



CANCER
RESEARCH
UK

CAMBRIDGE
CENTRE

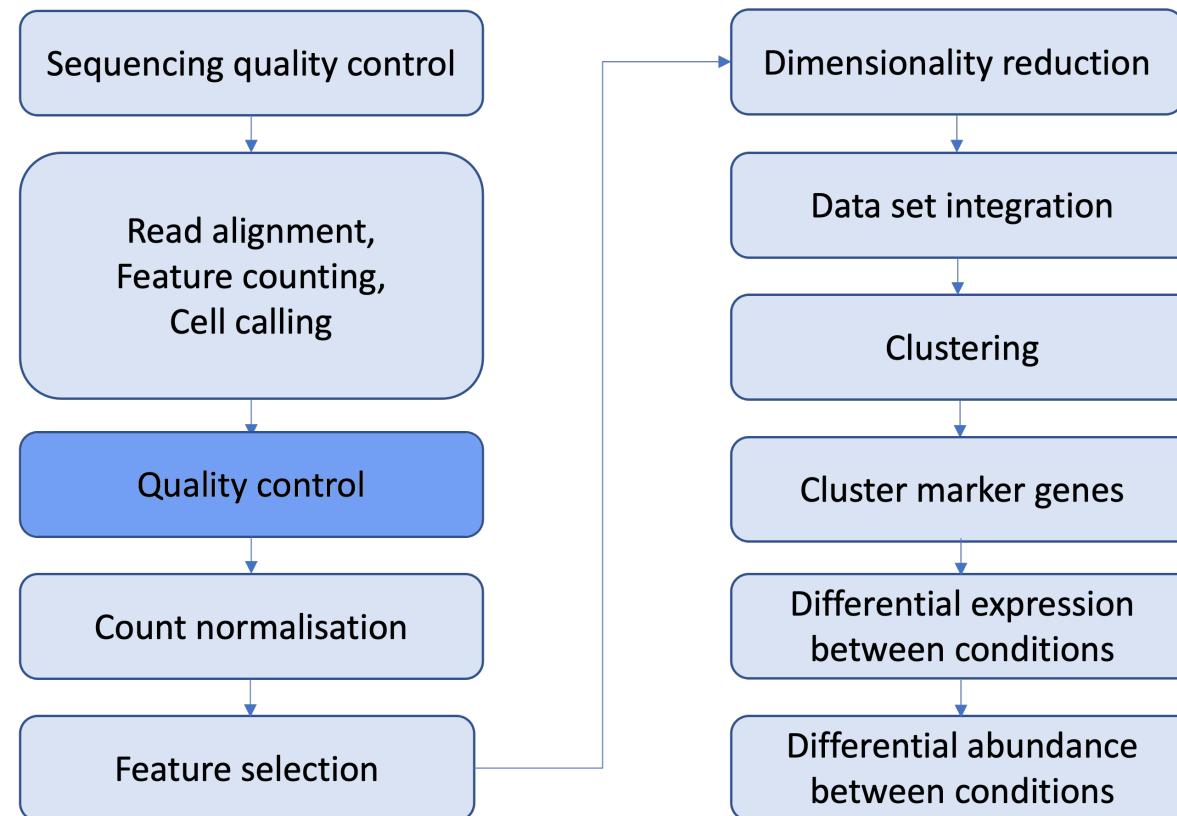
Introduction to single-cell RNA-seq analysis

Quality Control

Ashley Sawle

29th October 2021

Single Cell RNAseq Analysis Workflow



Quality Control

We will now check the quality of the data further:

- Mapping quality
- Cell counts
- Distribution of keys quality metrics

We will then:

- Filter genes with very low expression
- Identify low-quality cells
- Filter and/or mark low quality cells

Quality Control

We will now check the quality of the data further:

- Mapping quality
- Cell counts
- Distribution of keys quality metrics

We will then:

- Filter genes with very low expression
- Identify low-quality cells
- Filter and/or mark low quality cells



Quality Control

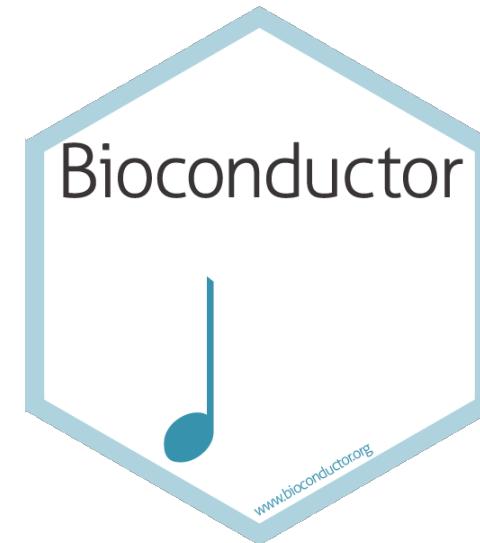
Bioconductor R packages:

- *scater*
- *DropletUtils*

Orchestrating Single-Cell Analysis with Bioconductor

Robert Amezquita, Aaron Lun, Stephanie Hicks, Raphael Gottardo

<http://bioconductor.org/books/release/OSCA/>



Read CellRanger outputs into R

```
%h%-$ ls SRR9264343/outs/raw_feature_bc_matrix  
barcodes.tsv.gz  
features.tsv.gz  
matrix.mtx.gz
```

Loading a single sample

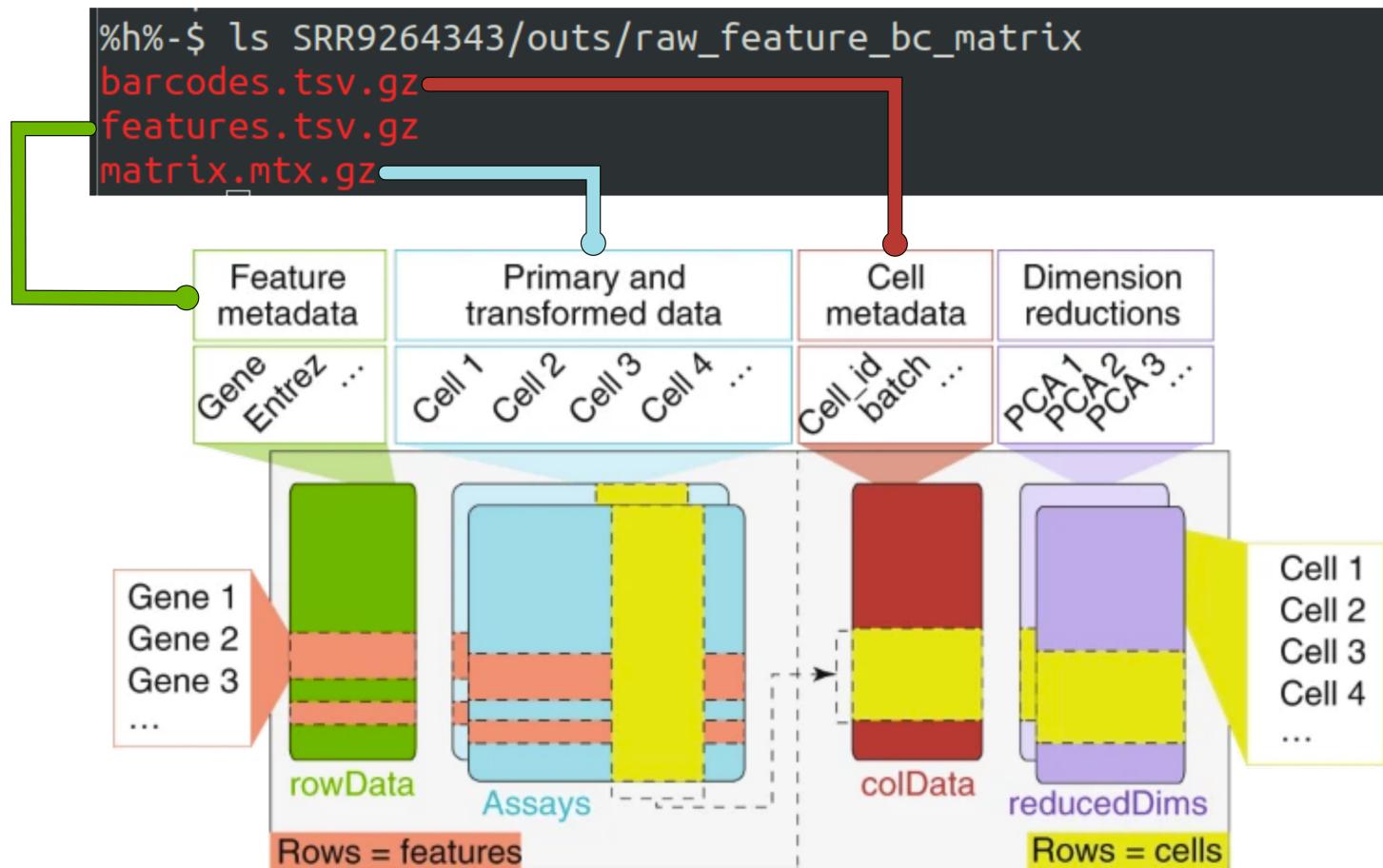
```
sample.path <- c(SRR9264343 = "CellRanger_Outputs/SRR9264343/outs/filtered_feature_bc_matrix/")  
sce <- read10xCounts(sample.path, col.names=TRUE)  
sce <- read10xCounts(sample.path, col.names=FALSE)
```

