Practical Example: NGS – data handling and single cell differentiation

Ivan G. Costa, Mingbo Cheng & James Nagai

Institute for Computational Genomics

Joint Research Centre for Computational Biomedicine

RWTH Aachen University, Germany

Institute for Computational Genomics
01011011010101
10100100101

RWTH AACHEN UNIVERSITY

Contact Information

MINGBO CHENG & JAMES NAGAI

james.nagai@gmail.com

mcheng@ukaachen.de

Overview

- DNA Sequencing: What's new?
  - New Generation Sequencing Methods
  - Sequencing Quality Check
  - Alignment of Sequenced Reads
- Single Cell Sequencing
  - Demultiplexing Reads
  - Single Cell Expression Matrix
  - From the Expression Matrix to Information Retrieval

Useful URL

- Course material: https://www.costalab.org/teaching/software-lab-in-bioinformatics-2020/
- Single Cell Broad Institute Workshop: https://broadinstitute.github.io/2020_scWorkshop/
- Best Practices in Single Cell Analysis: https://www.embopress.org/doi/10.15252/msb.20188746

DNA Sequencing

- Problem: Converting a DNA molecule to a string
  - string: sequence of bases (A, T, C, G,N)

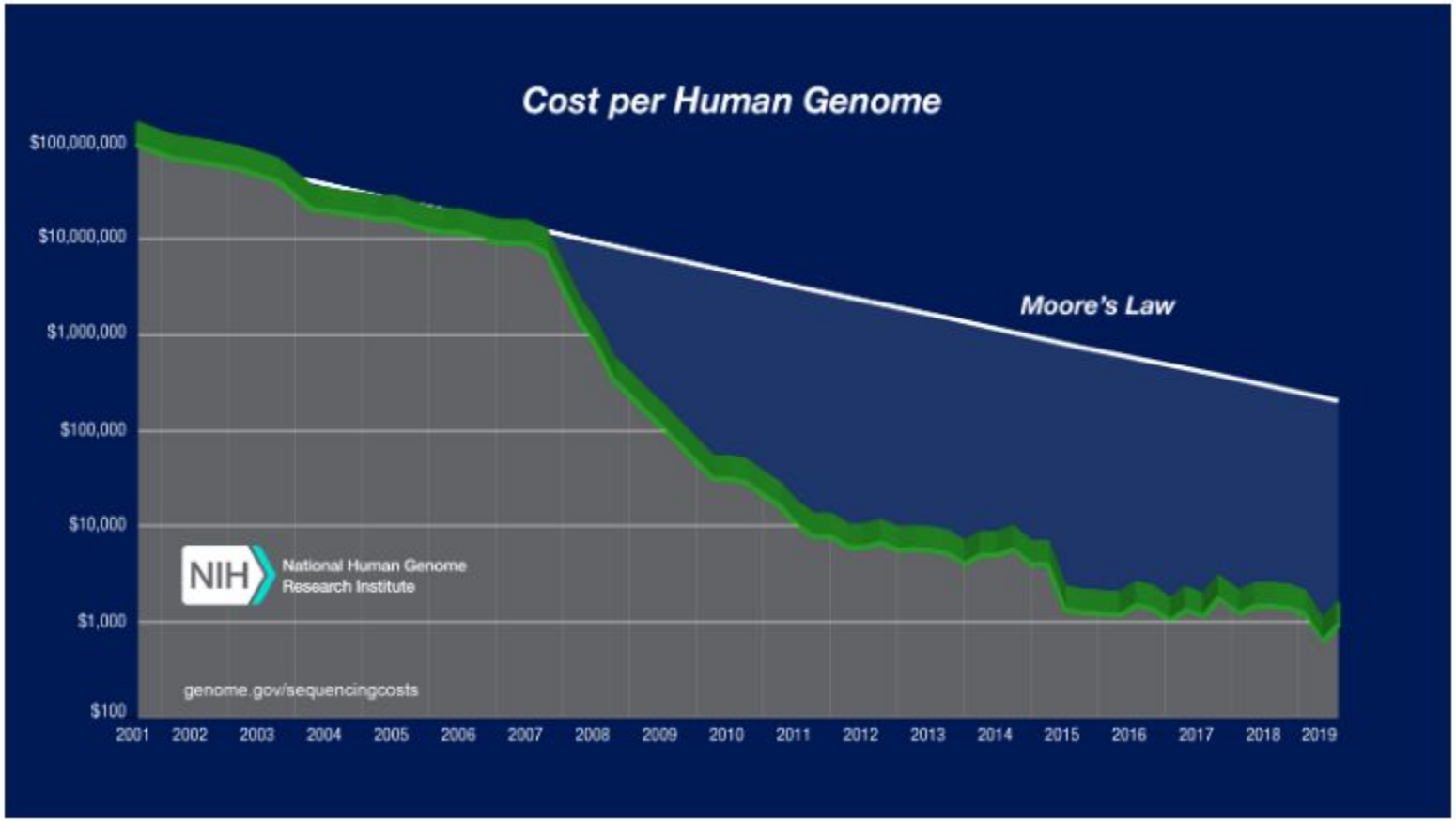
- Many possible sequencing techniques exist:

illumina®

PACBIO®

Oxford NANOPORE™ Technologies

DNA Cost



Cost per genome data

https://www.genome.gov/about-genomics/fact-sheets/Sequencing-Human-Genome-cost

Institute for Computational Genomics
01011011010101
10100100101

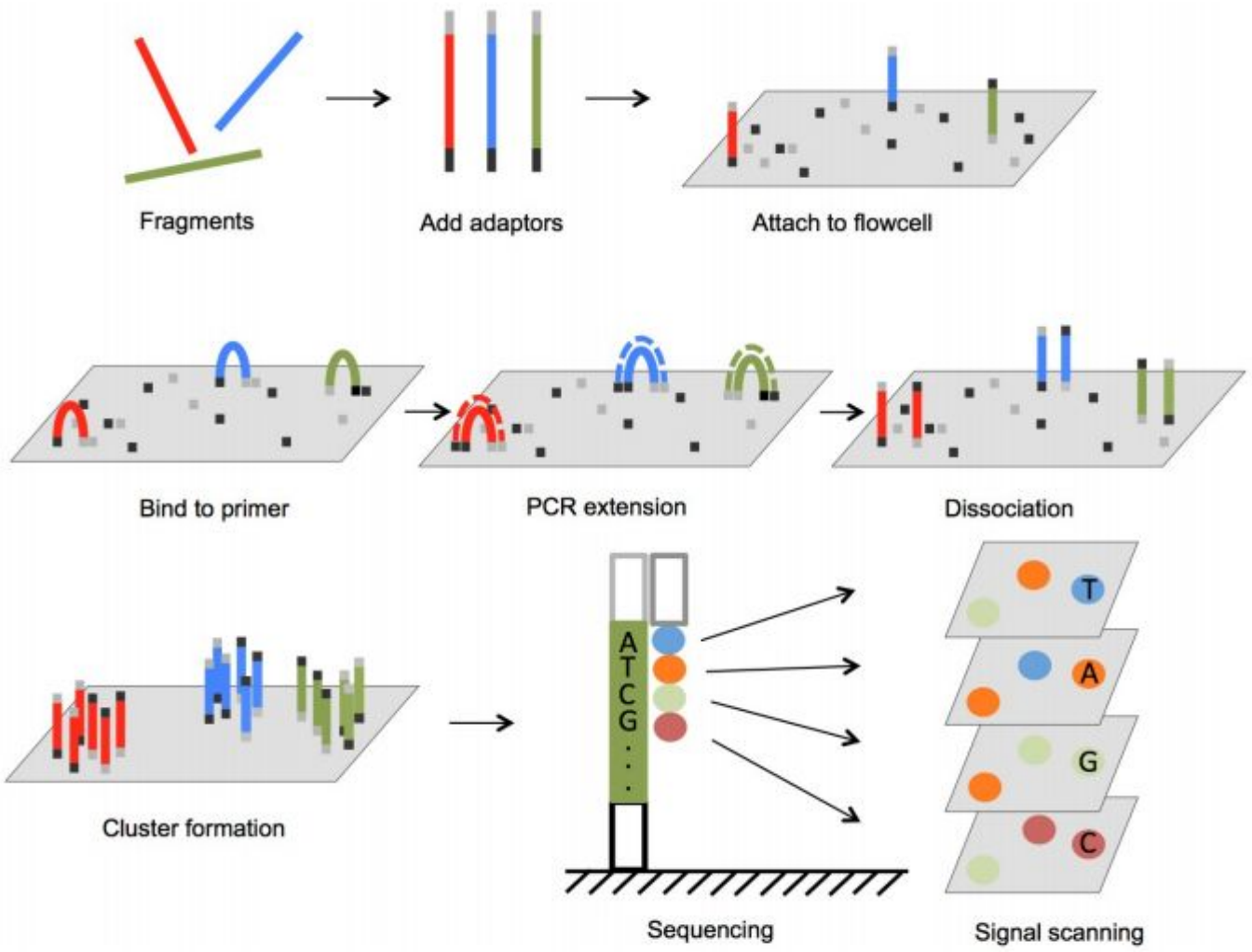
RWTH AACHEN UNIVERSITY

FASTA file

- Can be used to store DNA/protein sequence in a text-based file
- Mainly used to store large genomic sequences
- Header (lines that start with '>') + DNA sequence
- Alphabet: A, C, G, T, N

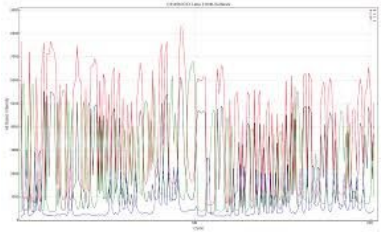
```
>NC_005248.1:3950-4810 Escherichia coli plasmid pIGAL1, complete sequence
ATGAGTATTCAACATTTCCGTGTCGCCCTTATTCCCTTTTTTGCGGCATTTTGCCTTCCTGTTTTTGCTC
ACCCAGAAACGCTGGTGAAAGTAAAAGATGCTGAAGATCAGTTGGGTGCACGAGTGGGTTACATCGAACT
GGATCTCAACAGCGGTAAGATCCTTGAGAGTTTTCGCCCCGAAGAACGTTTTCCAATGATGAGCACTTTT
AAAGTTCTGCTATGTGGCGCGGTATTATCCCGTGTTGACGCCGGGCAAGAGCAACTCGGTCGCCGCATAC
ACTATTCTCAGAATGACTTGGTTGAGTACTCACCAGTCACAGAAAAGCATCTTACGGATGGCATGACAGT
AAGAGAATTATGCAGTGCTGCCATAACCATGAGTGATAACACTGCGGCCAACTTACTTCTGACAACGATC
GGAGGACCGAAGGAGCTAACCGCTTTTTTGCACAACATGGGGGATCATGTAACTCGCCTTGATCGTTGGG
AACCGGAGCTGAATGAAGCCATACCAAACGACGAGCGTGACACCACGATGCCTGCAGCAATGGCAACAAC
GTTGCGCAAACTATTAACTGGCGAACTACTTACTCTAGCTTCCCGGCAACAATTAATAGACTGGATGGAG
```

Illumina



Lu, Yuan, et al. "Next Generation Sequencing in Aquatic Models." 2016.
https://www.youtube.com/watch?time_continue=258&v=fCd6B5HRaZ8&feature=emb_logo

From signal to string data



Signal: Ilumina HiSeq

How we can measure the quality of these algorithms?

Base detection algorithms

ACCTTCGAACGGCGGGGGTTACAA

FASTQ

● Also text-based. Mainly used to store short DNA sequences (reads) from NGS-based experiments. ● Line 1: Begins with '@' and is followed by a an identifier.

● Line 2: DNA sequence.

● Line 3: Begins with '+' and is optionally followed by the same sequence identifier (and any description) again.

● Line 4: Quality values for the sequence in Line 2, and must contain the same number of symbols as the sequence.

```
@NS500746:56:HLY2MAFXX:1:11101:10096:1016 1:N:0:1
GCCTGNCGCATTGCATTCATCAAACGCTGAATAGCAAAGCCTCTACGCGATTTCATAGTGGAGGCCTCCAGCAATCTTGAACACTC
ATCCTTAATACCTTTCTTTTTGGGGTAATTATACTCATCGCGAATATCCTTAAGAGGGCGTTCAGCAGCCAGCTTGCGG
+
AAAAA#EEEEEEEEEAEE/AEEEAEEEEEEEEEEAEE/EEEEEEEEEEA/EE<AEEEEEEEEEEEEAEAEAE/EEEEEE/E<AEE
<EE/EEE//E/AEEEEE<AAEEEEEEEEEEAEE/EAAA/AEAEEEEEE/EEEAA<AAEA<EEAEAEEE</AE<A/A<<<
```

FASTQ Evaluation – FastQC

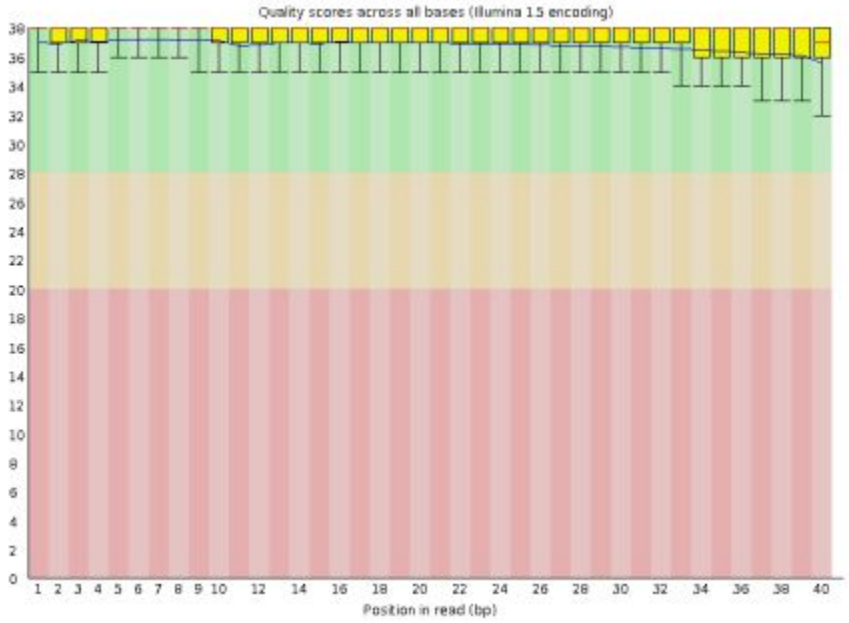
Fastq files can be very big with millions of (long) reads. Infeasible to investigate.

