Practical Example: NGS – data handling and single cell differentiation

Ivan G. Costa & Martin Manolov

Institute for Computational Genomics

Joint Research Centre for Computational Biomedicine

RWTH Aachen University, Germany

Contact Information

Martin Manolov

martin.manolov@rwth-aachen.de

Room 3.03

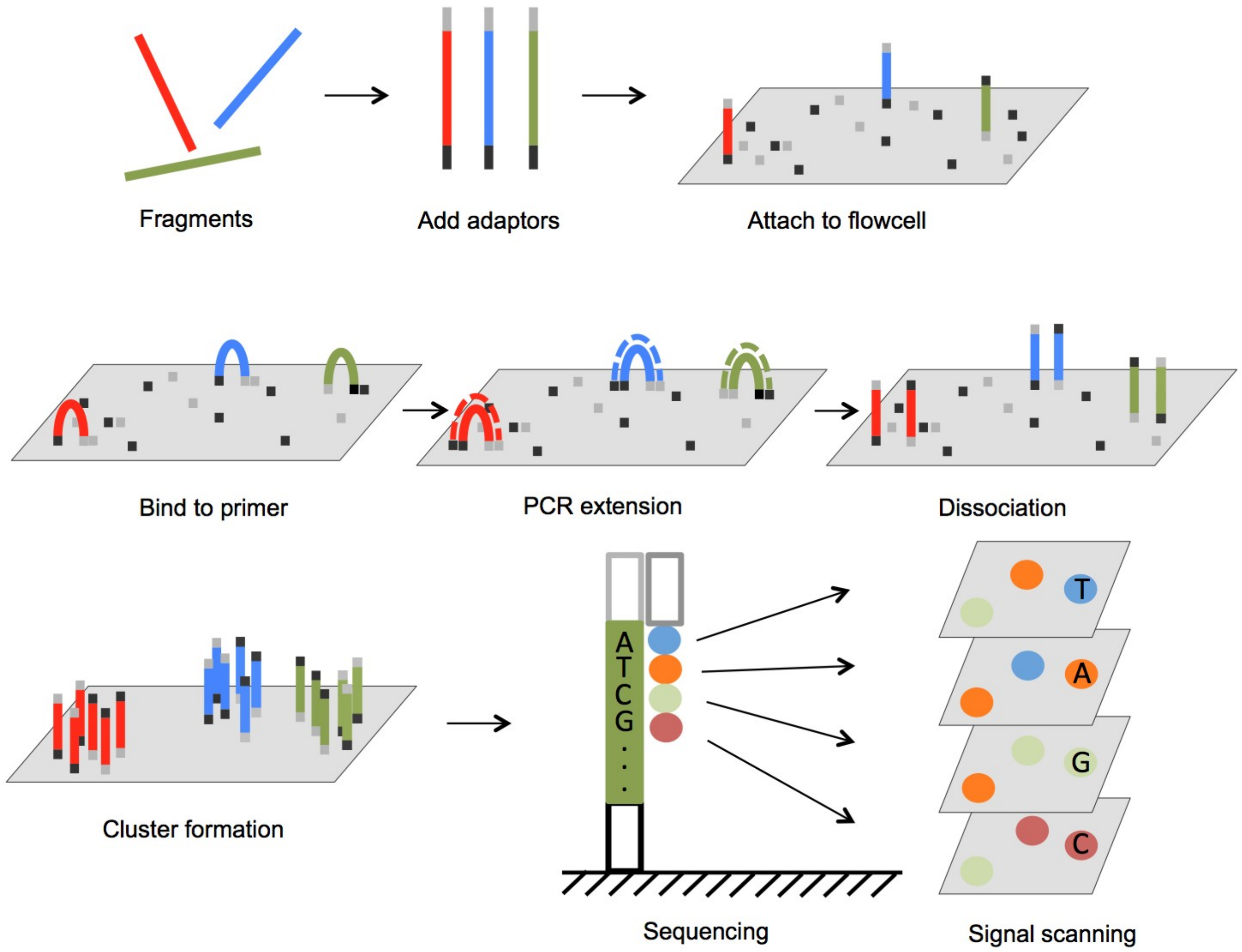
From Sequencing to Alignment

DNA Sequencing

- The problem of converting a DNA molecule to a string [sequence] of bases (C, A, G, T).

- Many possible sequencing techniques exist:
  - Illumina
  - PacBio
  - Nanopore

Illumina



Lu, Yuan, et al. "Next Generation Sequencing in Aquatic Models." 2016.

FASTA File

- Stores DNA sequences 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

```
>SEQ_1
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
>SEQ_2
AGCAGTTGGGGTTCATCGAATTTGGGGTTCATCCATTAAAGCAGAATCCATTTGATCAAT
```

FASTQ File

- 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.

```
@SEQ_ID
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
+
!''*((((***+))%%%++)(%%%%).1***-+*''))**55CCF>>>>>>CCCCCCC65
```

FASTQ Evaluation – FastQC

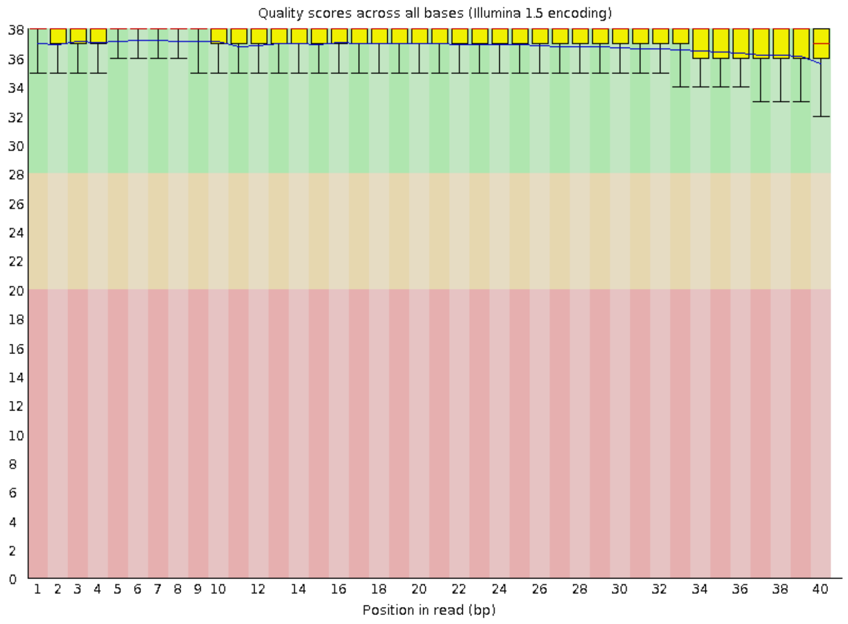
- Fastq files can be very big with millions of (long) reads. Infeasible to investigate.
- Phred-Score hard to read in ASCII form.

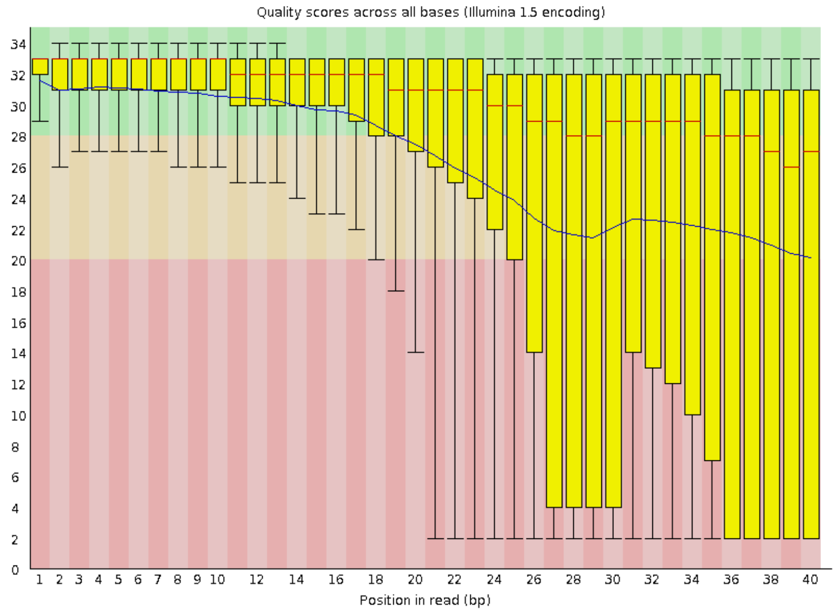
- FastQC (usually provided by NGS core facilities)
  - Tool to analyse quality of reads from sequencing.
  - Indicate problems in library preparation or sequencing steps.

- 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.

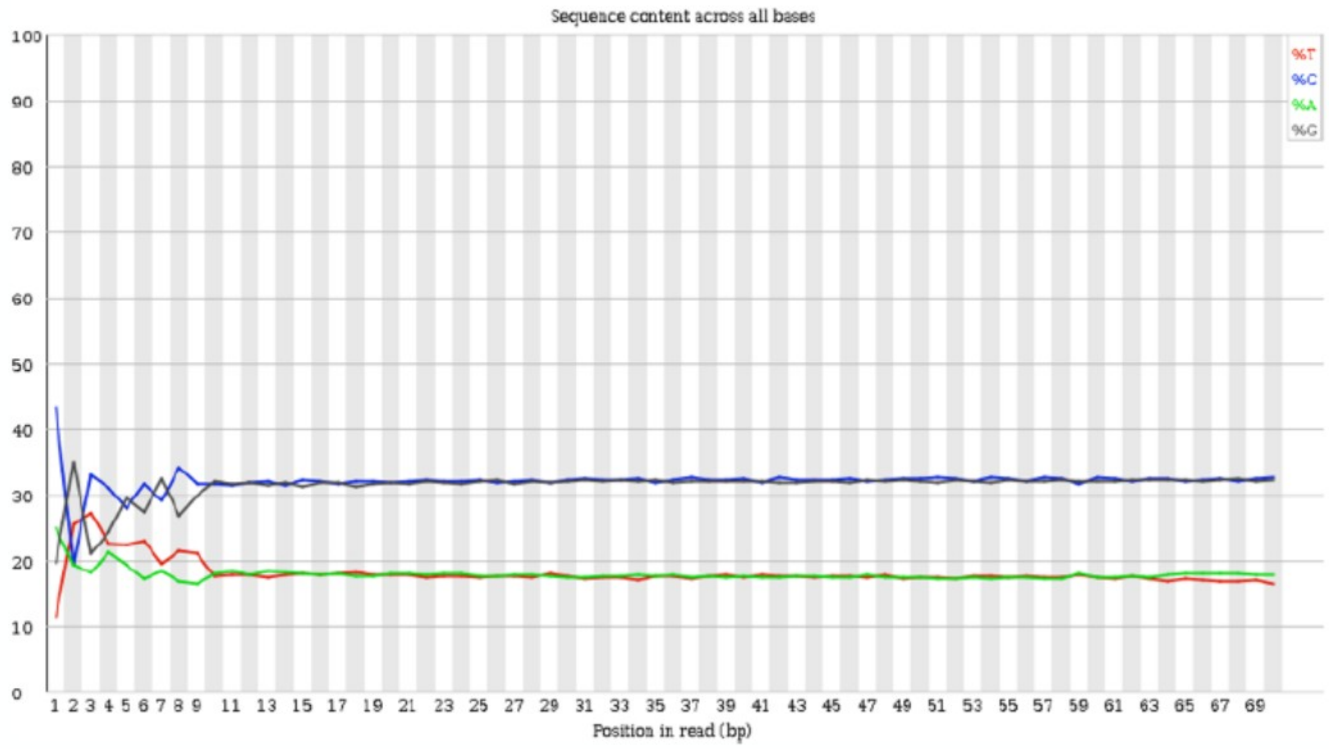




Solution: trim ends of reads, if quality is low.

