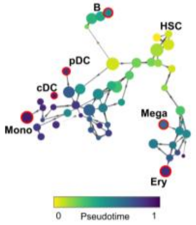
Differential Expression

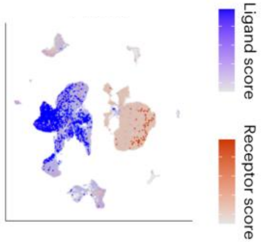
Trajectory Inference

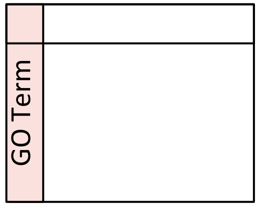
Cell-Cell Communication

Gene Set Enrichment (GSEA, GO, KEGG)



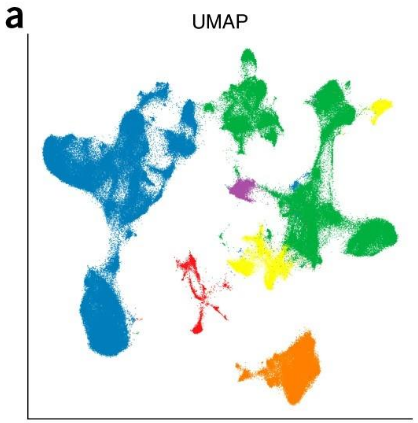


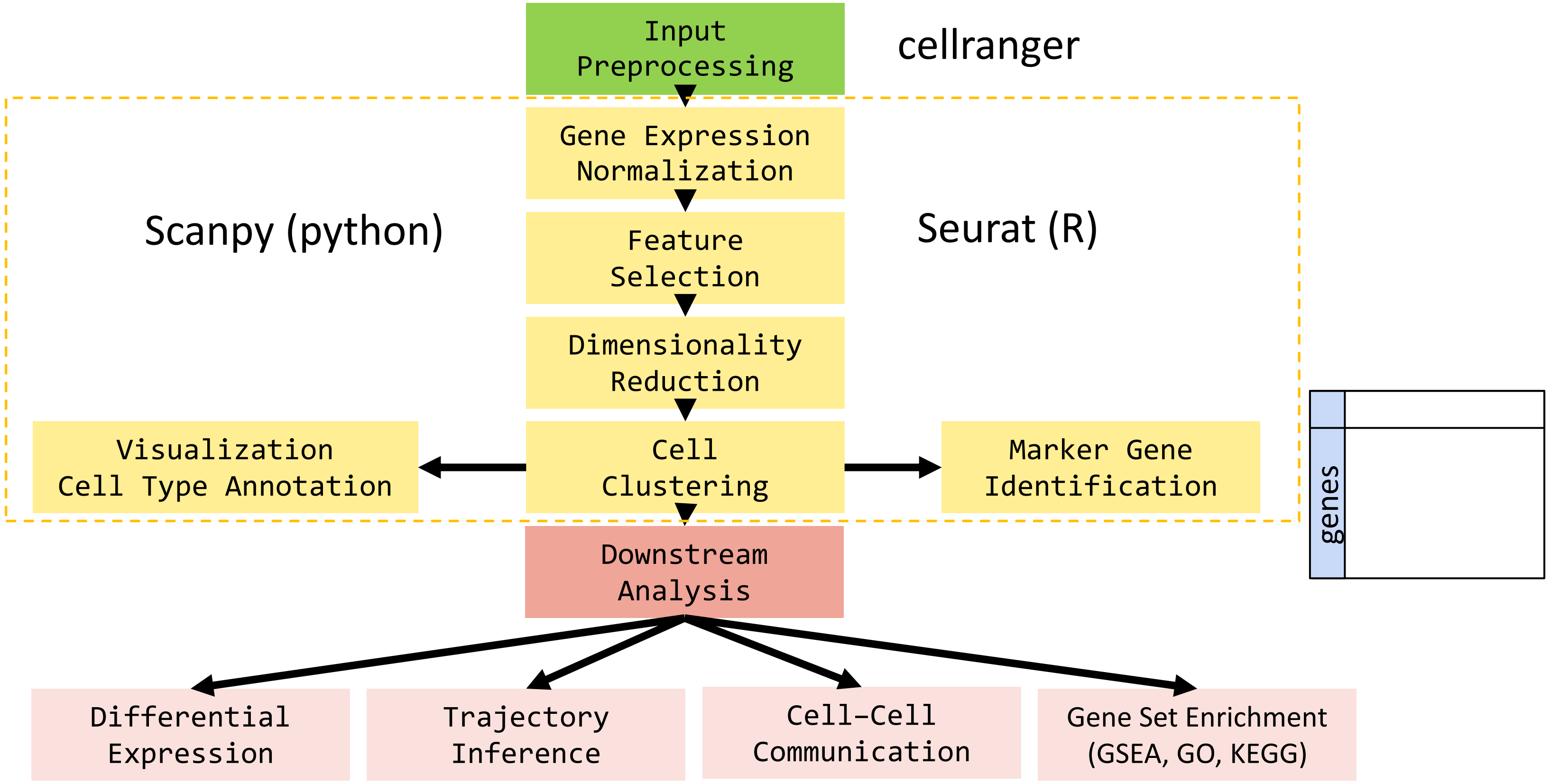




UNIVERSITY OF ALBERTA

The Pipeline of scRNA-seq Analysis




Input Preprocessing

cellranger

Gene Expression Normalization

Feature Selection

Dimensionality Reduction

Scanpy (python)

Seurat (R)

Visualization Cell Type Annotation

Cell Clustering

Marker Gene Identification

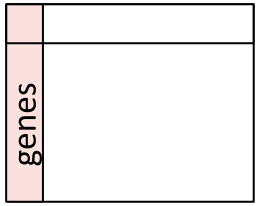
Downstream Analysis

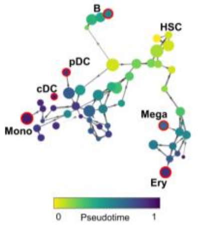
Differential Expression

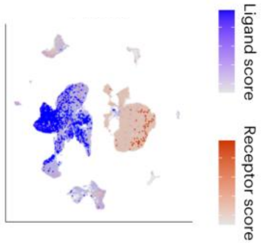
Trajectory Inference

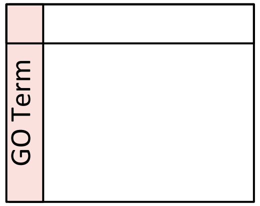
Cell-Cell Communication

Gene Set Enrichment (GSEA, GO, KEGG)







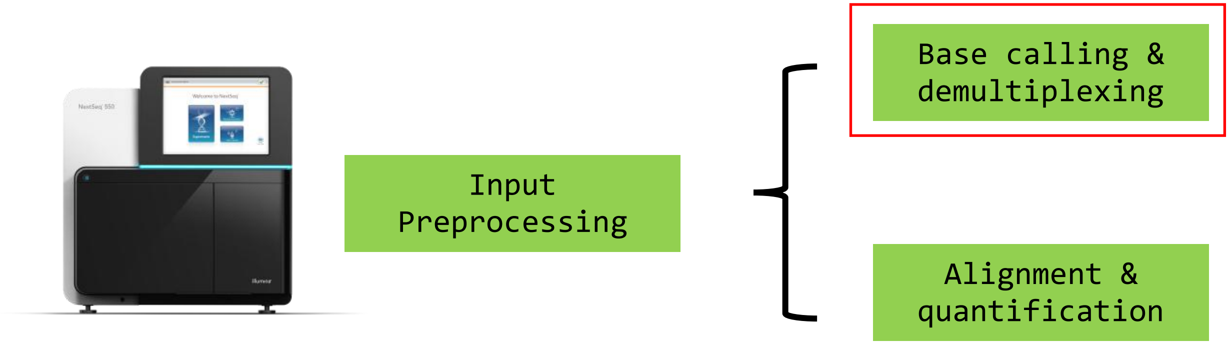


Question?

Input Processing

UNIVERSITY
OF ALBERTA

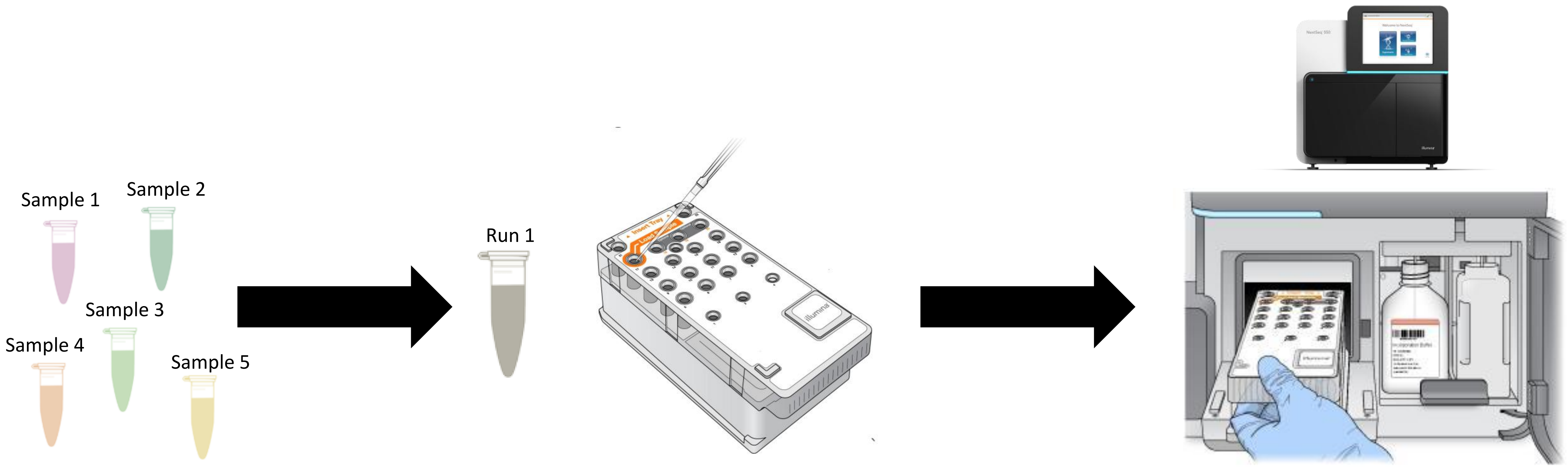
Input Preprocessing by Cell Ranger



Optional Tools:

Cell Ranger	**Most popular**
STARsolo	Open source so you can adjust the parameters, like the tolerance of mismatches
Alevin	Super fast

10x Chromium 3’ scRNA-seq



Base calling & demultiplexing

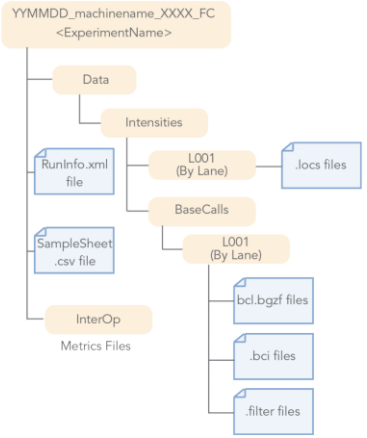


Figure 2 BCL Conversion Input Files from the MiniSeq or NextSeq System

SampleName1_S1_L001_R1_001.fastq.gz 10x barcode+UMI
SampleName1_S1_L001_R2_001.fastq.gz cDNA sequence
SampleName1_S1_L001_I1_001.fastq.gz index sequence

SampleName2_S2_L001_R1_001.fastq.gz 10x barcode+UMI
SampleName2_S2_L001_R2_001.fastq.gz cDNA sequence
SampleName2_S2_L001_I1_001.fastq.gz index sequence

Split chunk of reads (4M in default)

Sample Name (given by you when
filling the sample sheet

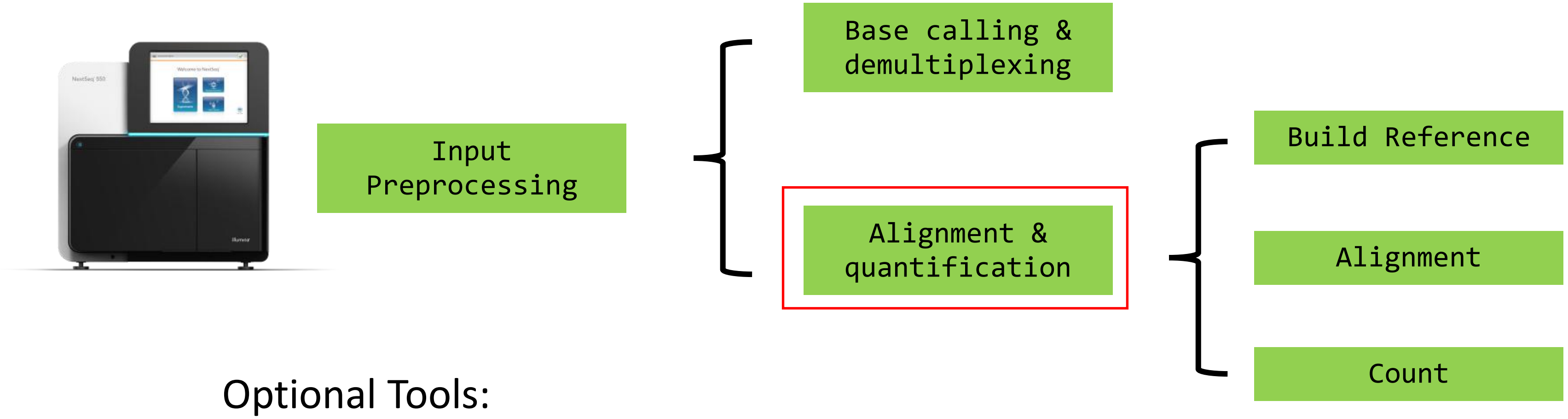
```
cellranger mkfastq \
  --id=sample_name_fastq \
  --run=/path/to/bcl_folder \
  --csv=sample_sheet.csv \
  --output-dir=/path/to/fastq/
```

Base calling & demultiplexing

- You need to do base calling & demultiplexing **ONLY WHEN** you have 10x genomics machine and sequencer.
- However, if you do need to run this step, here is a guide of computing power needed for your reference

Run size (rough input)	Typical reads	CPUs to request	RAM to request	Ballpark runtime*
Small (MiSeq/mini-runs; BCL ~30–50 GB)	~20–40 M	**8** cores	**16–32 GB**	**0.5–1.5 h**
Medium (NextSeq lane; BCL ~60–120 GB)	~80–150 M	**16** cores	**32–48 GB**	**1–3 h**
Large (NovaSeq X/6k/10k lane; BCL ~200–400 GB)	~300–600 M	**24–32** cores	**48–64 GB**	**2–6 h**
Very large (≥2 lanes; BCL ~400–800 GB)	~0.6–1.2 B	**32–48** cores	**64–128 GB**	**4–10 h**

Input Preprocessing by Cell Ranger



Input Preprocessing

Base calling & demultiplexing

Alignment & quantification

Build Reference

Alignment

Count

Optional Tools:

Cell Ranger	**Most popular**
STARsolo	Open source so you can adjust the parameters, like the tolerance of mismatches
Alevin	Super fast

Install Cellranger

https://www.10xgenomics.com/support/software/cell-ranger/downloads#download-links

Download and unzip

```
cd
cp -r projects/def-sponsor00/scRNA-seq/Sample_FASTQ/ .
```

Alignment & quantification – Build The Reference

```
cd
cp -r projects/def-sponsor00/scRNA-seq/Sample_FASTQ/ .
export PATH=/project/def-sponsor00/scRNA-seq/cellranger-9.0.1/bin:$PATH
cd Sample_FASTQ/ref
cellranger mkref --genome tp53 --fasta tp53.fa --genes tp53.gtf
```

Tips & best practices:

- For human reference genome: https://support.10xgenomics.com/single-cell-gene-expression/software/downloads/latest
- For big genome that 10x Genomics doesn’t have, run this step in a batch job

Genome (approx size)	Examples	CPUs to request	RAM to request	Runtime (ballpark)
Bacteria/virus (≤10 Mb)	E. coli, SARS-CoV-2	2–4	1–2 GB	minutes
Fish/amphib (~1–2 Gb)	Zebrafish	8	16–24 GB	~30–90 min
Human/Mouse (~3 Gb)	GRCh38, GRCm39	8–16	**32 GB**	**~1–3 h**
Large plant/complex (6–20 Gb)	Maize, wheat	16–32	**64–128 GB**	6-24 hours

Alignment & quantification

```
cd $HOME/Sample_FASTQ
vi cellranger.sh            # edit the job name
```

```
cellranger count \
  --id output_tp53 \                                    # Name of the output folder
  --transcriptome ref/tp53 \                            # Path to reference genome built for Cell Ranger
  --fastqs fastq \                                      # Folder containing FASTQ files
  --sample tp53test \                                   # Sample ID in FASTQ file names
  --localcores 1 \                                      # Number of CPU cores
  --localmem 2  \                                       # Amount of memory (GB)
  --create-bam true \                                   # whether create a bam file
  --chemistry=SC3Pv4                                    # only for testing in this case, delete it when
                                                          you run your analysis
```

```
sbatch cellranger.sh
```

The job will take a few minutes to finish

Alignment & quantification – check the output

```
cd $HOME/Sample_FASTQ/output_tp53
du -h | tail -1
du -h ../fastq
```

The output could be 1000x larger than the input!

Factors affect the output size:

1. Number of cells
2. Sequencing depth
3. Size of the reference genome
4. Introns included?
5. Bam files generated?

Alignment & quantification – Sequencing Depth

For typical human tissue studies	
Cells per sample	5,000 – 20,000
Reads per cell	20,000 – 100,000
Total reads per sample	100M – 1B
Genes detected per cell	A few hundreds to thousands

Factors affecting the sequencing depth:

- Transcriptome size and complexity
- The nature of sample: FFPE, fresh, frozen?
- Genes of interest: highly or lowly expressed? Isoform-specific transcripts?
- Your budget

Alignment & quantification – Computational Resources

Cells × Reads	Cores	RAM	Time
~3k cells, 100M reads	8 cores	32 GB	~1–1.5 hrs
~10k cells, 500M reads	16 cores	64 GB	~2–4 hrs
~50k cells, 1B+ reads	32 cores	128 GB	4–8+ hrs

Understand The Output

```
cd $HOME/Sample_FASTQ/output_tp53/outs
```

Gene Expression Profile

sample_raw_feature_bc_matrix

sample_filetered_feature_bc_matrix

Summary and Quality Assessment

web_summary.html

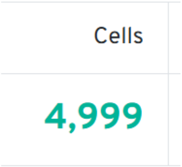
matrix_summary.csv

https://github.com/ualberta-rcg/scRNA-seq

Go to Google Drive link -> scRNA-seq -> 10x_data

Quick Assessment Of Cellranger Outputs

- Number of cells

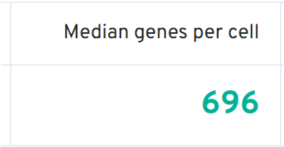


Cells

4,999

- Median genes / cell

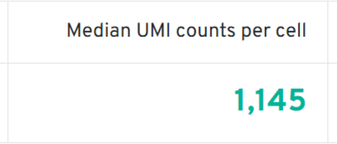
Cell Type / Sample	Expected Range
Fresh PBMCs	1,000–2,500
Solid tissue	500–1,500
FFPE or nuclei	200–800



Median genes per cell

696

- Median UMI Counts per Cell
  - Fresh PBMCs: ~5,000–15,000
  - FFPE or nuclei: ~1,000–5,000

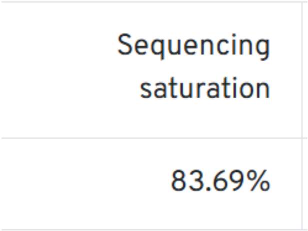


Median UMI counts per cell

1,145

- Sequencing Saturation

<50%	You can benefit from deeper sequencing
50-70%	Still can get some new UMIs
>70%	No need to sequence more



Sequencing saturation

83.69%

Understand The Output

```
cd $HOME/Sample_FASTQ/output_tp53/outs
```

📁 sample_raw_feature_bc_matrix

📁 sample_filetered_feature_bc_matrix

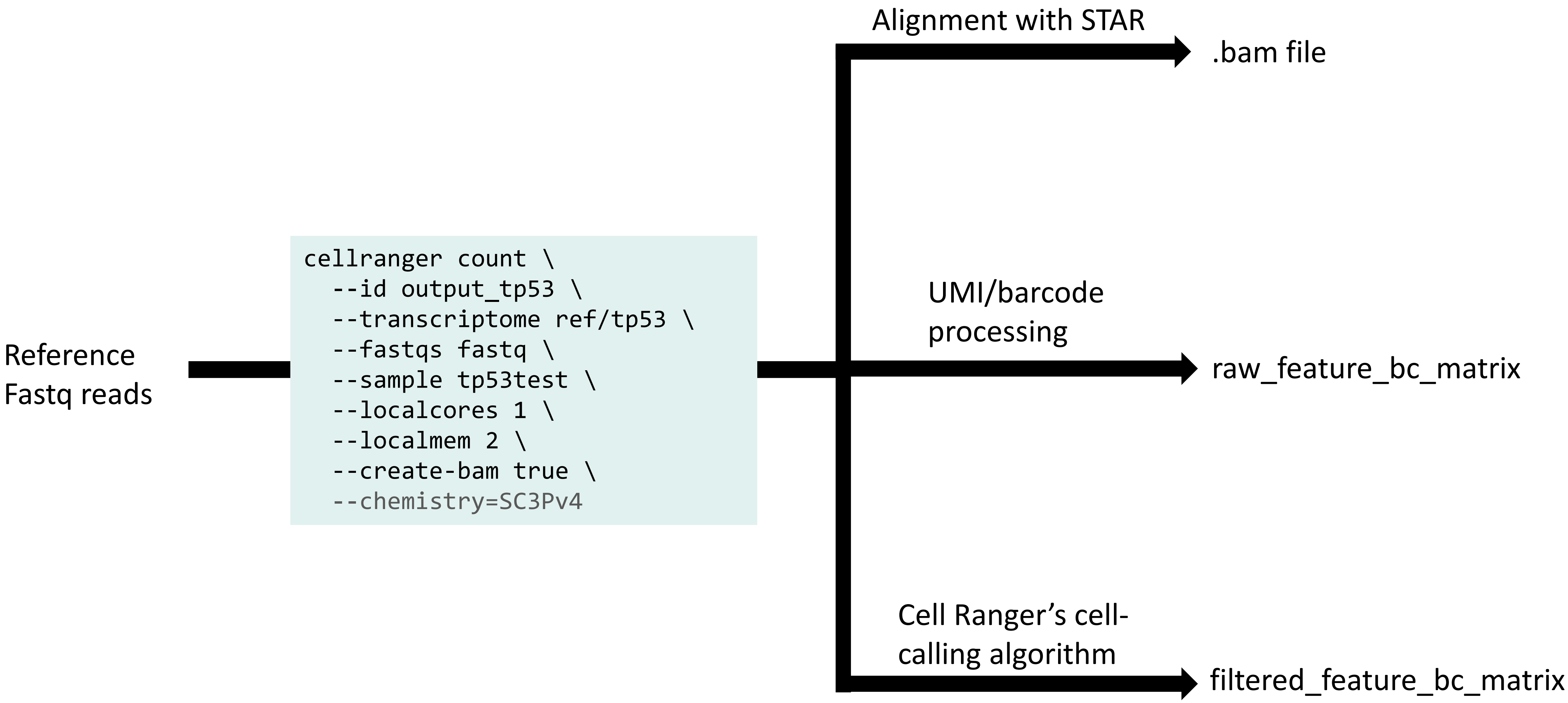
📄 web_summary.html

📄 matrix_summary.csv

https://github.com/ualberta-rcg/scRNA-seq

Go to Google Drive link -> scRNA-seq -> 10x_data

Understand The Output



Reference Fastq reads

```
cellranger count \
  --id output_tp53 \
  --transcriptome ref/tp53 \
  --fastqs fastq \
  --sample tp53test \
  --localcores 1 \
  --localmem 2 \
  --create-bam true \
  --chemistry=SC3Pv4
```