- Phred-Score hard to read in ASCII form.
- FastQC (usually provided by NGS core facilities)
- Tool to analyse quality of reads from sequencing.
- Indicate problems in library preparation or sequencing steps.

Example – **good quality sequences**
http://www.bioinformatics.babraham.ac.uk/projects/fastqc/good_sequence_short_fastqc.html
Example – **bad quality sequences**
http://www.bioinformatics.babraham.ac.uk/projects/fastqc/bad_sequence_fastqc.html

FASTQ Evaluation – FastQC

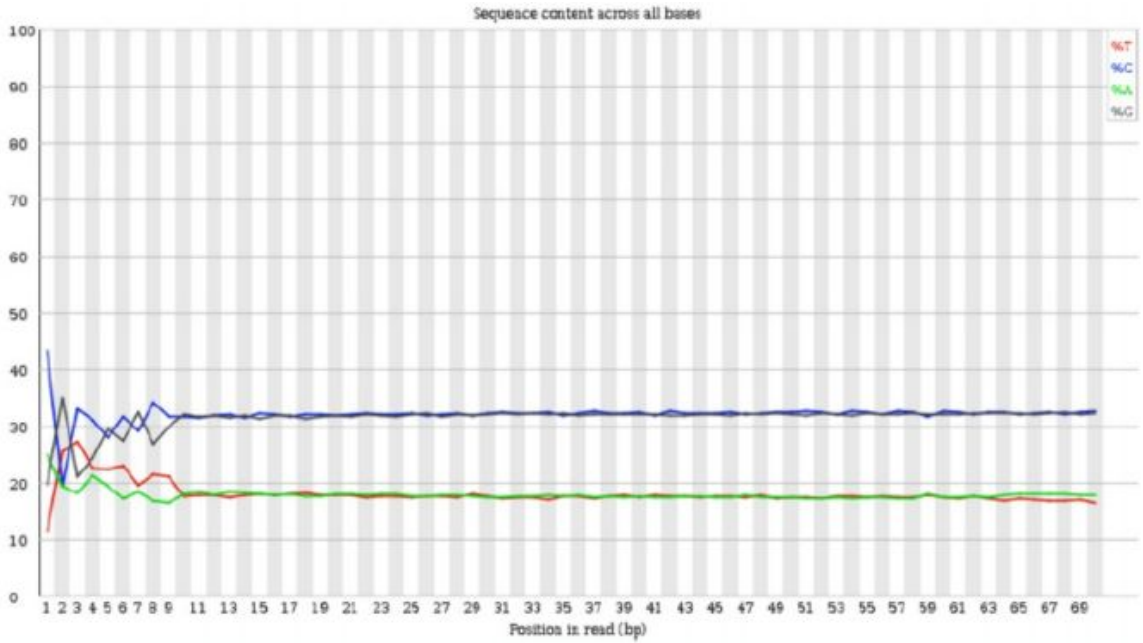
Sequencing quality decreases with size.



Solution: trim ends of reads, if quality is low.

FASTQ Evaluation – FastQC

Read position sequence bias.



Solution: Trim starts of reads.

Hands-on time

- Download data1.zip from the lecture website.

- Use FastQC to analyze the data:
  - create new directory “fastqc_results”
  - read the documentation of FastQC to understand how to export the files to the new directory:
  - fastqc -h
  - What do you see?
    What is the overall quality?
    Do we have any adapters?

- Trim the reads from the identified adapter using trim_galore (trim_galore –help) in a new folder “trimmed_results”. Again analyze the fastq. What do you see? Are the adapters gone?

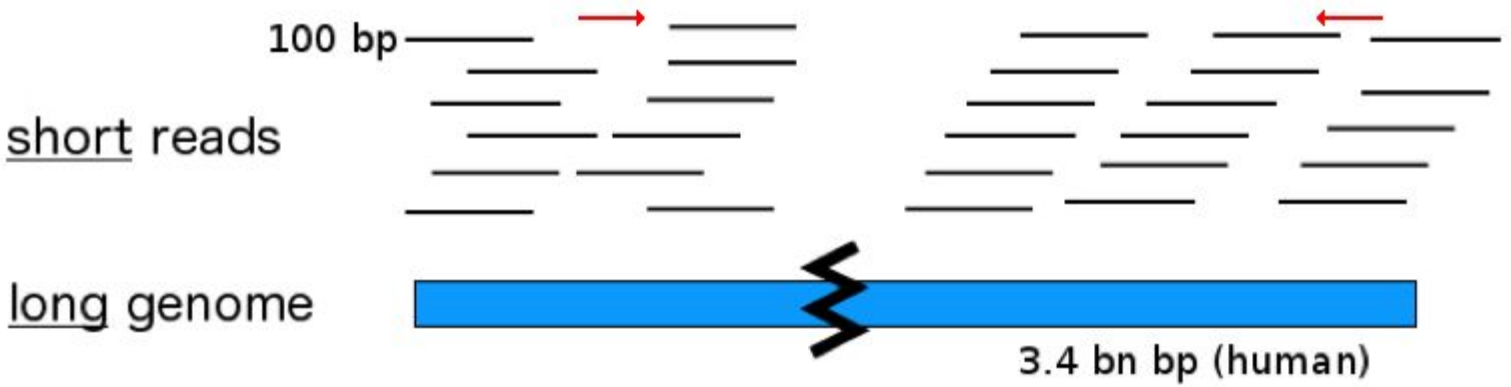
Hands-on time

- fastqc -o fastqc_results/ ERR522959_1.fastq.gz ERR522959_2.fastq.gz.

- trim_galore –nextera -o trimmed_results/ ERR522959_1.fastq.gz ERR522959_2.fastq.gz
- fastqc -o trimmed_results/ trimmed_results/ERR522959_1_trimmed.fq.gz trimmed_results/ERR522959_2_trimmed.fq.gz

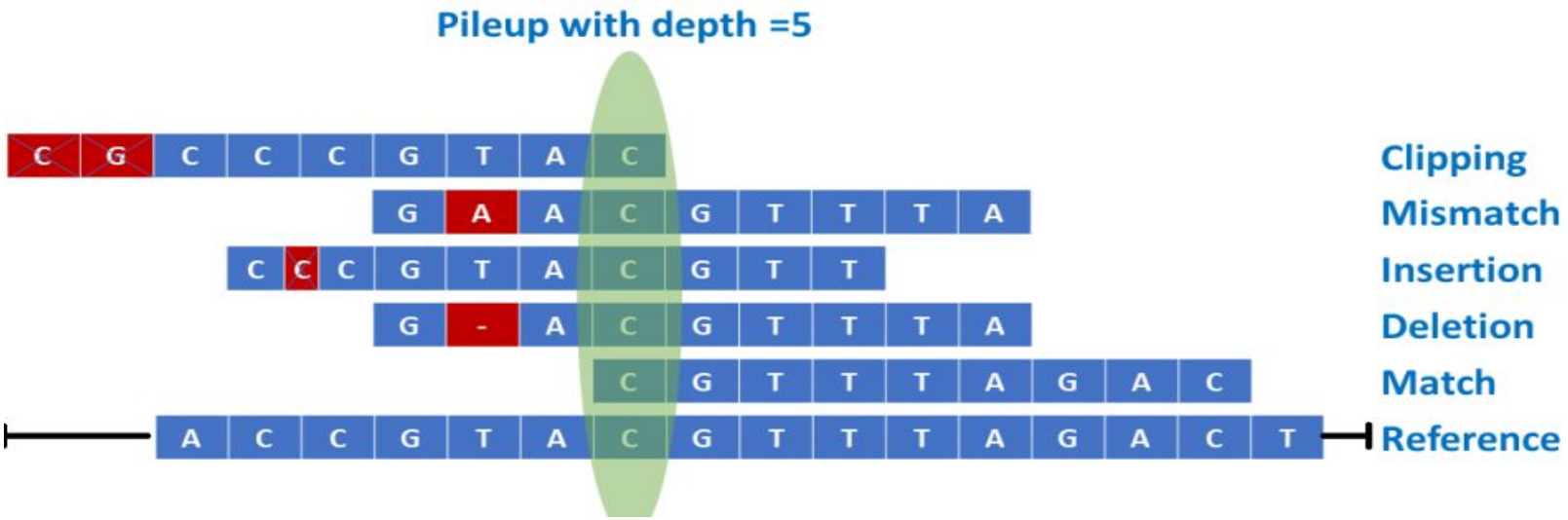
Alignment

Usually very large genomes (with repetitive regions) and very small reads.



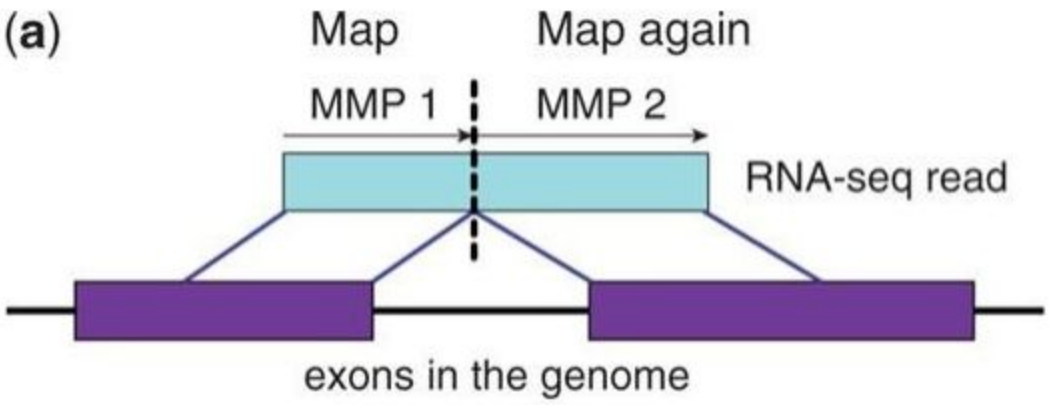
Alignment

Problem: aligning DNA sequence to a reference genome.



STAR

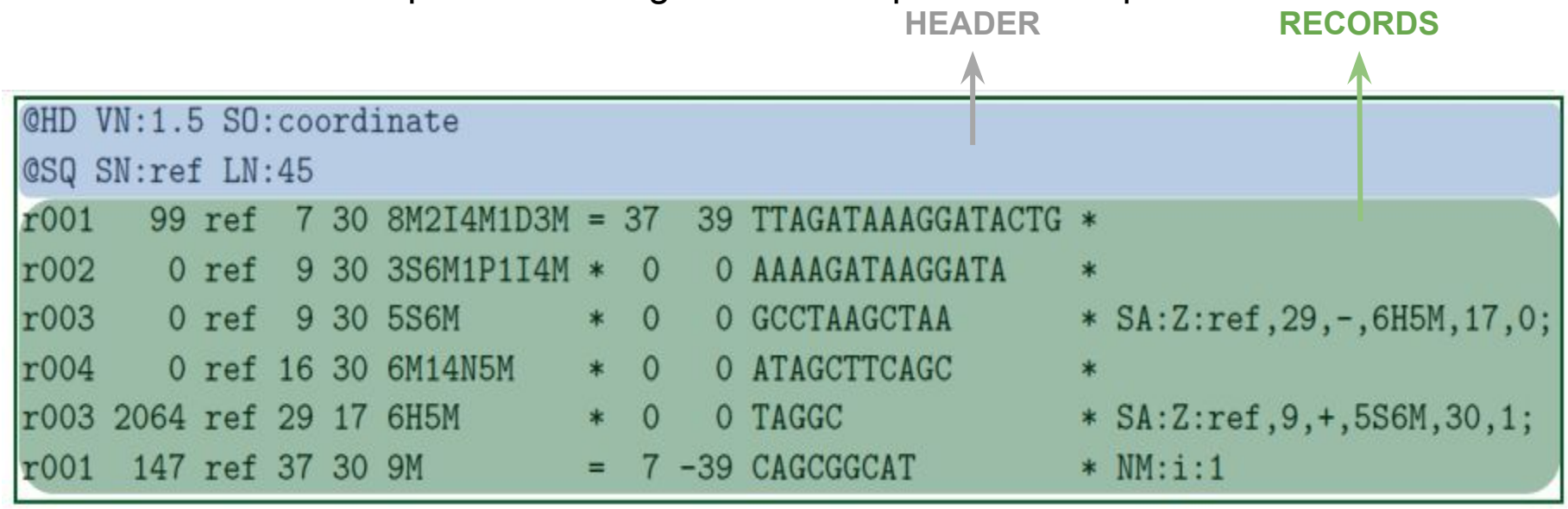
STAR allows a sequence to be split and aligned to different exons



Source: Dobin et al. (2013), Bioinformatics.