FASTQ Evaluation – FastQC

Read position sequence bias.



Solution: Trim starts of reads.

Exercise Time

- Download data1.zip from the lecture website.

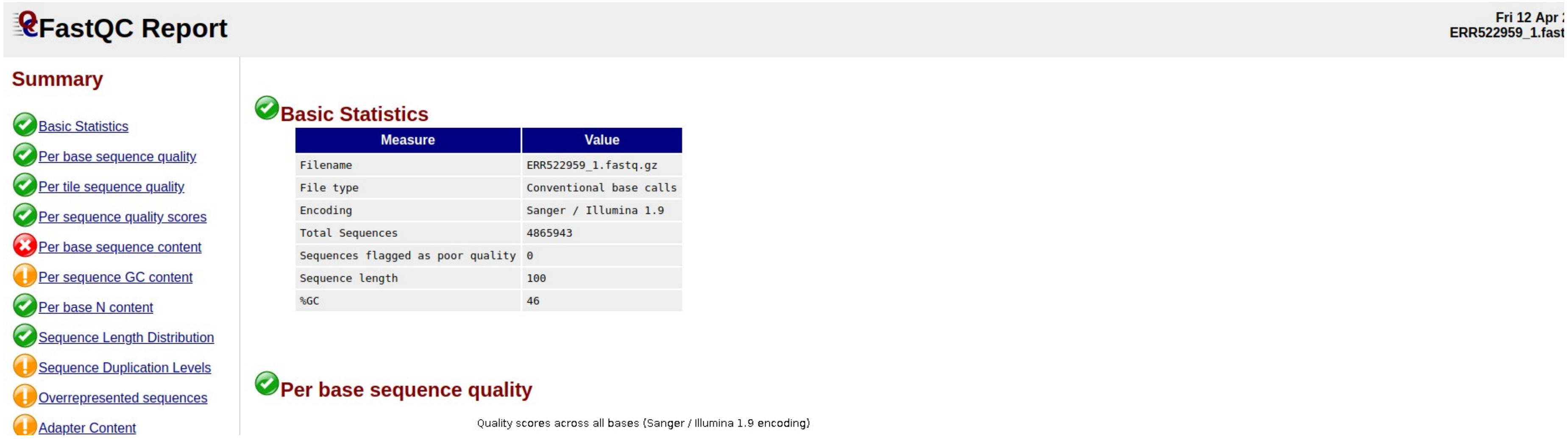
- Use FastQC to analyze the data:
  - create new directory “**fastqc_results**”
  - read the documentation of FastQC to understand how to export the files to the new directory:
    - *fastqc -h*

- 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?

Exercise Time

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



FastQC Report

Fri 12 Apr
ERR522959_1.fast

Summary

- ✓ Basic Statistics
- ✓ Per base sequence quality
- ✓ Per tile sequence quality
- ✓ Per sequence quality scores
- ✗ Per base sequence content
- ! Per sequence GC content
- ✓ Per base N content
- ✓ Sequence Length Distribution
- ! Sequence Duplication Levels
- ! Overrepresented sequences
- ! Adapter Content

Basic Statistics

Measure	Value
Filename	ERR522959_1.fastq.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	4865943
Sequences flagged as poor quality	0
Sequence length	100
%GC	46

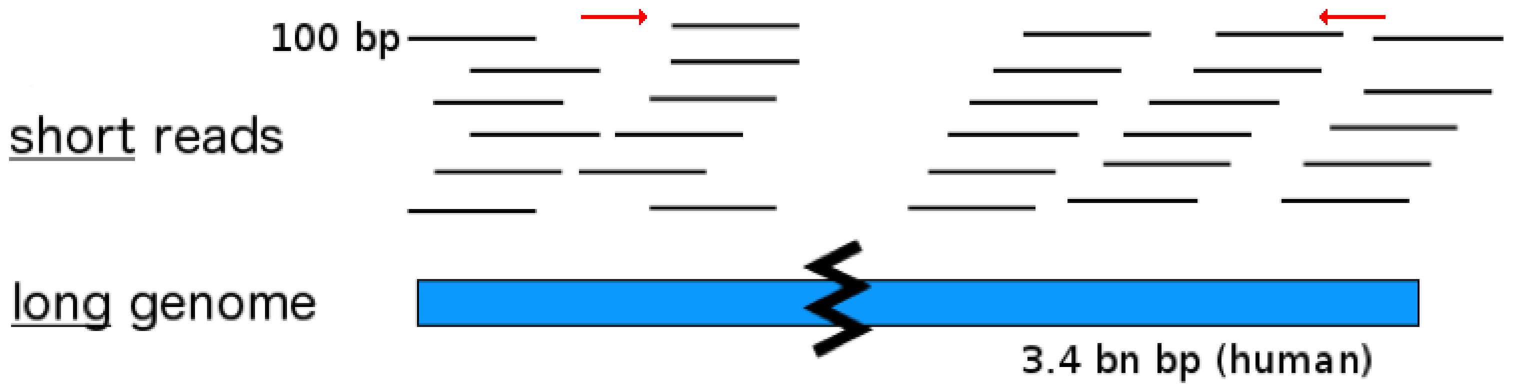
Per base sequence quality

Quality scores across all bases (Sanger / Illumina 1.9 encoding)

- *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*

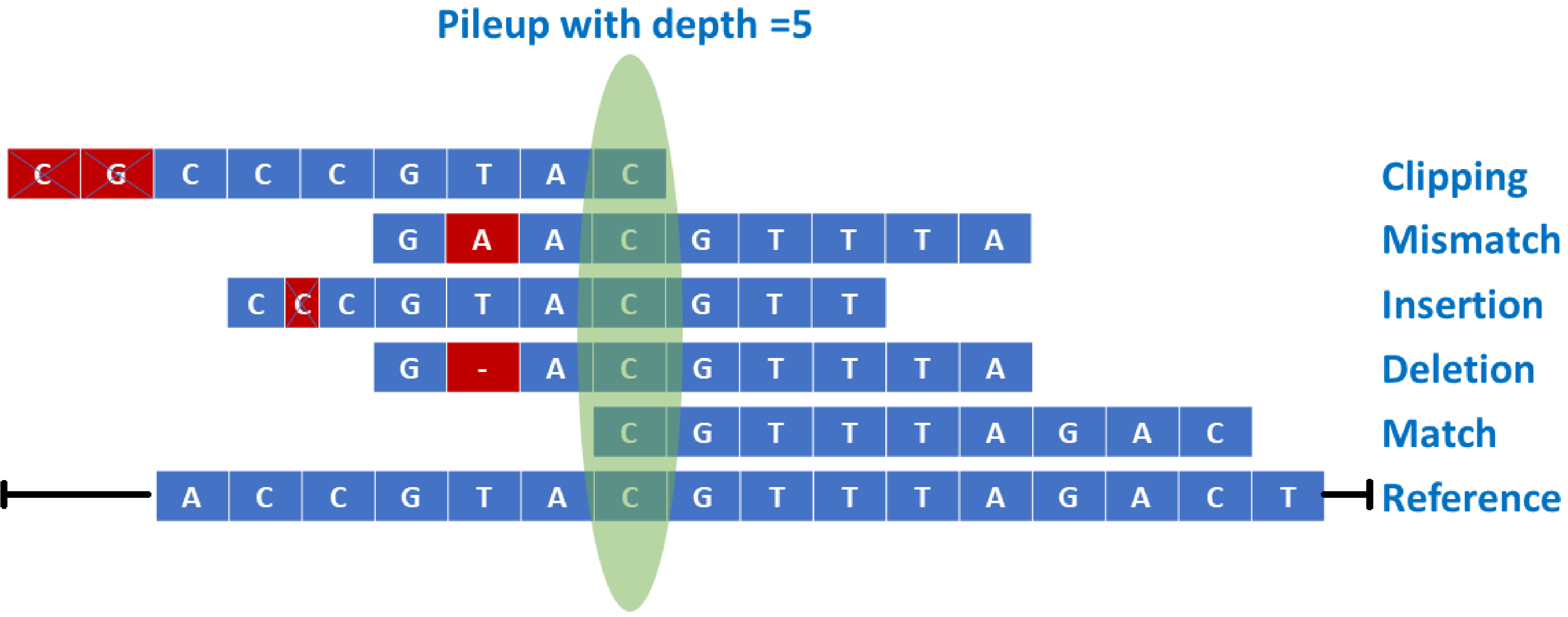
Alignment

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



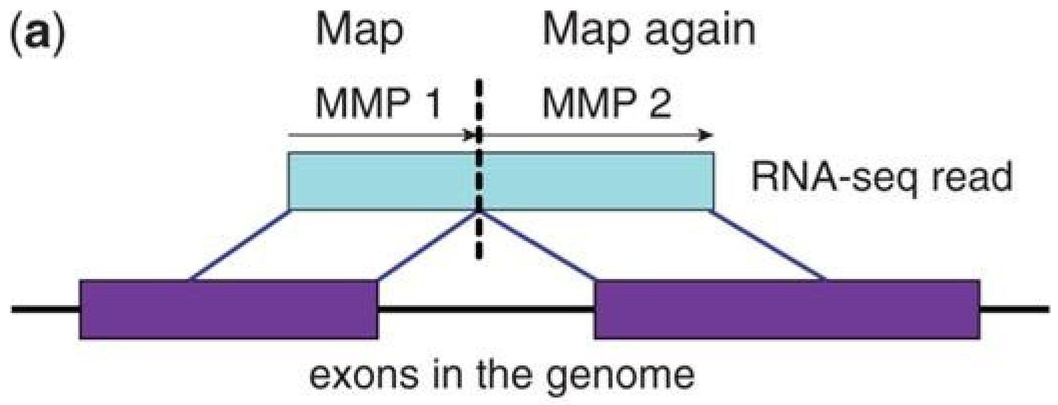
Alignment

- The problem of aligning DNA sequence to a reference genome.



STAR: Universal RNA-seq aligner.