Loading multiple samples

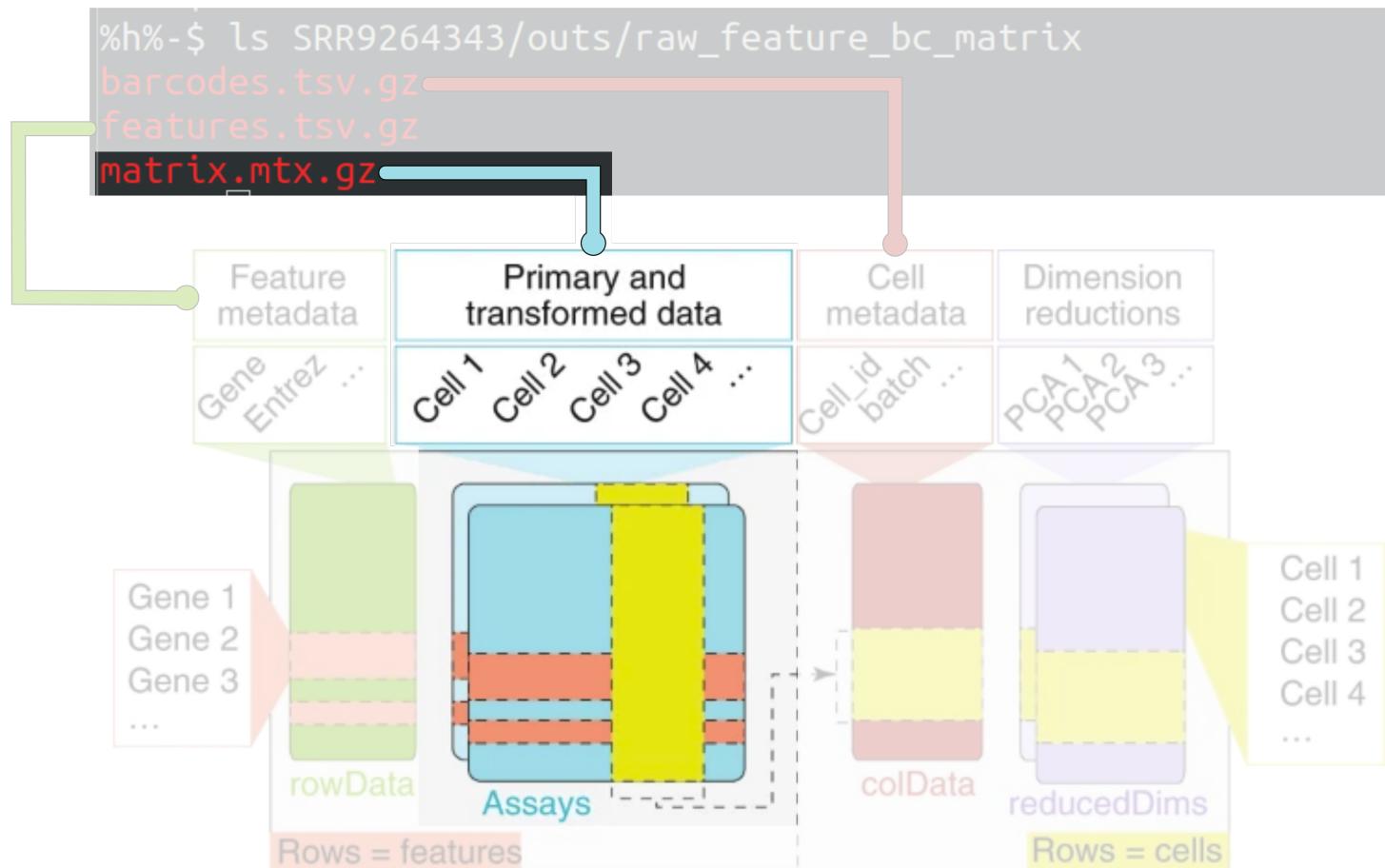
```
list_of_files <- c(SRR9264343 = "CellRanger_Outputs/SRR9264343/outs/filtered_feature_bc_matrix",  
                   SRR9264344 = "CellRanger_Outputs/SRR9264344/outs/filtered_feature_bc_matrix",  
                   SRR9264347 = "CellRanger_Outputs/SRR9264347/outs/filtered_feature_bc_matrix")  
sce <- read10xCounts(list_of_files, col.names=TRUE)
```