Alignment with STAR → .bam file

UMI/barcode processing → raw_feature_bc_matrix

Cell Ranger’s cell-calling algorithm → filtered_feature_bc_matrix

Understand The Output

What we want:

	Cell1	Cell2	Cell3
Gene1	0	20	3
Gene2	104	1	1

What we got:

```
cd $HOME/Sample_FASTQ/output_tp53/outs/raw_feature_bc_matrix
zcat matrix.mtx.gz
zcat barcodes.tsv.gz
zcat features.tsv.gz
```

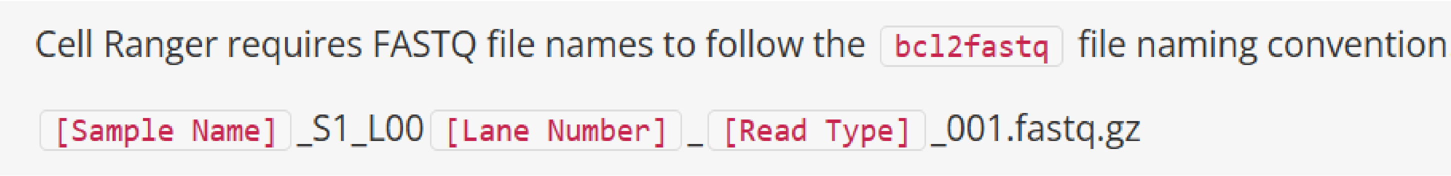
```
# matrix.mtx.gz
1 8 8               # In total, there are 1 gene, 8 cells, and 8 non-zero numbers
1 1 1               # Gene1  Cell1  1 read
1 2 1               # Gene1  Cell2  1 read
1 3 1               # Gene1  Cell3  1 read
1 4 1               # Gene1  Cell4  1 read
1 5 1               # Gene1  Cell5  1 read
1 6 1               # Gene1  Cell6  1 read
1 7 1               # Gene1  Cell7  1 read
1 8 1               # Gene1  Cell8  1 read
```

```
zcat ~/projects/def-sponsor00/scRNA-seq/10x_data/sample_raw_feature_bc_matrix/matrix.mtx.gz |head
zcat ~/projects/def-sponsor00/scRNA-seq/10x_data/sample_raw_feature_bc_matrix/barcodes.tsv.gz |head
zcat ~/projects/def-sponsor00/scRNA-seq/10x_data/sample_raw_feature_bc_matrix/features.tsv.gz |head
```

Summary of input preprocessing

```
SRR9291388_1.fastq.gz        # Read1
SRR9291388_2.fastq.gz        # Read2
SRR9291388_3.fastq.gz        # Index
```

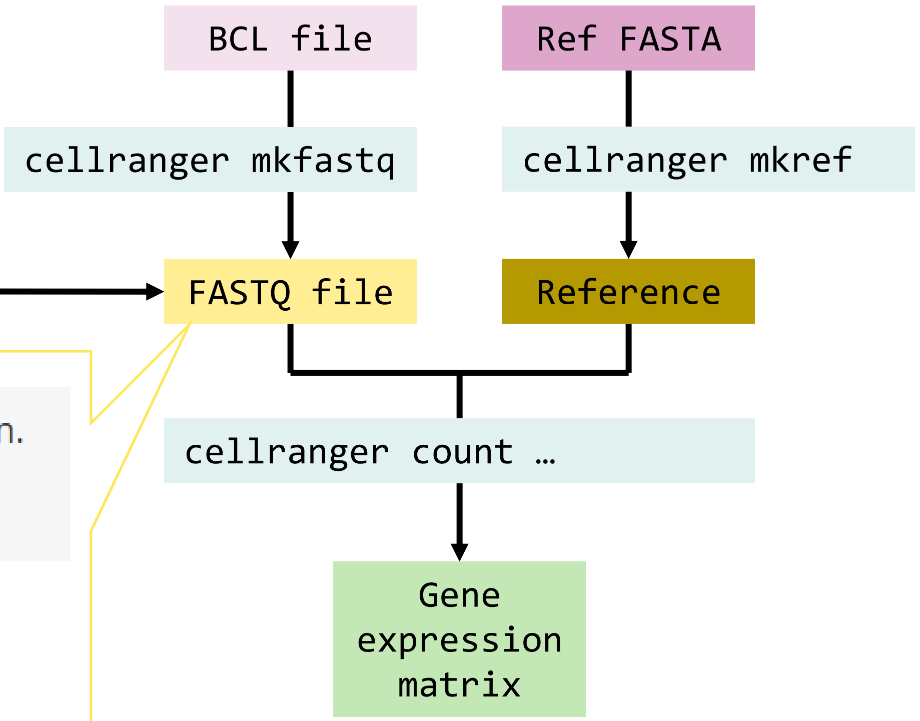
NCBI SRA



Cell Ranger requires FASTQ file names to follow the `bcl2fastq` file naming convention.

`[Sample Name]`_S1_L00`[Lane Number]`_`[Read Type]`_001.fastq.gz

```
SRR9291388_S1_L001_R1_001.fastq.gz    # Read1
SRR9291388_S1_L001_R2_001.fastq.gz    # Read2
SRR9291388_S1_L001_I1_001.fastq.gz    # Index
```



https://kb.10xgenomics.com/hc/en-us/articles/115003802691-How-do-I-prepare-Sequence-Read-Archive-SRA-data-from-NCBI-for-Cell-Ranger

Question?

Quality Control

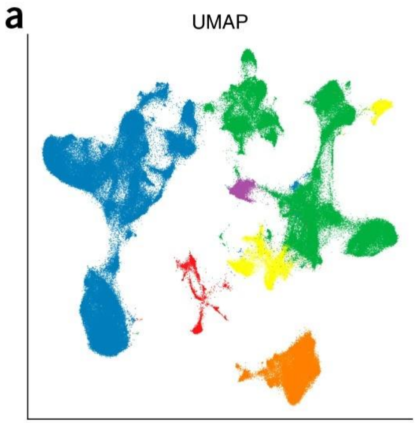
Start Rstudio

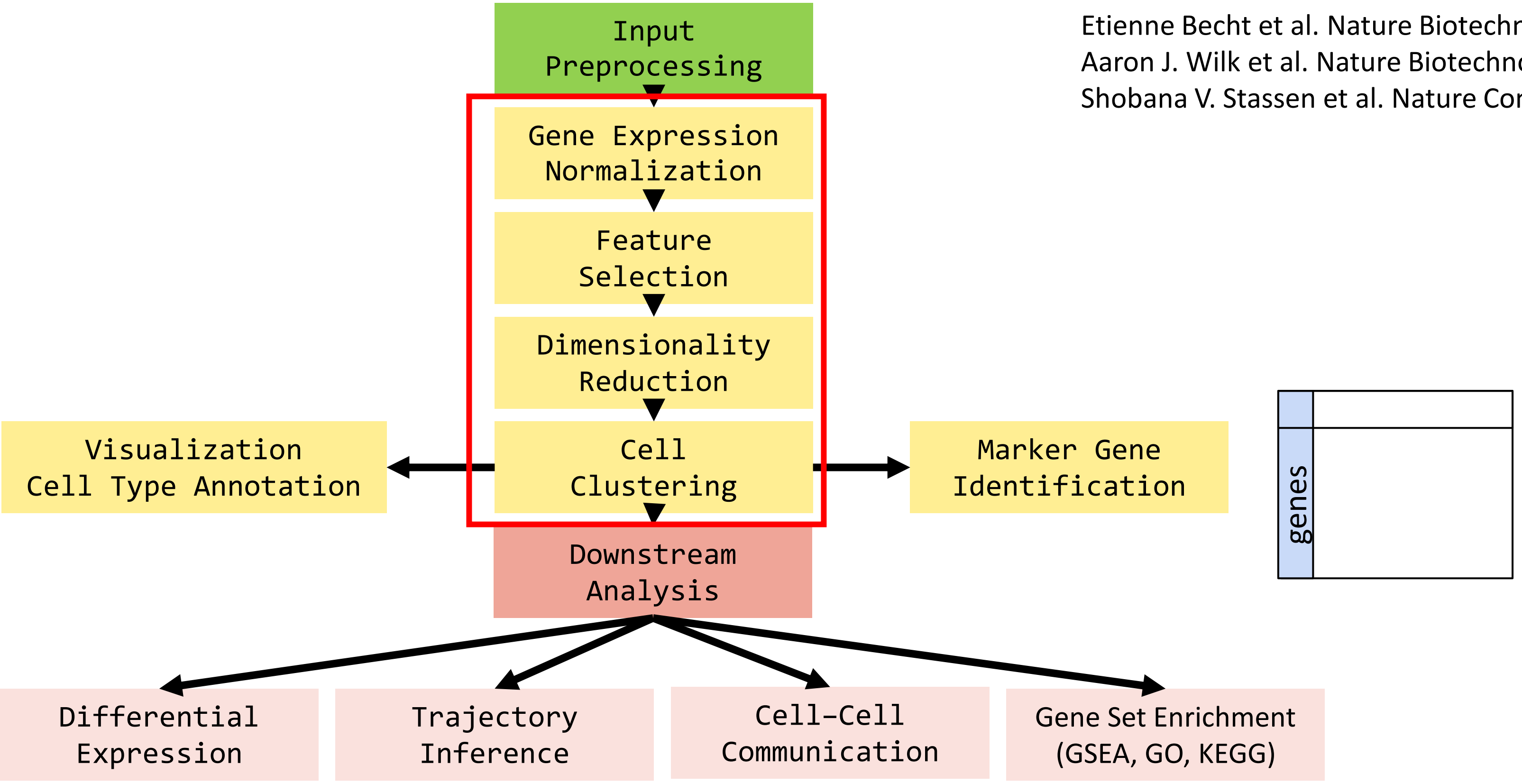
- https://github.com/ualberta-rcg/scRNA-seq

- Open the file “Analysis_with_Seurat.md”

The Pipeline of scRNA-seq Analysis

Etienne Becht et al. Nature Biotechnology. 37: 38–44 (2019)
Aaron J. Wilk et al. Nature Biotechnology. 42: 470–483 (2024)
Shobana V. Stassen et al. Nature Communications. 12: 5528 (2021)




Input Preprocessing

Gene Expression Normalization

Feature Selection

Dimensionality Reduction

Cell Clustering

Visualization Cell Type Annotation

Marker Gene Identification

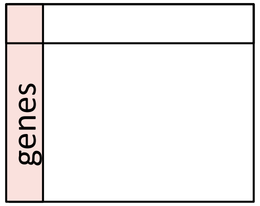
Downstream Analysis

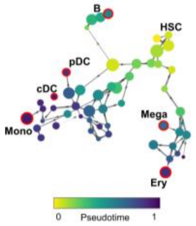
Differential Expression

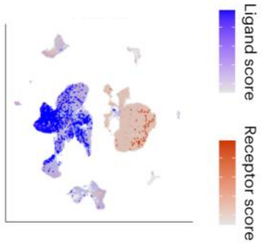
Trajectory Inference

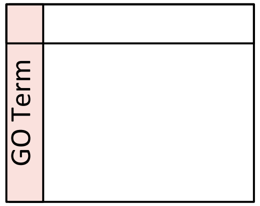
Cell-Cell Communication

Gene Set Enrichment (GSEA, GO, KEGG)

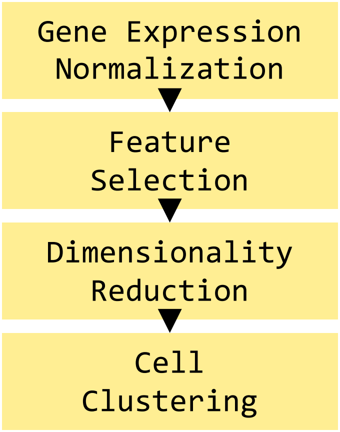








The Pipeline of scRNA-seq Analysis


Gene Expression Normalization

Feature Selection

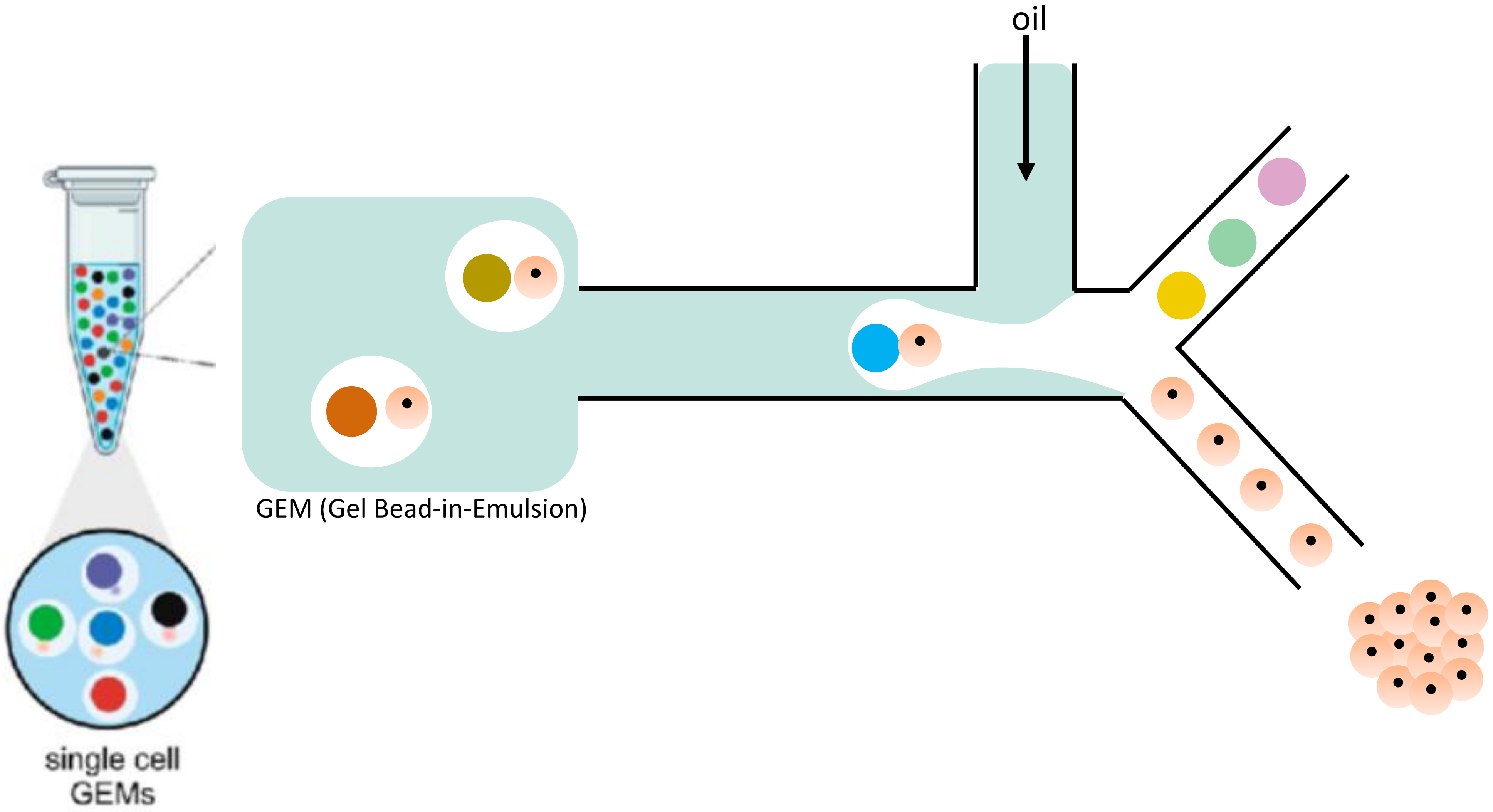
Dimensionality Reduction

Cell Clustering

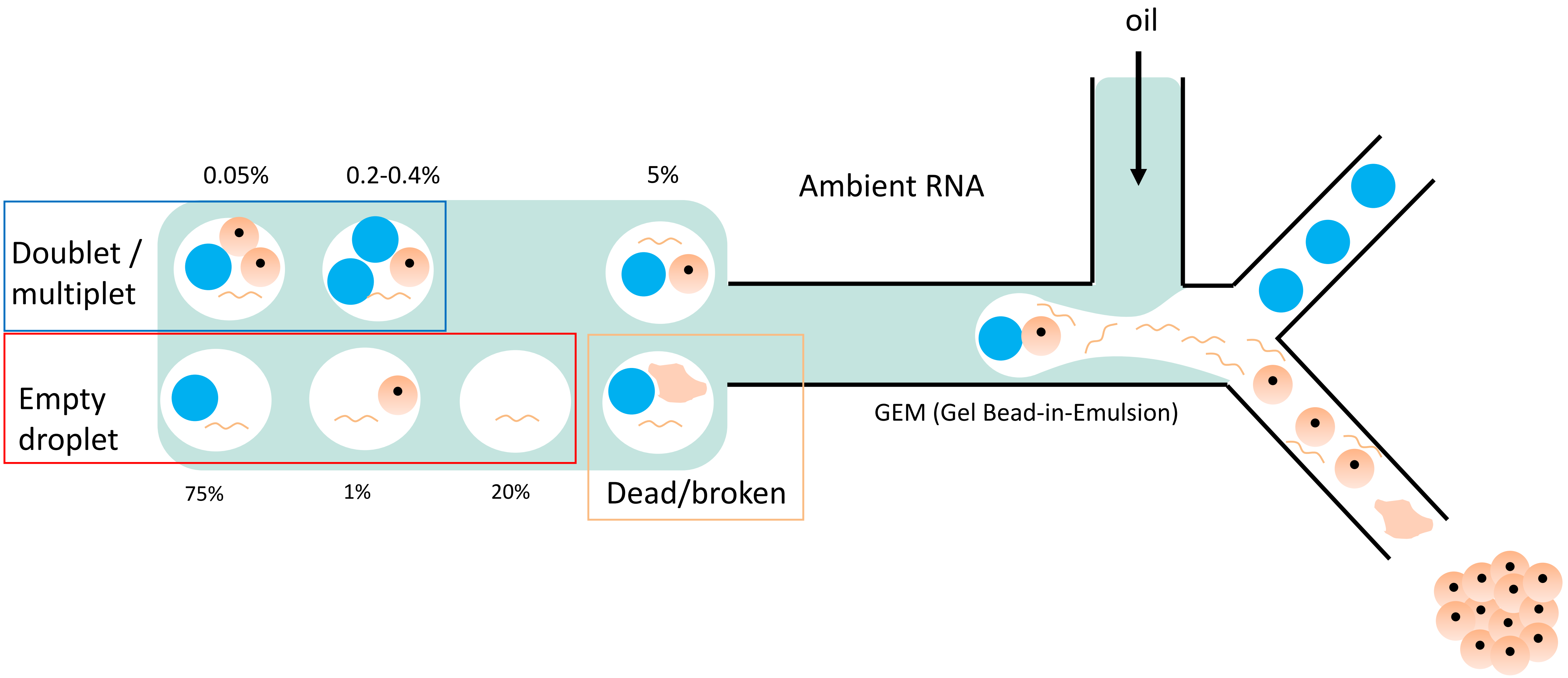
```
filter_dat <- Seurat::Read10X("filtered_feature_bc_matrix/")
seur_obj <- Seurat::CreateSeuratObject(filter_dat, min.cells=5, min.features=100)
seur_obj <- Seurat::NormalizeData(seur_obj)
seur_obj <- Seurat::FindVariableFeatures(seur_obj)
seur_obj <- Seurat::ScaleData(seur_obj)
seur_obj <- Seurat::RunPCA(seur_obj)
seur_obj <- Seurat::FindNeighbors(seur_obj)
seur_obj <- Seurat::FindClusters(seur_obj)
```