SAM File

- Sequence Alignment/Map format.
- Text-based tab-delimited file.
- Header + records (aligned reads)
  - Information: https://samtools.github.io/hts-specs/SAMv1.pdf



SAM File

Col	Field	Type	Regexp/Range	Brief description
1	QNAME	String	`[!-?A-~]{1,255}`	Query template NAME
2	FLAG	Int	`[0,2^16-1]`	bitwise FLAG
3	RNAME	String	`\*|[!-()+-<>-~][!-~]*`	Reference sequence NAME
4	POS	Int	`[0,2^31-1]`	1-based leftmost mapping POSition
5	MAPQ	Int	`[0,2^8-1]`	MAPping Quality
6	CIGAR	String	`\*|([0-9]+[MIDNSHPX=])+`	CIGAR string
7	RNEXT	String	`\*|=|[!-()+-<>-~][!-~]*`	Ref. name of the mate/next read
8	PNEXT	Int	`[0,2^31-1]`	Position of the mate/next read
9	TLEN	Int	`[-2^31+1,2^31-1]`	observed Template LENgth
10	SEQ	String	`\*|[A-Za-z=.]+`	segment SEQuence
11	QUAL	String	`[!-~]+`	ASCII of Phred-scaled base QUALity+33

```
@HD VN:1.5 SO:coordinate
@SQ SN:ref LN:45
r001   99 ref  7 30 8M2I4M1D3M = 37  39 TTAGATAAAGGATACTG *
r002    0 ref  9 30 3S6M1P1I4M *  0   0 AAAAGATAAGGATA    *
r003    0 ref  9 30 5S6M       *  0   0 GCCTAAGCTAA       * SA:Z:ref,29,-,6H5M,17,0;
r004    0 ref 16 30 6M14N5M    *  0   0 ATAGCTTCAGC       *
r003 2064 ref 29 17 6H5M       *  0   0 TAGGC             * SA:Z:ref,9,+,5S6M,30,1;
r001  147 ref 37 30 9M         =  7 -39 CAGCGGCAT         * NM:i:1
```

BAM File

- Binary Alignment/Map format – compressed version of SAM.
- Compression: BGZF block compression.
- Efficient random access: UCSC bin/chunk scheme.
- BAI index files.
- More Information:
  http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2723002/
  http://www.ncbi.nlm.nih.gov/pmc/articles/PMC186604/

In summary

Reference sequence (FASTA)

```
>chr1
TACCTCCAGGGGGCATCCTCCCCCCCAATTCG
AAACACAATCGTAGCCCCTGGCACTACCTATG
TGTGTCAATTCGGAGAGAGAGAGATTCACGAA
AAAAAAGTCTGGACTCAACTAGGATACACACA
TTCGGCTACAGATACCAAAAAAAAAAAAAAAA
AAATTTTCACCATTGAGGCACCACCTTCTCGT
CGCTGCGTCGCTCTGCTCGCTTCGGCTAAAAA
TTCGCGCAATACATTCGGCTACAGATACCAAA
```

Unaligned reads

```
@seq1
ATTCGAAACA...
+
DDED88(999...
@seq2
CCCCGTTTCA...
+
AAC887BBAC...
```

Alignment/ Mapping

SAM/BAM format aligned reads

```
seq1   99     1      3666901      60
149M   =      3666935      185
ATTCGAAACA...DDED88(999    MC:Z:151M
MD:Z:149      RG:Z:15-0017315_1   NM:i:0
MQ:i:60       AS:i:149      XS:i:44
seq2   147    1      3666935      60
151M   =      3666901      -185
CCCCGTTTCA...AAC887BBAC...MC:Z:149M
MD:Z:151      RG:Z:15-0017315_1   NM:i:0
MQ:i:60       AS:i:151      XS:i:59
```

Samtools

- Provides various utilities for manipulating alignments in the SAM format.

- Tools useful for quality check and bias correction.

- More Information:
  Paper: http://www.ncbi.nlm.nih.gov/pubmed/19505943
  Website: http://samtools.sourceforge.net/

Hands-on time

- Download data2.zip from the lecture website.

- Use STAR to align the reads to the supplied small reference genome (smaller_reference.fa) and output sam file

- FIRST! Index the genome: STAR --runThreadN 4 --runMode genomeGenerate --genomeDir output_dir/ --genomeFastaFiles smaller_reference.fa STAR –help # for manuals

- Convert the SAM file to BAM (samtools view –help)

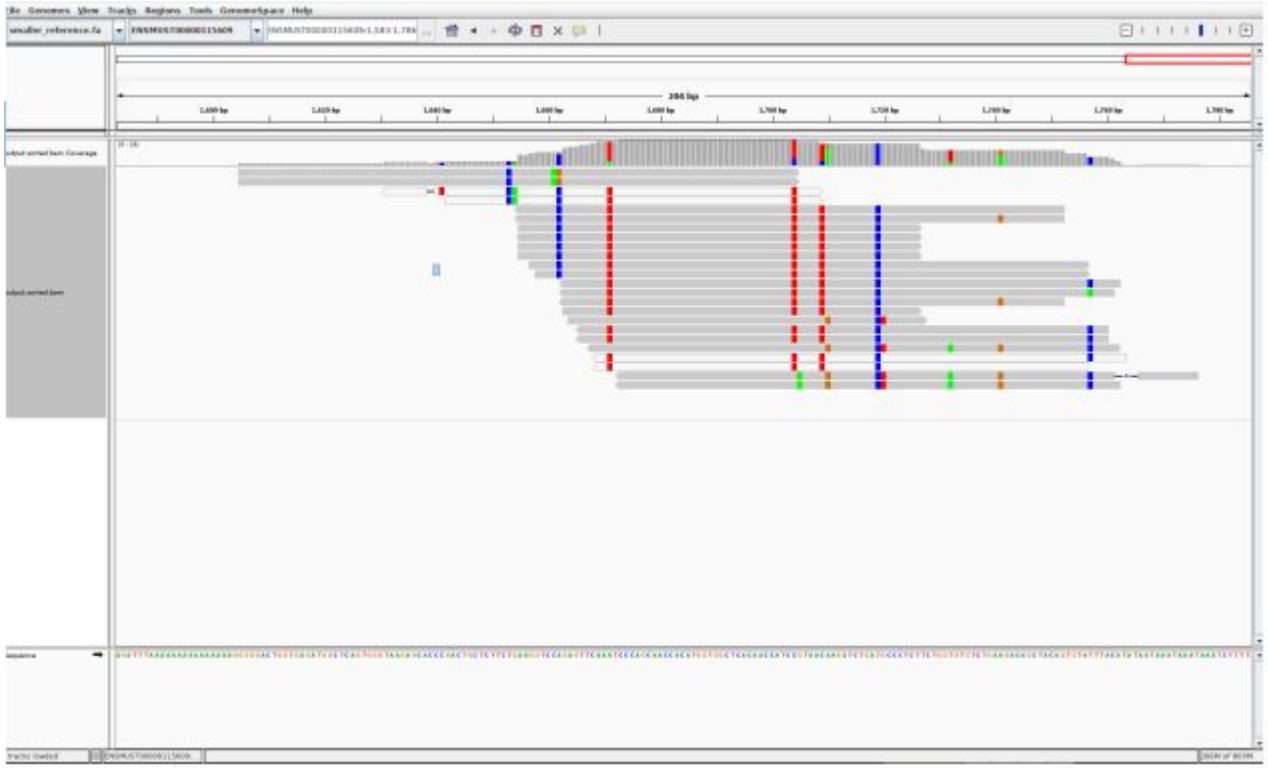
- Sort and index (samtools sort; samtools index)

IGV

- Tool for visualising sequences, reads and/or variants at genomic locations

- Open IGV. From menu: Genomes → Load genomes from file. → Navigate to genome fasta file

  - File →
  - Load from File →
  - Navigate to indexed Bam file.



Single Cell Sequencing

Institute for Computational Genomics
01011011010101
10100100101
RWTH AACHEN UNIVERSITY

Single Cell Analysis

- Extract sequences from a specific cell for the purpose of discovering differences in gene expression level

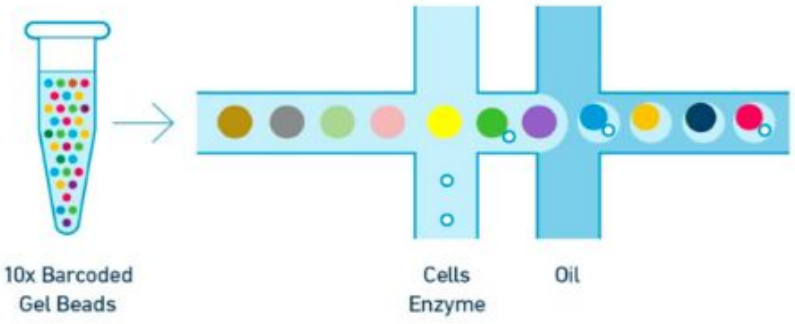
- Every sample is prepared by artificially adding a barcode and (preferably) Unique Molecule Identifier (UMI)

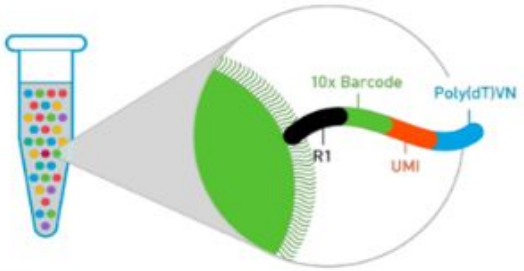
- All molecules from the same batch have the same barcode

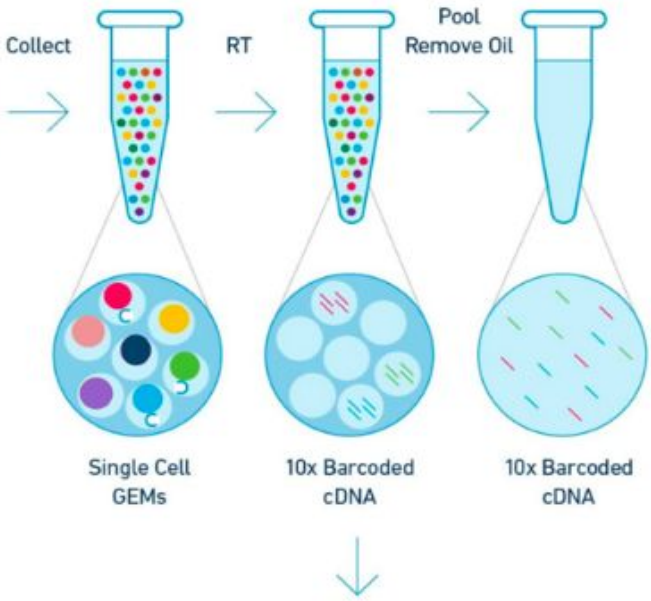
- Every individual molecule has a separate UMI

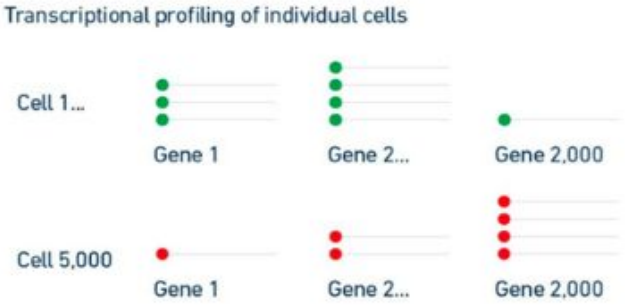
- Because of sequencing errors, we need to make sure that we can correct small amount of bases (1-2) and still have the same barcode – by maximizing the Hamming distance

Single Cell Analysis



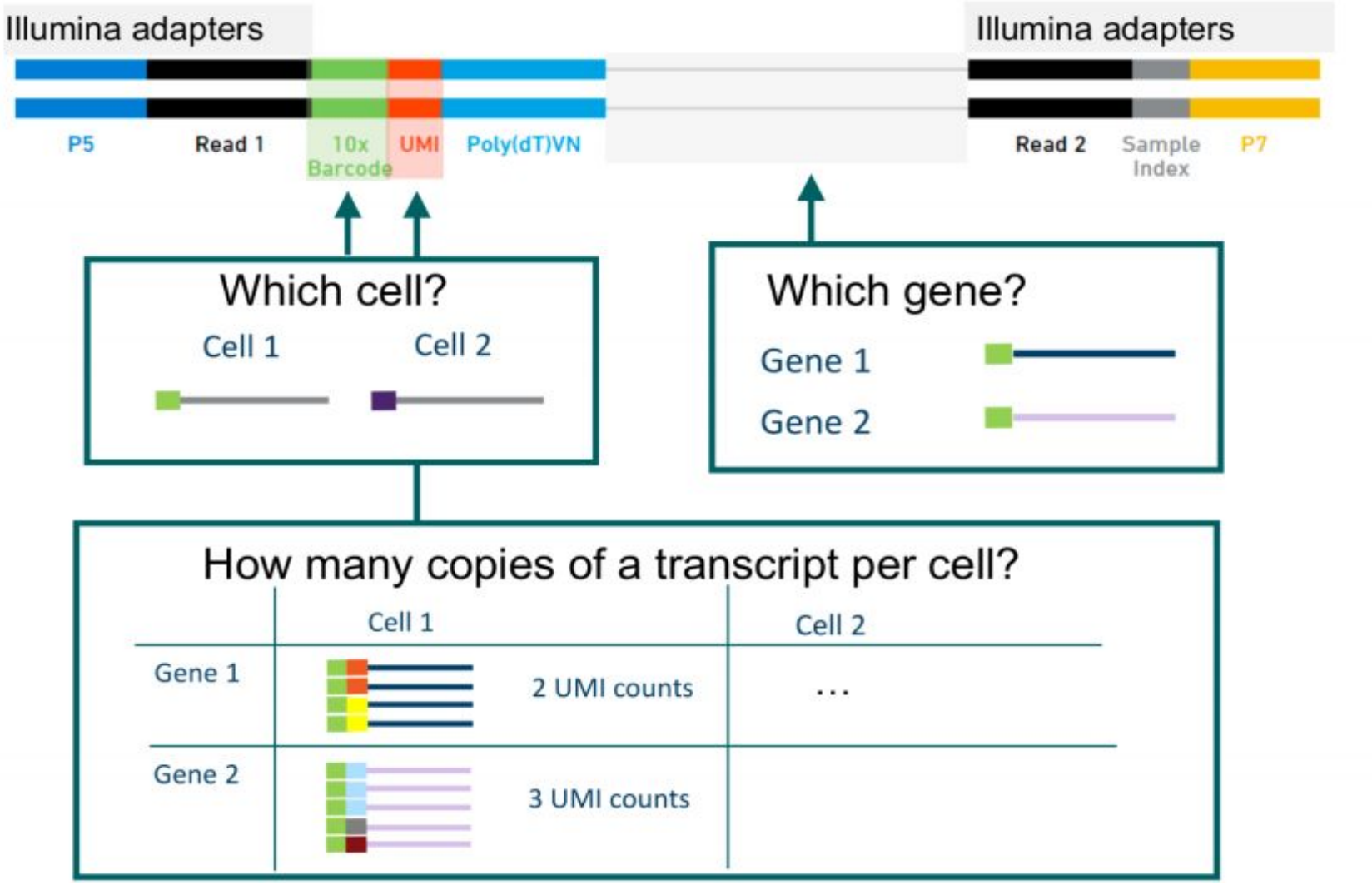






- Input: Single cells in suspension + 10x Gel Beads and Reagents
- Output: Digital gene expression profiles from every partitioned cell