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

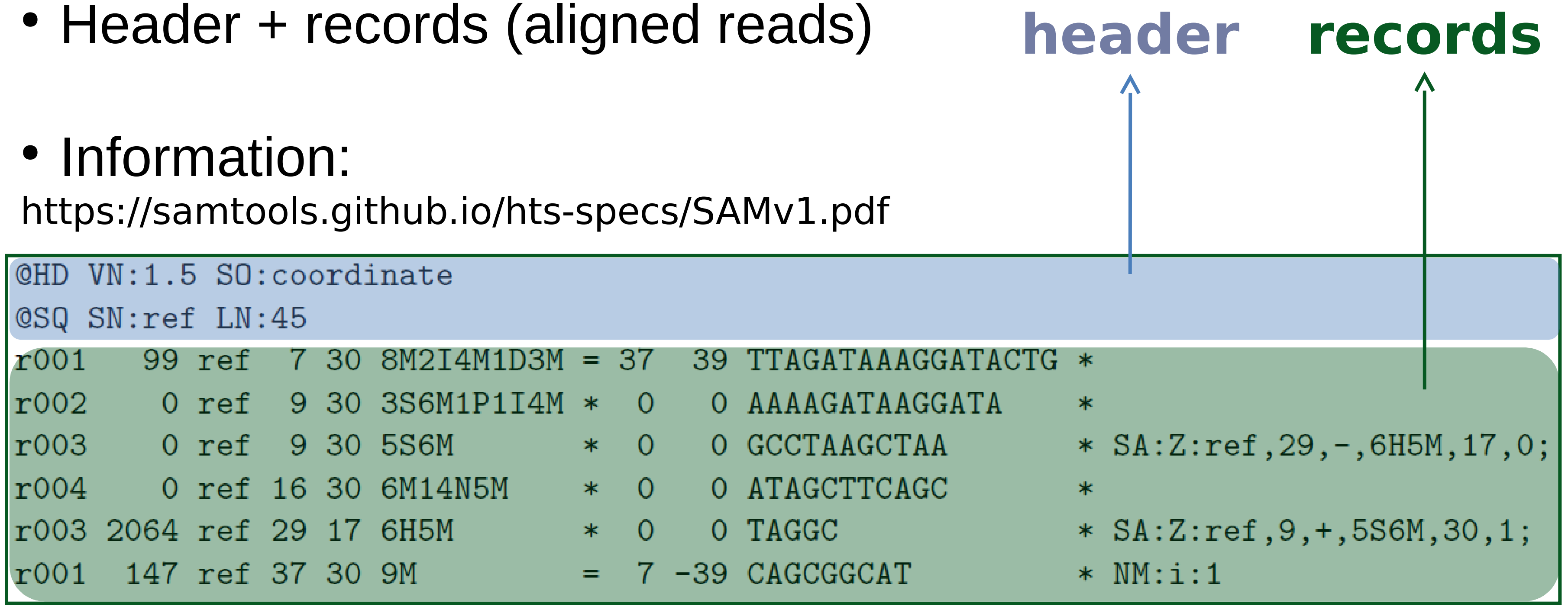


Source: Dobin et al. (2013), Bioinformatics.

SAM File

- Sequence Alignment/Map format.
- Text-based tab-delimited file.
- Header + records (aligned reads)
- Information:

https://samtools.github.io/hts-specs/SAMv1.pdf



```
@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
```