→ *SingleCellExperiment* object

The *SingleCellExperiment* object



The Counts Matrix

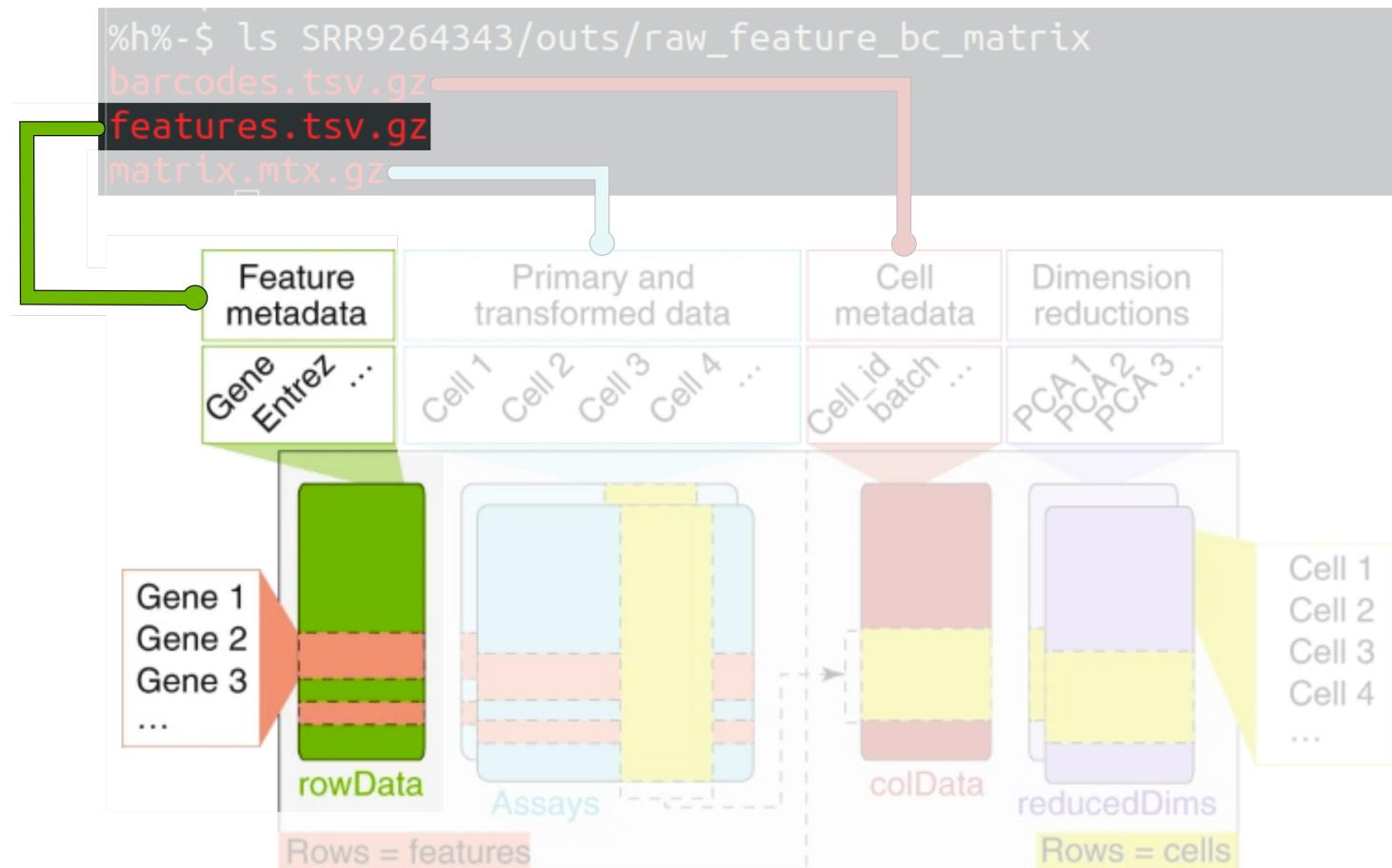


The Counts Matrix

```
counts(sce)
```

```
## 36601 x 3094 sparse Matrix of class "dgCMatrix"
##
## ENSG00000175756 1 2 . 1 1 . . 1 . . 2 1 . 1 2 2 1 . . 1 . 1 3 1 1 2 1 2 .....
## ENSG00000221978 . 2 . . . . . 3 . . . 1 . . . . . 1 . . . .
## ENSG00000224870 . . . . . . . . 1 1 . . . . . . . .
## ENSG00000242485 1 2 . 1 2 . . 1 . 1 1 4 . . 1 . 1 2 . 2 . 1 2 . . . 1 .
## ENSG00000272455 . . . . . . . . . . . . . .
## ENSG00000235098 . . . . . . . . . . . . .
## ENSG00000225905 . . . . . . . . . . . . .
## ENSG00000205116 . . . . . . . . . . . . .
## ENSG00000225285 . . . . . . . . . . . . .
## ENSG00000179403 . . . . . . . . . . . . .
## ENSG00000215915 . . . . . . . . . . . . .
## ENSG00000160072 . . . . . . . . . . . . .
## ENSG00000197785 1 . . . . . . . . . . . 1 .
## ENSG00000205090 . . . . . . . . . . . . .
## ENSG00000160075 2 1 . . . 1 3 1 4 2 3 1 . . 1 1 . 2 2 1 . 4 2 1 1 1 .
## ENSG00000215014 . . . . 1 . . . . . . . 1 .
## ENSG00000228594 . . . . . . . . . . . . .
## ENSG00000286989 . . . . . . . . . . . . .
## ENSG00000272106 . . . . . . . . . . . . .
## ENSG00000197530 . . . . . . . . . 1 . . 1 .
## ENSG00000189409 . . . . . . . . . . . . .
## ENSG00000248333 . 1 . . . 1 2 . . . . 1 1 .
## ENSG00000272004 . . . . . . . . . . . . .
## ENSG00000189339 . . . . 1 . . . . 1 . . 1 .
## ENSG00000008128 . . . . . . . . . . . . 1 .
## ENSG00000215790 . . . . . . . . . . . . 1 .
## ENSG00000008130 . . . . 1 . 1 1 . . 1 . . 1 .
```