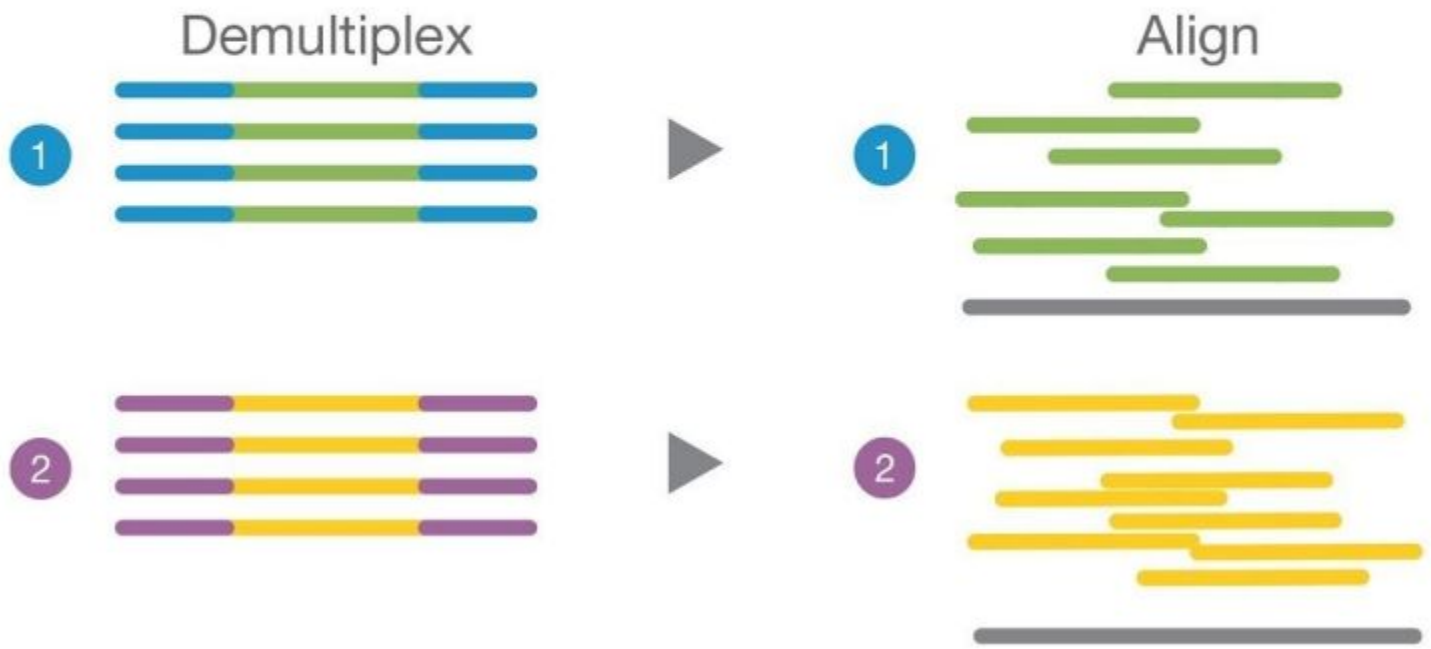
Demultiplexing



Source: 10x genomics

Demultiplexing

Distinguishing different DNA samples based on added barcode



Source: Illumina webinar

Hamming Distance

- A measure of similarity between two strings of equal length

- Measured by the amount substitutions needed to derive the second string from the first

B	I	O	I	N	F	O	R	M	A	T	I	C	S	
B	I	O	I	N	F	O	R	M	A	T	I	K	K	H = 2
F	O	R	M	S	B	I	O	L	O	G	I	E	S	H = 12

Hamming Distance - Example

A	C	T	G	G	G	A	C	G	T	Barcode 1
G	A	C	T	T	A	C	G	G	A	Barcode 2
A	C	T	G	G	G	A	C	G	A	Read 1 – H(1) = 1; H(2) = 9
T	A	T	C	A	G	C	C	G	A	Read 2 – H(1) = 6; H(7) = 6
T	A	C	T	T	G	C	G	G	A	Read 3 – H(1) = 7; H(2) = 2

Designing a set of equidistant barcodes for optimal error correction is NP-complete problem

Demultiplexing

- Demultiplexing both:
  - Barcode
  - UMI (Unique Molecule Identifier)

- Usually UMI is added to read of the paired read.

- This results in one Fastq File per barcode

Demultiplexing - Example

- For simplicity a demultiplexing script is provided as well as sample data - data3.zip.

Use it to extract demultiplexed reads and get familiar with the inputs and outpus.

mkdir data3/results

./demultiplexing.py -b data3/10cells_barcodes.txt -f data3/10cells_read1.fastq -r data3/10cells_read2.fastq -o data3/results/

Expression Matrix

- After performing QC we align the reads and count UMIs for specific barcodes and positions to create an Expression Matrix (mxn).
  n can vary to ~100 to ~ 1M

- Columns represent a cell
- Rows represent a gene (transpose used by some authors)

Seurat

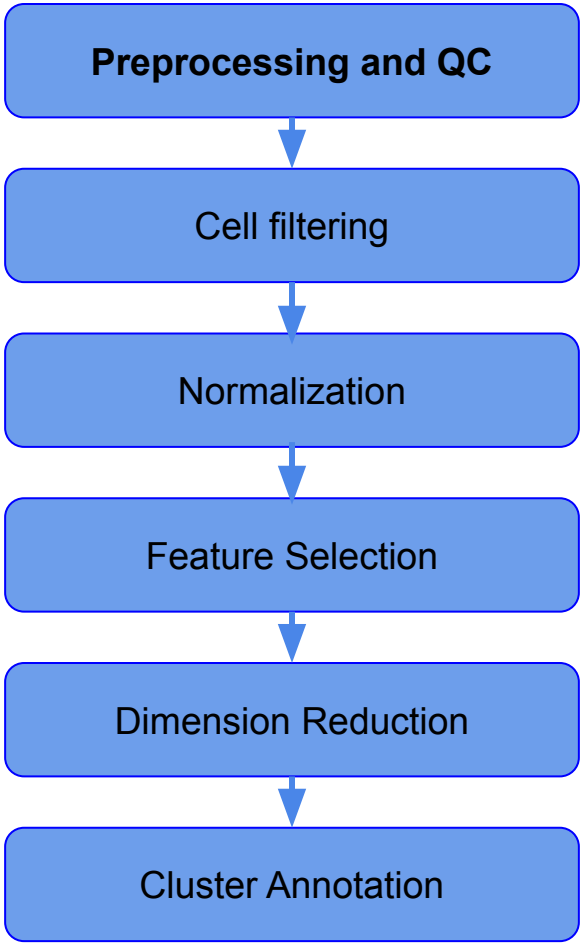
Seurat

- An R package designed for higher level analysis and exploration of single-cell RNA-seq data.

- Current version: 3.1.5

- Allows various functions like PCA and clustering and supports an array of different plotting capabilities.

Seurat-pipeline



Seurat-download data

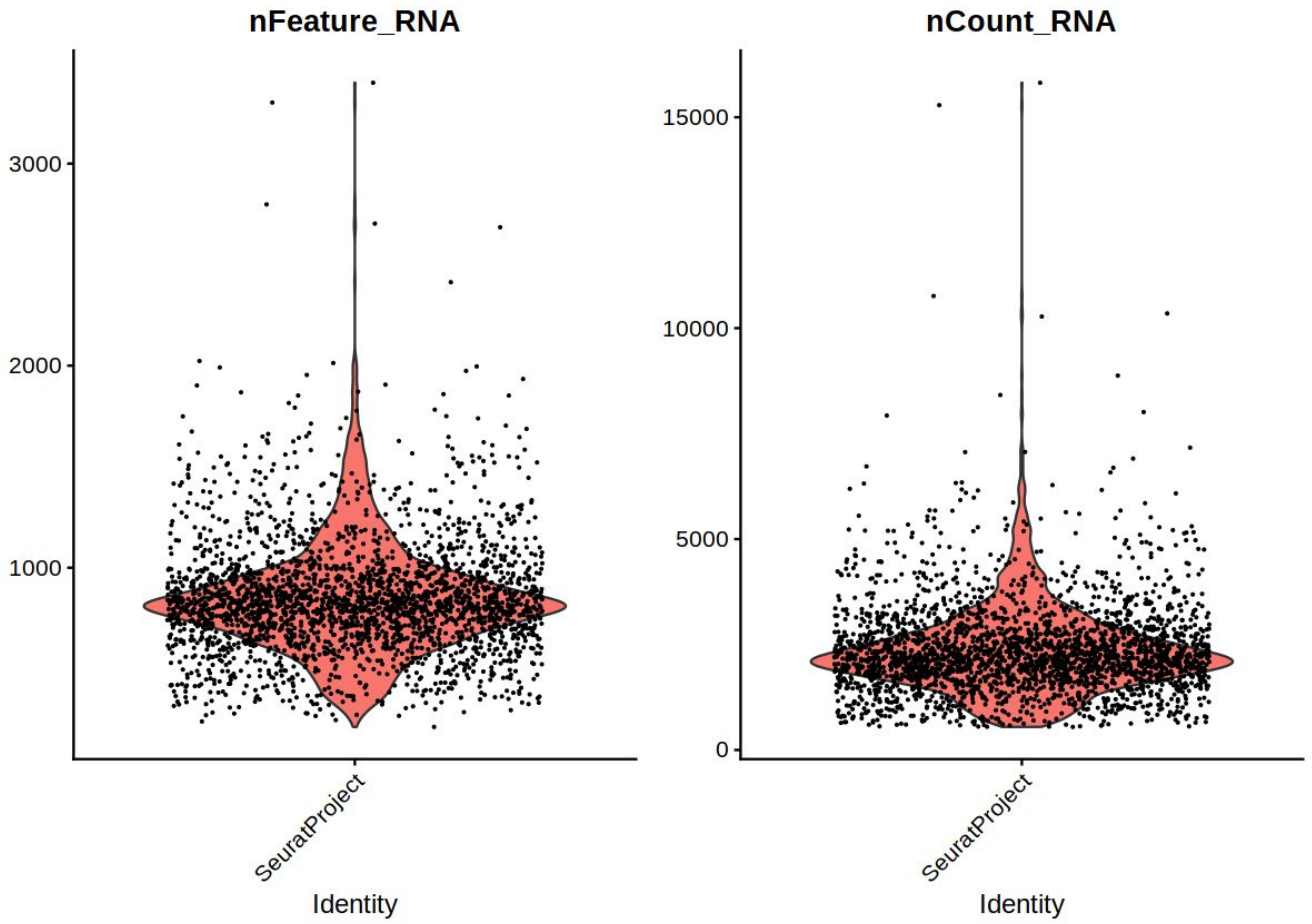
- Download the seurat_data.tar.gz and extract data: tar xzvf seurat_data.tar.gz
- open R (or Rstudio) and load the data in a seurat object.

```
library(Seurat)
library(dplyr)
seuobj.data <- Read10X(data.dir = "filtered_gene_bc_matrices/hg19/")
# create a Seurat object
seuobj <- CreateSeuratObject(
 counts = seuobj.data,
 min.cells = 3,
 min.features = 200
)
```