- It is straight forward if the web lab experiment is ideal

10x Chromium 3' scRNA-seq – GEM Formation



10x Chromium 3’ scRNA-seq – In Real World



Lareau et al. (2020) doi: 10.1038/s41467-020-14667

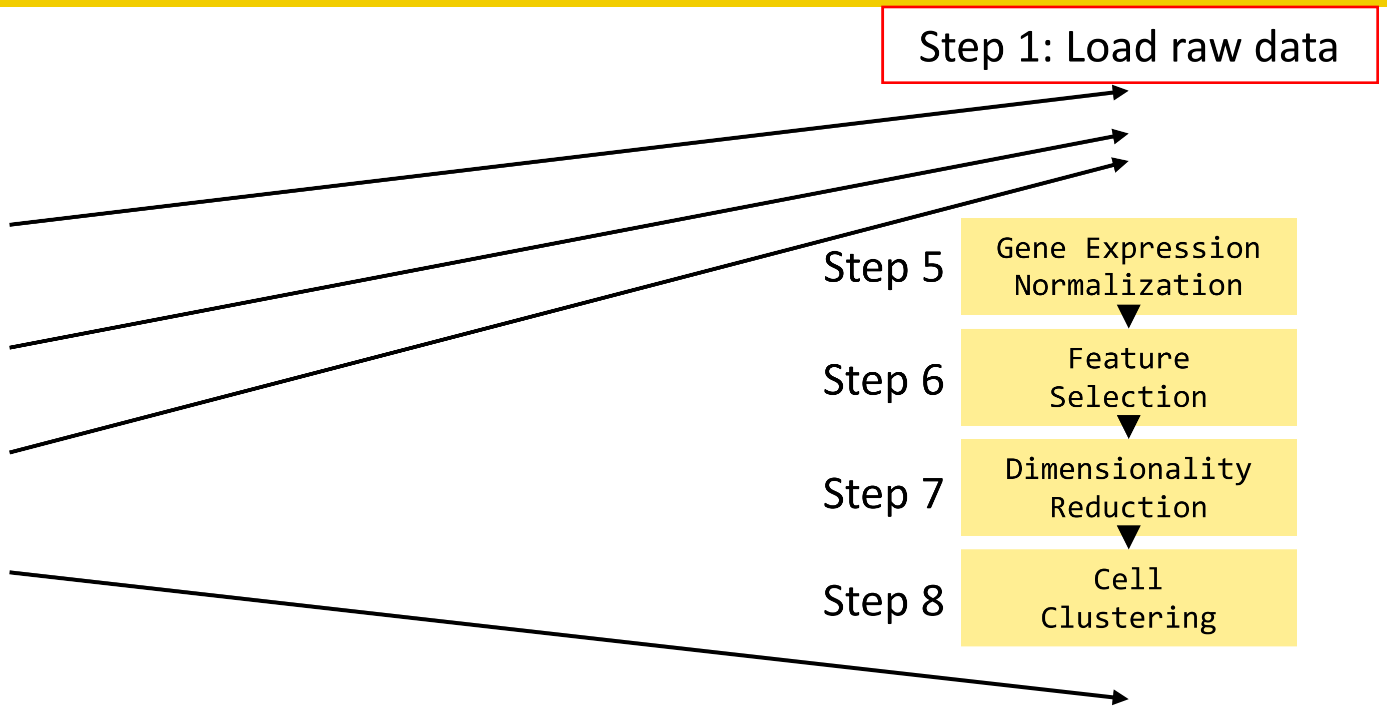
Quality Control

Step 2 • Remove empty droplets (QC1)

Step 3 • Filter out Ambient RNA (QC2)

Step 4 • Identify dead/broken cells (QC3)

Step 9 • Identify doublets (QC4)



Step 1: Load raw data

Step 5 Gene Expression Normalization

Step 6 Feature Selection

Step 7 Dimensionality Reduction

Step 8 Cell Clustering

(Code) Step 1: Load the raw matrix

What we want:

	Cell1	Cell2	Cell3
Gene1	0	20	3
Gene2	104	1	1

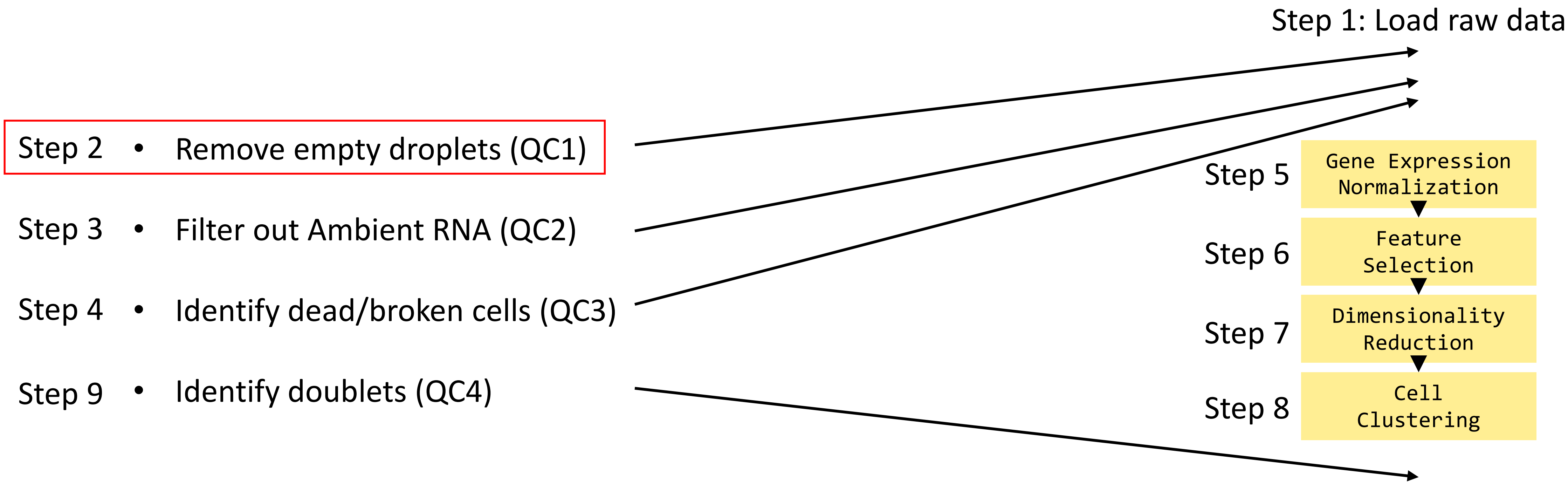
What we got:

```
# matrix.mtx.gz
1 8 8             # In total, there are 1 gene, 8 cells, and 8 non-zero numbers
1 1 1             # Gene1  Cell1  1 read
1 2 1             # Gene1  Cell2  1 read
1 3 1             # Gene1  Cell3  1 read
1 4 1             # Gene1  Cell4  1 read
1 5 1             # Gene1  Cell5  1 read
1 6 1             # Gene1  Cell6  1 read
1 7 1             # Gene1  Cell7  1 read
1 8 1             # Gene1  Cell8  1 read
```

```
......
raw_dat <- Seurat::Read10X(data.dir = "/usr/local/10x_data/sample_raw_feature_bc_matrix")
......
```

	AAACCTGAGATAGGAG-1	AAACCTGAGATCCTGT-1	AAACCTGAGATTACAA-1	...
TP53	6	0	2	
KRAS	3	0	7	
…				

Quality Control



Step 1: Load raw data

Step 2 • Remove empty droplets (QC1)

Step 3 • Filter out Ambient RNA (QC2)

Step 4 • Identify dead/broken cells (QC3)

Step 9 • Identify doublets (QC4)

Step 5 Gene Expression Normalization

Step 6 Feature Selection

Step 7 Dimensionality Reduction

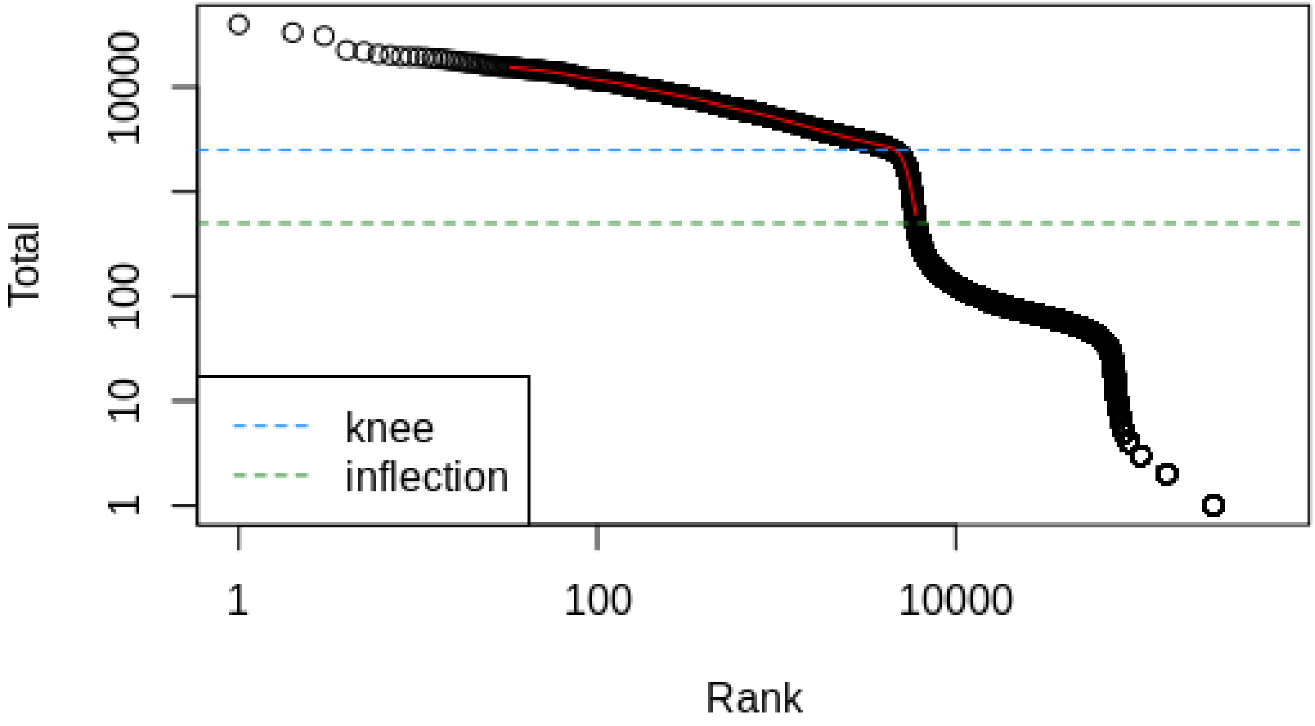
Step 8 Cell Clustering

Step 2: Remove the empty droplets

- Three methods:
  - Cell Ranger Strategy
  - Knee/Inflection
  - Poisson

Step 2: Remove the empty droplets

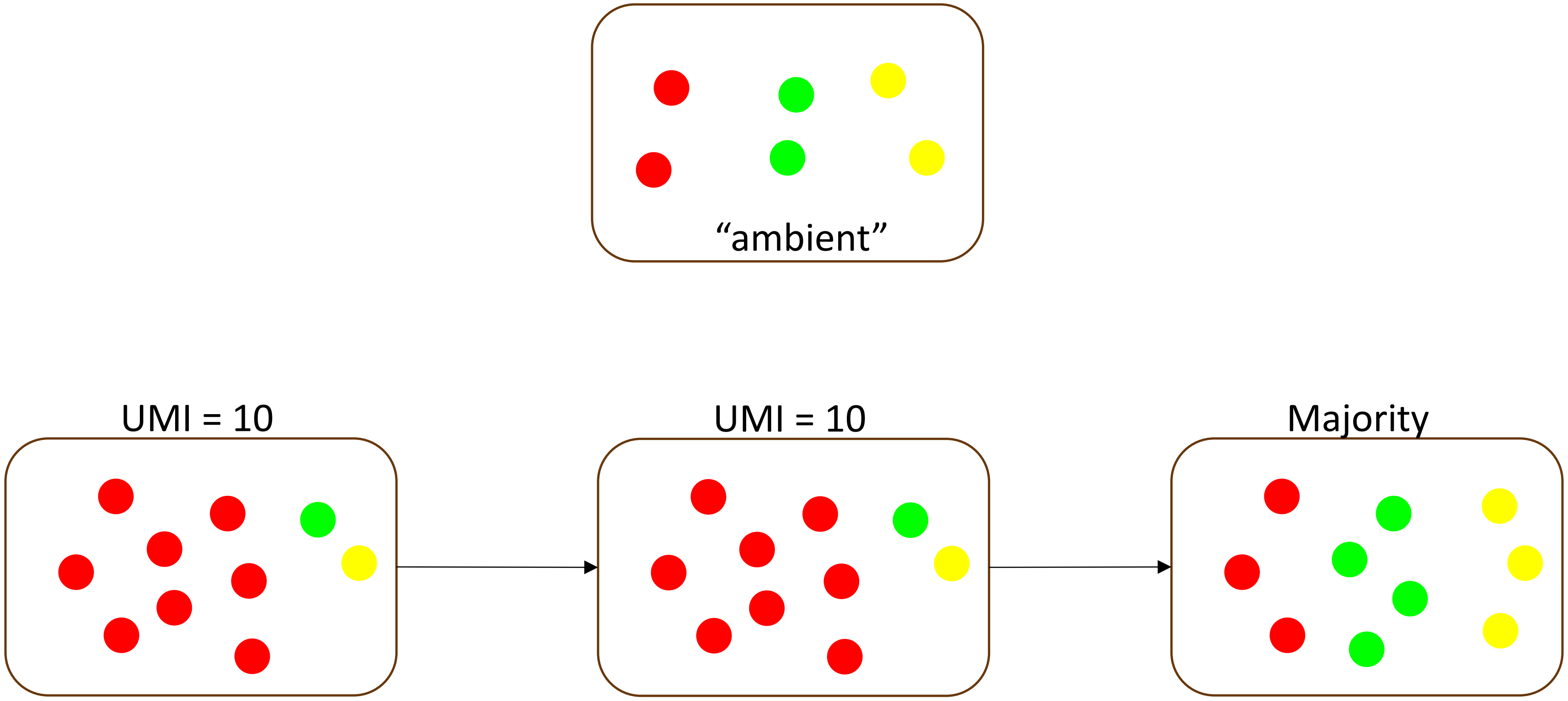
- Barcode Rank Plot



Knee: min 2nd derivative (Max curve bending)

Step 2: Remove the empty droplets

```
e.out <- emptyDrops(raw_dat, lower=100, niters=10000, ignore=NULL, retain=2*br.out$knee)
```



“ambient”

UMI = 10

UMI = 10

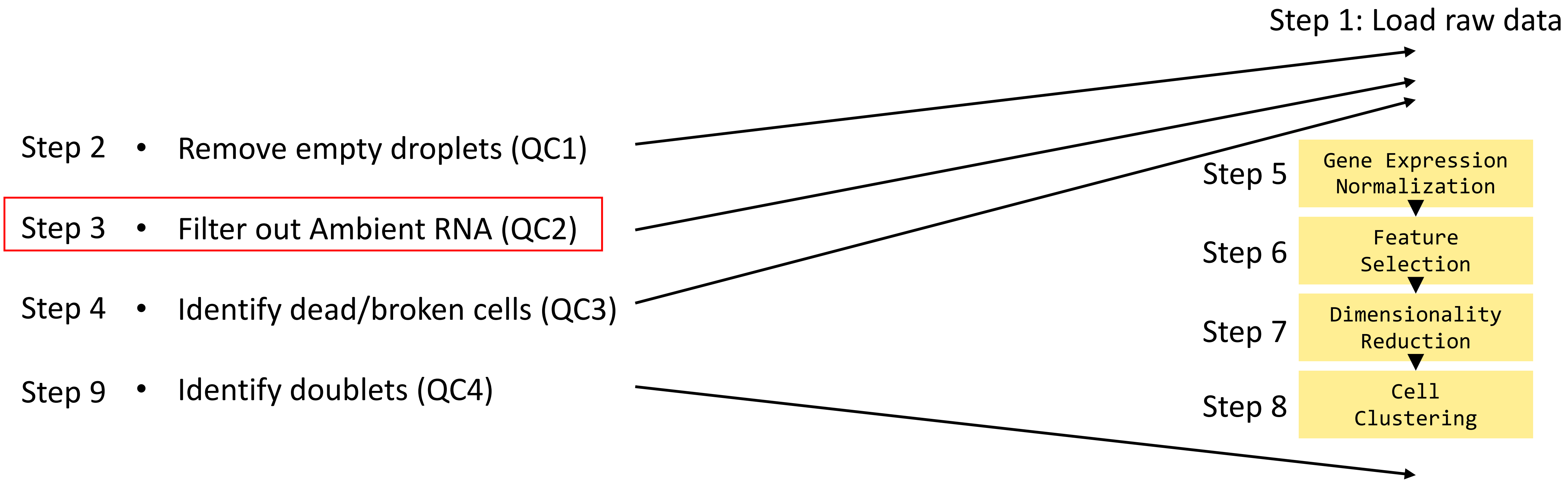
Majority

Monte Carlo
10000 Iteration

Step 2: Remove the empty droplets

- Cell ranger uses the similar idea

Step 3: Filter Out Ambient RNA (QC2)



Step 1: Load raw data

Step 2 • Remove empty droplets (QC1)

Step 3 • Filter out Ambient RNA (QC2)

Step 4 • Identify dead/broken cells (QC3)

Step 9 • Identify doublets (QC4)

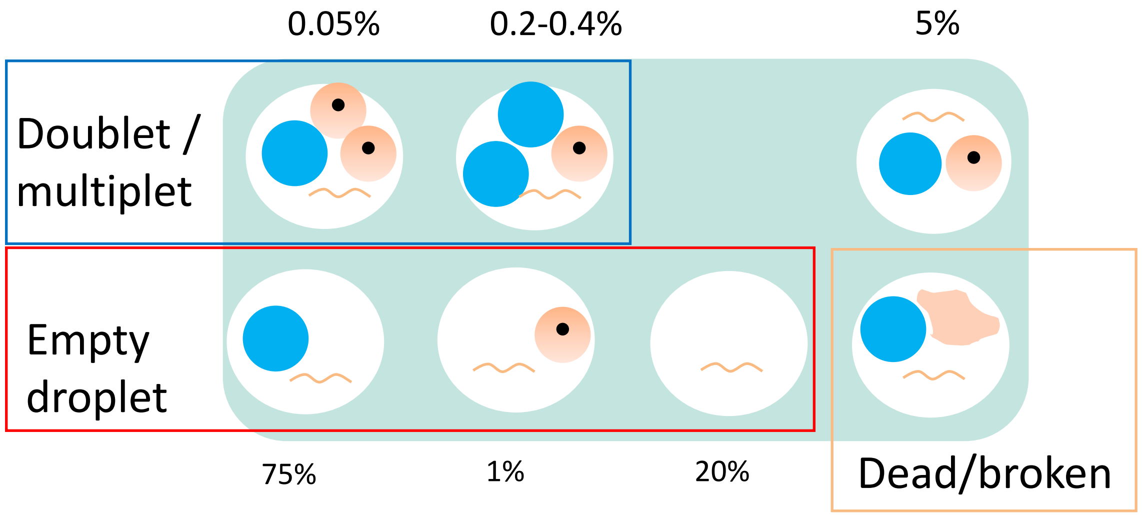
Step 5 Gene Expression Normalization

Step 6 Feature Selection

Step 7 Dimensionality Reduction

Step 8 Cell Clustering

Step 3: Filter out ambient RNA (QC2)



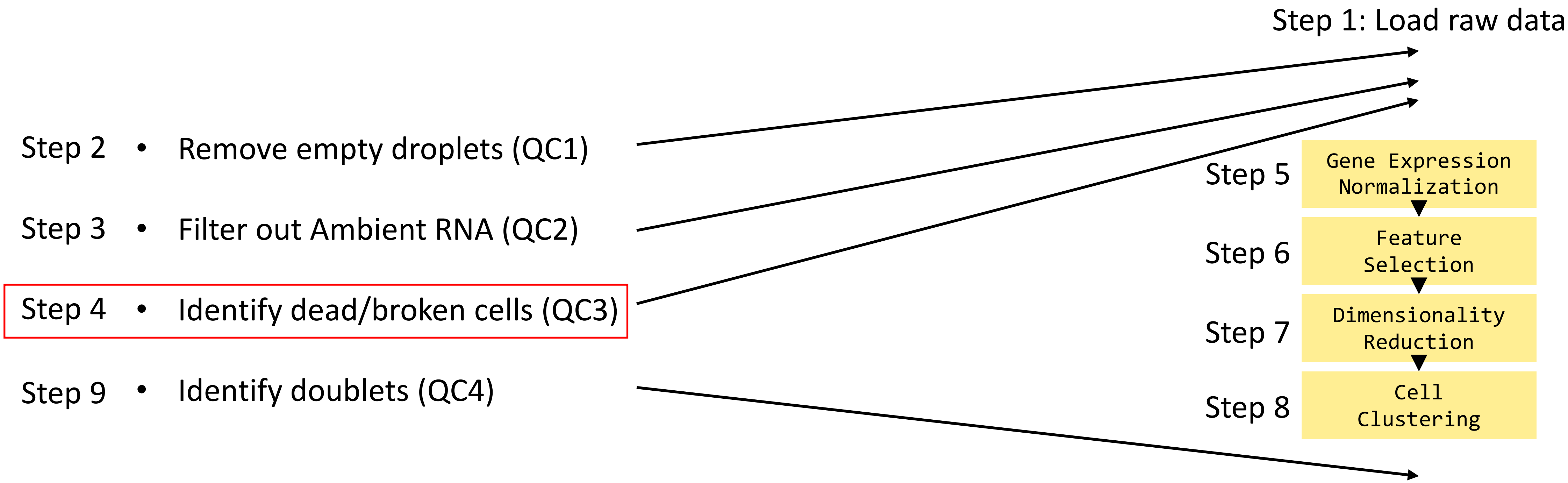
- There are multiple tools/ways.
  We are using DecontX today because:
  - It is easy to be integrated into the pipeline.
  - It doesn't need GPU.
  - The syntax is simple and clean.

- What DecontX does:
  - Group cells into clusters and estimates cluster-specific expression profiles
  - For each cell, tune the cell-specific contamination fraction to make the gene expression profile match the observation and the cluster as much as possible.