Feature metadata

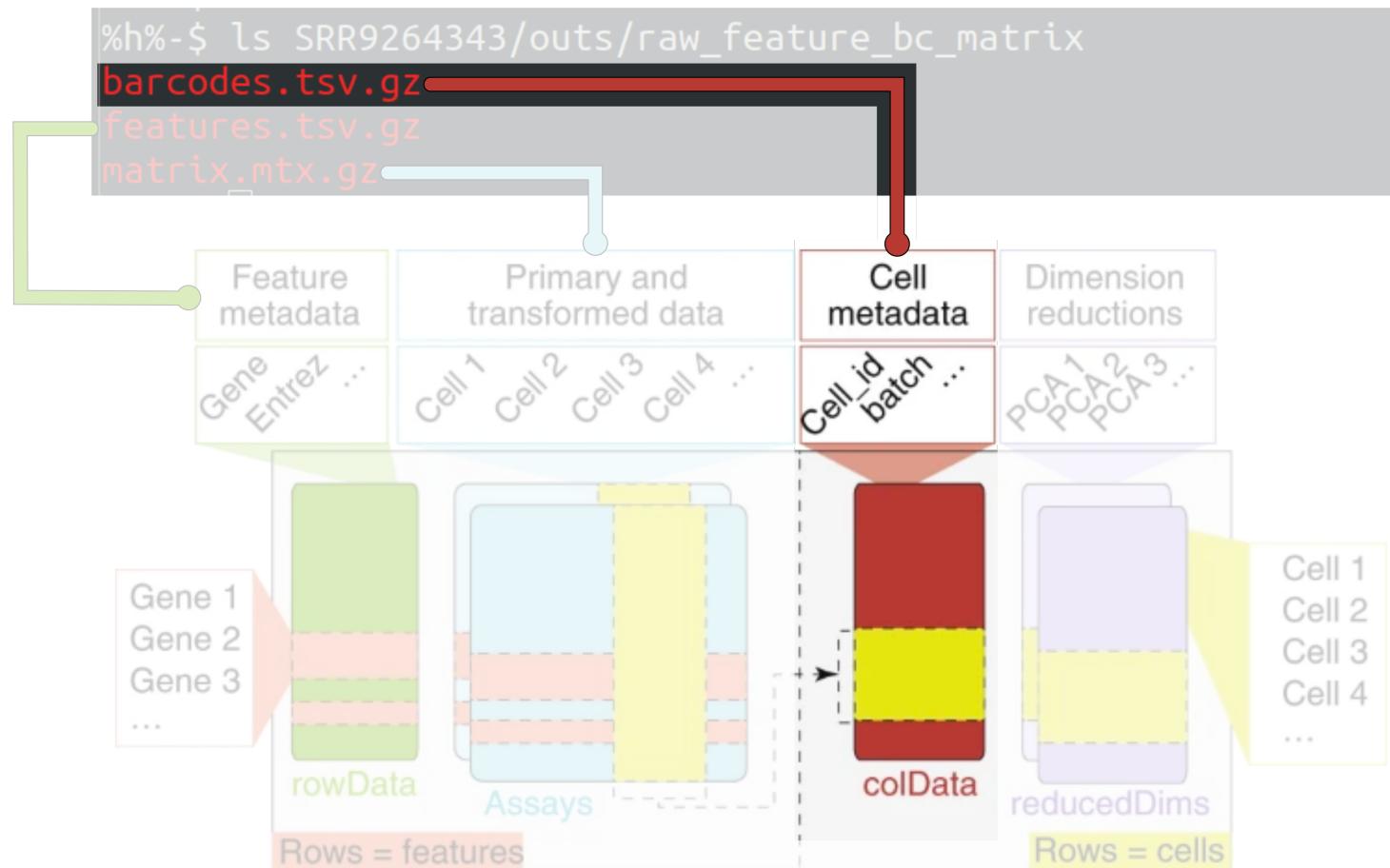


Feature metadata

```
rowData(sce)
```

```
## DataFrame with 36601 rows and 3 columns
##           ID      Symbol        Type
## <character> <character> <character>
## ENSG00000175756 ENSG00000175756 AURKAIP1 Gene Expression
## ENSG00000221978 ENSG00000221978 CCNL2 Gene Expression
## ENSG00000224870 ENSG00000224870 MRPL20-AS1 Gene Expression
## ENSG00000242485 ENSG00000242485 MRPL20 Gene Expression
## ENSG00000272455 ENSG00000272455 AL391244.2 Gene Expression
## ...
##       ...     ...
## ENSG00000240731 ENSG00000240731 AL139287.1 Gene Expression
## ENSG00000224051 ENSG00000224051 CPTP Gene Expression
## ENSG00000169962 ENSG00000169962 TAS1R3 Gene Expression
## ENSG00000107404 ENSG00000107404 DVL1 Gene Expression
## ENSG00000162576 ENSG00000162576 MXRA8 Gene Expression
```

Droplet annotation (Cell metadata)



Droplet annotation (Cell metadata)

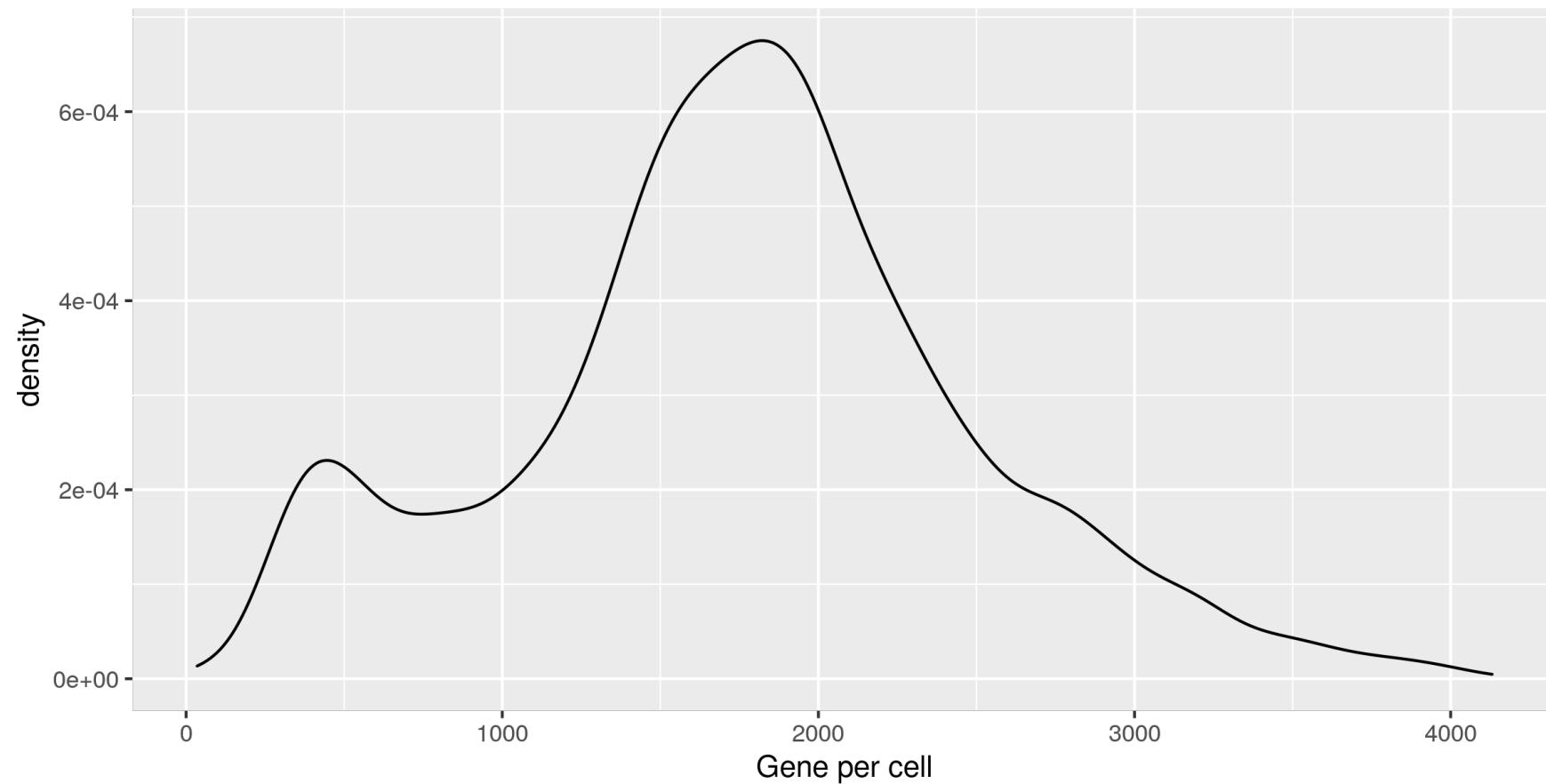
```
colData(sce)
```

```
## DataFrame with 3094 rows and 2 columns
##   Sample           Barcode
##   <character>     <character>
## 1 SRR9264343 AAACCTGAGACTTCG-1
## 2 SRR9264343 AAACCTGGTCTTCAAG-1
## 3 SRR9264343 AAACCTGGTGCAACTT-1
## 4 SRR9264343 AAACCTGGTGTGAGG-1
## 5 SRR9264343 AAACCTGTCCCAAGTA-1
## ...
## 3090 SRR9264343 TTTGGTTCTTAGGG-1
## 3091 SRR9264343 TTTGTCAAGAACGAG-1
## 3092 SRR9264343 TTTGTCAAGGACGAAA-1
## 3093 SRR9264343 TTTGTCACAGGCTCAC-1
## 3094 SRR9264343 TTTGTCAGTTCGGCAC-1
```