Seurat-preprocessing

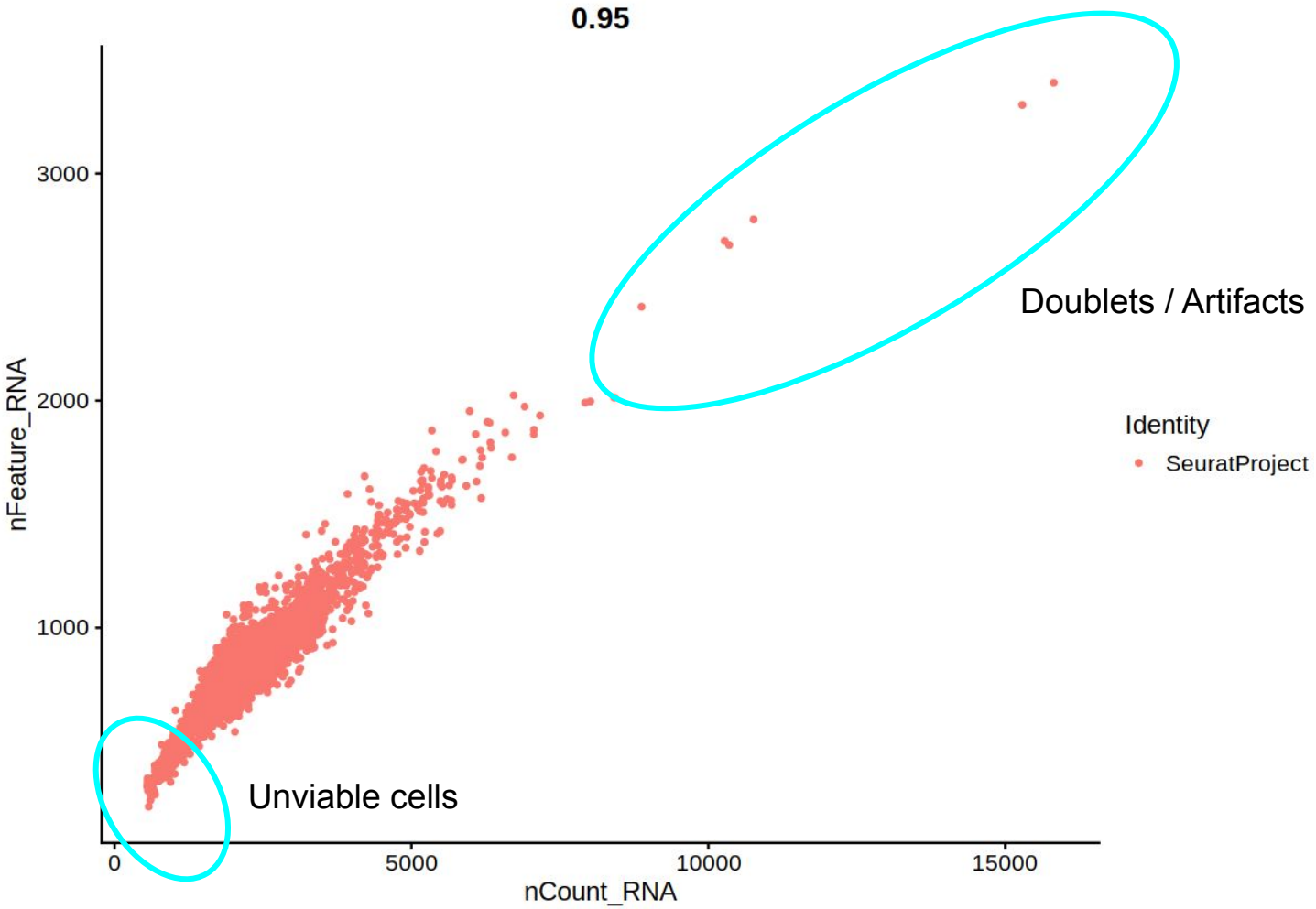
```
## An object of class Seurat
## 13714 features across 2700 samples within 1 assay
## Active assay: RNA (13714 features)
## 2 dimensional reductions calculated: pca, umap
# Plot the expression level
VlnPlot(
 object = seuobj,
 features = c("nFeature_RNA", "nCount_RNA"),
 ncol = 2,
 pt.size = 0.1
)
# Plot the feature correlation
FeatureScatter(
 object = seuobj,
 feature1 = "nCount_RNA",
 feature2 = "nFeature_RNA"
)
```

Seurat-preprocessing

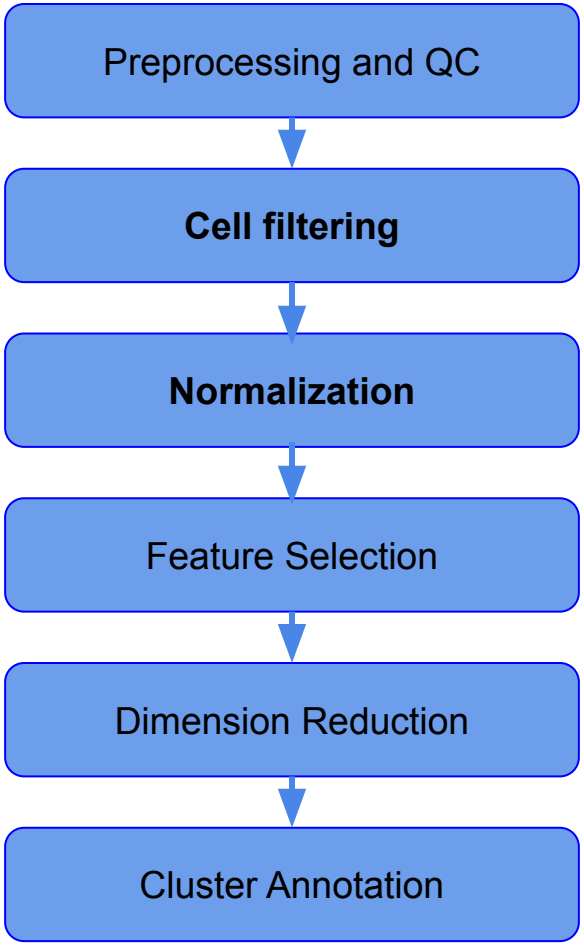


Cells with too high (doublets) or low counts (not viable cells) can be artifacts!

Seurat-preprocessing



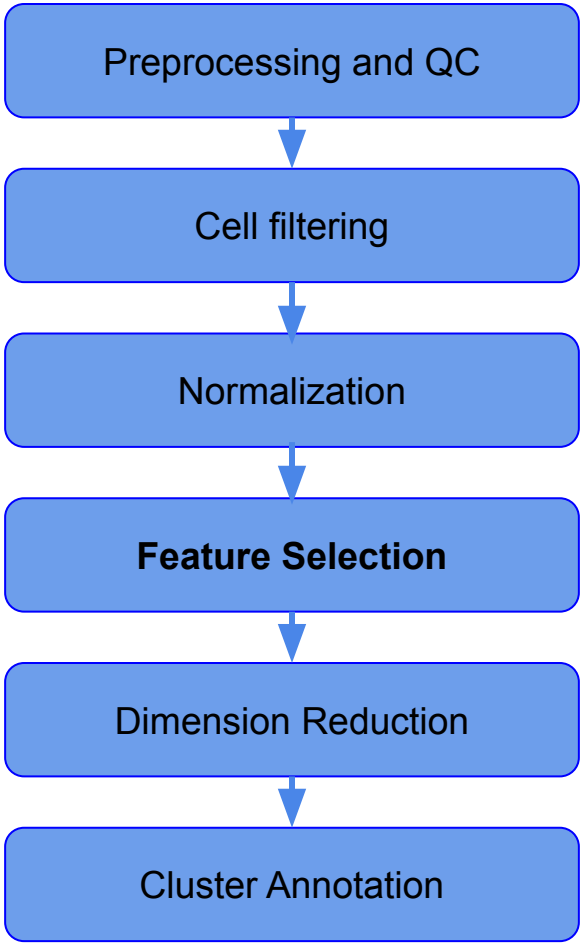
Seurat-pipeline



Seurat-Normalization

```
# Filter cells with outlier number of read counts
seuobj <- subset(
 x = seuobj,
 subset = nFeature_RNA < 2500 & nFeature_RNA > 200
)

# Perform Log-Normalization with scaling factor 10,000
seuobj <- NormalizeData(
 object = seuobj,
 normalization.method = "LogNormalize",
 scale.factor = 10000
)
```

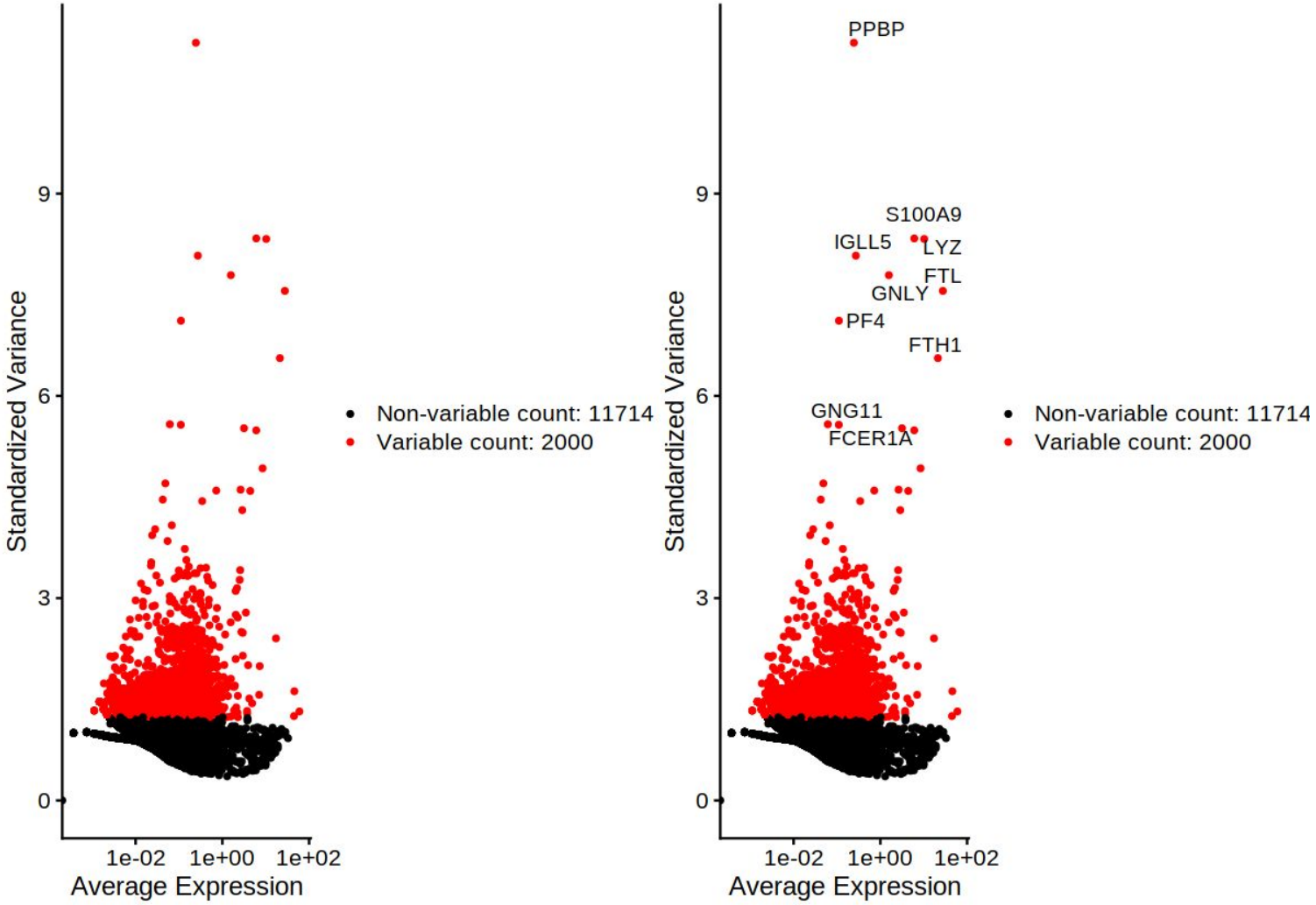
Seurat-pipeline



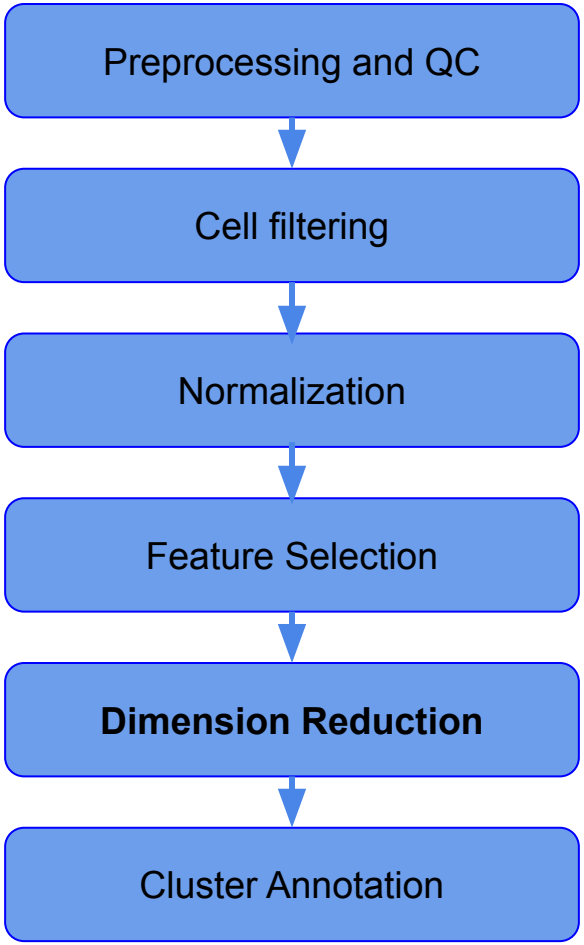
Seurat-Features

```
# Identification of highly variable features
seuobj <- FindVariableFeatures(
 object = seuobj,
 mean.function = ExpMean,
 dispersion.function = LogVMR
)


# Identify the 10 most highly variable genes
top10 <- head(x = VariableFeatures(object = seuobj), 10)
# plot variable features with and without labels
plot1 <- VariableFeaturePlot(object = seuobj)
plot2 <- LabelPoints(plot = plot1, points = top10, repel = TRUE)
CombinePlots(plots = list(plot1, plot2))
```