Step 3: Filter out ambient RNA (QC2)

	Cell A	Profile A	Ambient <- B	Cell B	Profile B	Ambient <- A
Gene1	90	0.545	0.046 x 15	6	0.046	0.545 x 10
Gene2	60	0.363	0.030 x 15	4	0.030	0.363 x 10
Gene3	10	0.060	0.538 x 15	70	0.538	0.060 x 10
Gene4	5	0.030	0.385 x 15	50	0.385	0.030 x 10
Total	165		165 x 9.1% = 15	130		130 x 7.7% = 10
Contamination with Bayesian Mixture model	9.1%			7.7%		

Step 4: Identify Dead/Broken Cells (QC3)



Step 1: Load raw data

- Step 2 • Remove empty droplets (QC1)
- Step 3 • Filter out Ambient RNA (QC2)
- Step 4 • Identify dead/broken cells (QC3)
- Step 9 • Identify doublets (QC4)

Step 5 Gene Expression Normalization

Step 6 Feature Selection

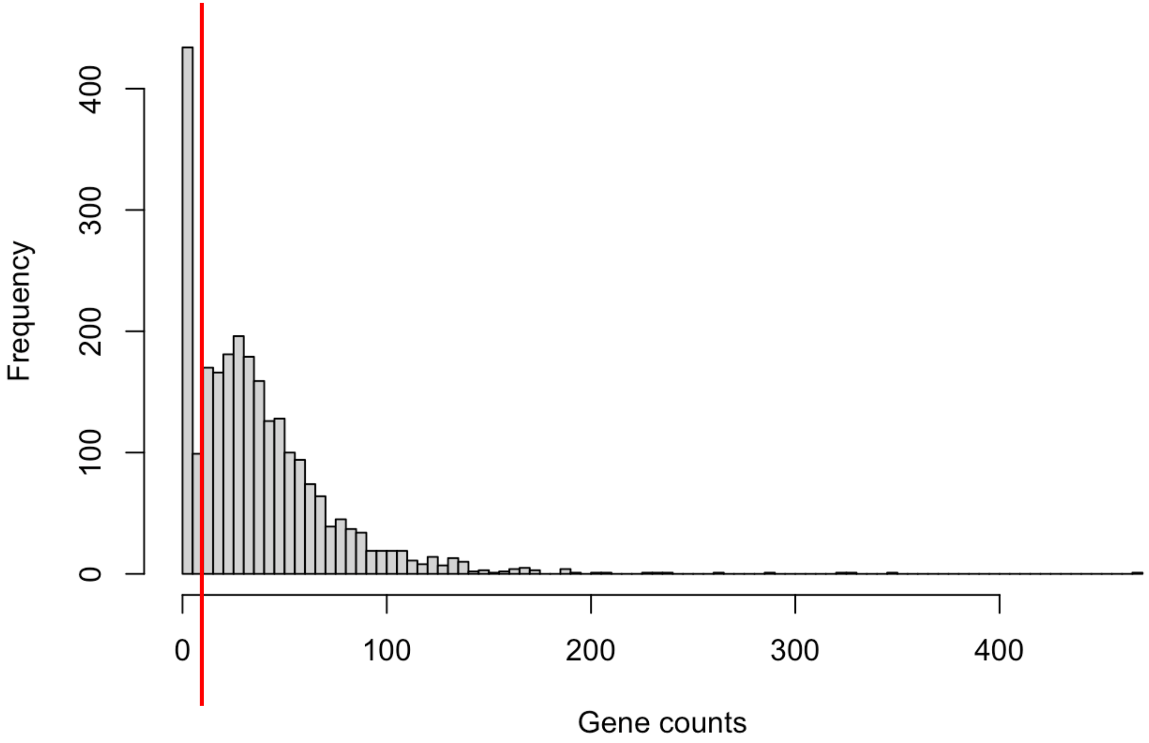
Step 7 Dimensionality Reduction

Step 8 Cell Clustering

Step 4: Identify the dead / broken cells (QC3)

- Through Mitochondrial genes.

Consideration	Guidance
Human PBMCs / tumors / organs	200–500 gene threshold is typical. MT% cutoff: 30%
Low-input / fragile tissues (e.g., FFPE, brain)	Use a **lower threshold** (100–300). MT% cutoff: 40-60%
Visual inspection	Use hist(sce$detected) or violin plots to **see natural separation**



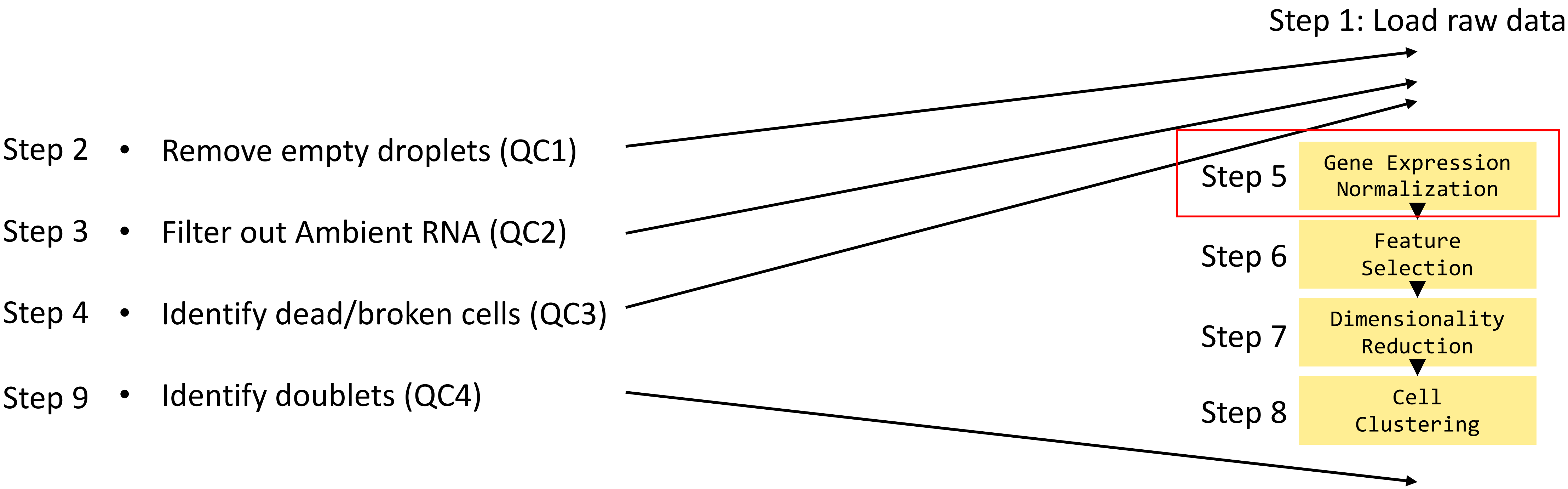
Choose your genes based on your project, for example, for PBMC sample, you may want to remove those red blood cells with high hemoglobin mRNAs (>0.5-1%).

UNIVERSITY OF ALBERTA

Step 4: Identify dead/broken cells (QC3)

```
......
......
is.mt <- grepl("^MT-", rowData(sce)$Symbol)
......
......
......
cell_filter_detect <- sce$detected < 100
cell_filter_MT <- sce$subsets_Mito_percent > 30
```

Step 5: Gene Expression Normalization



Step 1: Load raw data

Step 2 • Remove empty droplets (QC1)

Step 3 • Filter out Ambient RNA (QC2)

Step 4 • Identify dead/broken cells (QC3)

Step 9 • Identify doublets (QC4)

Step 5 Gene Expression Normalization

Step 6 Feature Selection

Step 7 Dimensionality Reduction

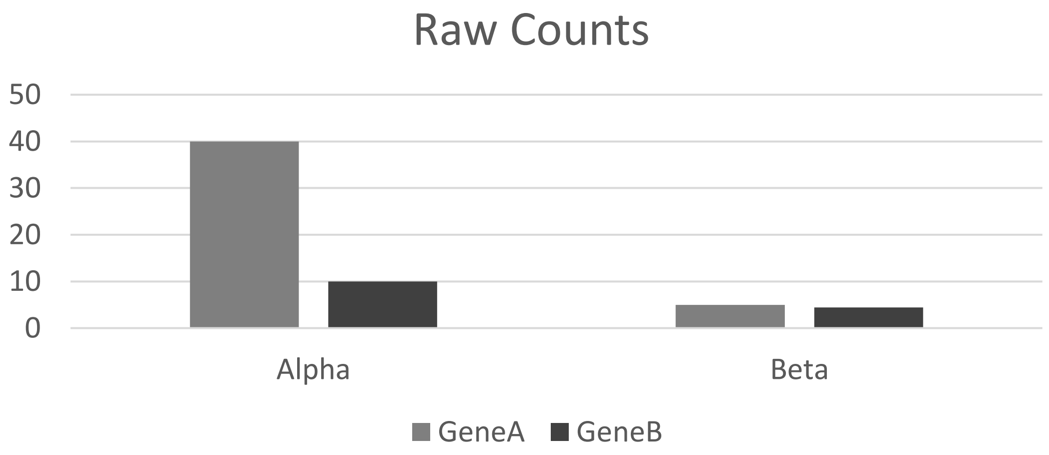
Step 8 Cell Clustering

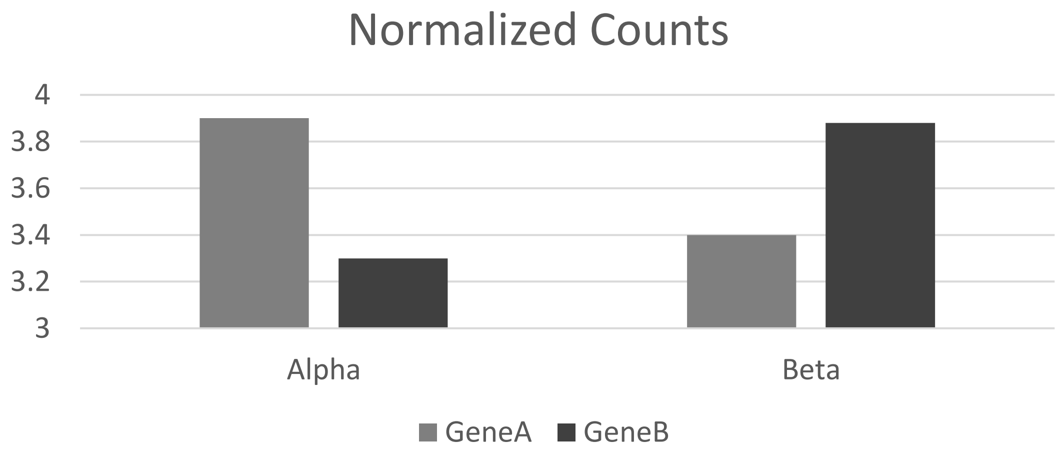
UNIVERSITY OF ALBERTA

Step 5: Gene Expression Normalization

```
seur_filtered <- NormalizeData(seur_filtered, normalization.method = "LogNormalize", scale.factor = 10000)
```

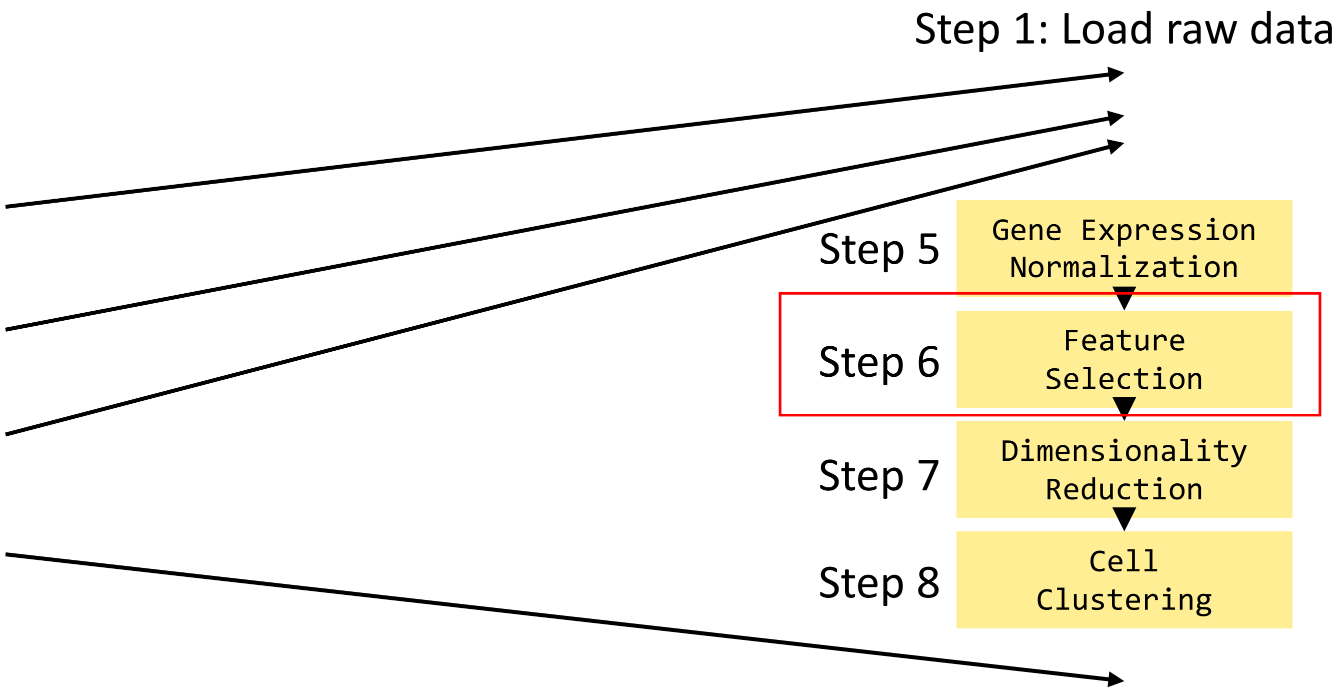
Cell	Gene	UMIs	Scaling	Log Transformation
Alpha	A	40	40/(40+10)*10000 = 8000	log(1+8000) = 3.90
	B	10	10/(40+10)*10000 = 2000	log(1+2000) = 3.30
Beta	A	5	5/(5+15)*10000 = 2500	log(1+2500) = 3.40
	B	15	15/(5+15)*10000 = 7500	log(1+7500) = 3.88





Step 6: Feature Selection

- Step 2 • Remove empty droplets (QC1)
- Step 3 • Filter out Ambient RNA (QC2)
- Step 4 • Identify dead/broken cells (QC3)
- Step 9 • Identify doublets (QC4)



Step 1: Load raw data

Step 5 Gene Expression Normalization

Step 6 Feature Selection

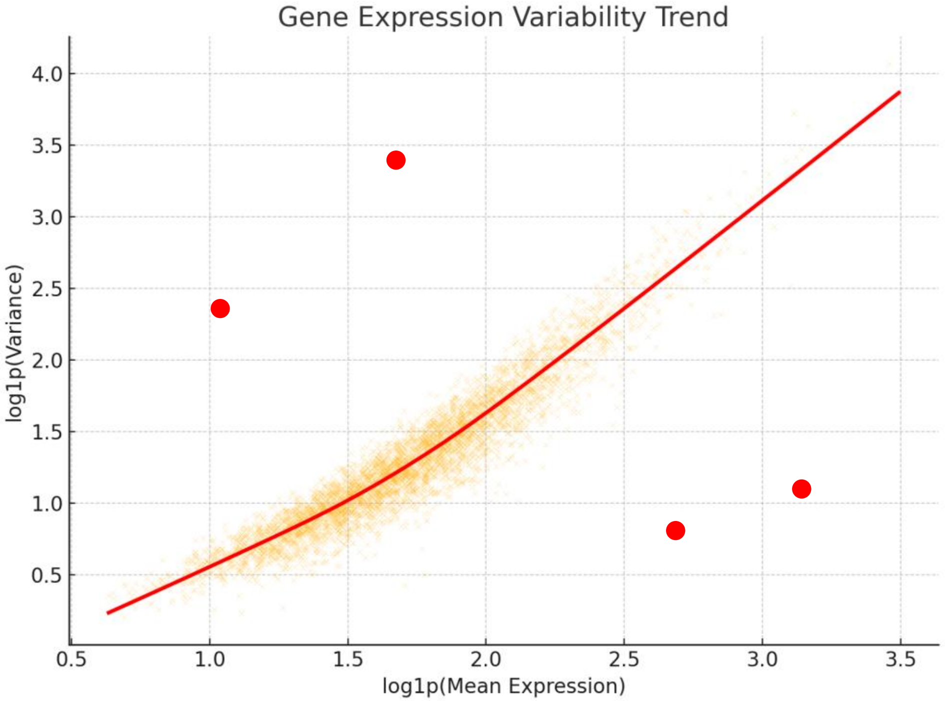
Step 7 Dimensionality Reduction

Step 8 Cell Clustering

Step 6: Feature Selection - Identify highly variable genes

```
seur_filtered <- FindVariableFeatures(seur_filtered, selection.method = "vst", nfeatures = 500)
```

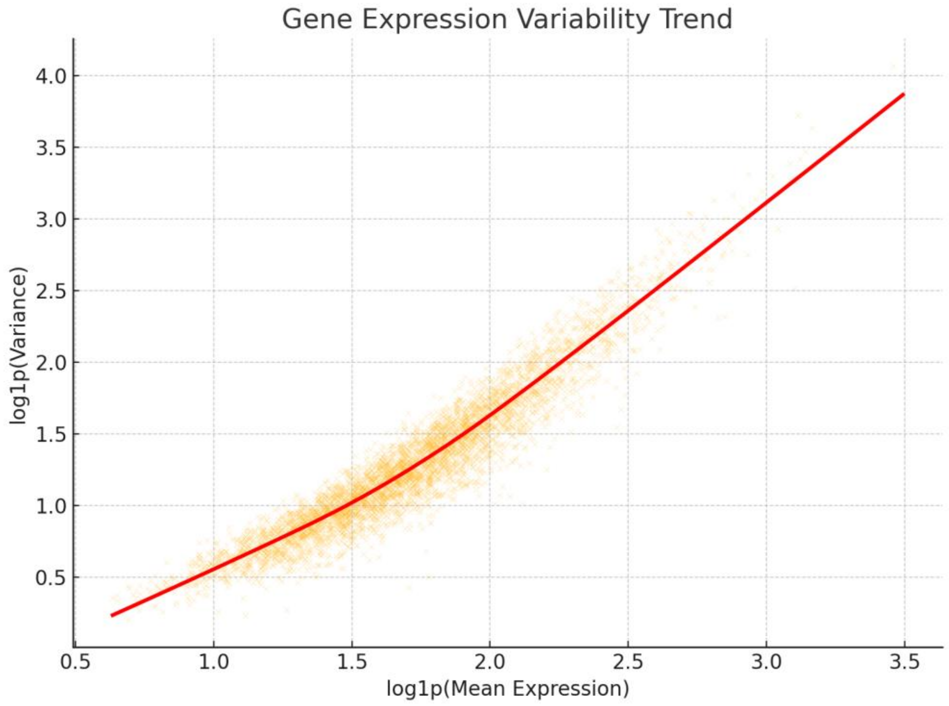
Identify 500 most variable genes through “Variance Stabilizing Transformation” (VST).



Step 6: Feature Selection - Identify highly variable genes

```
seur_filtered <- FindVariableFeatures(seur_filtered, selection.method = "vst", nfeatures = 500)
```

Identify 500 most variable genes through “Variance Stabilizing Transformation” (VST).



How to determine “nfeatures”?

Total genes detected per cell	Suggested *nfeatures*
< 500	300–500
~1000	500–1000
>2000	Up to 2000

```
summary(seur_filtered$nFeature_originalexp)
```

Step 6: Feature Selection - Identify highly variable genes

```
seur_filtered <- FindVariableFeatures(seur_filtered, selection.method = "vst", nfeatures = 500)
```

Identify 500 most variable genes through “Variance Stabilizing Transformation” (VST).

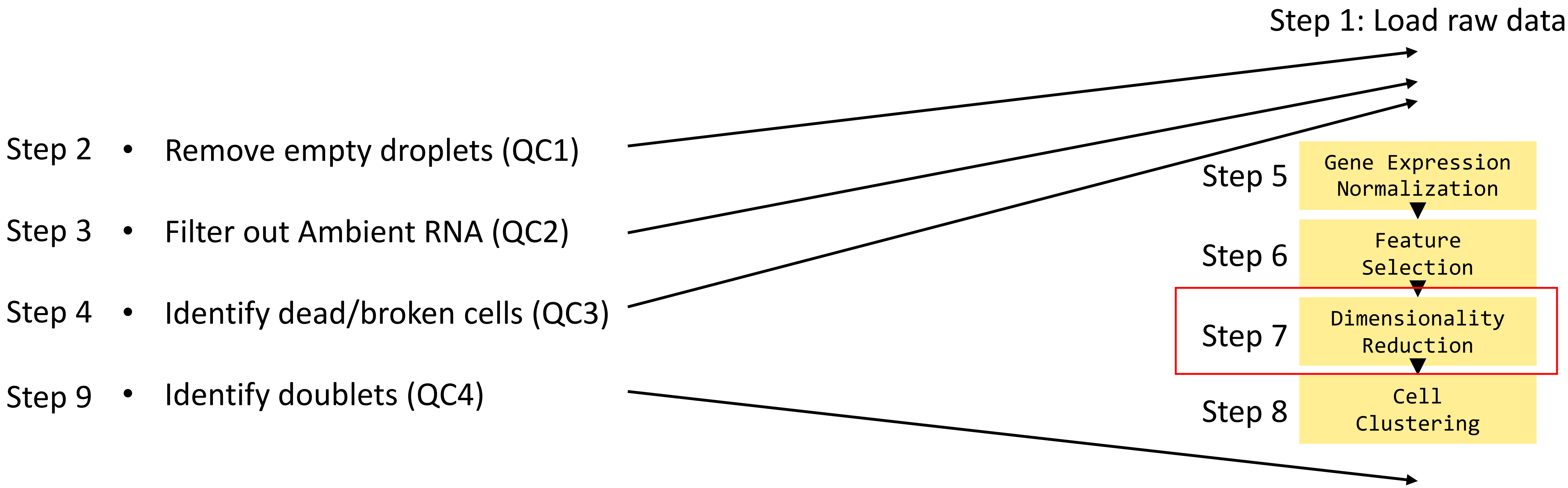
- This is arbitrary.
- You can play with the number and see which makes more sense.
- Usually, it wouldn’t change much from 2000 to 2100, but it is worth testing from 500 to 400/600.

How to determine “nfeatures”?