SAM Fields

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/

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/

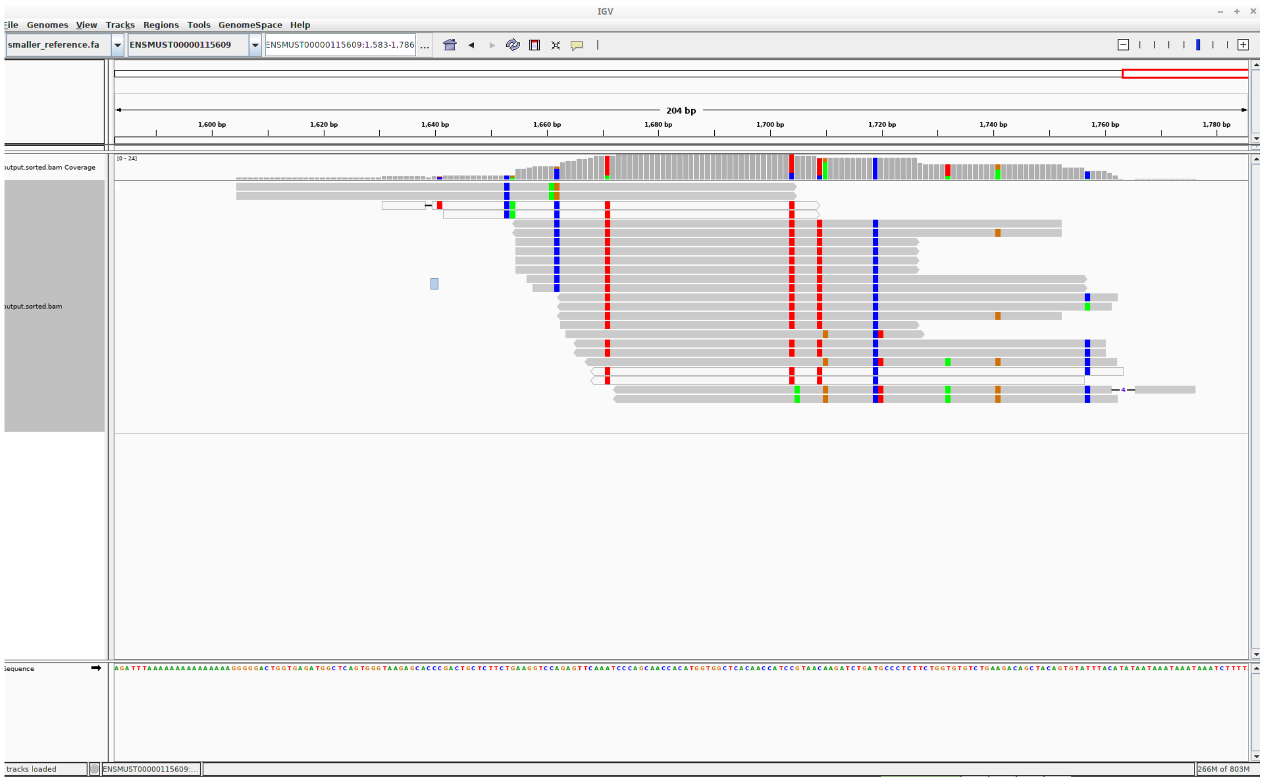
Exercise Time

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

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



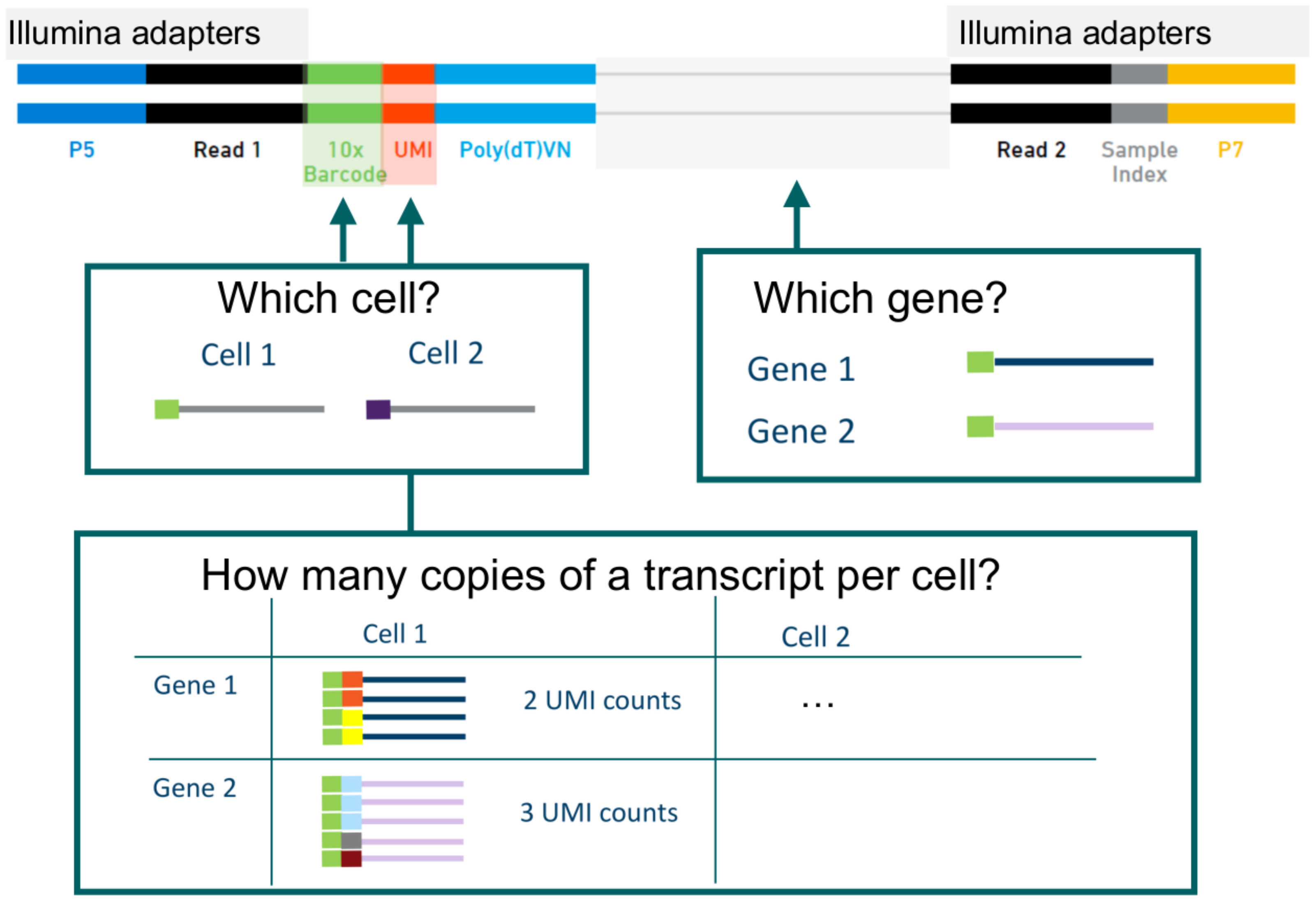
Single Cell Analysis

- 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

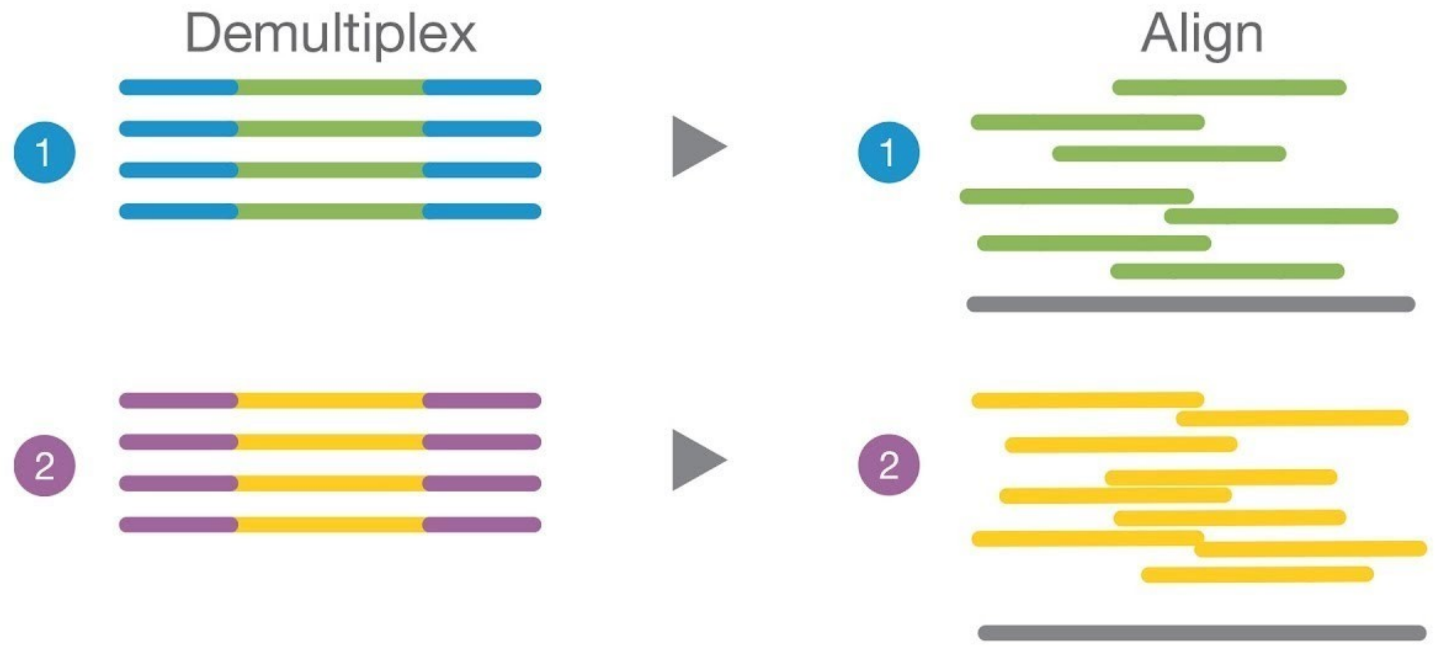
Demultiplexing



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

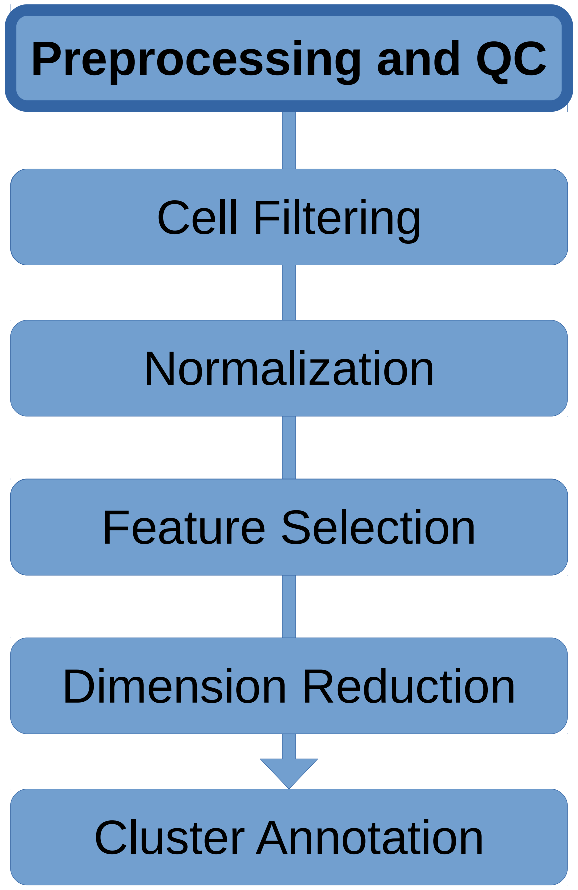
- After performing QC we align the reads and count UMIs for specific barcodes and positions to create an Expression Matrix (mxn).

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

Seurat

- An R package designed for higher level analysis and exploration of single-cell RNA-seq data.
- Current version: 3.0.0
- 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
```

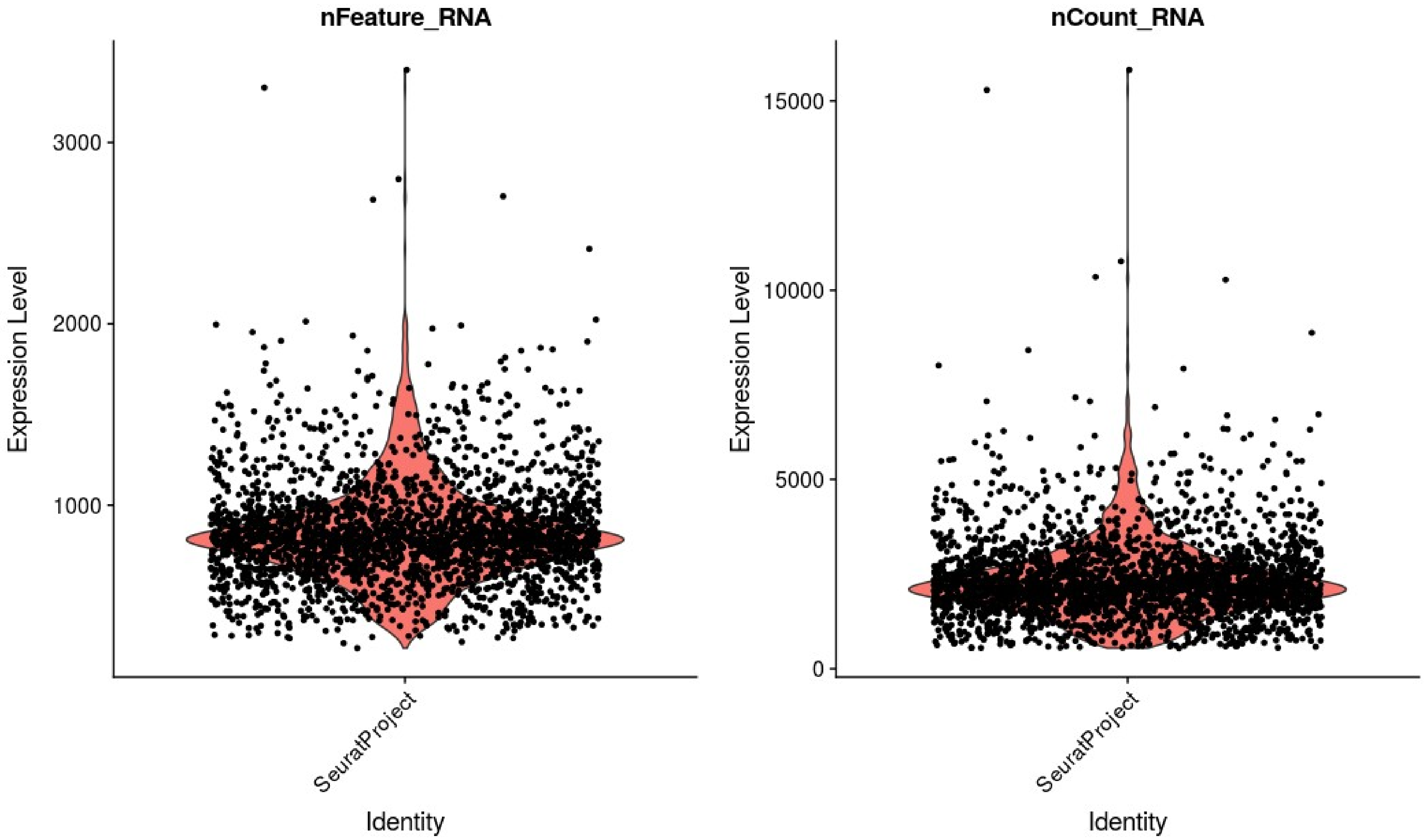
- open R (or Rstudio) and load the data in a seurat object.

```
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
)
```

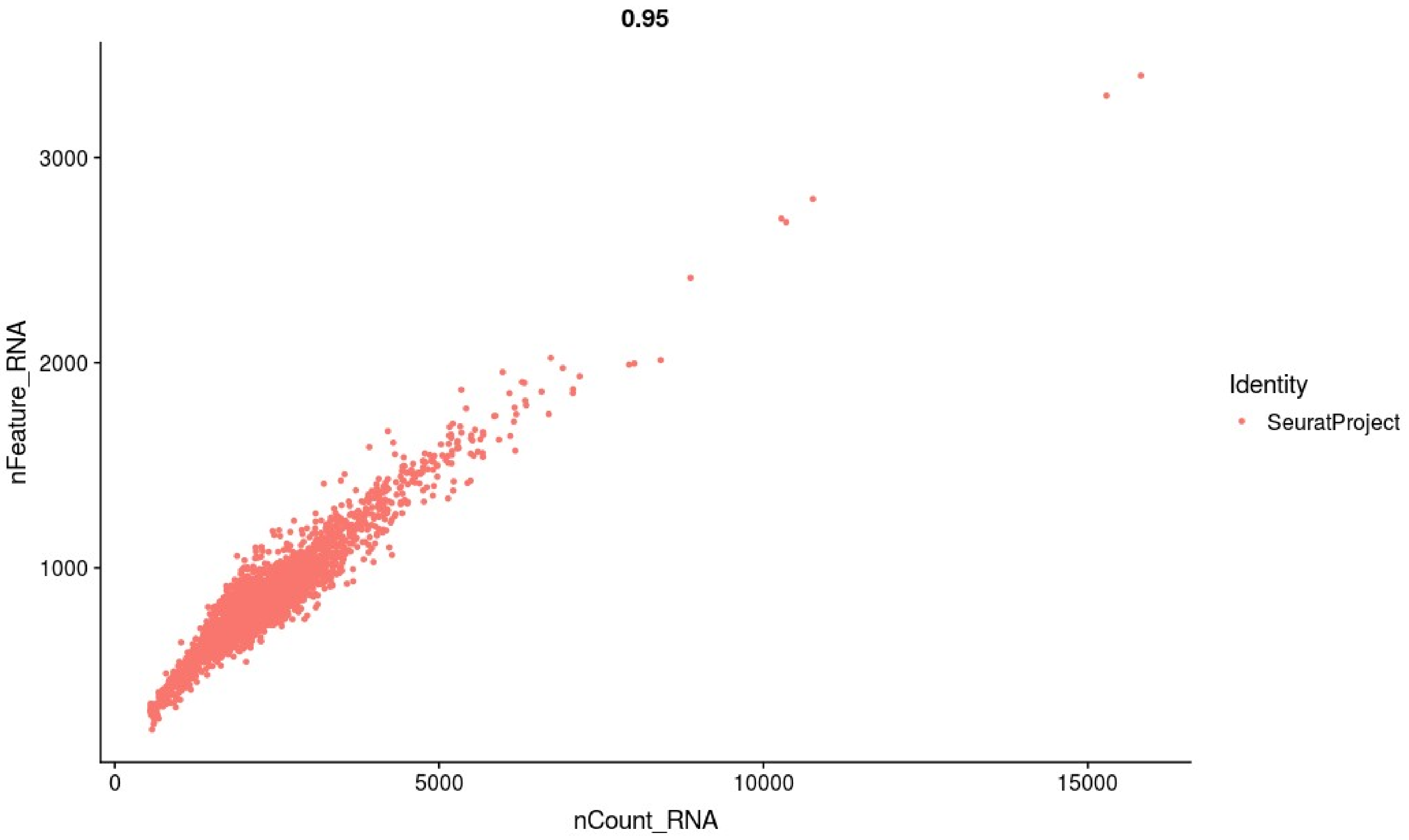
Seurat – Preprocessing