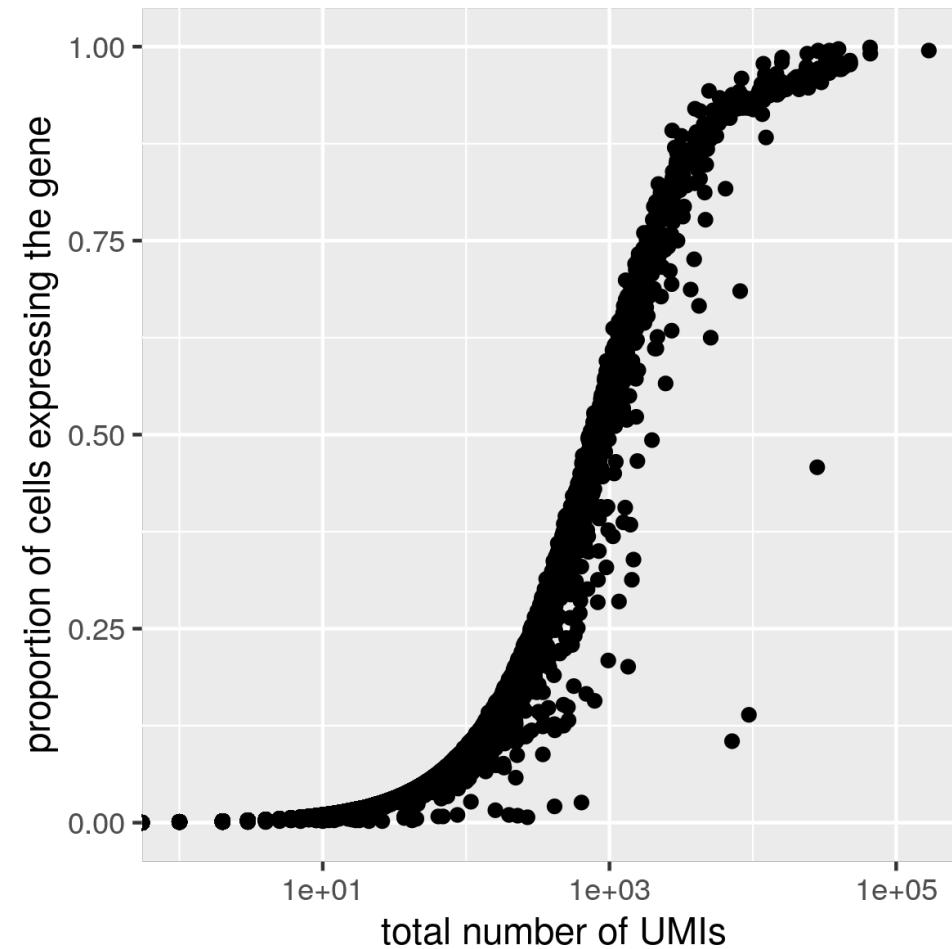
```
colnames(counts(sce))
```

```
## NULL
```

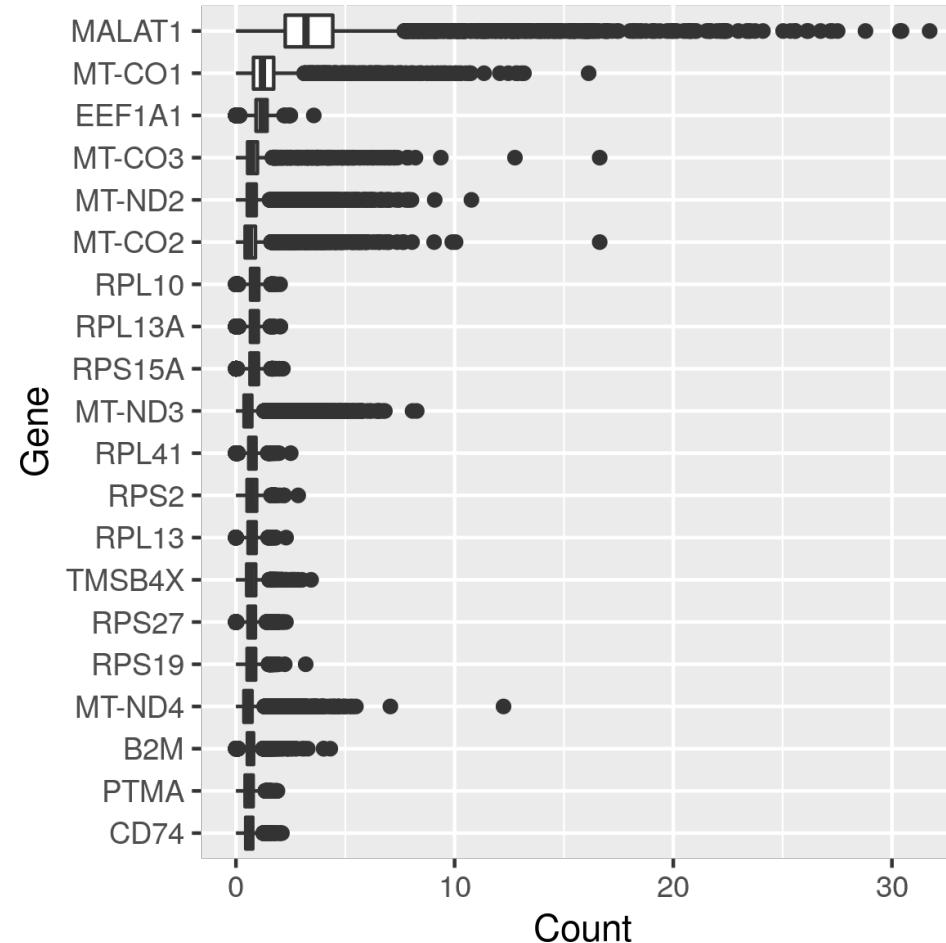
Properties of RNAseq data - Number of genes detected per cell