Total genes detected per cell	Suggested *nfeatures*
< 500	300–500
~1000	500–1000
>2000	Up to 2000

```
summary(seur_filtered$nFeature_originalexp)
```

Step 7: Dimensionality Reduction



Step 1: Load raw data

Step 2 • Remove empty droplets (QC1)

Step 3 • Filter out Ambient RNA (QC2)

Step 4 • Identify dead/broken cells (QC3)

Step 9 • Identify doublets (QC4)

Step 5 Gene Expression Normalization

Step 6 Feature Selection

Step 7 Dimensionality Reduction

Step 8 Cell Clustering

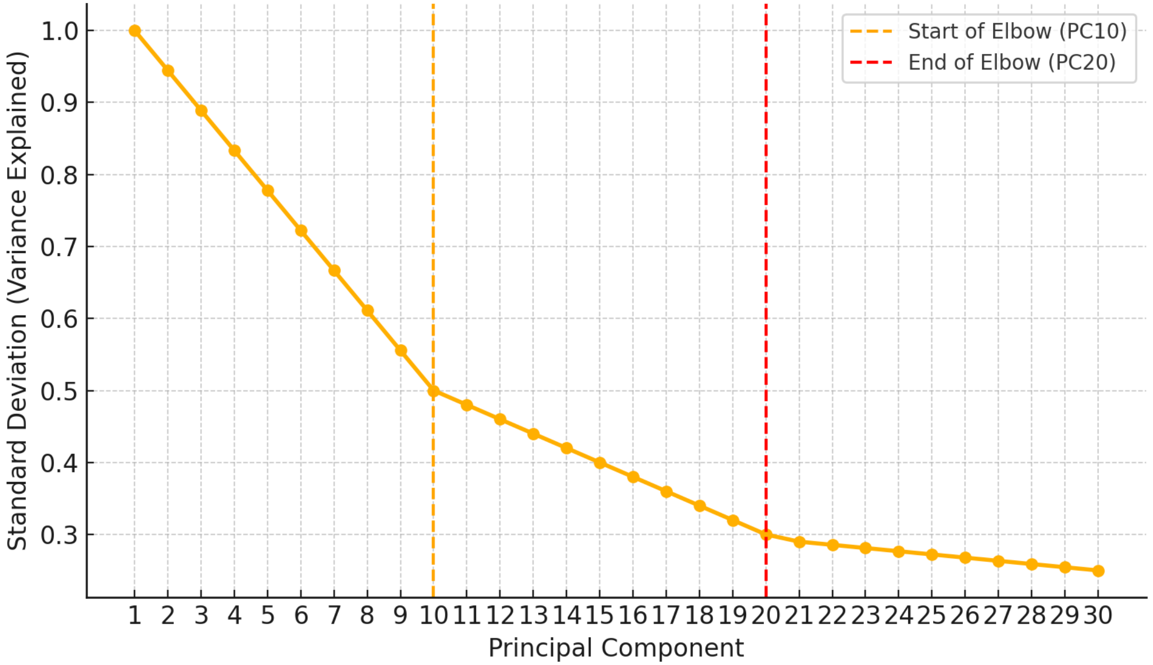
Step 7: Dimensionality Reduction

- Principle Component Analysis (PCA) is the most popular way to reduce dimensionality.

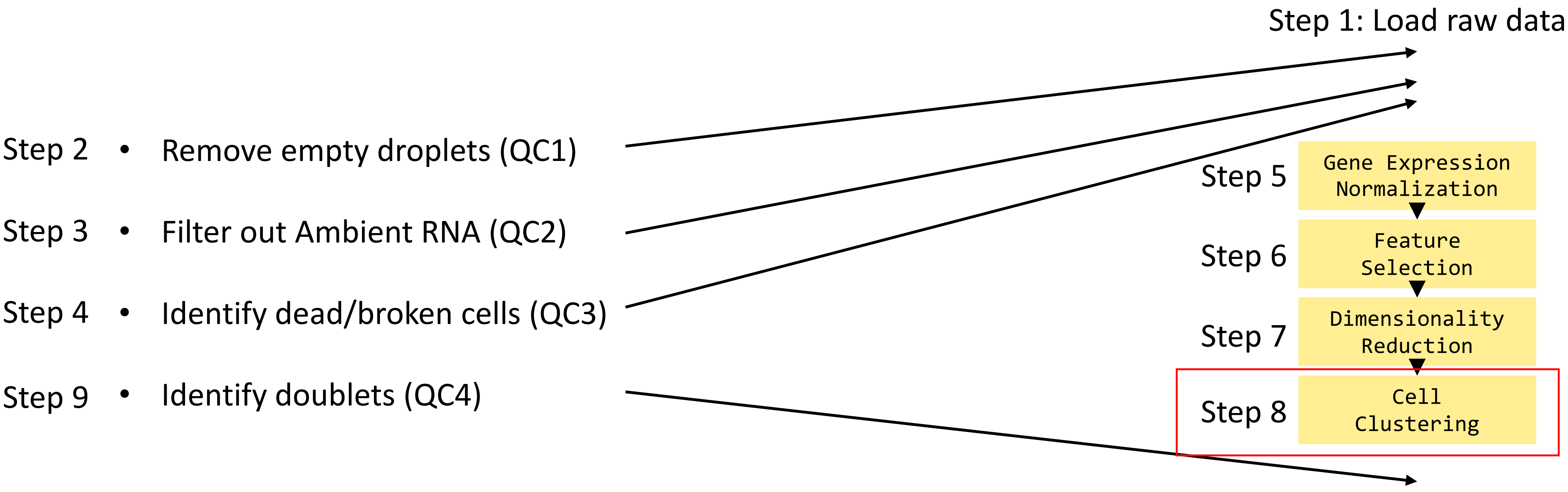
- Why do we want to reduce dimensionality?
  - Remove noise
  - Speed up computation
  - Visualize the data
  - Reveal biological structure

- Determine how many PCs via ElbowPlot

```
ElbowPlot(seur_filtered)
PCs <- 7
```



Step 8: Cell Clustering



Step 1: Load raw data

Step 2 • Remove empty droplets (QC1)

Step 3 • Filter out Ambient RNA (QC2)

Step 4 • Identify dead/broken cells (QC3)

Step 9 • Identify doublets (QC4)

Step 5 Gene Expression Normalization

Step 6 Feature Selection

Step 7 Dimensionality Reduction

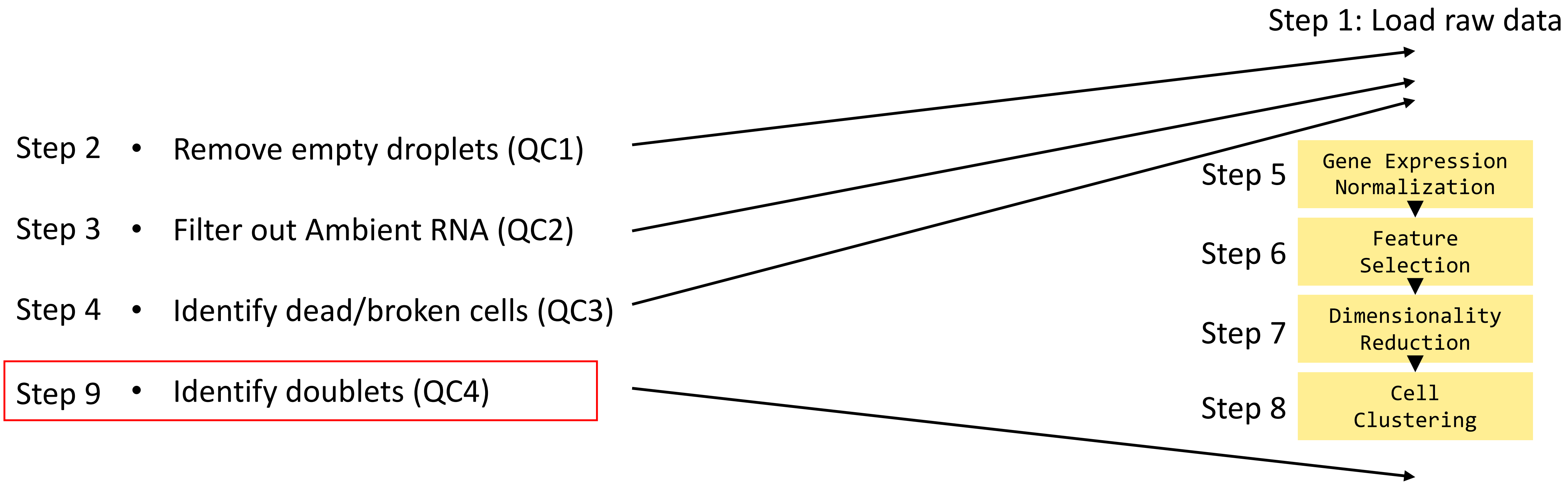
Step 8 Cell Clustering

Step 8: Cell Clustering

```
seur_filtered <- FindNeighbors(seur_filtered, dims = 1:PCs)
seur_filtered <- FindClusters(seur_filtered, resolution = 0.3)
```

Resolution value	**Effect**
0.1	Large, coarse clusters
0.3	Moderate clusters (default-ish)
0.8	Smaller, finer clusters
≥1.0	Many small clusters (may over-split)

Step 9: Identify Doublets (QC4)



Step 1: Load raw data

Step 2 • Remove empty droplets (QC1)

Step 3 • Filter out Ambient RNA (QC2)

Step 4 • Identify dead/broken cells (QC3)

Step 9 • Identify doublets (QC4)

Step 5 Gene Expression Normalization

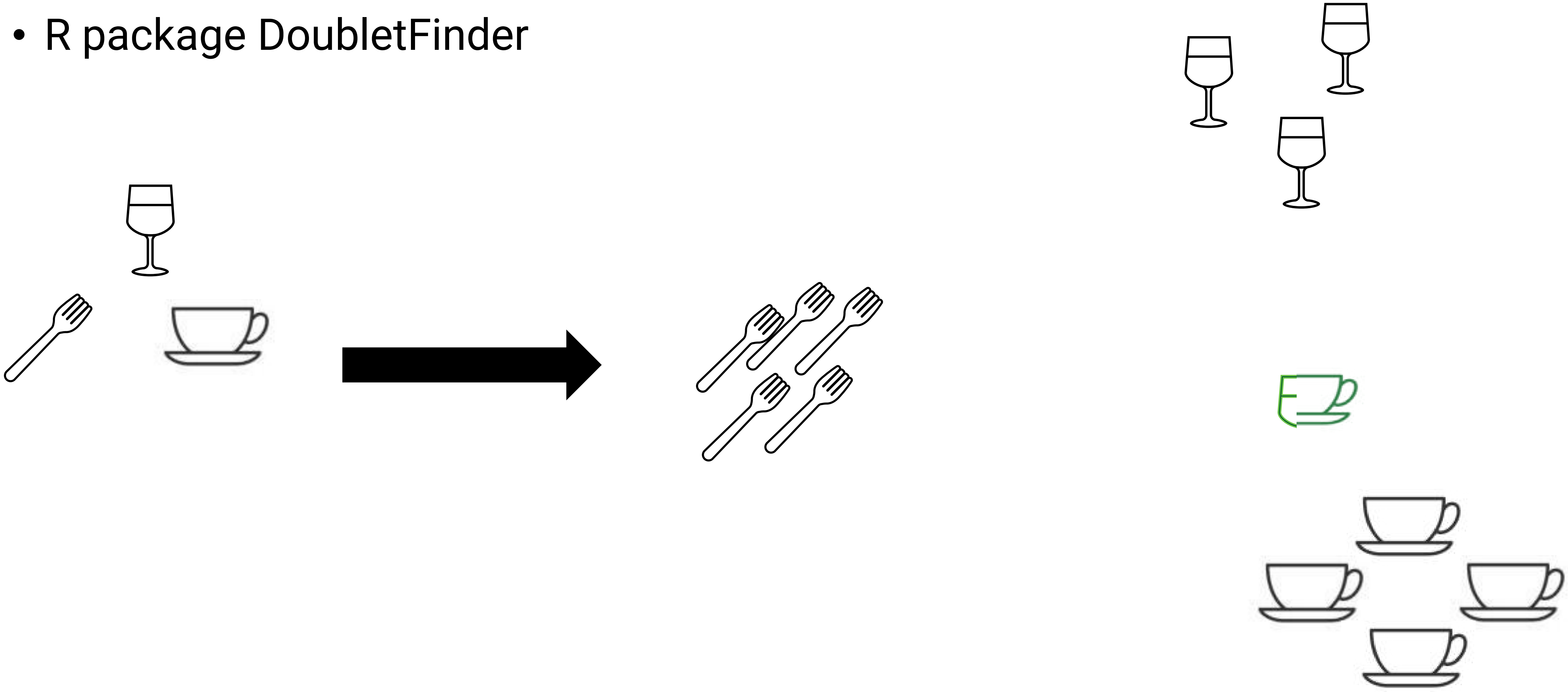
Step 6 Feature Selection

Step 7 Dimensionality Reduction

Step 8 Cell Clustering

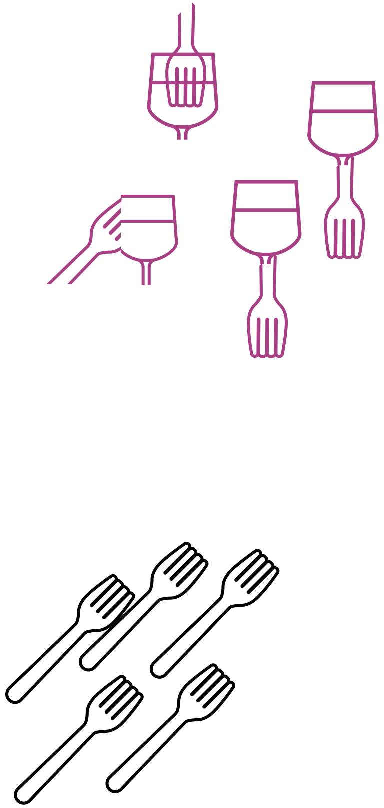
Step 9: Identify the doublets (QC4)

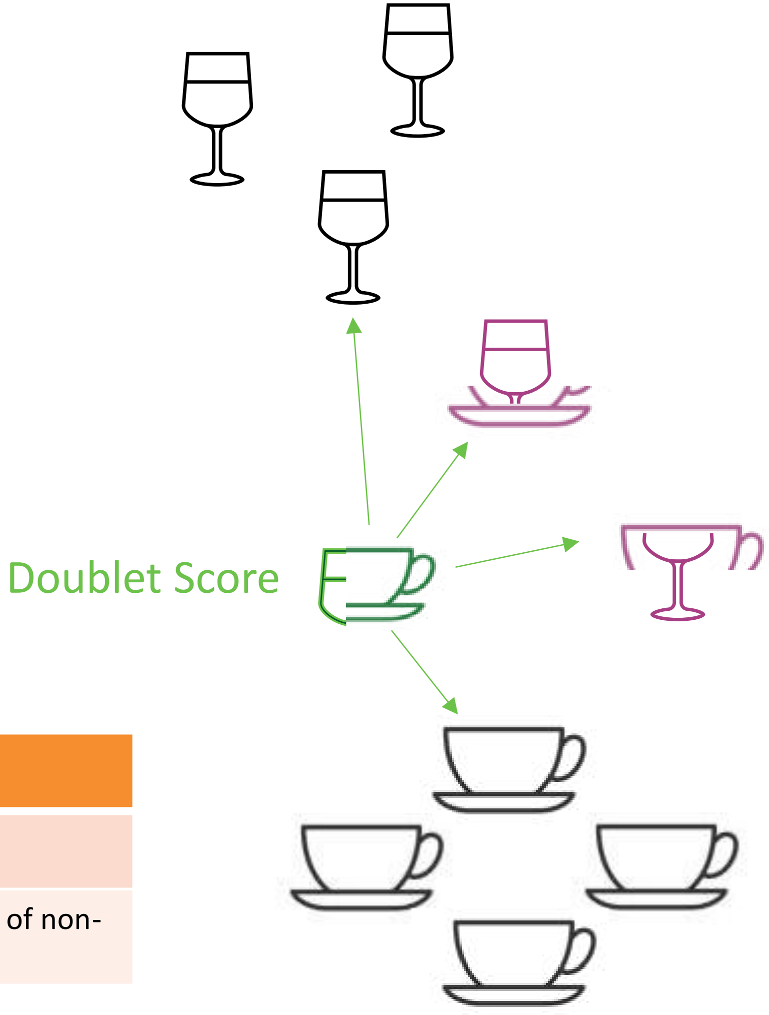
- R package DoubletFinder



Step 9: Identify the doublets (QC4)

- R package DoubletFinder

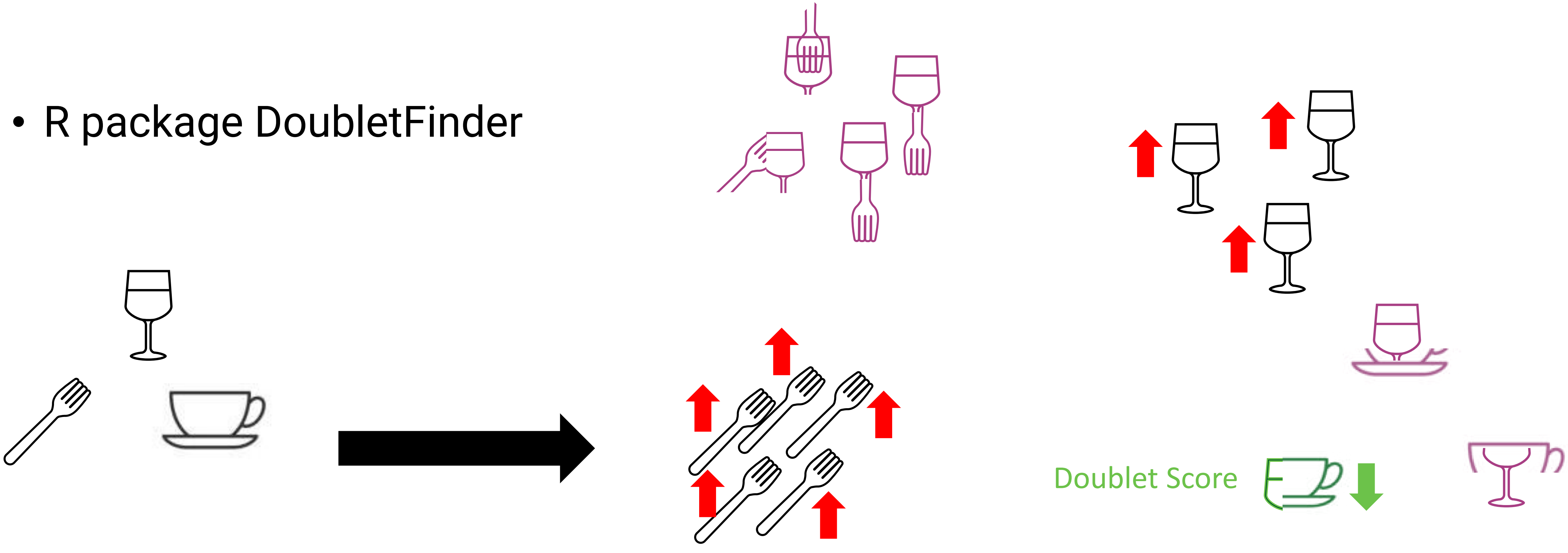




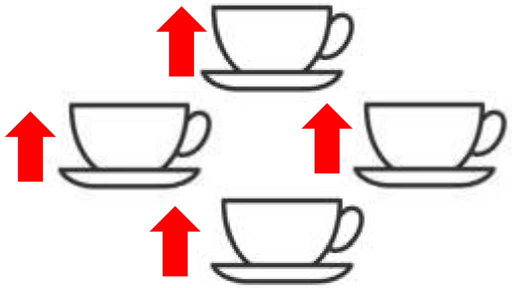
Parameter	Description
nEXP	Expected number of doublets
pN	how many artificial doublets are generated relative to the number of non-empty droplets

Step 9: Identify the doublets (QC4)

- R package DoubletFinder



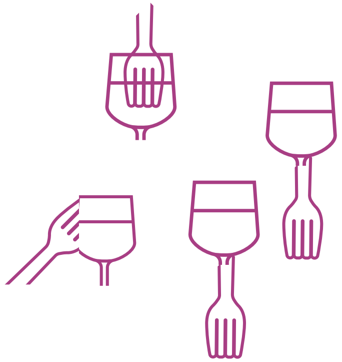
Parameter	Description
nEXP	Expected number of doublets
pN	how many artificial doublets are generated relative to the number of non-empty droplets

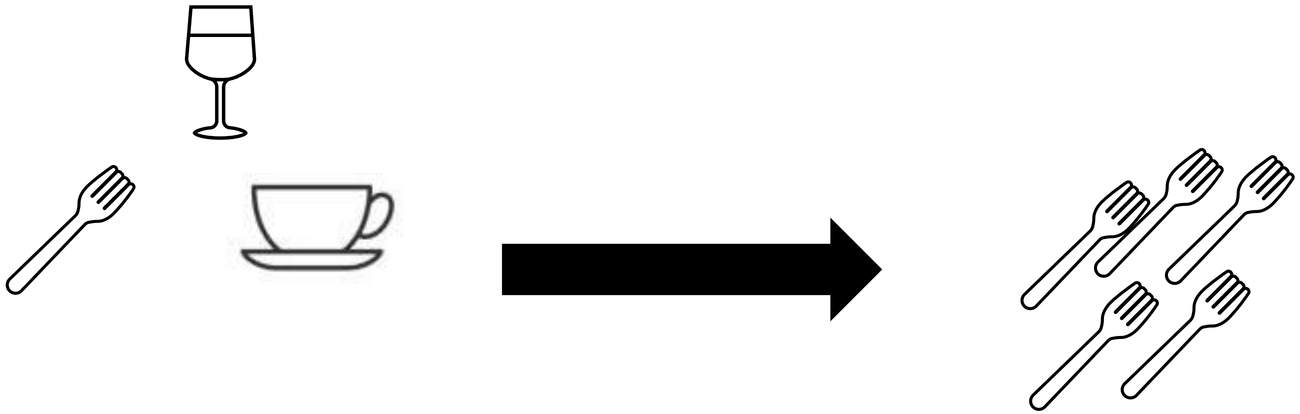


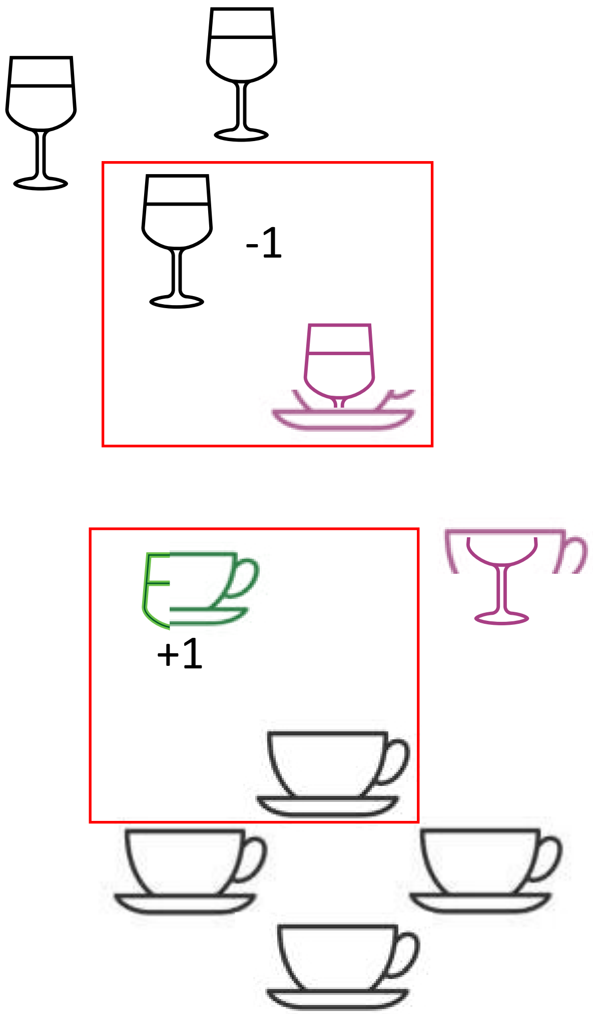
Step 9: Identify the doublets (QC4)