```
## An object of class Seurat
## 13714 features across 2700 samples within 1 assay
## Active assay: RNA (13714 features)
##   2 dimensional reductions calculated: pca, tsne

# Plot  the expression level
VlnPlot(
  object = seuobj,
  features = c("nFeature_RNA", "nCount_RNA"),
  ncol = 2
)

# Plot the feature correlation
FeatureScatter(
  object = seuobj,
  feature1 = "nCount_RNA",
  feature2 = "nFeature_RNA"
)
```

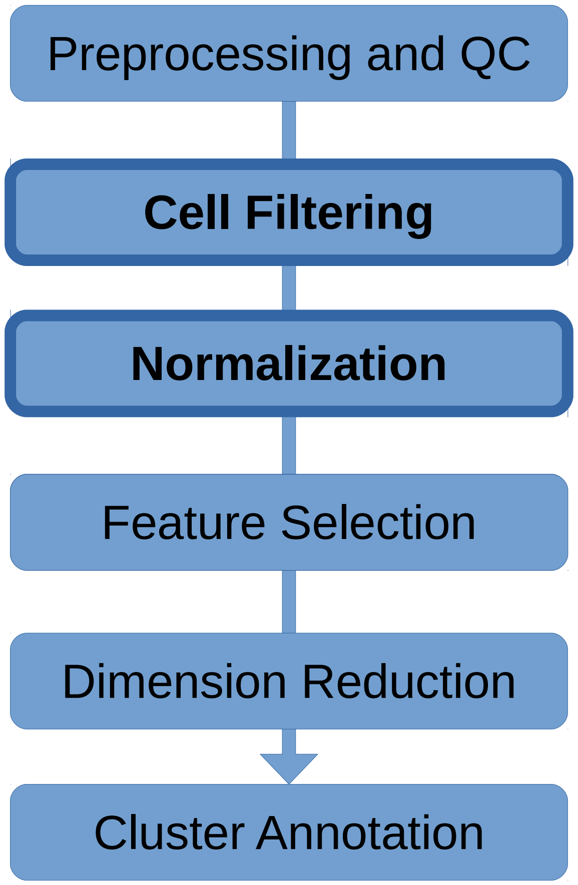
Seurat – Preprocessing



Seurat – Preprocessing



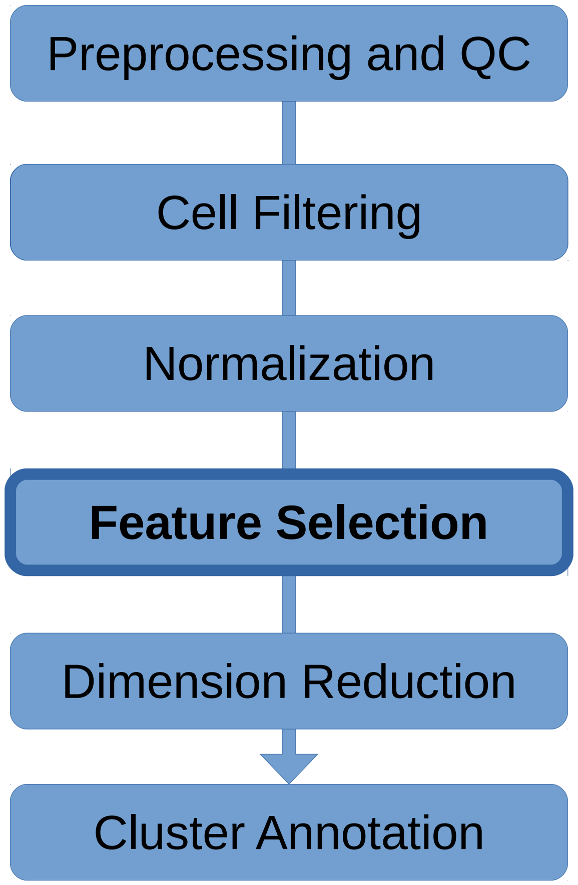
Seurat – pipeline



Seurat – Data normalization

```
# Filter cells with outlier number of read counts
seuobj <- subset(
  x = seuobj,
  subset = nFeature_RNA < 2500 & nFeature_RNA > 200
) # Currently a problem in development version. If you need to apply this,
install Seurat from CRAN (install.packages(Seurat))