Seurat-Identifying Highly Variable Features



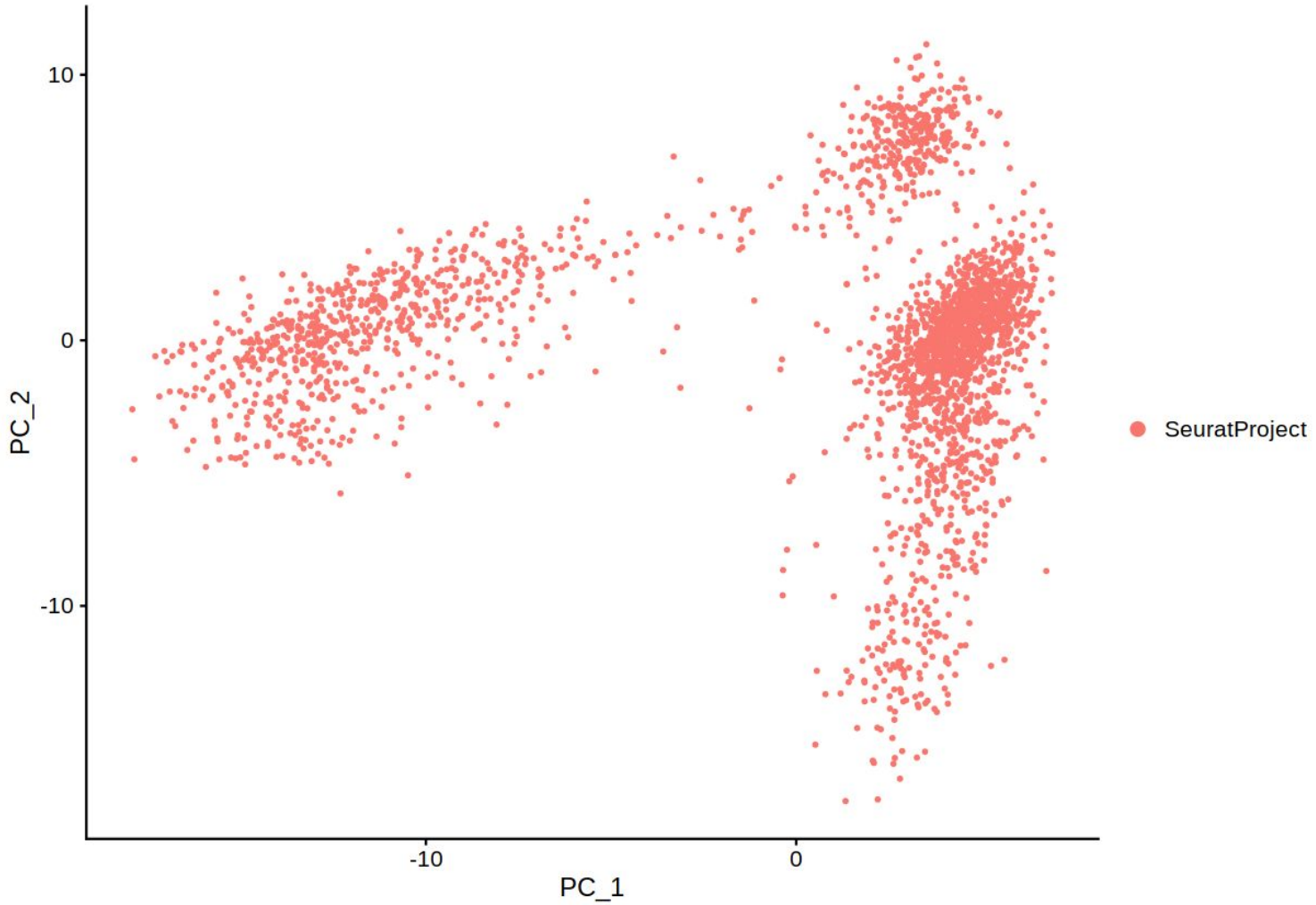
Seurat-pipeline



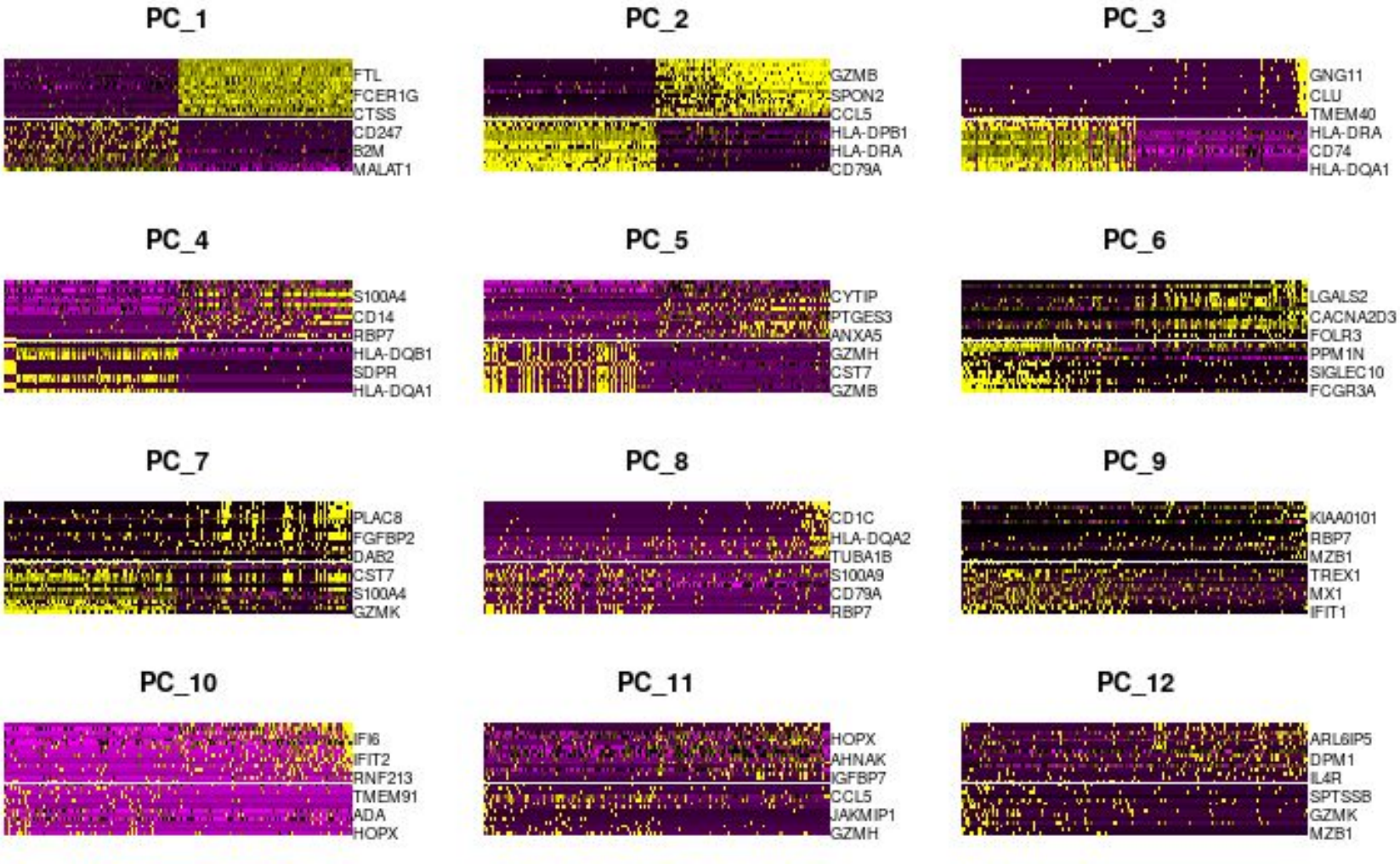
Seurat-Scale and Dimension Reduction

```
# Scale the data
all.genes <- rownames(x = seuobj)
seuobj <- ScaleData(object = seuobj, features = all.genes)

# Perform linear dimensional reduction
seuobj <- RunPCA(object = seuobj,
            features = VariableFeatures(object = seuobj))

# Visualize PCA
DimPlot(object = seuobj, reduction = "pca")
DimHeatmap(object = seuobj, dims = 1:12, cells = 500, balanced = TRUE)
ElbowPlot(object = seuobj)
```