pK is too small

- R package DoubletFinder



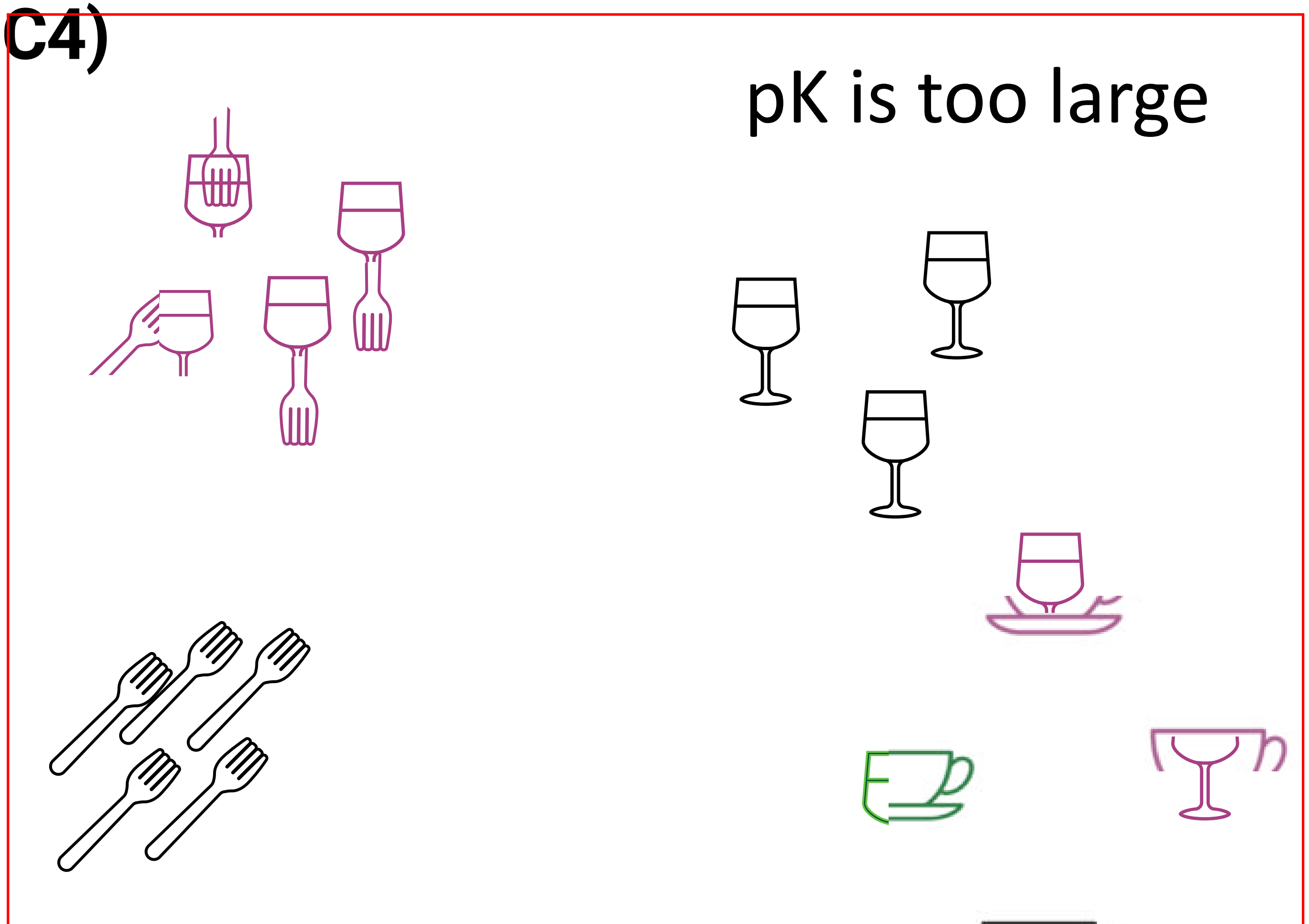




Parameter	Description
nEXP	Expected number of doublets
pN	how many artificial doublets are generated relative to the number of non-empty droplets
pK	the neighborhood size parameter, which determines for a droplet, how many nearby droplets are included to calculate its doublet score

Step 9: Identify the doublets (QC4)

- R package DoubletFinder

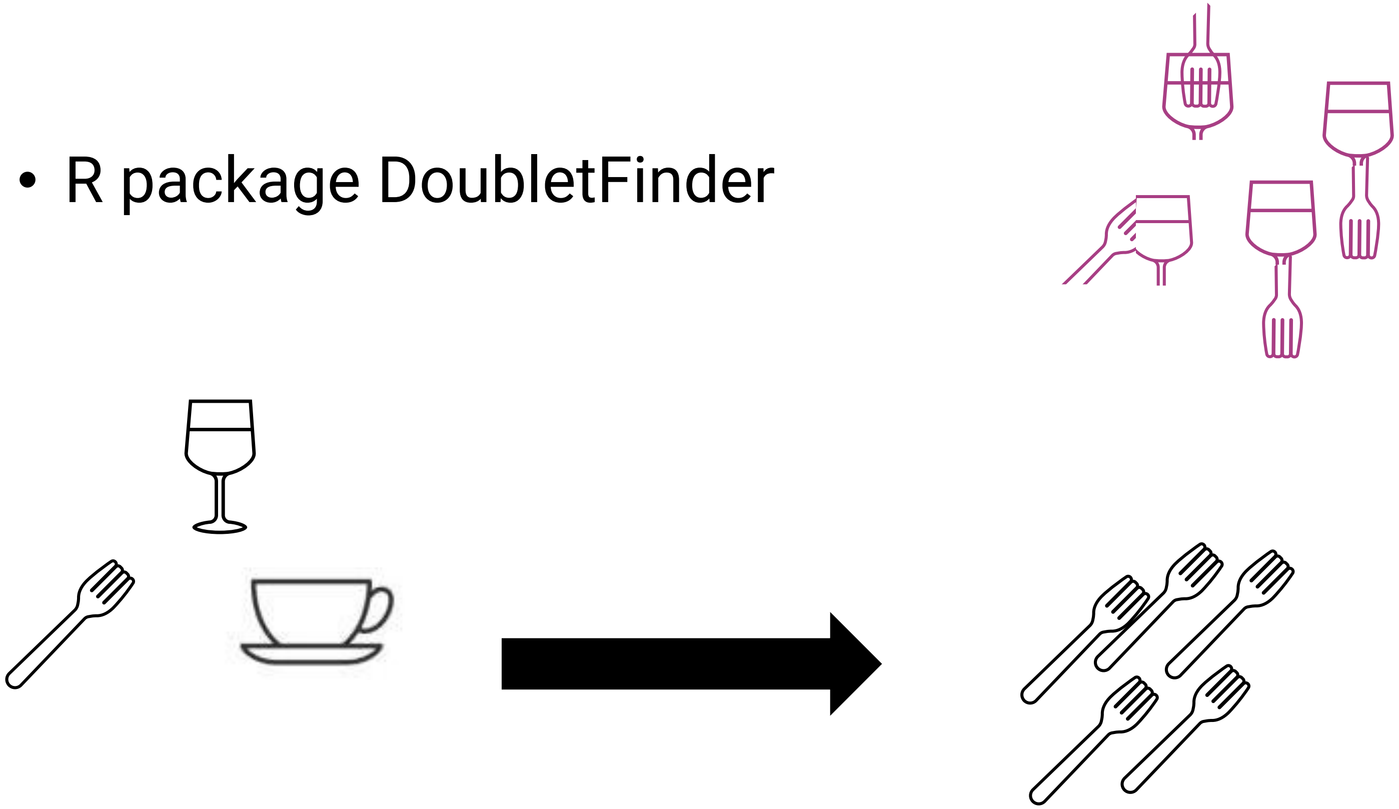


pK is too large

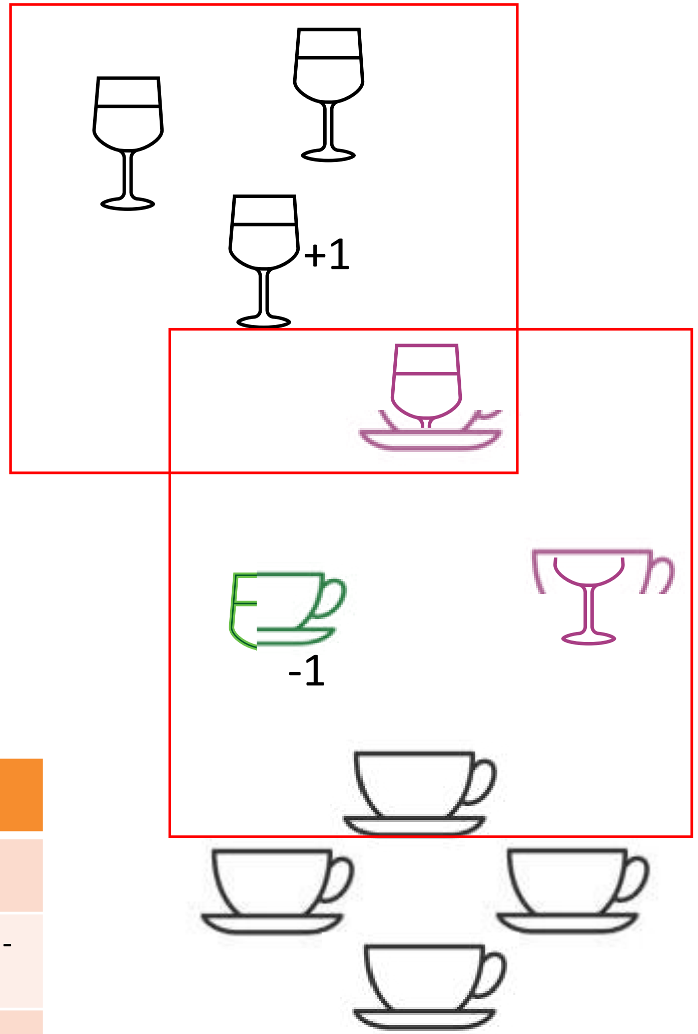
Parameter	Description
nEXP	Expected number of doublets
pN	how many artificial doublets are generated relative to the number of non-empty droplets
pK	the neighborhood size parameter, which determines for a droplet, how many nearby droplets are included to calculate its doublet score

Step 9: Identify the doublets (QC4)

- R package DoubletFinder



pK is just right



Parameter	Description
nEXP	Expected number of doublets
pN	how many artificial doublets are generated relative to the number of non-empty droplets
pK	the neighborhood size parameter, which determines for a droplet, how many nearby droplets are included to calculate its doublet score

UNIVERSITY OF ALBERTA

Step 9: Identify the doublets (QC4)

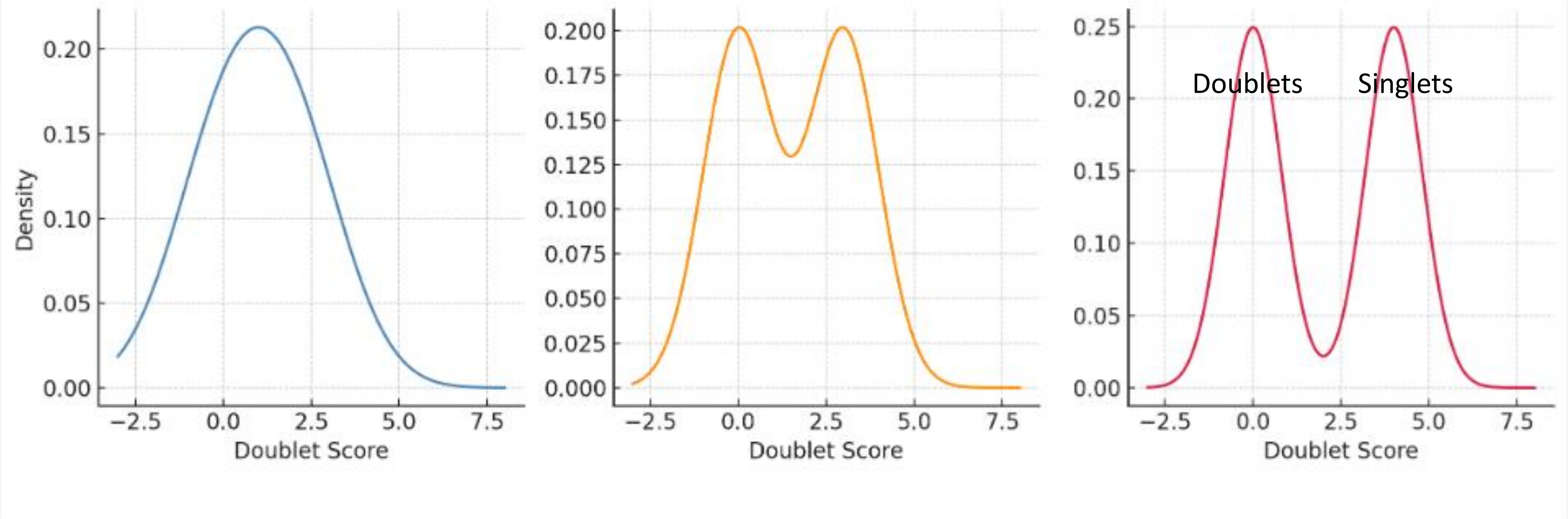
- A series of pK are tested. The best pK is the one that generates highest BC.

Bimodality Coefficient

BC = 0.2

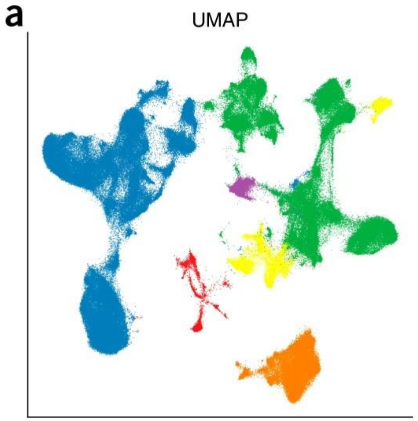
BC = 0.5

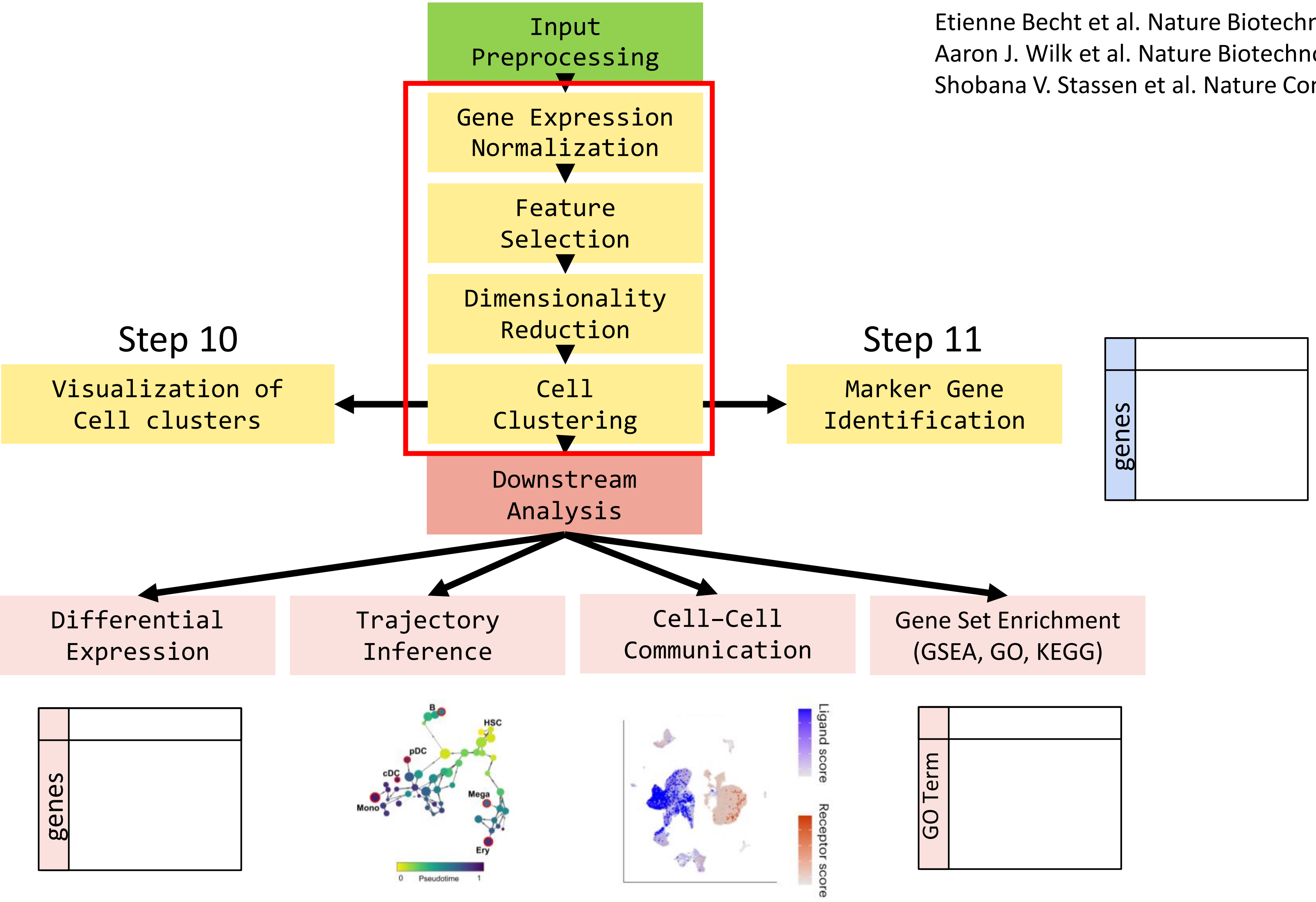
BC = 0.8



Cell Type Annotation

Etienne Becht et al. Nature Biotechnology. 37: 38–44 (2019)
Aaron J. Wilk et al. Nature Biotechnology. 42: 470–483 (2024)
Shobana V. Stassen et al. Nature Communications. 12: 5528 (2021)





Input Preprocessing

Gene Expression Normalization

Feature Selection

Dimensionality Reduction

Cell Clustering

Step 10

Visualization of Cell clusters

Step 11

Marker Gene Identification

Downstream Analysis

Differential Expression

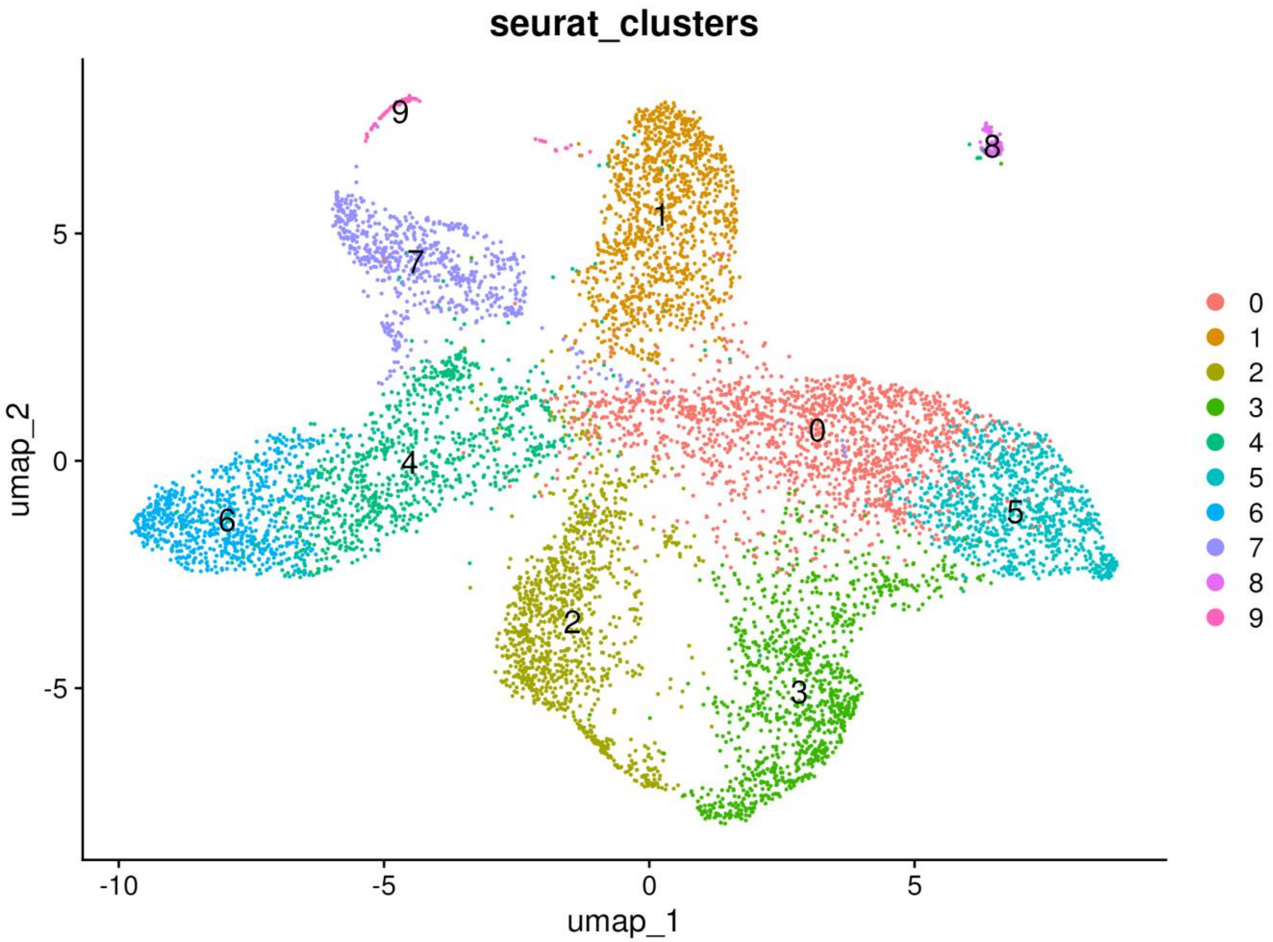
Trajectory Inference

Cell-Cell Communication

Gene Set Enrichment (GSEA, GO, KEGG)

Step 11: Visualization of Cell Clusters

```
seur_filtered <- RunUMAP(seur_filtered, dims = 1:PCs)
picture <- DimPlot(seur_filtered, reduction = "umap", group.by = "seurat_clusters", label = TRUE, label.size = 5)
ggsave("umap_cluster_plot.png", plot = p, width = 8, height = 6, dpi = 300)
```