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

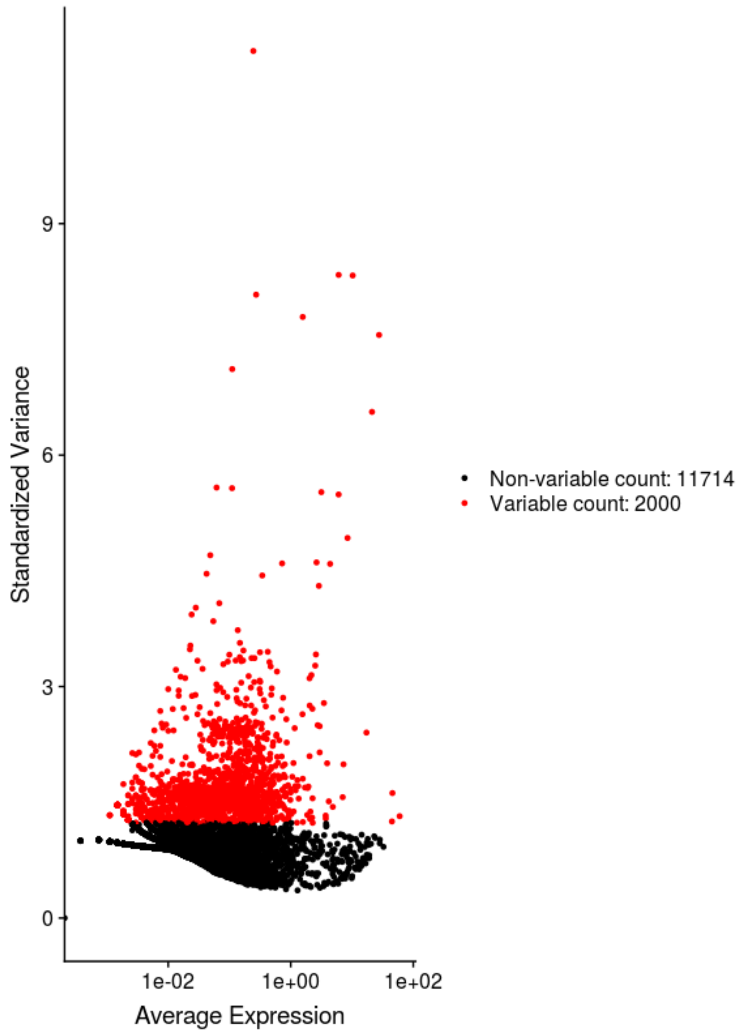
Seurat – pipeline

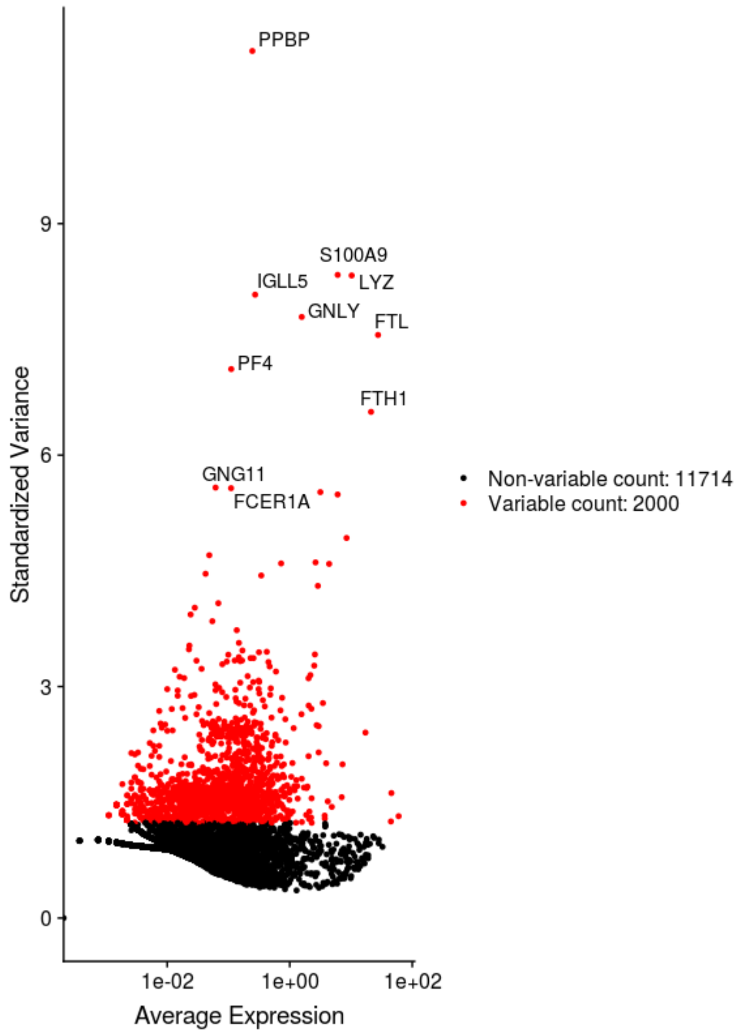


Features

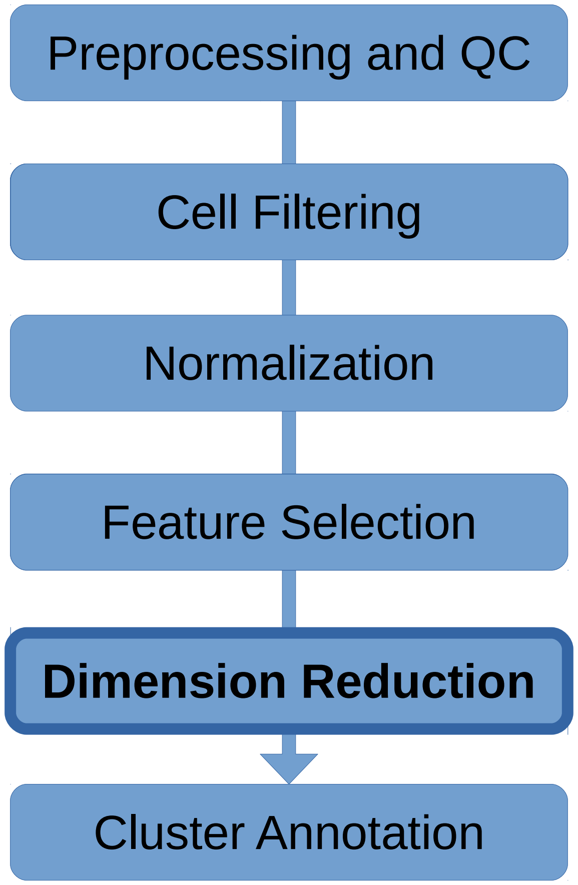
```
# Identification of highly variable features
seuobj <- FindVariableFeatures(
  object = seuobj,
  mean.function = ExpMean,
  dispersion.function = LogVMR,
  x.low.cutoff = 0.0125,
  x.high.cutoff = 3,
  y.cutoff = 0.5
)

# 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





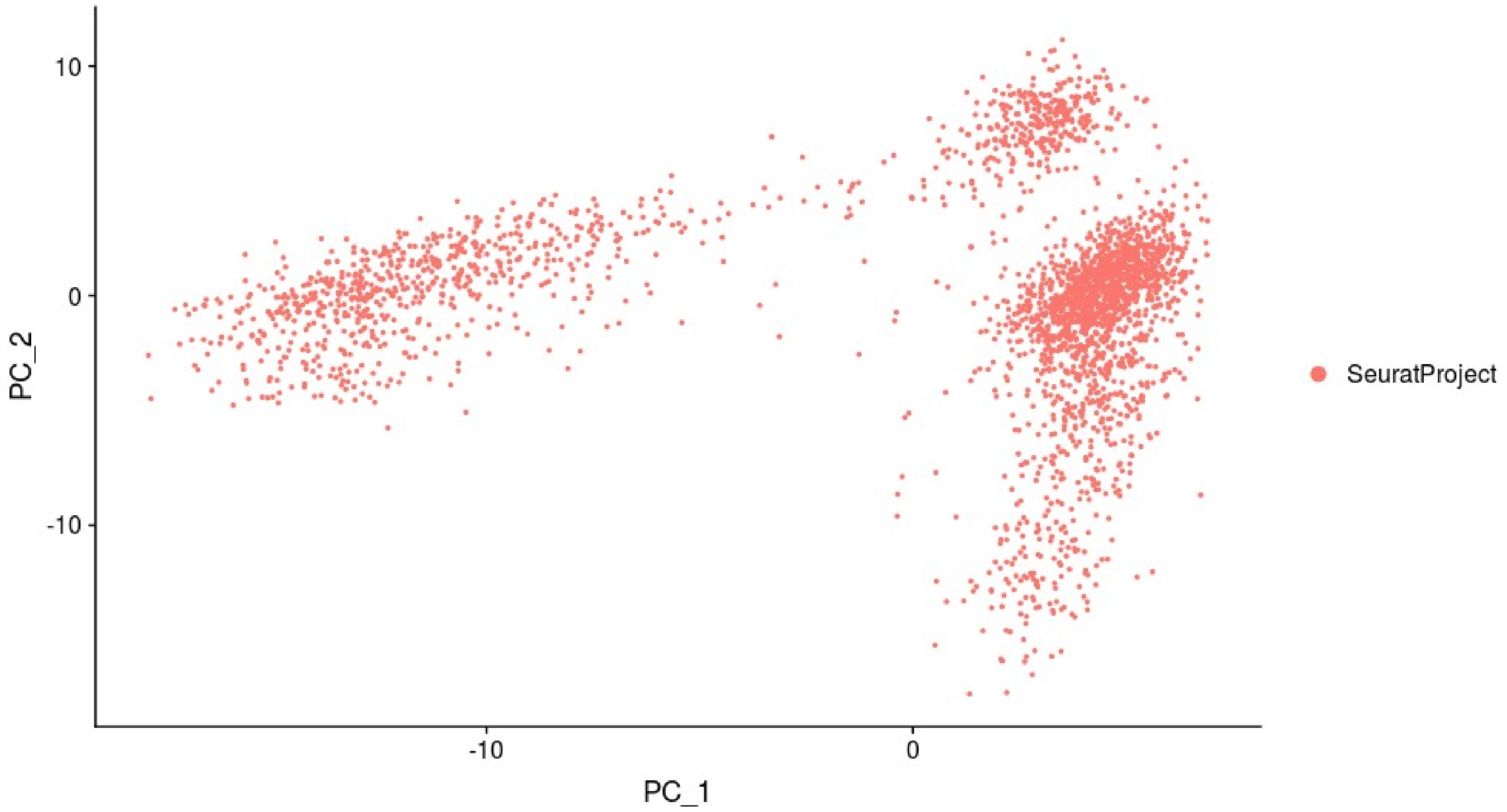
Seurat – pipeline



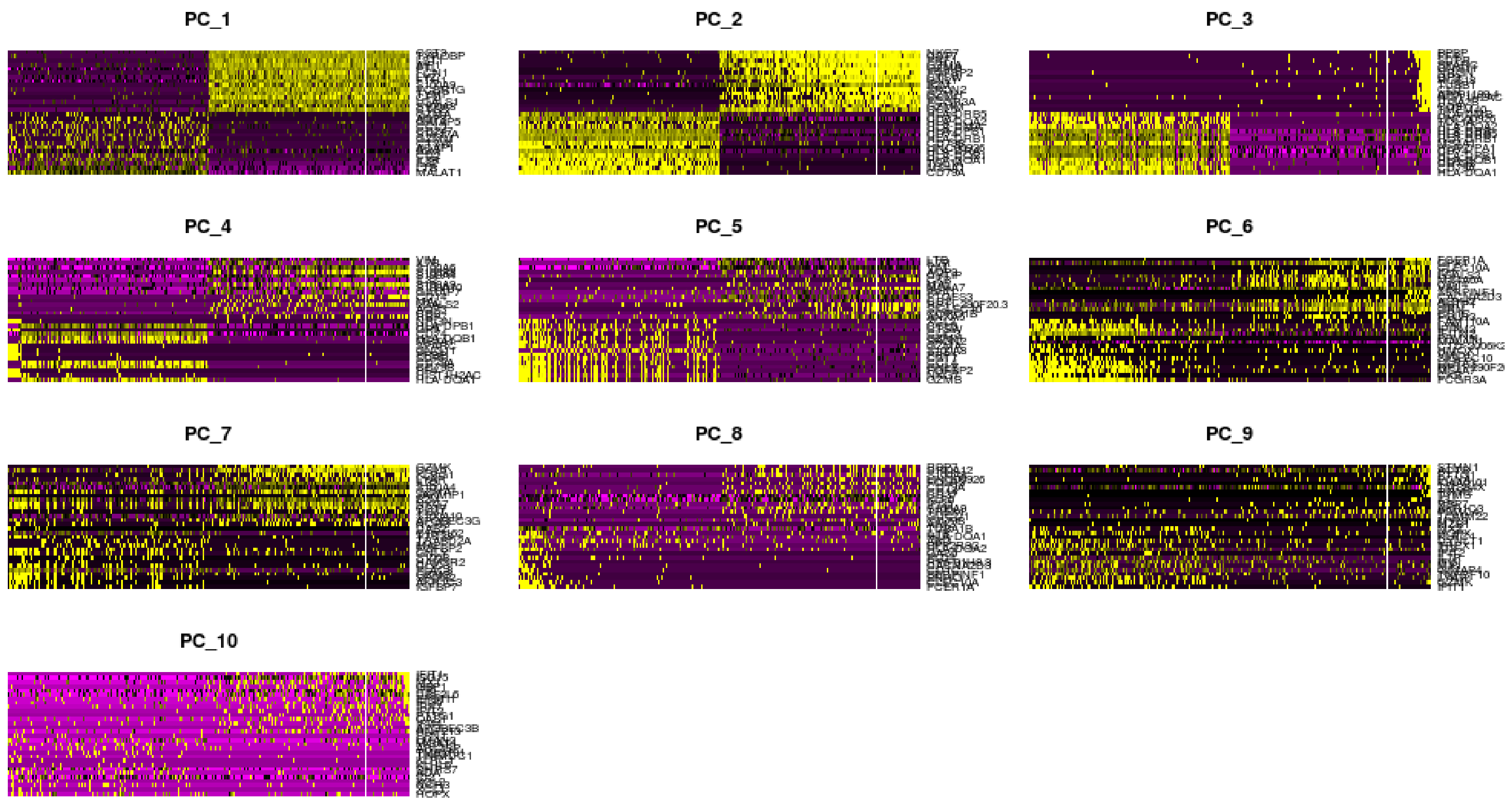
Seurat – Scale and Dimension Reduction