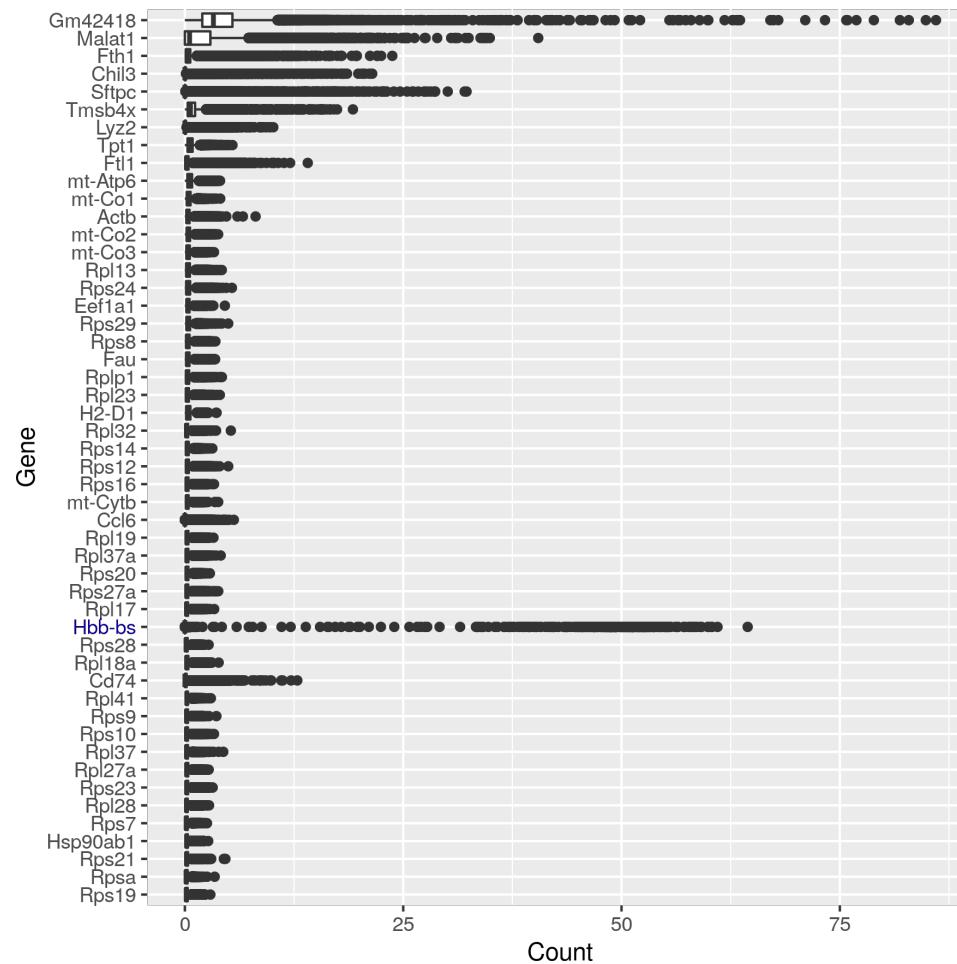
Properties of RNAseq data - Total UMIs



Properties of RNAseq data - Distribution of counts for a gene across cells



Properties of RNAseq data - Distribution of counts for a gene across cells



Remove undetected genes

Although the count matrix has 36601 genes, many of these will not have been detected in any droplet. We can remove these to reduce the size of the count matrix.

```
undetected_genes <- rowSums(counts(sce)) == 0
sce <- sce[!undetected_genes,]
sce

## class: SingleCellExperiment
## dim: 19938 3094
## metadata(1): Samples
## assays(1): counts
## rownames(19938): ENSG00000175756 ENSG00000221978 ...
##   ENSG00000169962 ENSG00000107404
## rowData names(3): ID Symbol Type
## colnames: NULL
## colData names(2): Sample Barcode
## reducedDimNames(0):
## mainExpName: NULL
## altExpNames(0):
```

Quality Control

- Not all of the droplets called as cells by CellRanger will contain good quality cells
- Poor quality droplets will adversely affect downstream analysis
- We can use QC metrics to filter out poor quality droplets:
 - Total UMIs (Library size)
 - Number of gene detected
 - Proportion of UMIs mapping to mitochondrial genes

Quality Control

- Add gene annotation to identify Mt genes - AnnotationHub

```
rowData(sce)
```

```
## DataFrame with 19938 rows and 4 columns
##           ID      Symbol       Type Chromosome
##           <character> <character> <character> <character>
## ENSG00000175756 ENSG00000175756 AURKAIP1 Gene Expression 1
## ENSG00000221978 ENSG00000221978 CCNL2 Gene Expression 1
## ENSG00000224870 ENSG00000224870 MRPL20-AS1 Gene Expression 1
## ENSG00000242485 ENSG00000242485 MRPL20 Gene Expression 1
## ENSG00000272455 ENSG00000272455 AL391244.2 Gene Expression 1
## ...
##           ...      ...       ...      ...
## ENSG00000212907 ENSG00000212907 MT-ND4L Gene Expression MT
## ENSG00000198886 ENSG00000198886 MT-ND4 Gene Expression MT
## ENSG00000198786 ENSG00000198786 MT-ND5 Gene Expression MT
## ENSG00000198695 ENSG00000198695 MT-ND6 Gene Expression MT
## ENSG00000198727 ENSG00000198727 MT-CYB Gene Expression MT
```

Quality Control

```
is.mito <- which(rowData(sce)$Chromosome=="MT")
sce <- addPerCellQC(sce, subsets = list(Mito = is.mito))
```

Adds six columns to the droplet annotation:

- **sum**: total UMI count
- **detected**: number of features (genes) detected
- **subsets_Mito_sum**: number of UMIs mapped to mitochondrial transcripts
- **subsets_Mito_detected**: number of mitochondrial genes detected
- **subsets_Mito_percent**: percentage of UMIs mapped to mitochondrial transcripts
- **total**: also the total UMI count

Quality Control

```
is.mito <- which(rowData(sce)$Chromosome=="MT")
sce <- addPerCellQC(sce, subsets = list(Mito = is.mito))
```

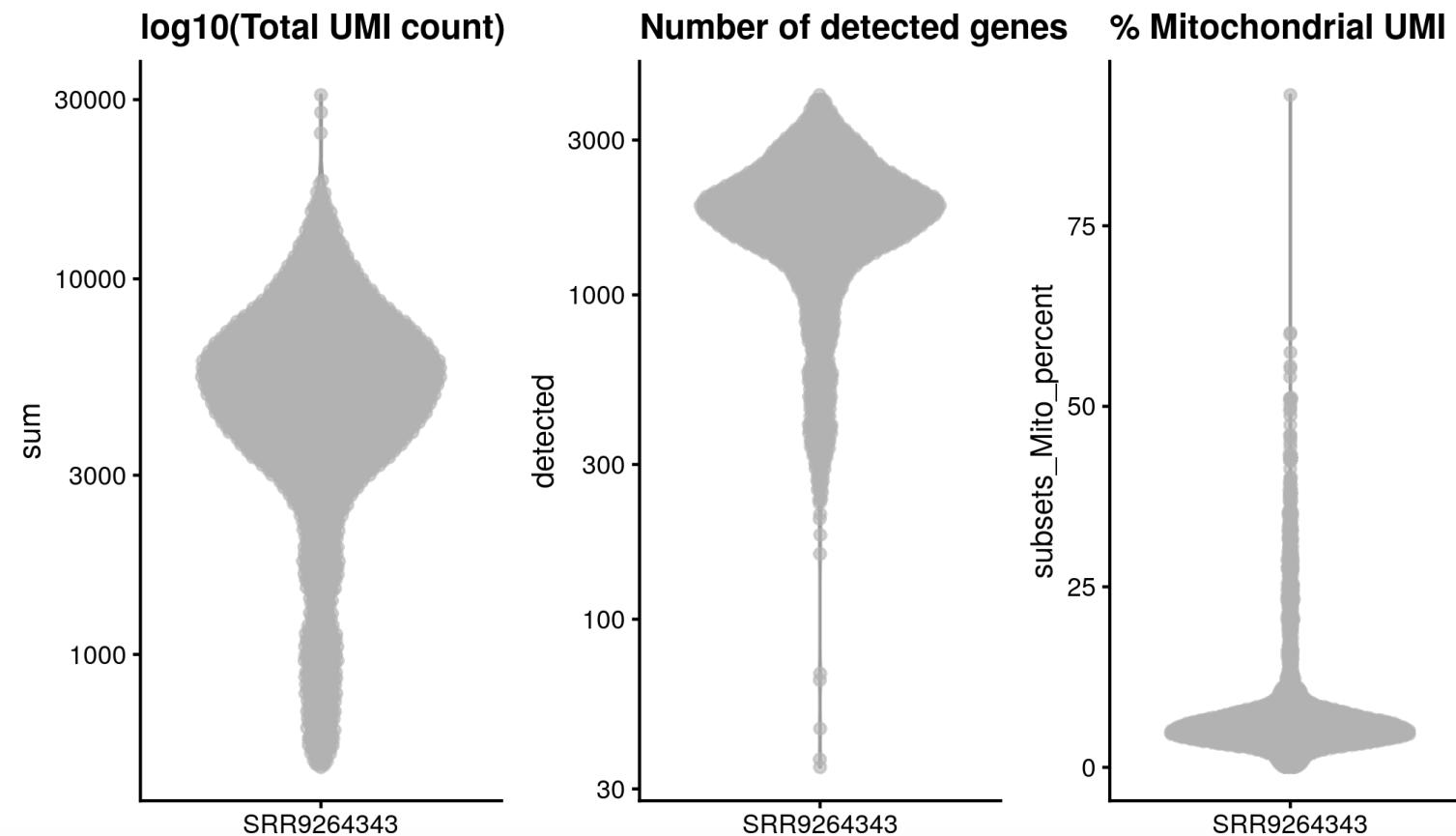
```
colData(sce)
```

```
## DataFrame with 3094 rows and 8 columns
##           Sample        Barcode      sum detected subsets_Mito_sum
## AACCTGAGACTTCG-1 SRR9264343 AACCTGAGACTTCG-1    6677    2056          292
## AACCTGGTCTTCAAG-1 SRR9264343 AACCTGGTCTTCAAG-1   12064    3177          575
## AACCTGGTGCAACTT-1 SRR9264343 AACCTGGTGCAACTT-1     843     363          428
## AACCTGGTGTGAGG-1  SRR9264343 AACCTGGTGTGAGG-1    8175    2570          429
## AACCTGTCCCAAGTA-1 SRR9264343 AACCTGTCCCAAGTA-1   8638    2389          526
## ...
## TTTGGTTTCTTAGGG-1 SRR9264343 TTTGGTTTCTTAGGG-1   3489    1600          239
## TTTGTCAAGAACGAG-1 SRR9264343 TTTGTCAAGAACGAG-1   7809    2415          548
## TTTGTCAAGGACGAAA-1 SRR9264343 TTTGTCAAGGACGAAA-1   9486    2589          503
## TTTGTCACAGGCTCAC-1 SRR9264343 TTTGTCACAGGCTCAC-1   1182     591          224
## TTTGTCAGTCGGCAC-1 SRR9264343 TTTGTCAGTCGGCAC-1   10514    2831          484
##           subsets_Mito_detected subsets_Mito_percent      total
## AACCTGAGACTTCG-1                12       4.37322      6677
## AACCTGGTCTTCAAG-1               12       4.76625     12064
## AACCTGGTGCAACTT-1              11      50.77106      843
## AACCTGGTGTGAGG-1               12       5.24771      8175
## AACCTGTCCCAAGTA-1              13       6.08937     8638
## ...
## TTTGGTTTCTTAGGG-1               11       6.85010      3489
## TTTGTCAAGAACGAG-1               12       7.01754      7809
## TTTGTCAAGGACGAAA-1              12       5.30255      9486
```