Step 12: Marker Gene Identification

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

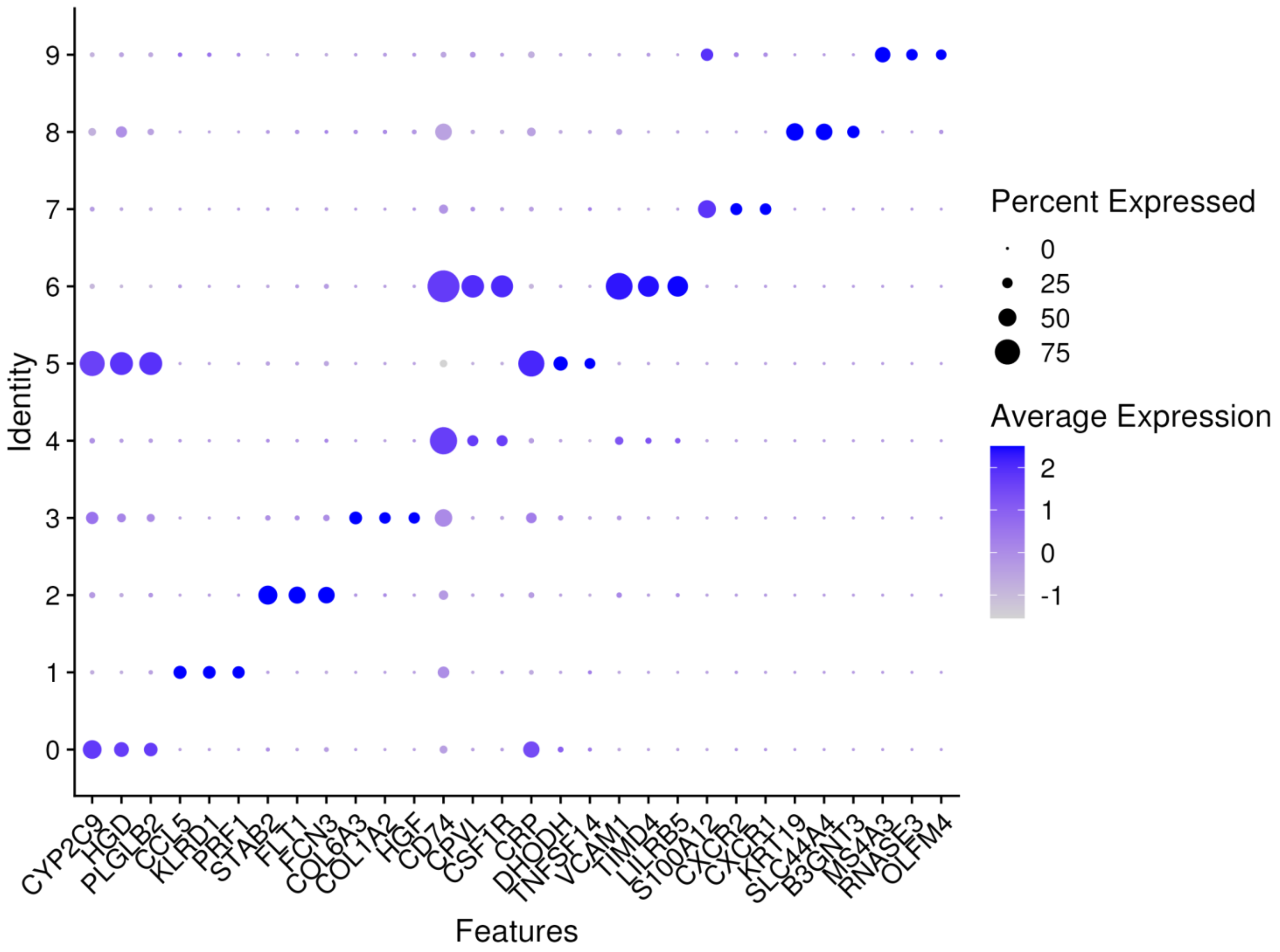
- min.pct = 0.25 Only test genes expressed in ≥25% of cells in either cluster
- Logfc.threshold = 0.25 Only report genes with >1.2-fold change in average expression ($\log_2 X >= 0.25$)

Situation	Suggested Change
You have **rare or small clusters**	Lower min.pct to 0.1
You're interested in **subtle expression differences***	Lower logfc.threshold to 0.1 or 0.15
You're doing exploratory marker discovery	Lower both slightly to get more candidates

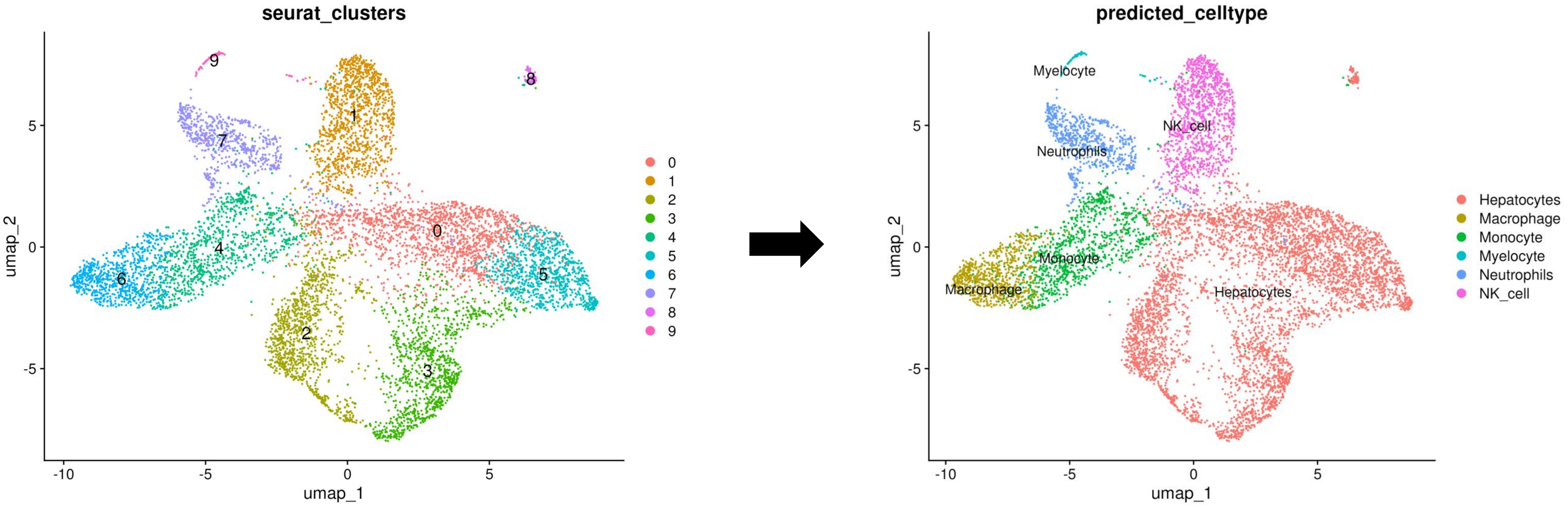
* Early development stage, immune cell activation, autoimmune disease, drug responses/resistance etc

Step 12: Marker Gene Identification

Top 3 highly expressed genes in each cluster



Step 13: Automated Cell-Type Annotation by SingleR



UNIVERSITY OF ALBERTA

Keep In Mind:

- A good understanding of your sample is essential
  - How was it prepared?
  - What are the expected cell types?
  - What genes should be there and what should not?

- A good understanding of the methodology can help you:
  - Optimize the parameters
  - Assess your results
  - Develop new methods

- There are multiple options/tools for each step. Each has pros & cons with different focus and strength. When choosing the tools, you may want to ask:
  - Does it do a good job with your sample and project?
  - Is it easy to integrated into my pipeline?
  - Does it need customization? If so, does it worth my time?

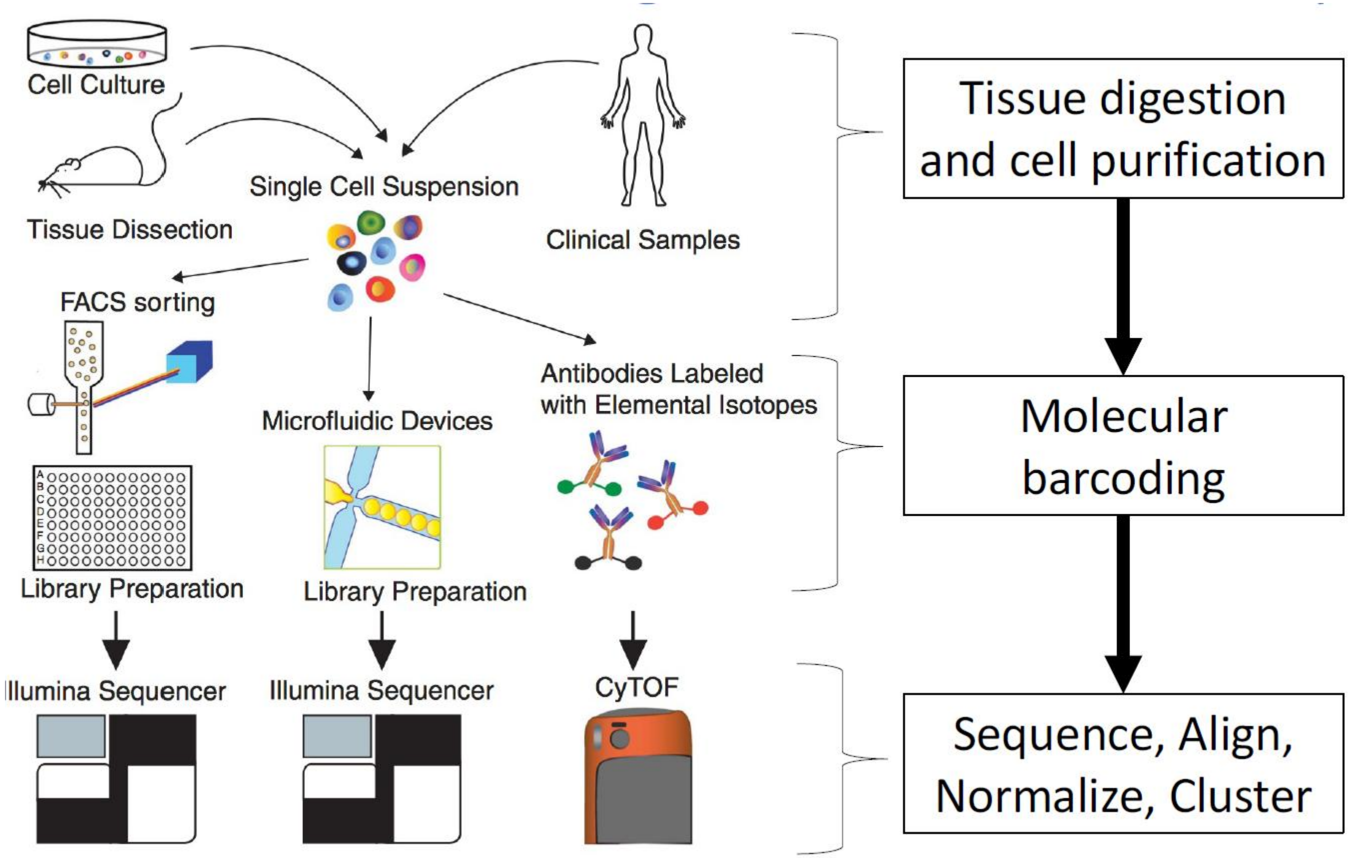
Question?

UNIVERSITY
OF ALBERTA

Thank you!

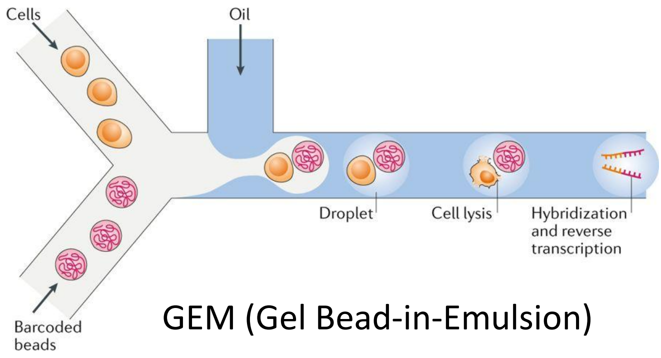
UNIVERSITY OF ALBERTA

scRNA-seq Wet Lab Pipeline



Proserpio and Lönnberg. Immunol Cell Biol. 2016 Mar;94(3):225-9.

Single Cell Isolation



GEM (Gel Bead-in-Emulsion)

Droplet-based
S. Steven Potter, Nature Reviews Nephrology volume 14, pages479–492 (2018)

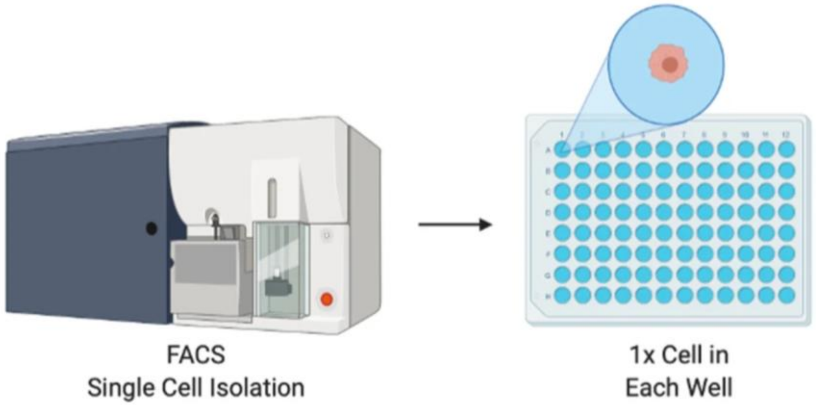
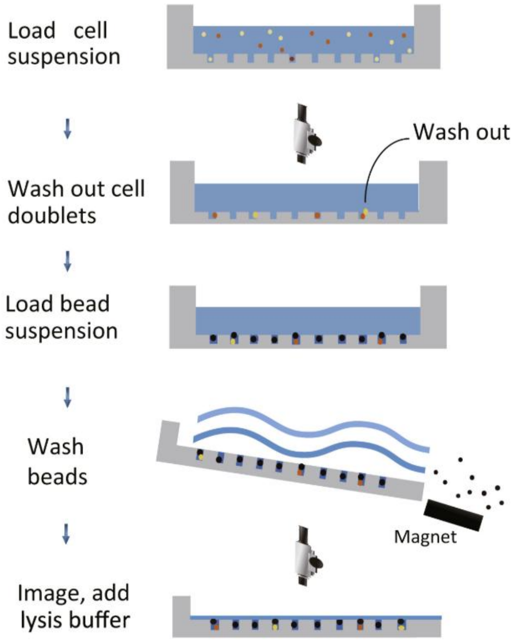
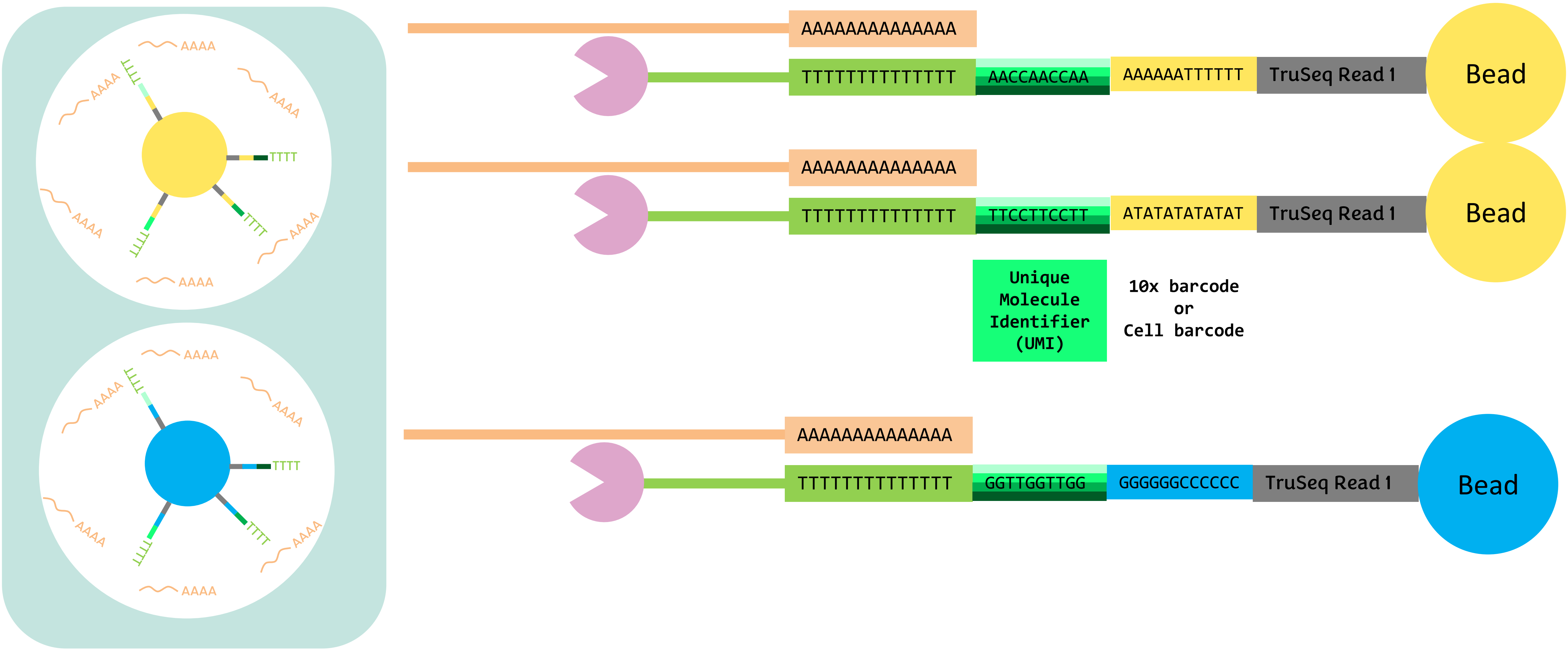


Plate-based
Probst et al., BMC Genomics, 23: 860 (2022)



Microwell-based
Han et al., Cell, 172(5): 1091-1107.e17 (2018)

10x Chromium 3’ scRNA-seq – Reverse Transcription



AAAAAAAAAAAAAA

TTTTTTTTTTTTTTT AACCAACCAA AAAAAATTTTTT TruSeq Read 1 Bead

AAAAAAAAAAAAAA

TTTTTTTTTTTTTTT TTCCTTCCTT ATATATATATAT TruSeq Read 1 Bead

Unique Molecule Identifier (UMI)

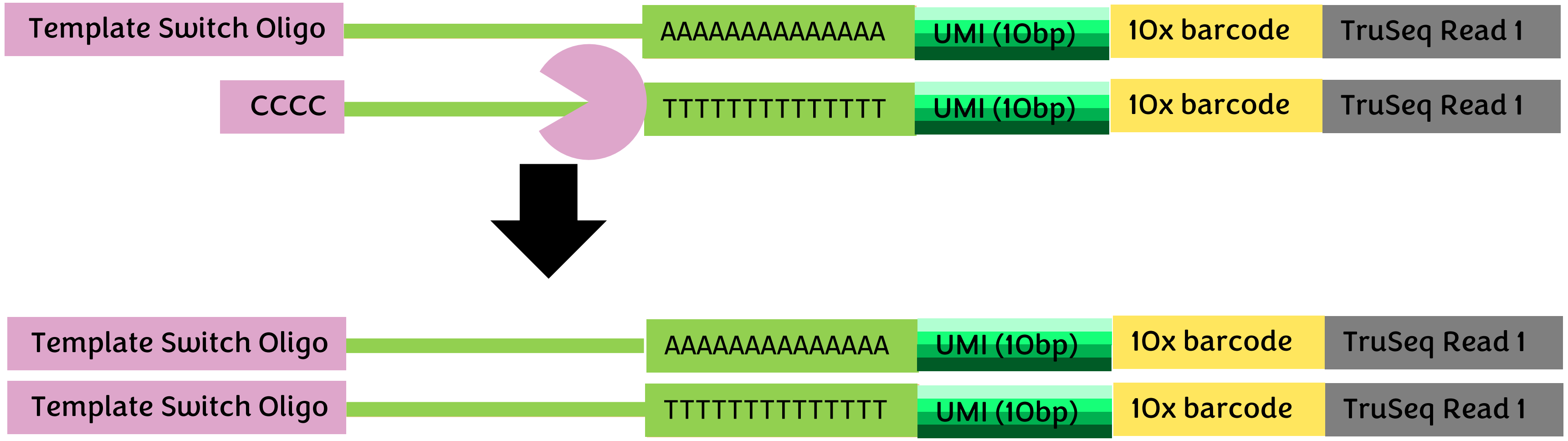
10x barcode or Cell barcode

AAAAAAAAAAAAAA

TTTTTTTTTTTTTTT GGTTGGTTGG GGGGGGCCCCCC TruSeq Read 1 Bead

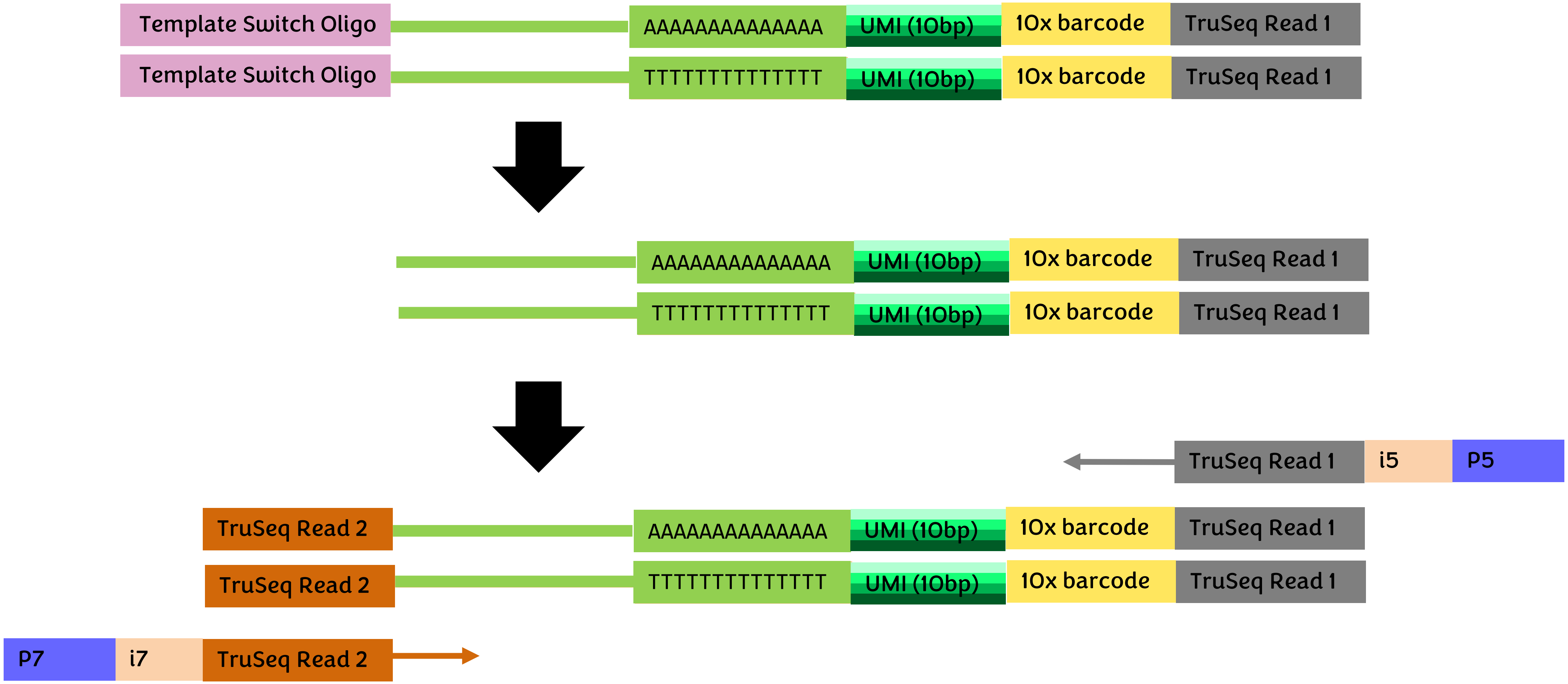
https://www.10xgenomics.com/support/universal-three-prime-gene-expression/documentation/steps/library-prep/chromium-gem-x-single-cell-3-v4-gene-expression-user-guide