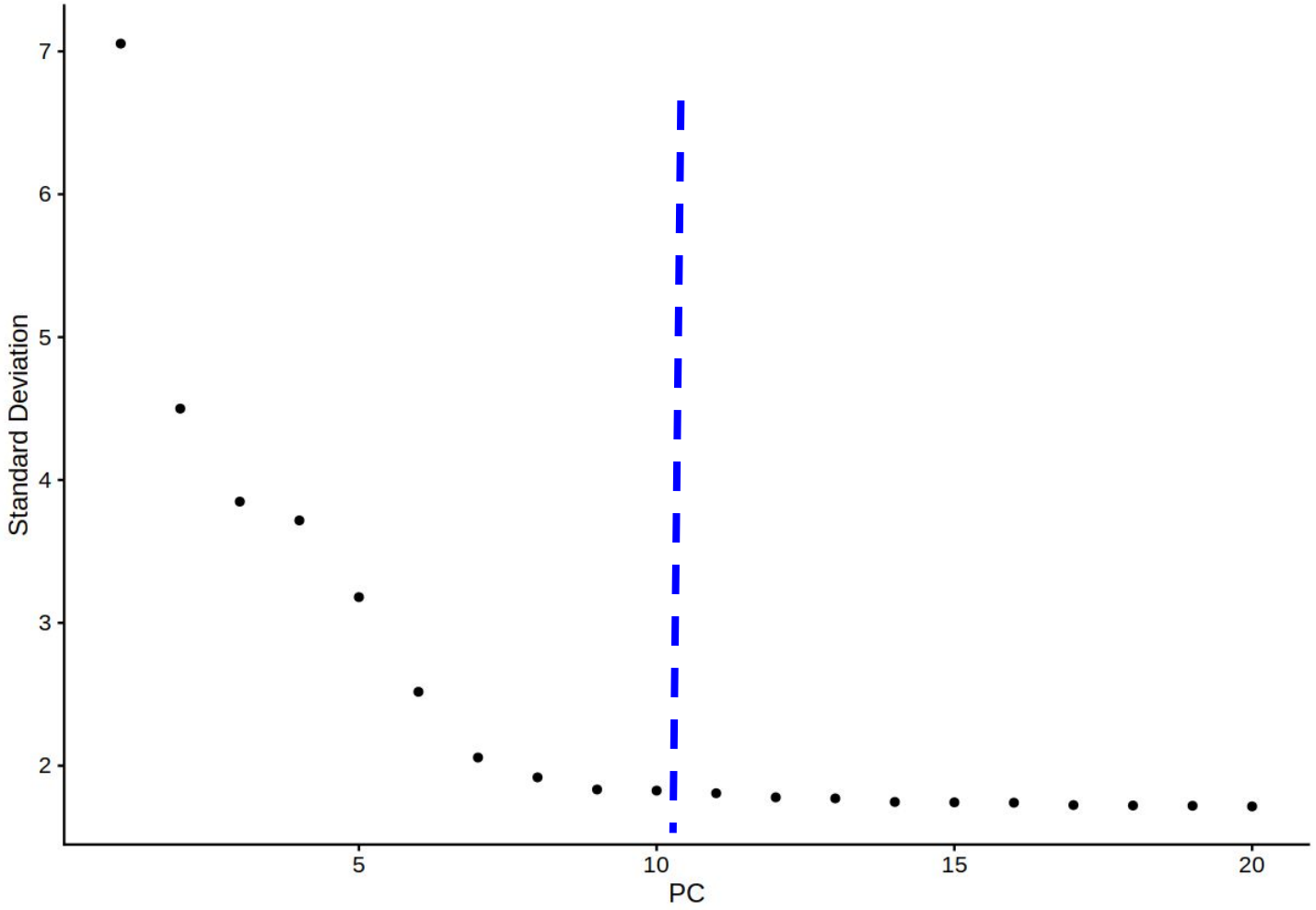
Seurat-Scale and Dimension Reduction



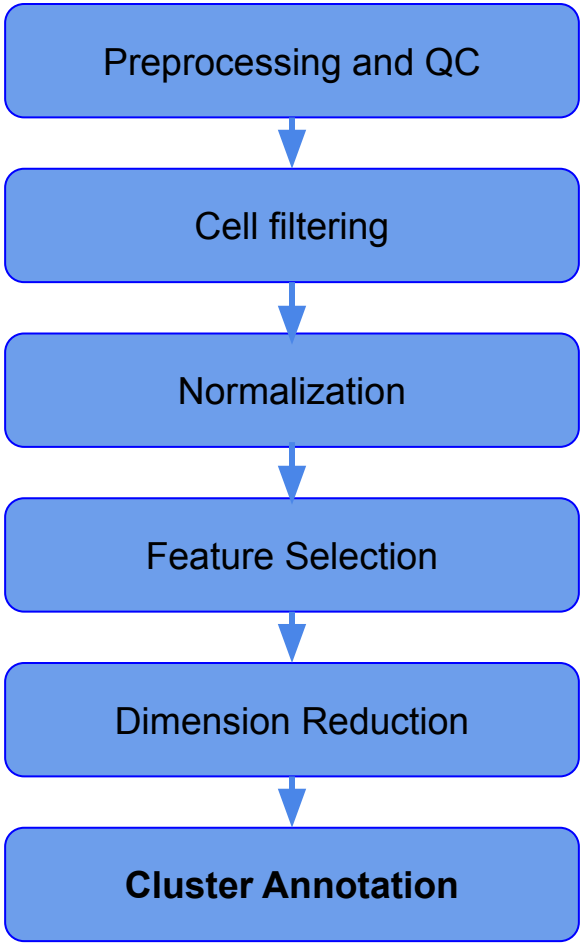
Seurat-Scale and Dimension Reduction



Seurat-Scale and Dimension Reduction



Seurat-pipeline

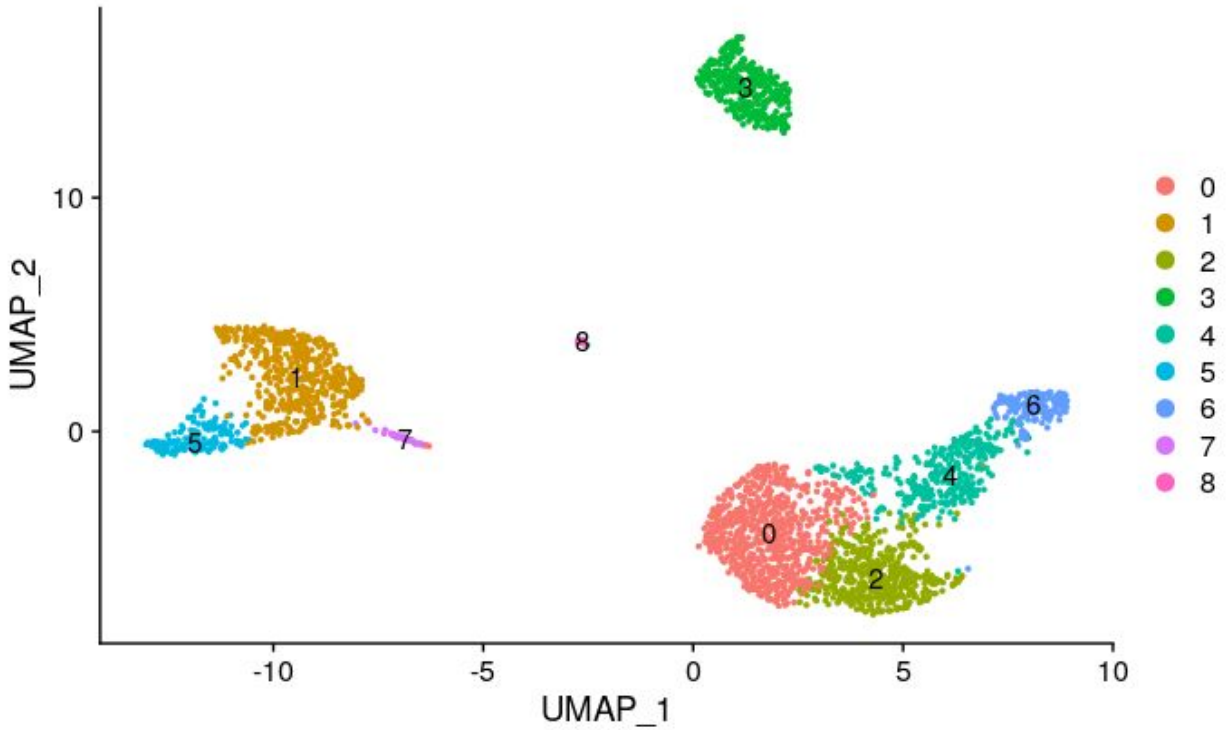


Seurat-Cluster cells

```
# Clustering Cells
seuobj <- FindNeighbors(object = seuobj, dims = 1:10)
seuobj <- FindClusters(object = seuobj, resolution = 0.5)
```

Seurat-Cluster cells

```
# Run UMAP dimension reduction
seuobj <- RunUMAP(
    object = seuobj,
    dims = 1:10
)
DimPlot(object=seuobj, label=T, group.by="seurat_clusters")
```

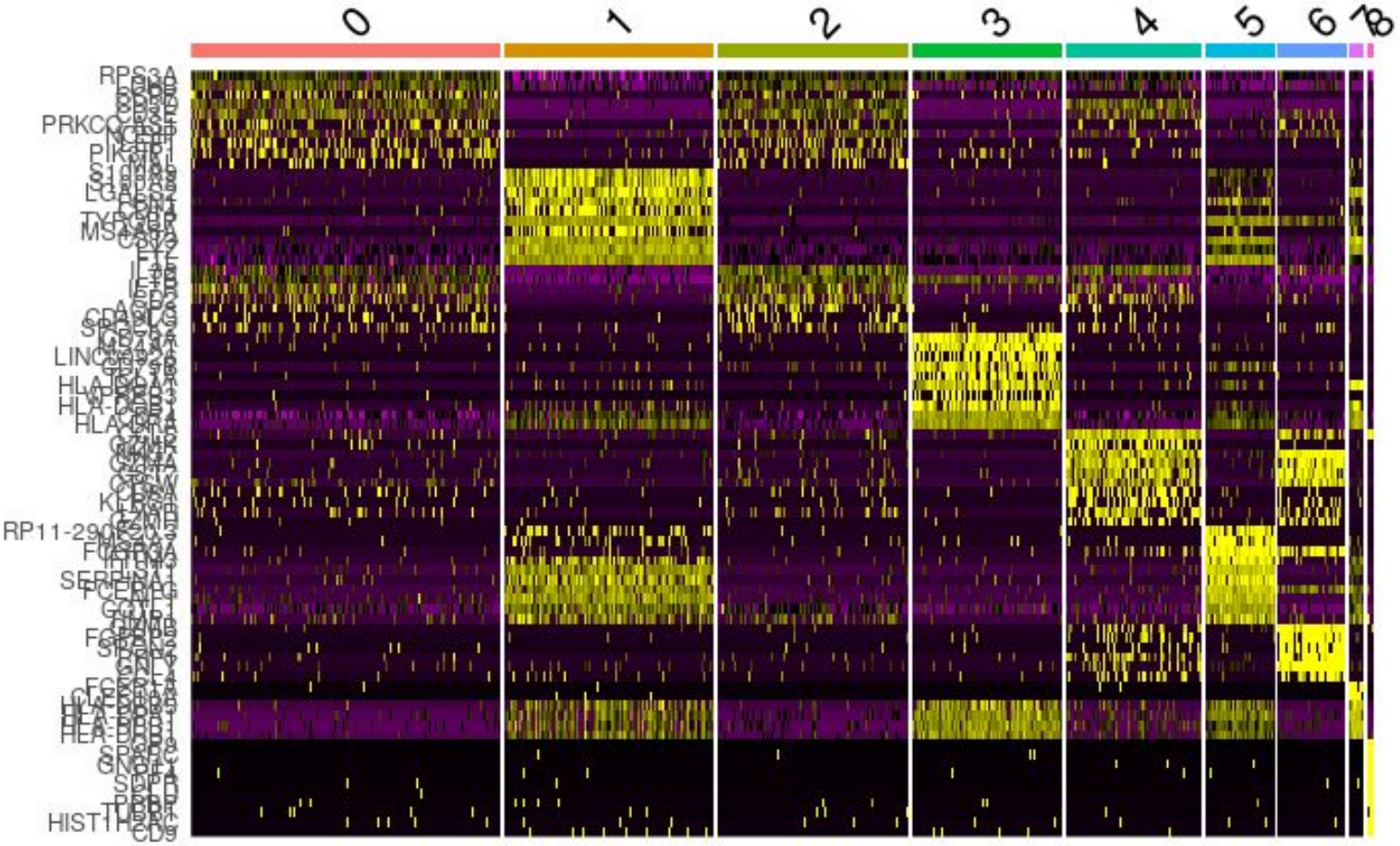


Seurat- Identify markers for cells

```
# Find markers for specific clusters
cluster1.markers <- FindMarkers(object = seuobj,
                    ident.1 = 0, min.pct = 0.25)

# Display first 10 markers found for cluster 1
head(x = cluster1.markers, n = 10)

# Find best markers for each cluster in the dataset
seuobj.markers <- FindAllMarkers(object = seuobj, only.pos = TRUE,
                min.pct = 0.25, logfc.threshold = 0.25)
```

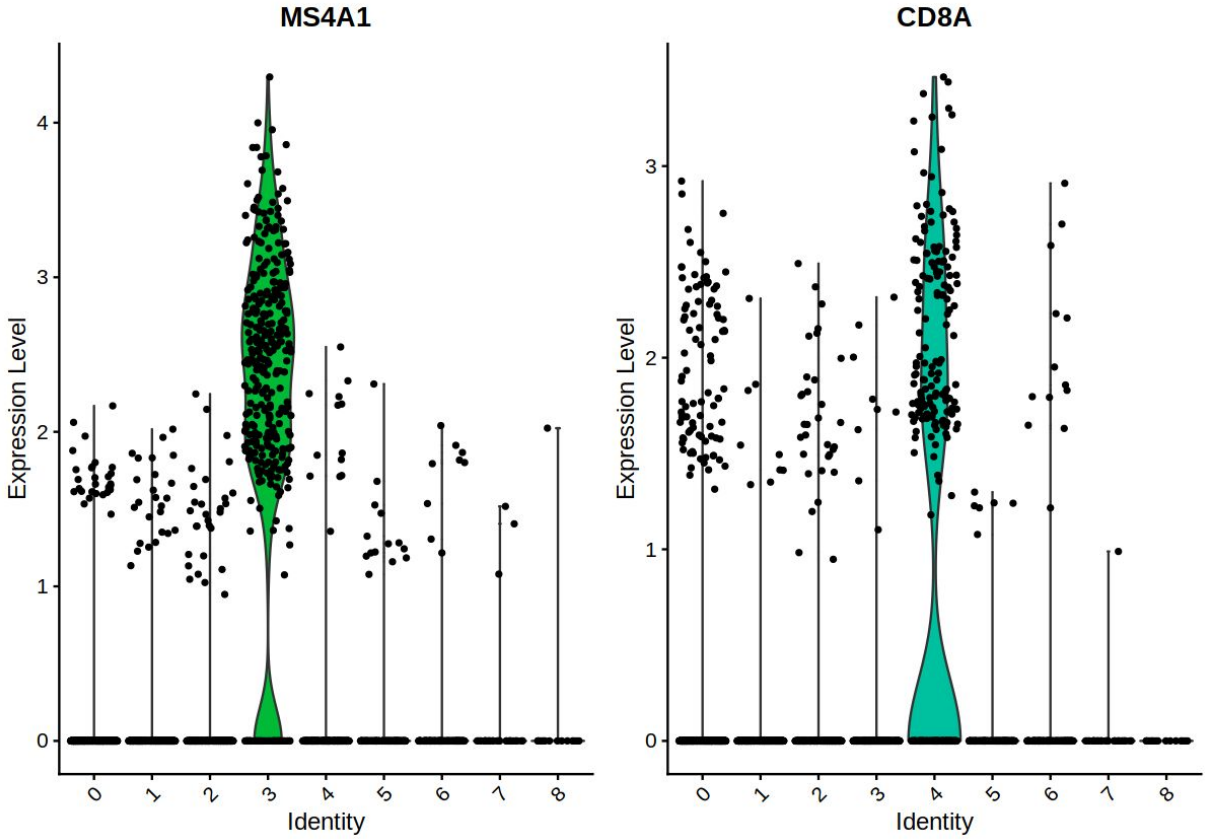
Seurat- Unbiased cluster identification

```
# Identifty top 10 markers for all genes and plot a heatmap
top10 <- seuobj.markers %>% group_by(cluster) %>% top_n(n = 10, wt = avg_logFC)
DoHeatmap(object = seuobj, features = top10$gene) + NoLegend()
```



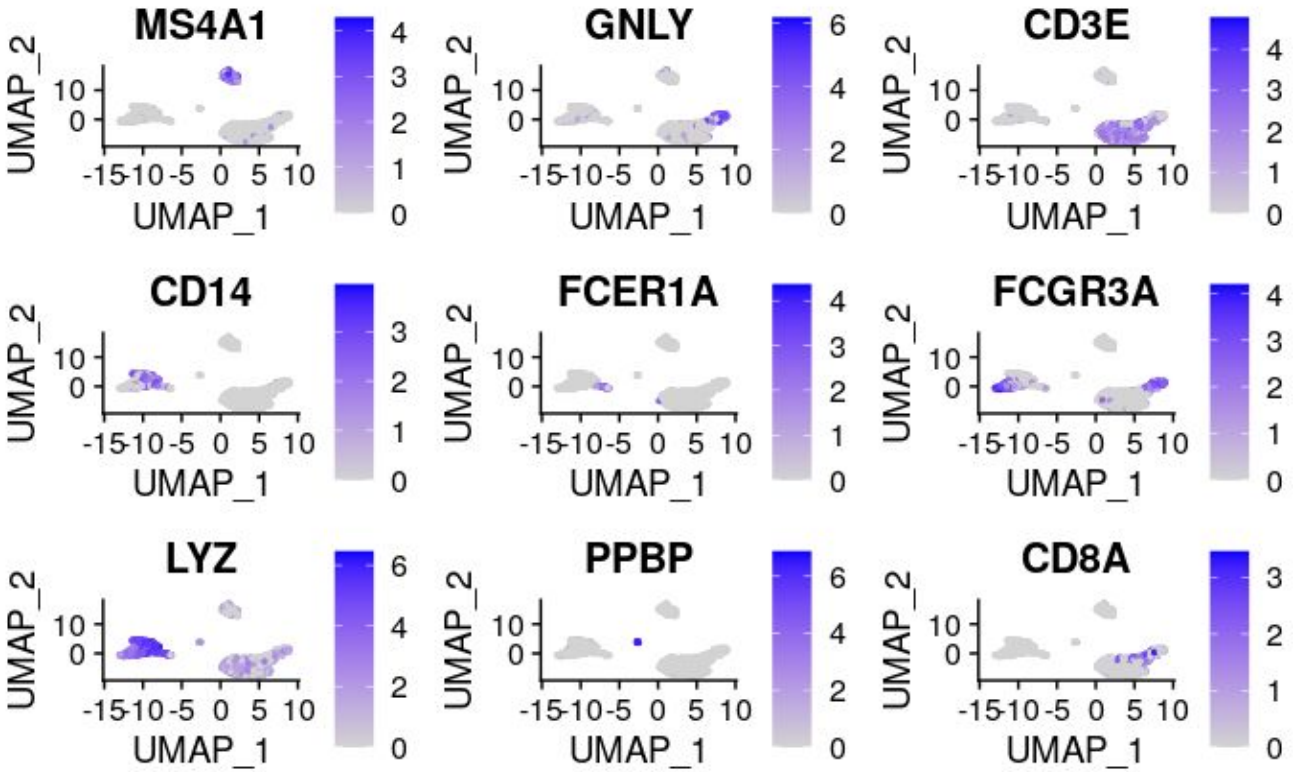
Seurat- Expert-based cluster annotation

```
# Using a known marker plot cluster responses
VlnPlot(object = seuboj, features = c("MS4A1", "CD8A"))
```