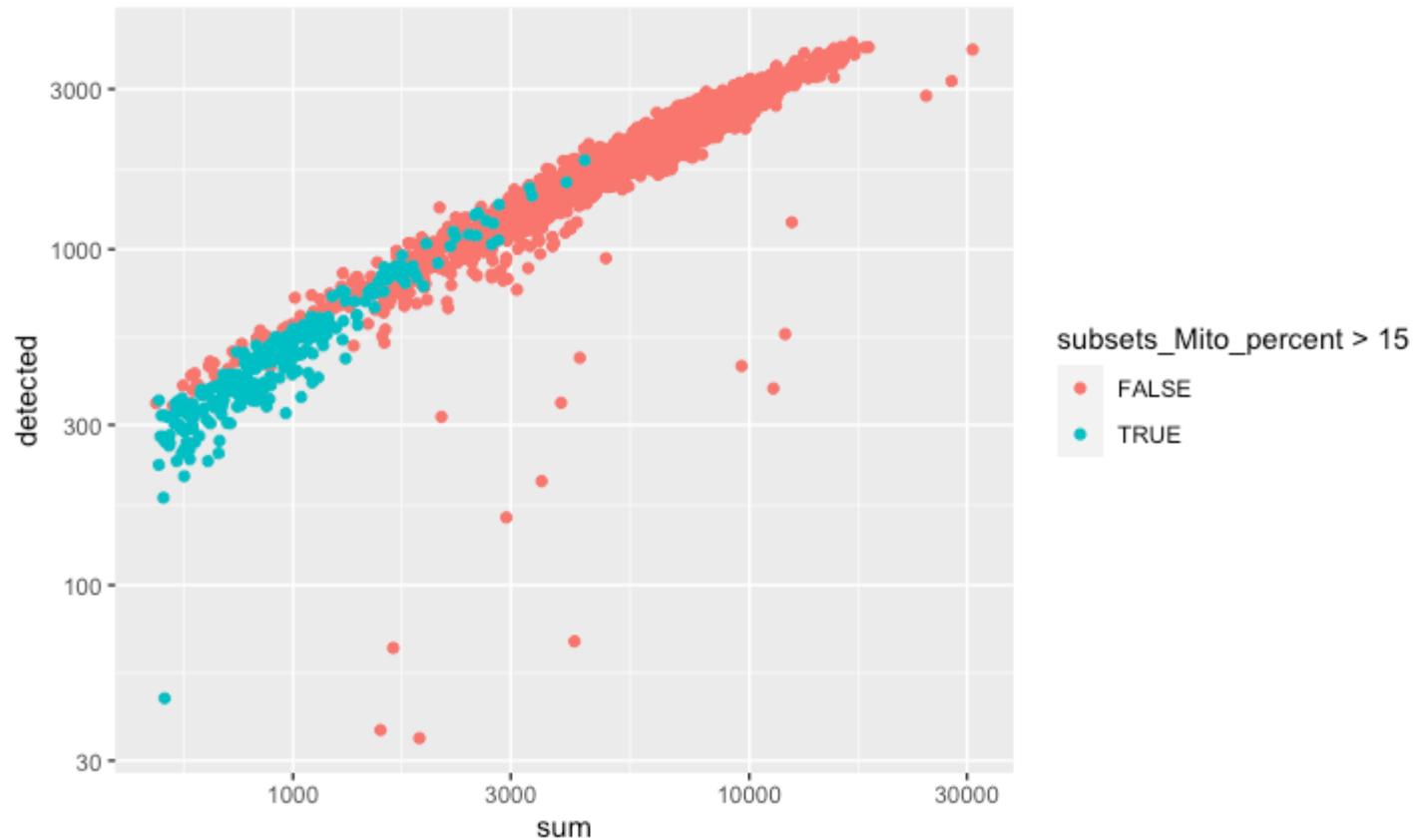


QC metrics - distribution

```
plotColData(sce, x="Sample", y="sum") + scale_y_log10()  
plotColData(sce, x="Sample", y="detected") + scale_y_log10()  
plotColData(sce, x="Sample", y="subsets_Mito_percent")
```