10x Chromium 3’ scRNA-seq – Second Strand cDNA



https://www.10xgenomics.com/support/universal-three-prime-gene-expression/documentation/steps/library-prep/chromium-gem-x-single-cell-3-v4-gene-expression-user-guide

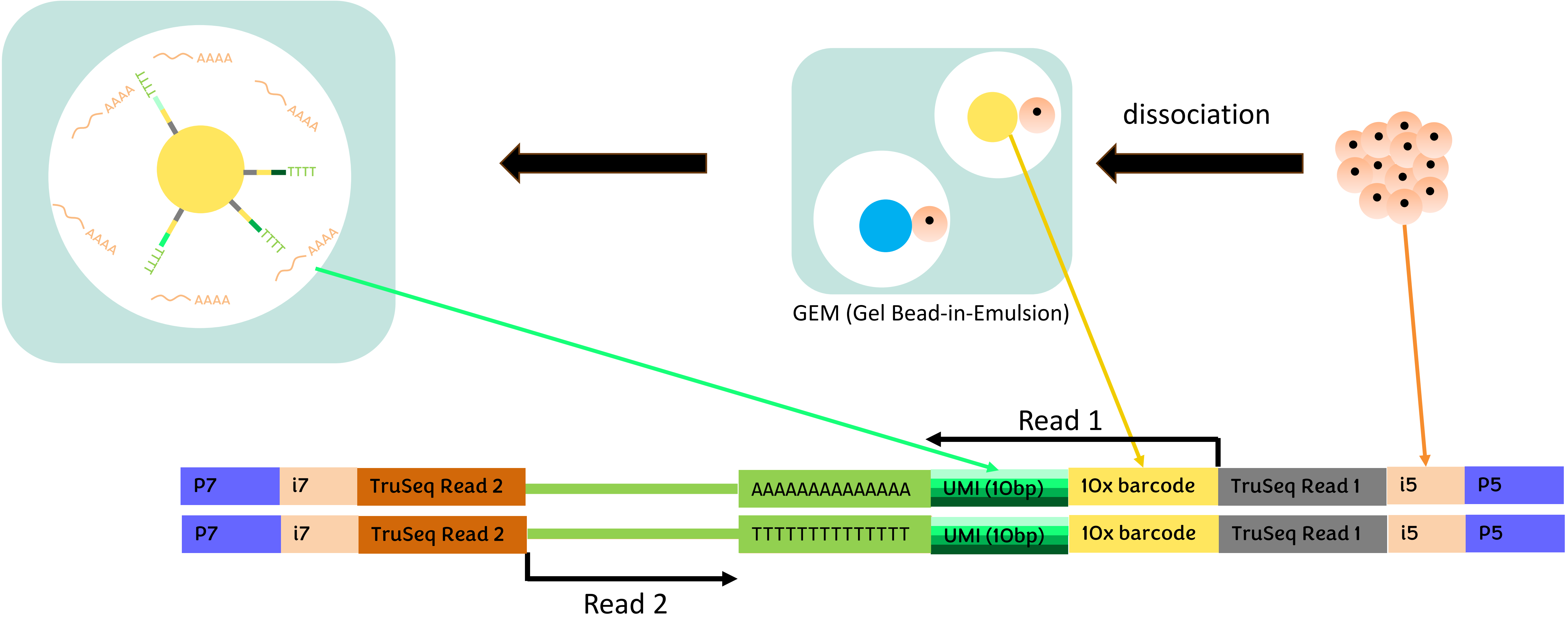
10x Chromium 3’ scRNA-seq – Adding sequencing adapter



UNIVERSITY OF ALBERTA

https://www.10xgenomics.com/support/universal-three-prime-gene-expression/documentation/steps/library-prep/chromium-gem-x-single-cell-3-v4-gene-expression-user-guide

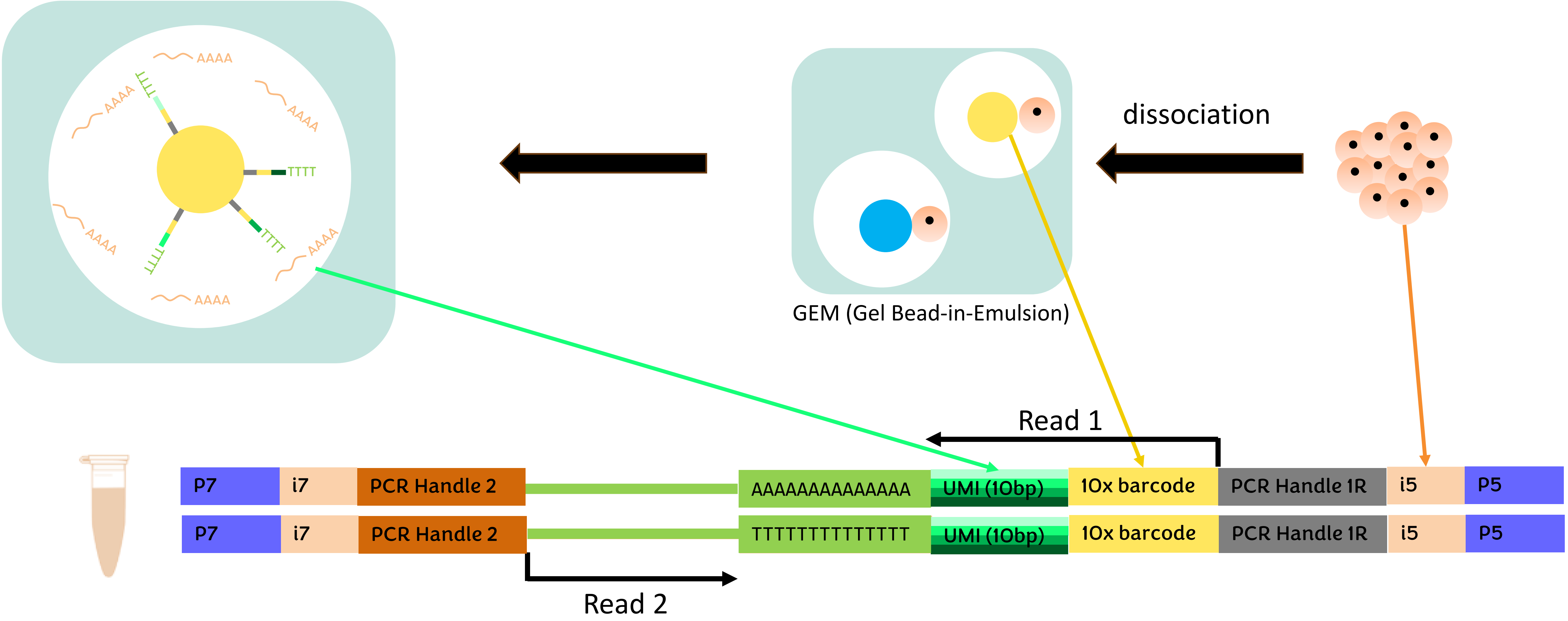
10x Chromium 3’ scRNA-seq – Sample, Cell, Molecule



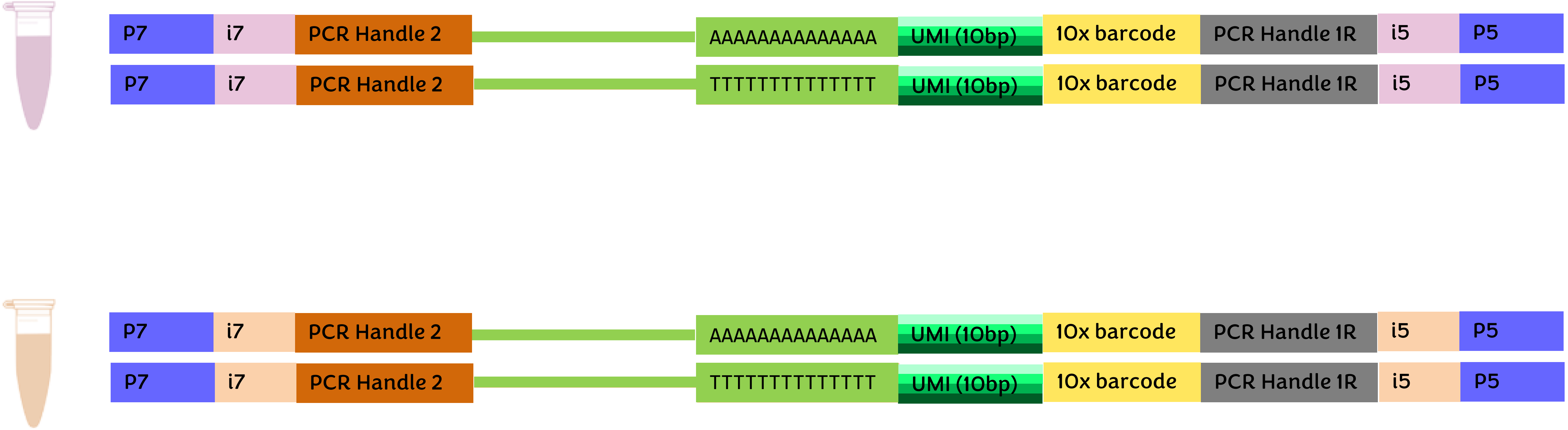
UNIVERSITY OF ALBERTA

https://www.10xgenomics.com/support/universal-three-prime-gene-expression/documentation/steps/library-prep/chromium-gem-x-single-cell-3-v4-gene-expression-user-guide

10x Chromium 3’ scRNA-seq – One Sample



10x Chromium 3’ scRNA-seq – Multiple Sample



P7 | i7 | PCR Handle 2 | AAAAAAAAAAAAAA | UMI (10bp) | 10x barcode | PCR Handle 1R | i5 | P5

P7 | i7 | PCR Handle 2 | TTTTTTTTTTTTTT | UMI (10bp) | 10x barcode | PCR Handle 1R | i5 | P5

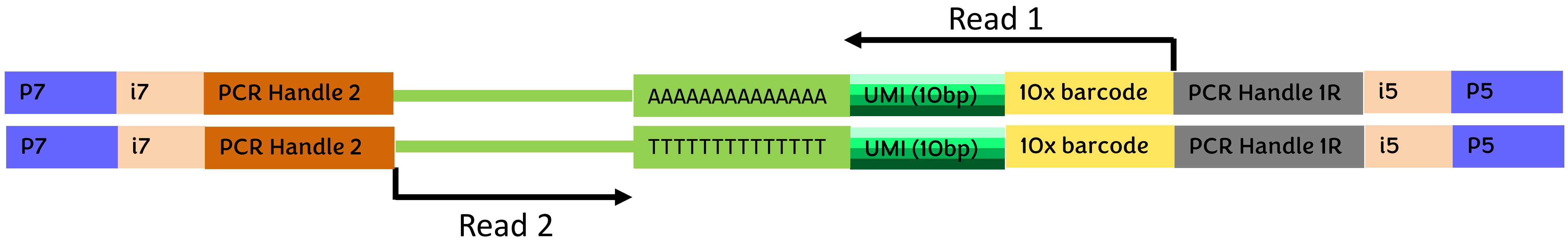
P7 | i7 | PCR Handle 2 | AAAAAAAAAAAAAA | UMI (10bp) | 10x barcode | PCR Handle 1R | i5 | P5

P7 | i7 | PCR Handle 2 | TTTTTTTTTTTTTT | UMI (10bp) | 10x barcode | PCR Handle 1R | i5 | P5

Your Input File: Compressed Fastq

Machine Run ID Lane Tile Coordinate

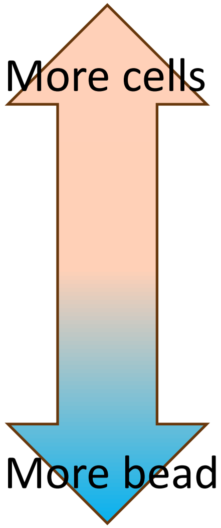
```
@A00469:87:H5WY2DRX2:1:1101:6247:10324 1:N:0:ATCACG
CCGTATGCGGGGCTCCGATTCCATGTCG
+
FFFFFFFFFFFFFFFFFFFFFFFFFFFF
```



```
@A00469:87:H5WY2DRX2:1:1101:6247:10324 2:N:0:ATCACG
AGACCGGCGGAGGGGCTGGGCGGAGGCTCCGAGAGAGCTGAATGAGGCCTTGGAACTCAAGGATGCCCAGGAGGAGCCGCAGTCAGATCCTAGCGTCGA
+
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
```

10x Chromium 3’ scRNA-seq – In Real World

Doublets/multiplets



More cells

More beads

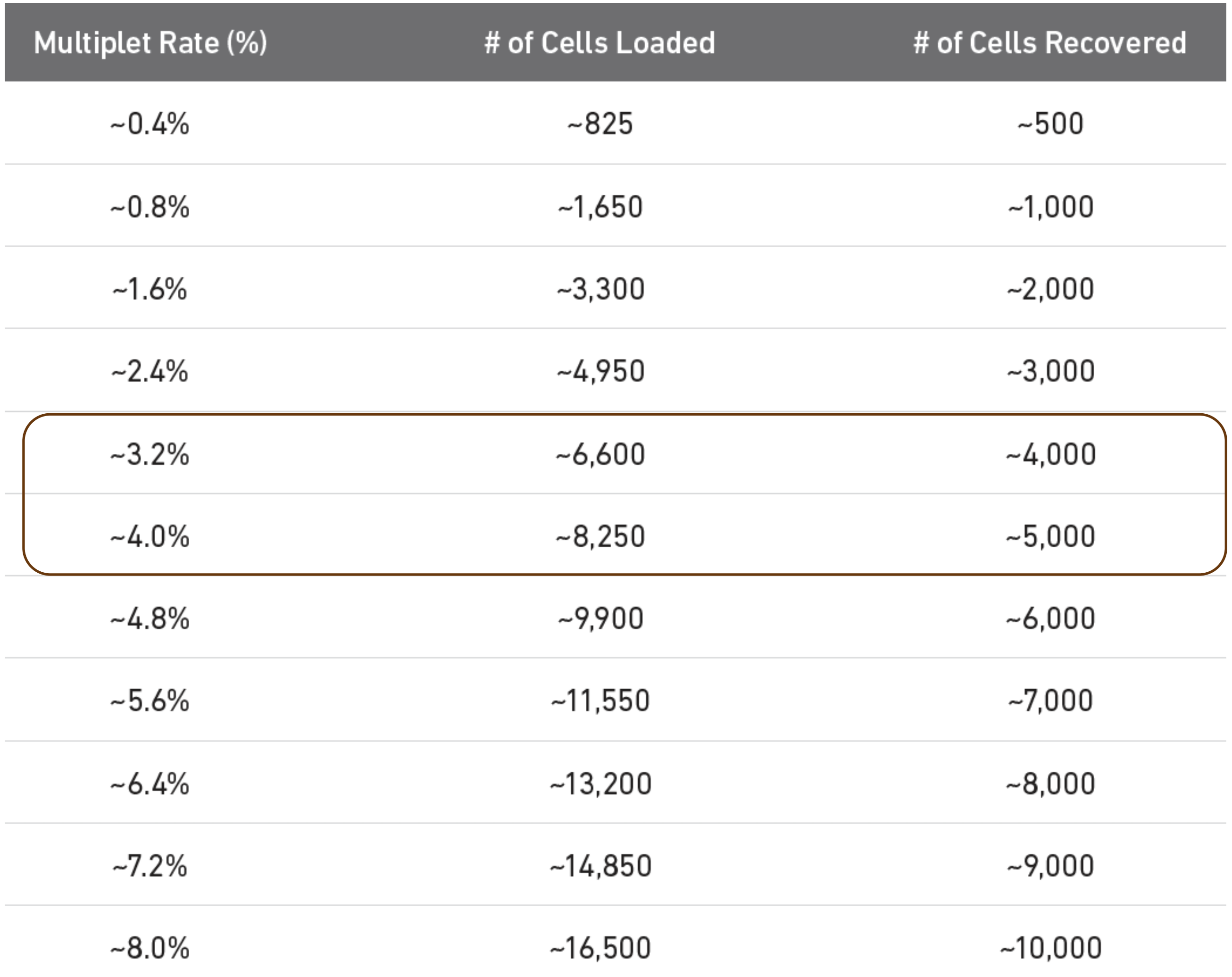
Empty droplets

Ambient RNA

- Sample type
- Tissue dissociation strategy
- Storage and transport conditions
- Wet-lab strategy
- Whether you did a good job

Broken/dead cells

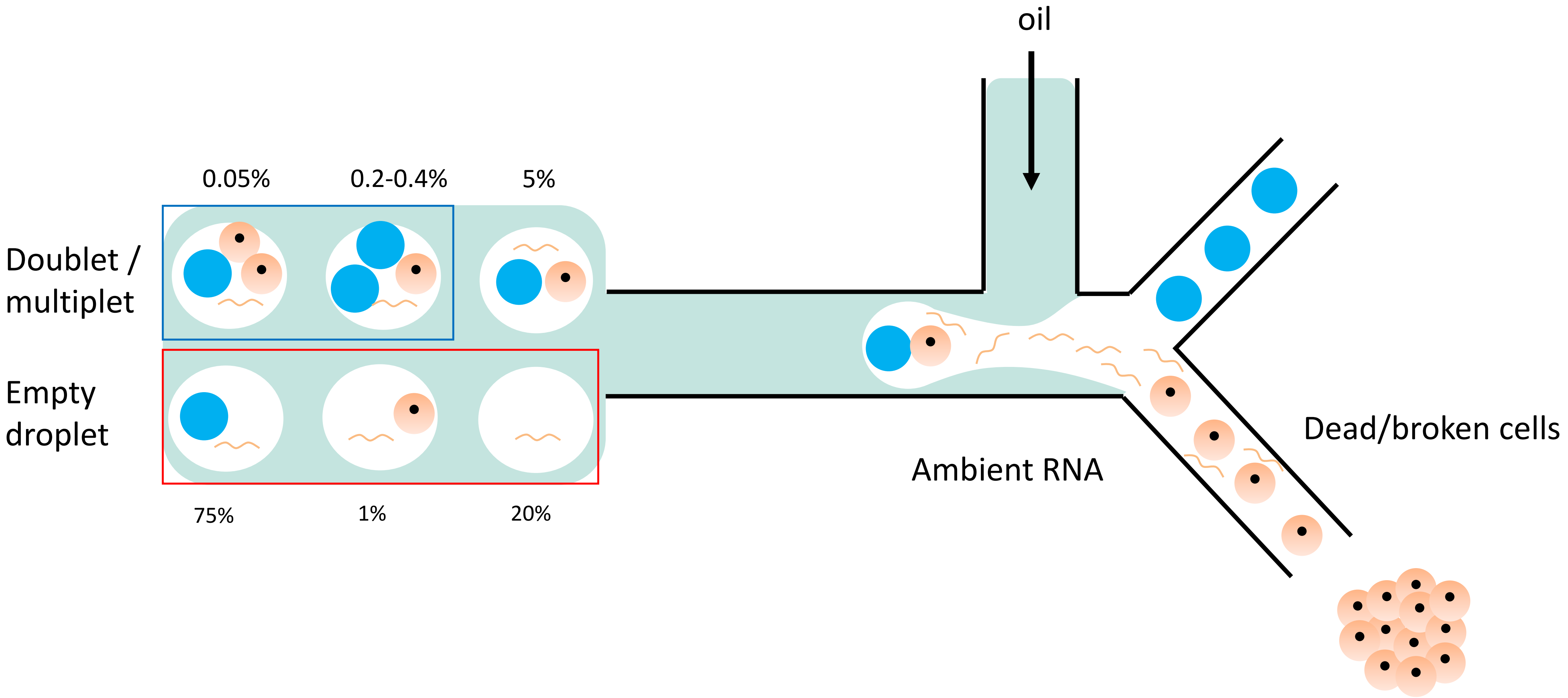
10x Chromium 3’ scRNA-seq – Multiplets vs. Throughput



Multiplet Rate (%)	# of Cells Loaded	# of Cells Recovered
~0.4%	~825	~500
~0.8%	~1,650	~1,000
~1.6%	~3,300	~2,000
~2.4%	~4,950	~3,000
~3.2%	~6,600	~4,000
~4.0%	~8,250	~5,000
~4.8%	~9,900	~6,000
~5.6%	~11,550	~7,000
~6.4%	~13,200	~8,000
~7.2%	~14,850	~9,000
~8.0%	~16,500	~10,000

https://kb.10xgenomics.com/hc/en-us/articles/360001378811-What-is-the-maximum-number-of-cells-that-can-be-profiled

Quality Control



Lareau et al. (2020) doi: 10.1038/s41467-020-14667