Seurat- Expert-based cluster annotation

```
# Show known markers in UMAP plot
FeaturePlot(object = seuobj, features = c("MS4A1", "GNLY", "CD3E",
                "CD14","FCER1A", "FCGR3A", "LYZ", "PPBP", "CD8A"))
```



Seurat- Expert-based cluster annotation

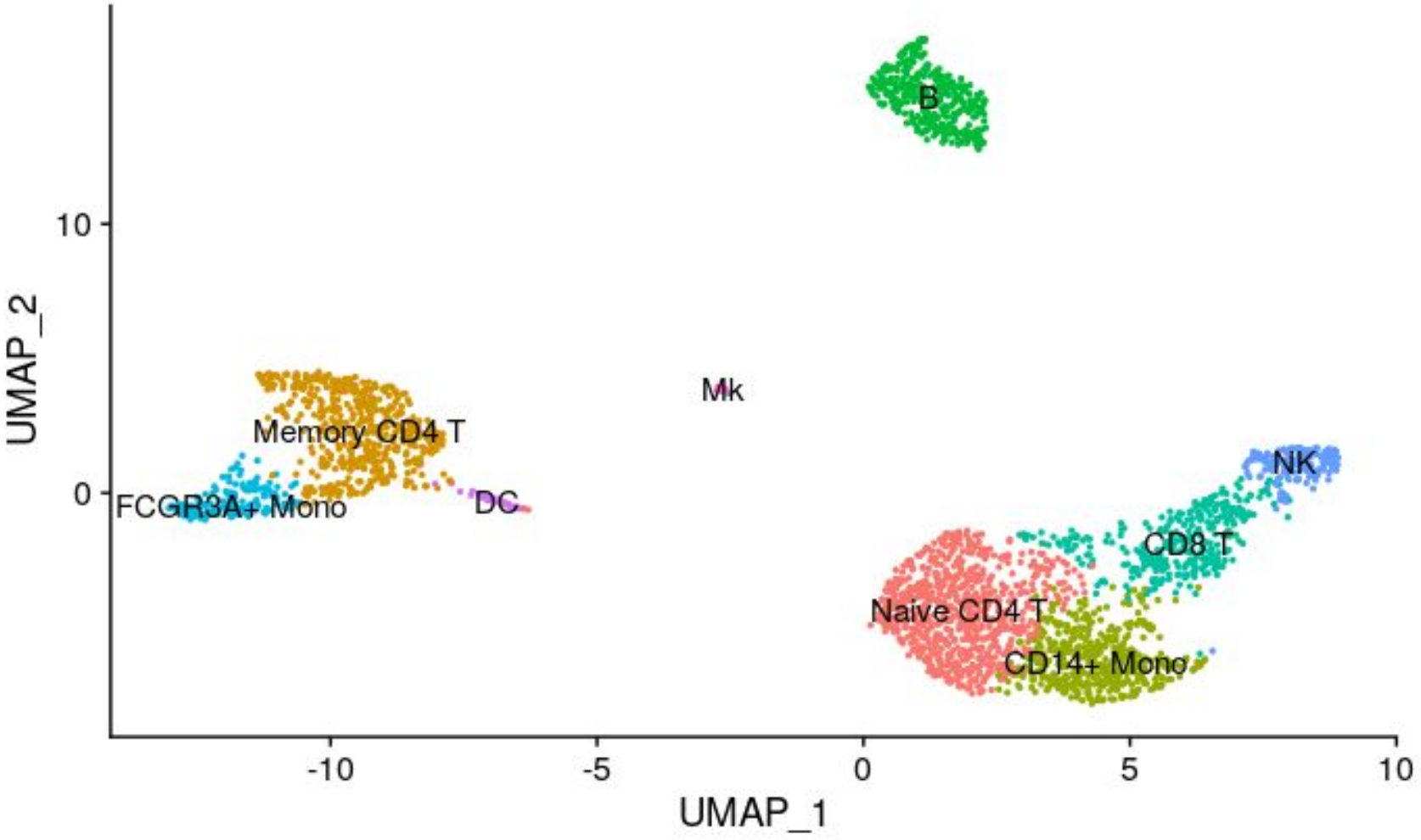
```
# Using a known marker identify clusters
```

Markers	Cell Type	Identified Cluster
IL7R, CCR7	Naive CD4+ T	0
IL7R, S100A4	Memory CD4+	1
CD14, LYZ	CD14+ Mono	2
MS4A1	B	3
CD8A	CD8+ T	4
FCGR3A, MS4A7	FCGR3A+ Mono	5
GNLY, NKG7	NK	6
FCER1A, CST3	DC	7
PPBP	Mk	8

Seurat- Expert-based cluster annotation

```
# Plot UMAP with new cluster IDs
new.cluster.ids <- c("Naive CD4 T", "Memory CD4 T",
            "CD14+ Mono", "B","CD8 T", "FCGR3A+ Mono",
             "NK", "DC", "Mk")

names(x = new.cluster.ids) <- levels(x = seuobj)
seuobj <- RenameIdents(object = seuboj, new.cluster.ids)
DimPlot(object = seuobj, reduction = "umap",
    label = TRUE, pt.size = 0.5) + NoLegend()
```

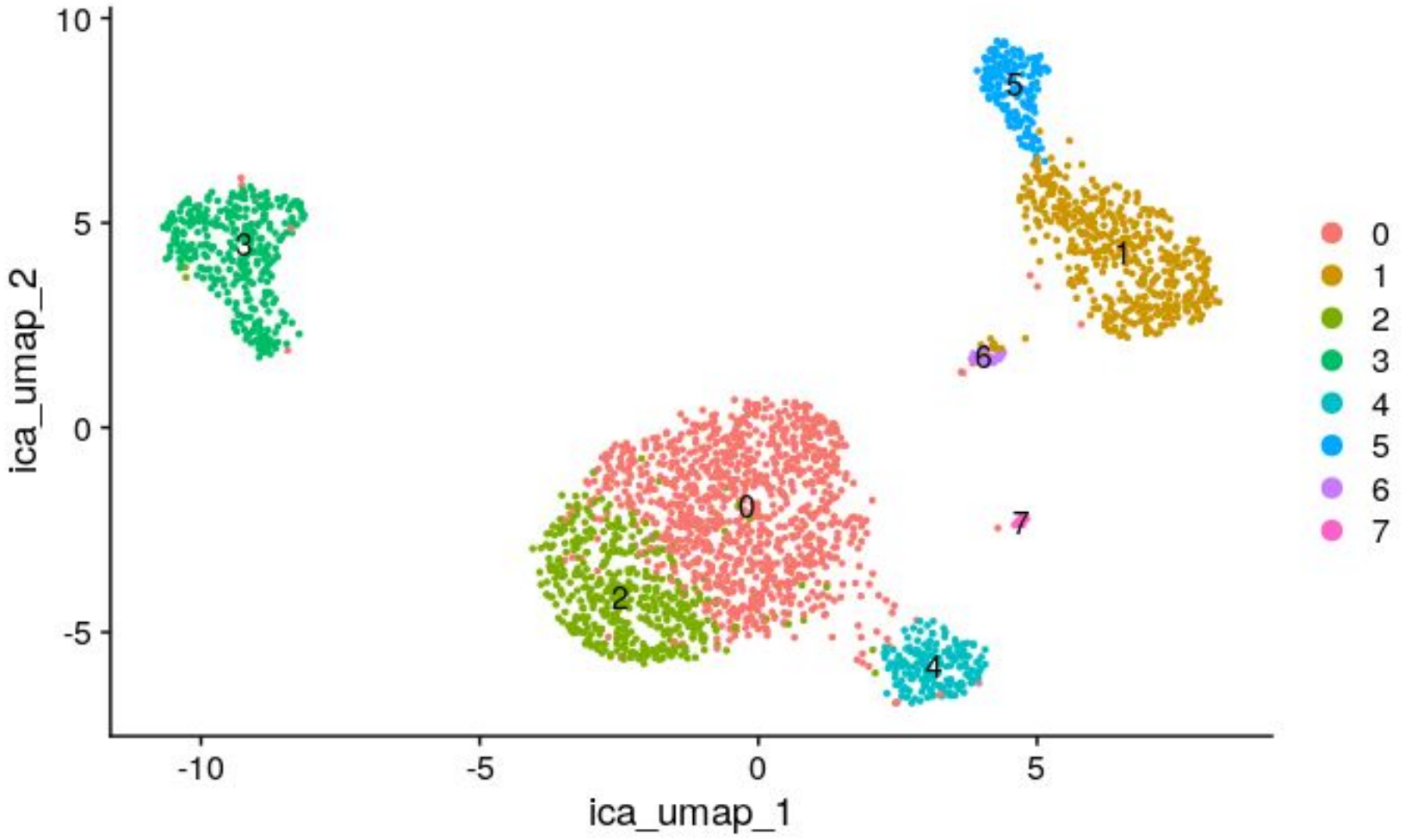
Seurat- Expert-based cluster annotation



Seurat- More reductions-ICA

```
seuobj <- RunICA(object = seuobj, features =
                  VariableFeatures(object = seuobj))
seuobj <- FindNeighbors(object = seuobj, dims = 1:10,
                 reduction="ica")

seuobj <- FindClusters(object = seuobj, resolution = 0.5)

seuobj <- RunUMAP(
  object = seuobj,
  dims = 1:10,
  reduction="ica",
  reduction.name = "ica_umap"
)

DimPlot(seuobj, reduction = "ica_umap",
                    group.by="seurat_clusters", label=T)
```

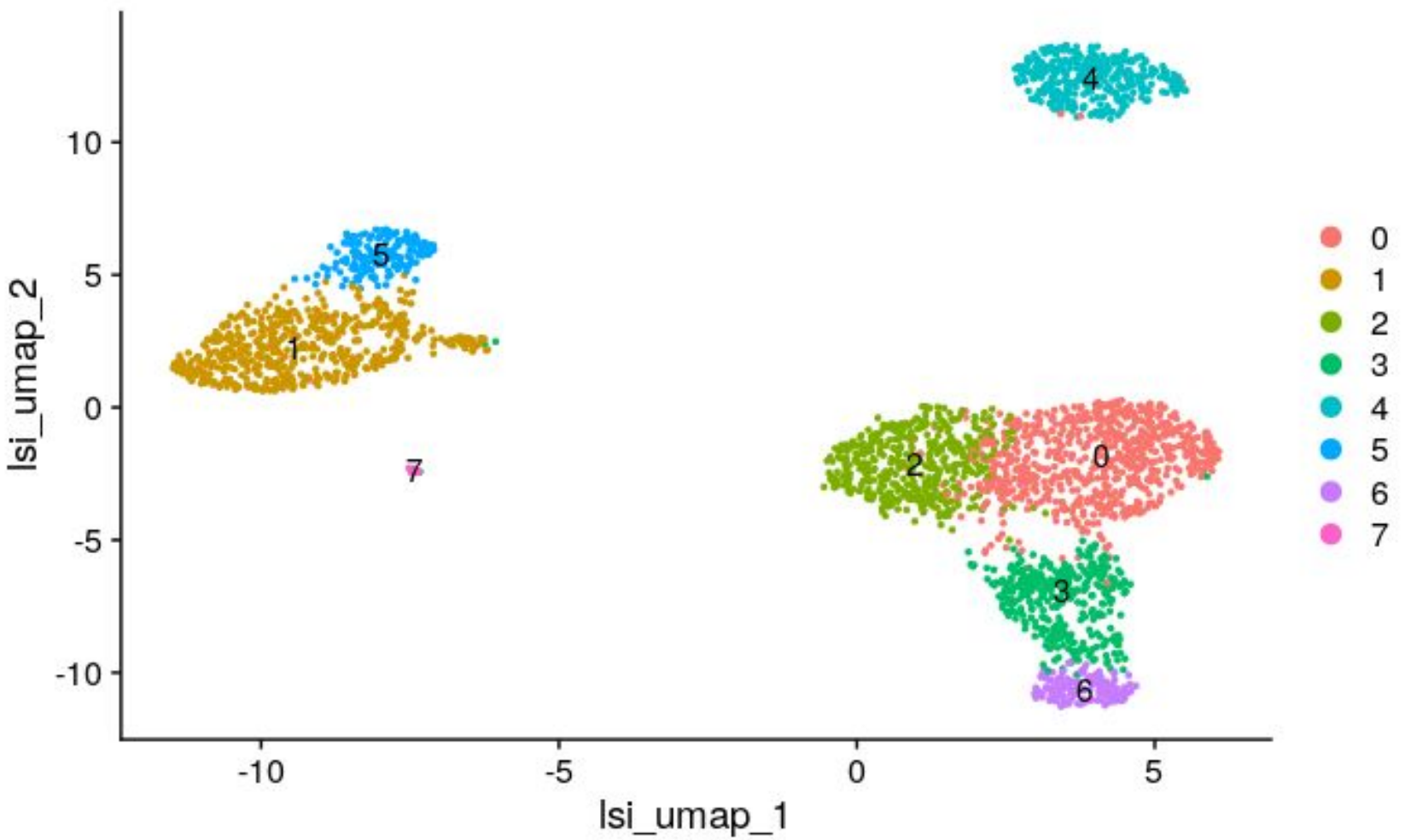
Seurat- More reductions-ICA



Seurat- More reductions-LSI

```
seuobj <- RunLSI(object = seuobj, features =
                  VariableFeatures(object = seuobj))
seuobj <- FindNeighbors(object = seuobj, dims = 1:10,
                 reduction="lsi")

seuobj <- FindClusters(object = seuobj, resolution = 0.5)

seuobj <- RunUMAP(
  object = seuobj,
  dims = 1:10,
  reduction="lsi",
  reduction.name = "lsi_umap"
)

DimPlot(seuobj, reduction = "lsi_umap",
                    group.by="seurat_clusters", label=T)
```

Seurat- More reductions-LSI



Alternative in Python - scanpy

An scanpy example:

https://costalab.ukaachen.de/open_data/SOSE_2020/scanpy.html

Thanks for your attention