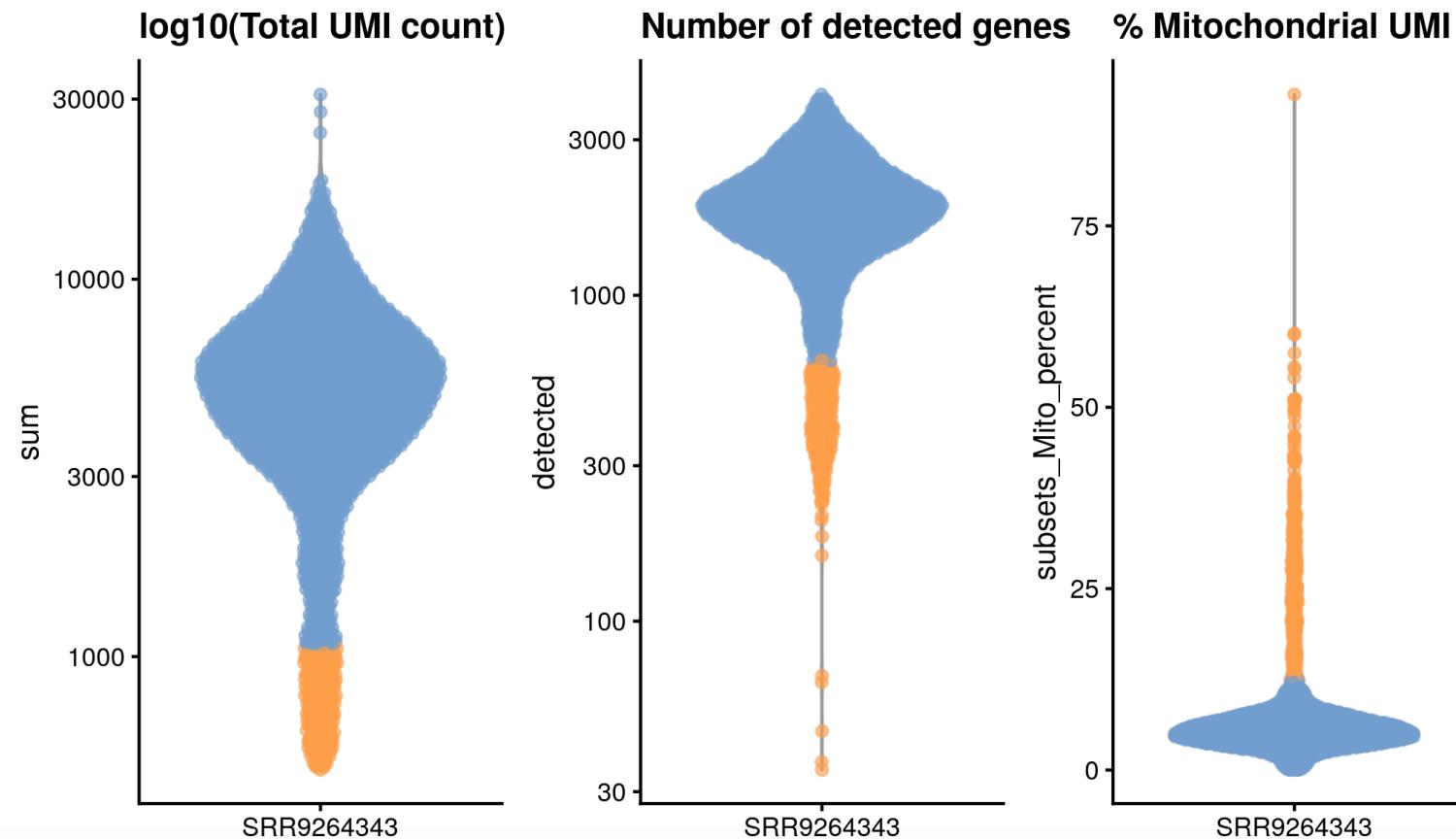


QC metrics - relationship



Identification of low-quality cells with adaptive thresholds

```
sce$low_lib_size <- isOutlier(sce$sum, log=TRUE, type="lower")
sce$low_n_features <- isOutlier(sce$detected, log=TRUE, type="lower")
sce$high_Mito_percent <- isOutlier(sce$subsets_Mito_percent, type="higher")
```



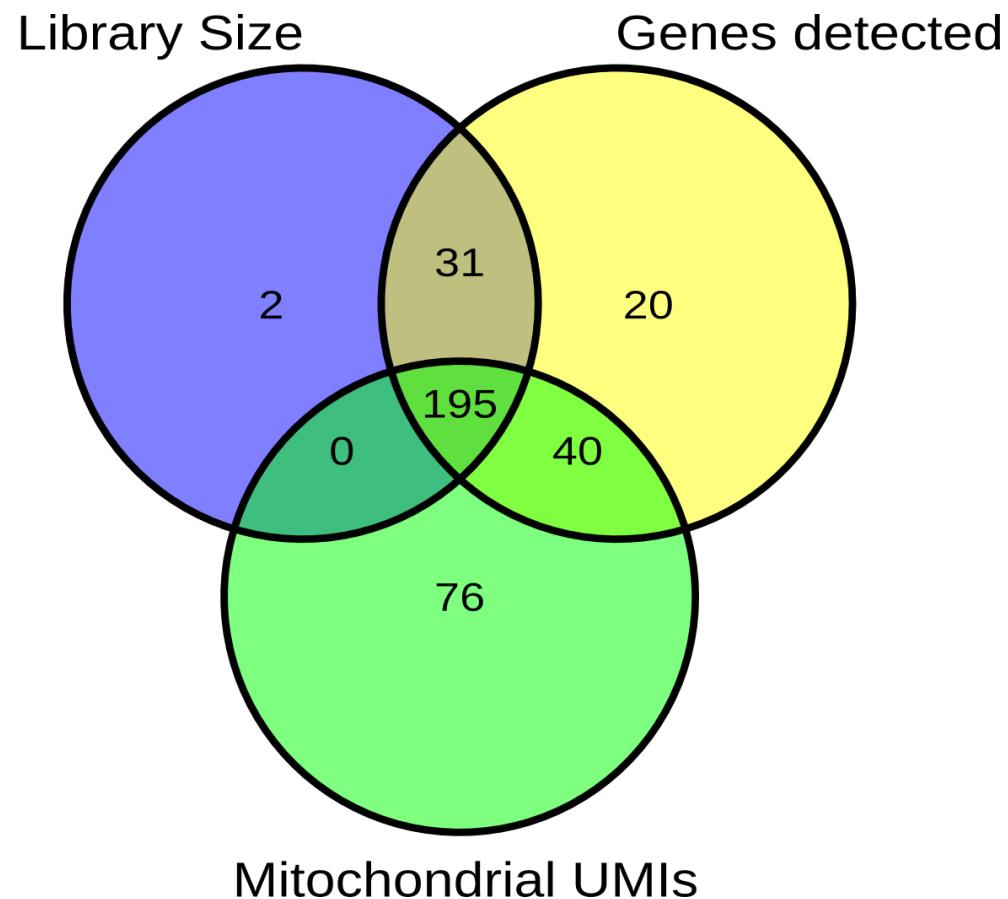
All three filter steps at once

```
cell_qc_results <- quickPerCellQC(colData(sce), percent_subsets=c("subsets_Mito_percent"))
```

```
## DataFrame with 3094 rows and 4 columns
##   low_lib_size  low_n_features high_subsets_Mito_percent
##   <outlier.filter> <outlier.filter> <outlier.filter>
## 1      FALSE        FALSE        FALSE
## 2      FALSE        FALSE        FALSE
## 3      TRUE         TRUE         TRUE
## 4      FALSE        FALSE        FALSE
## 5      FALSE        FALSE        FALSE
## ...
## 3090     ...        ...        ...
## 3091     FALSE        FALSE        FALSE
## 3092     FALSE        FALSE        FALSE
## 3093     FALSE         TRUE         TRUE
## 3094     FALSE        FALSE        FALSE
##   discard
##   <logical>
## 1      FALSE
## 2      FALSE
## 3      TRUE
## 4      FALSE
## 5      FALSE
## ...
## 3090     ...
## 3091     FALSE
## 3092     FALSE
## 3093      TRUE
## 3094      FALSE
```

All three filter steps at once

```
cell_qc_results <- quickPerCellQC(colData(sce), percent_subsets=c("subsets_Mito_percent"))
```



Filter the Single Cell Object

- Filter cells according to QC metrics.

```
sce.Filtered <- sce[, !cell_qc_results$discard]  
sce.Filtered
```

```
## class: SingleCellExperiment  
## dim: 19938 2730  
## metadata(1): Samples  
## assays(1): counts  
## rownames(19938): ENSG00000175756 ENSG00000221978 ...  
##   ENSG00000169962 ENSG00000107404  
## rowData names(4): ID Symbol Type Chromosome  
## colnames: NULL  
## colData names(8): Sample Barcode ... subsets_Mito_percent total  
## reducedDimNames(0):  
## mainExpName: NULL  
## altExpNames(0):
```