```
# 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 = 1:10, cells = 500, balanced = TRUE)
ElbowPlot(object = seuobj)
```

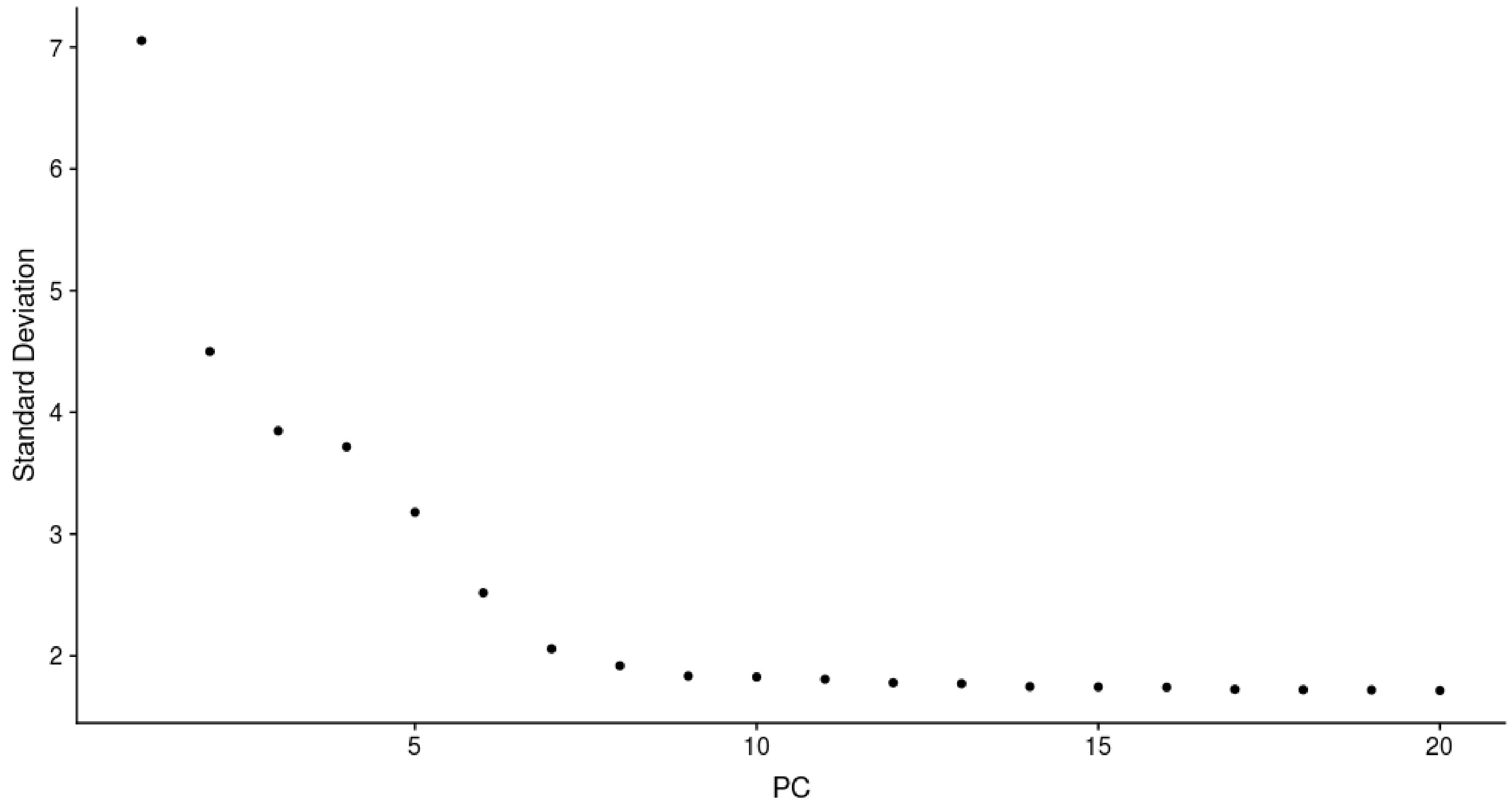
Seurat – Scale and Dimension Reduction



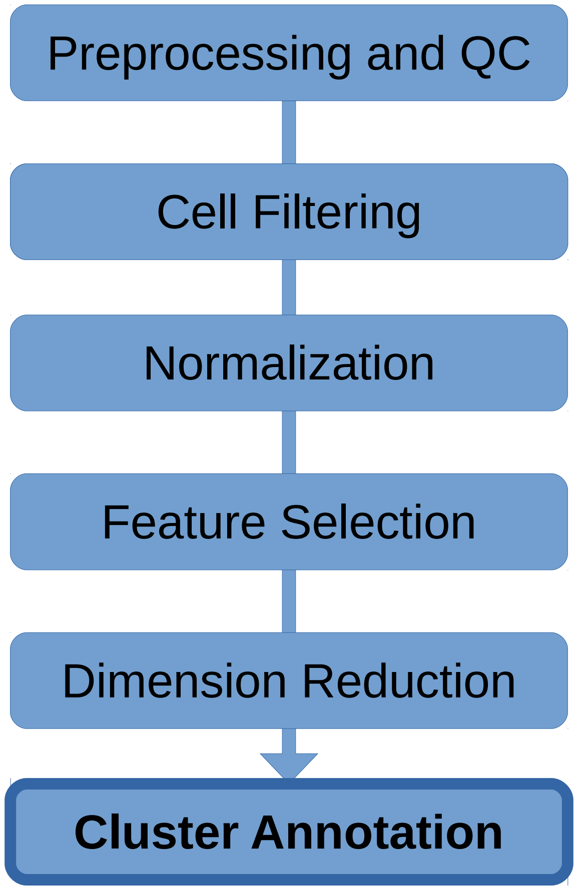
Seurat – Scale and Dimension Reduction



Seurat – Scale and Dimension Reduction



Seurat – pipeline

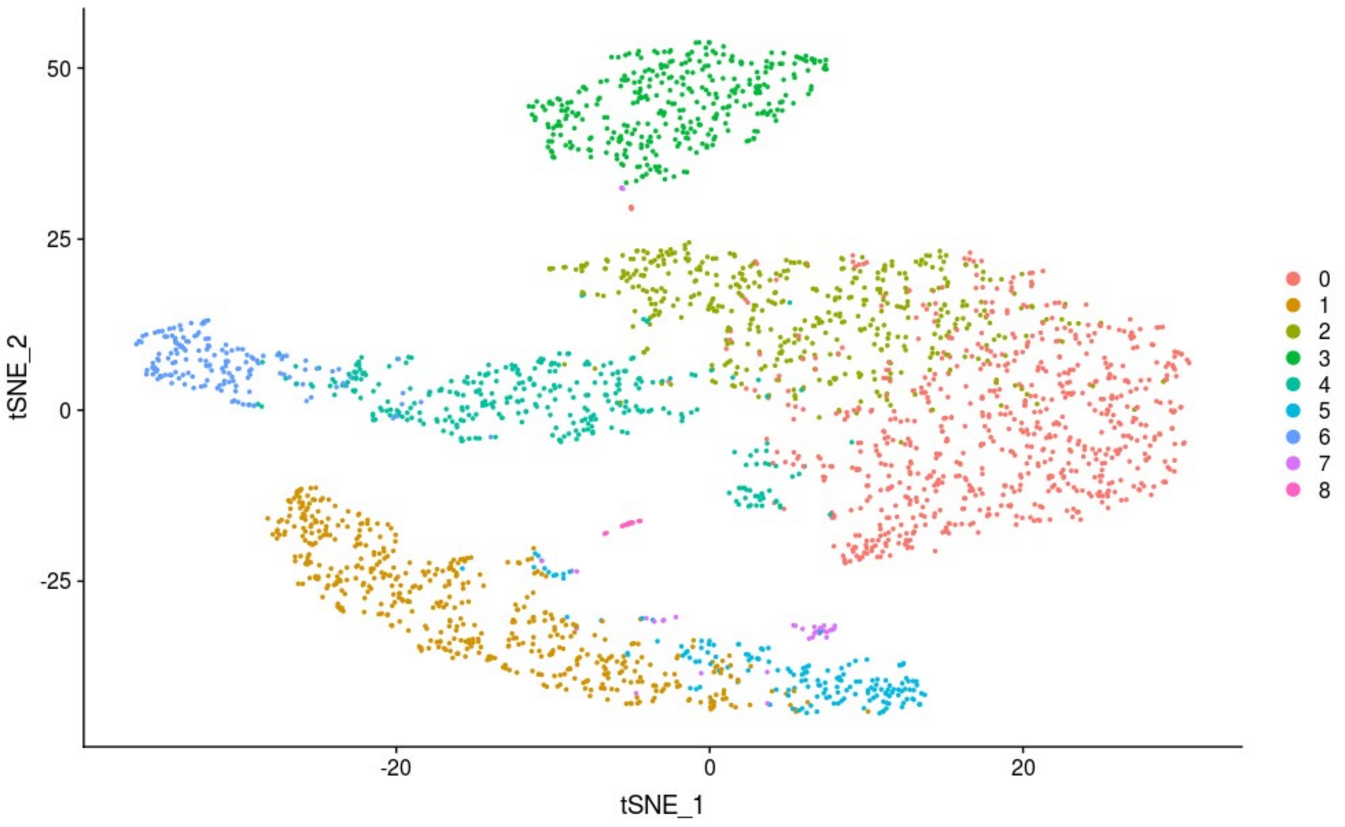


Seurat – Cluster Cells

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

## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 2695
## Number of edges: 97555
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.8746
## Number of communities: 9
## Elapsed time: 0 seconds
```

Seurat – Plot

```
# Run TSNE dimension reductions
seuobj <- RunTSNE( object = seuobj, dims.use = 1:8, do.fast = TRUE)
TSNEPlot(object = seuobj)
```

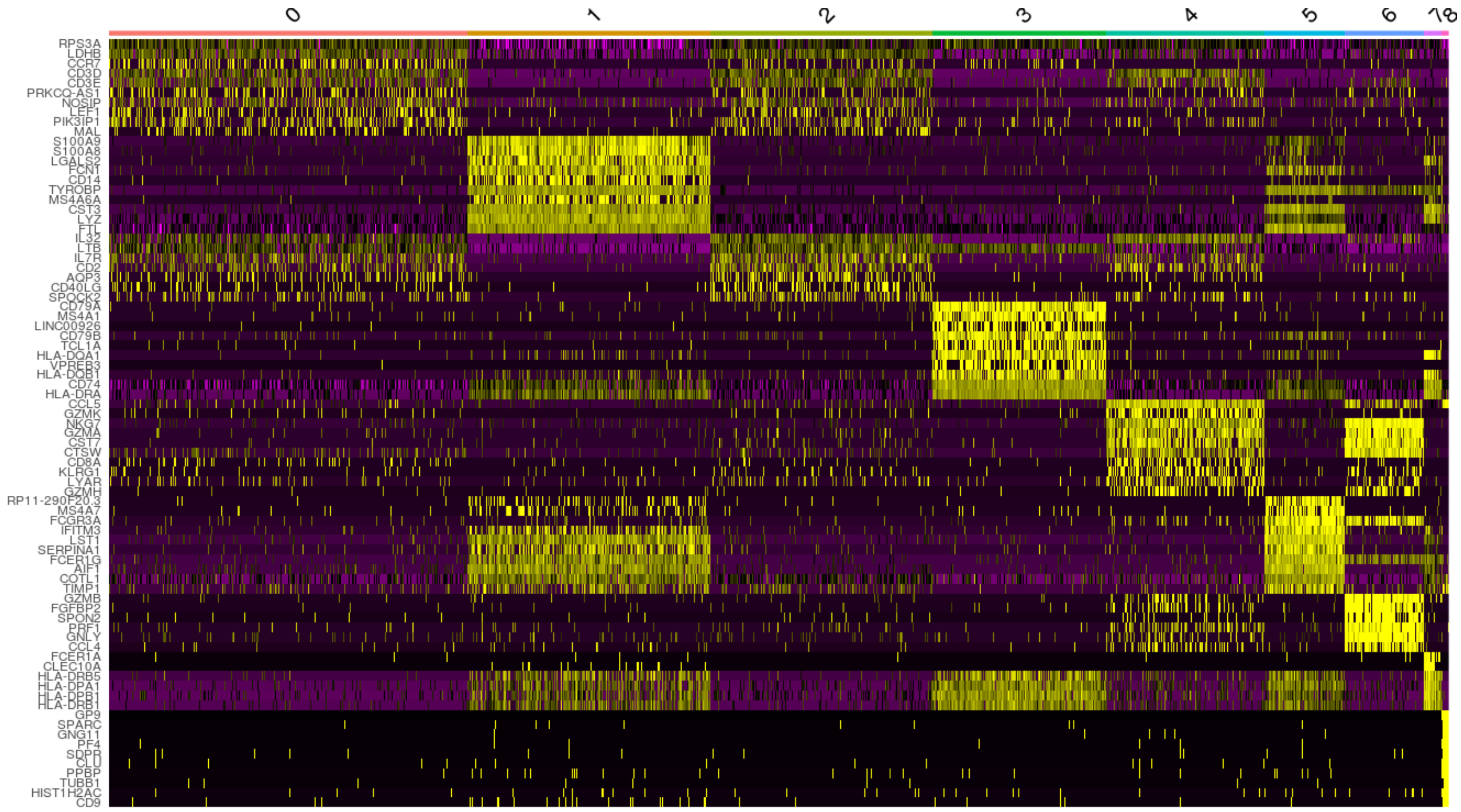


Seurat – Identify markers for cells

```
# 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)
# Sort by influnce and group by cluster
seuobj.markers %>% group_by(cluster) %>% top_n(n = 2, wt = avg_logFC)
```

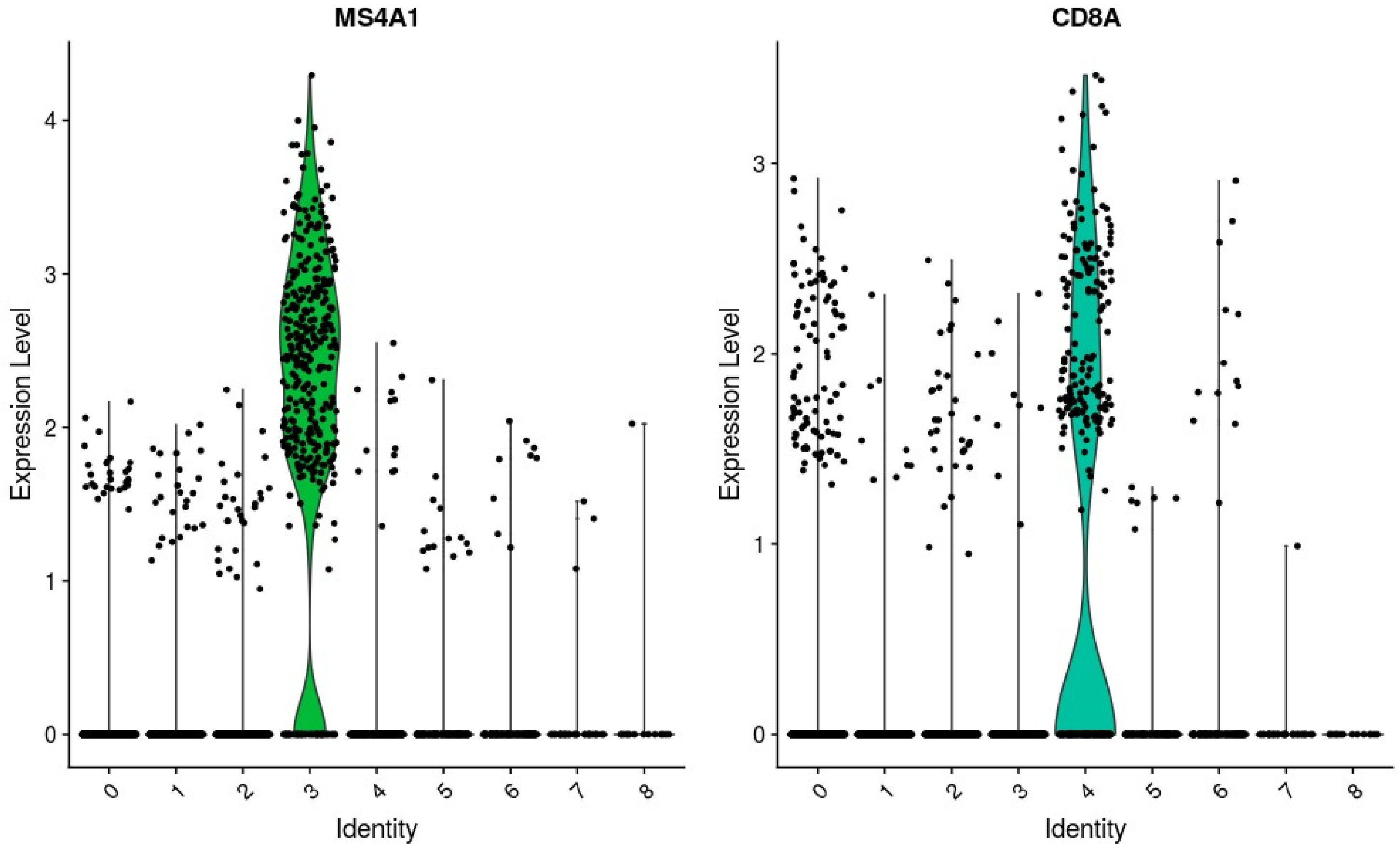
Seurat – Unbiased cluster identification

```
# 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()
```



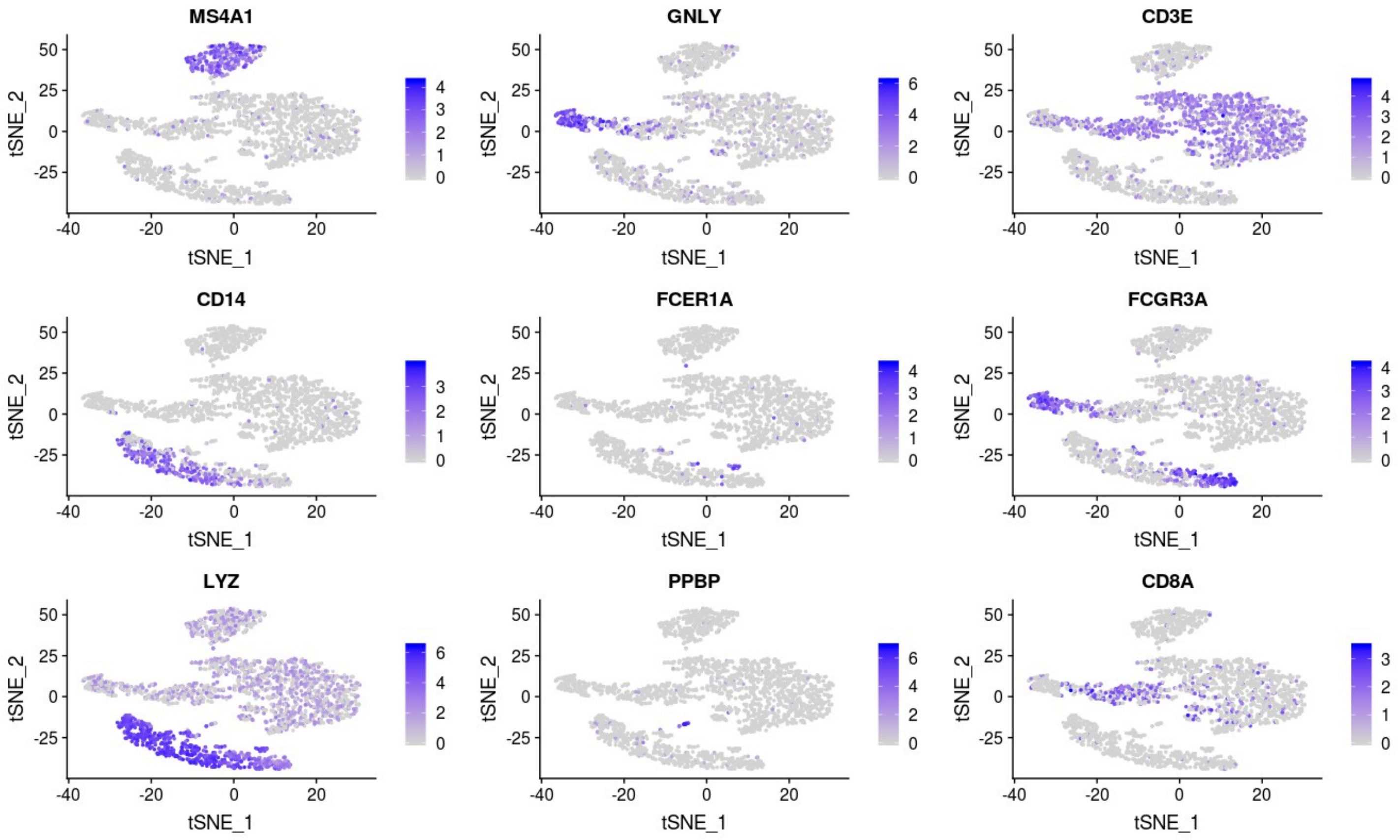
Seurat – Expert-based cluster annotation

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



Seurat – Expert-based cluster annotation

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



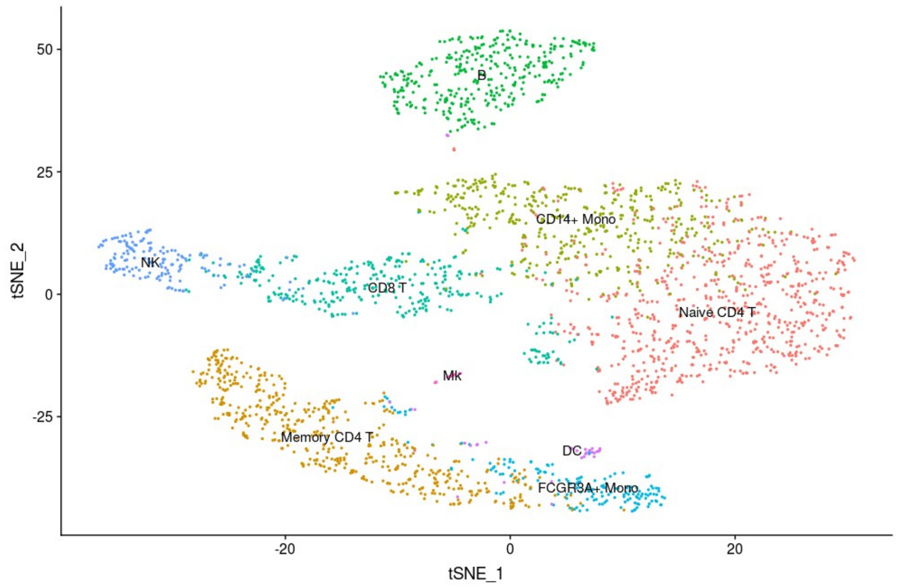
Seurat – Expert-based cluster annotation

```
# Using a known marker identify clusters
seuobj.markers["MS4A1",]$cluster
```

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 tSNE 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 = pbmc)
pbmc <- RenameIdents(object = pbmc, new.cluster.ids)
DimPlot(object = pbmc, reduction = "umap", label = TRUE, pt.size = 0.5) +
NoLegend()
```



Thank You for the attention