BestPracticesSTBiocBook

Table of contents

Welcome

Package: BestPracticesSTBiocBook **Authors:** First Last [aut, cre] **Compiled:** 2024-09-26 **Package version:** 0.98.0 **R version:** R version 4.4.1 (2024-06-14) **BioC version:** 3.20 **License:** MIT + file LICENSE

This is the website for the online book **'Best Practices for Spatial Transcriptomics Analysis with Bioconductor'**.

This book provides discussion and interactive examples on best practices for computational analysis workflows for spatial transcriptomics data, using the Bioconductor framework within R. The chapters contain details on individual analysis steps as well as complete example workflows, with interactive example datasets and R code.

The book is organized into several parts, including introductory materials, analysis steps, and example workflows.

Additional details on analysis workflows for non-spatial single-cell data as well as further introductory materials on R and Bioconductor can be found in the related book Orchestrating Single-Cell Analysis with Bioconductor (OSCA).

Docker image

A `Docker` image built from this repository is available here:

ghcr.io/lmweber/bestpracticesstbiocbook

Get started now

You can get access to all the packages used in this book in < 1 minute, using this command in a terminal:

Listing 0.1 `bash`

```bash
docker run -it ghcr.io/lmweber/bestpracticesstbiocbook:devel R
```

RStudio Server

An RStudio Server instance can be initiated from the `Docker` image as follows:

Listing 0.2 `bash`

```bash
docker run \
    --volume <local_folder>:<destination_folder> \
    -e PASSWORD=OHCA \
    -p 8787:8787 \
    ghcr.io/lmweber/bestpracticesstbiocbook:devel
```

The initiated RStudio Server instance will be available at https://localhost:8787.

Session info

ℹ Click to expand

Part I

Introduction

1 Introduction

1.1 Overview

Bioconductor

1.2 Contents

-
-
-
-

1.3 Scope and who this book is for

Visium Data Preprocessing

1.4 Bioconductor

Bioconductor

1.5 Additional resources

- Orchestrating Single-Cell Analysis with Bioconductor (OSCA)
- R for Data Science
- Data Carpentry Software Carpentry

- detailed guide YouTube videos

- Visium Data Preprocessing

1.6 Contributions

GitHub issues

References

2 Spatial transcriptomics

2.1 Overview

Method of the Year 2020

2.2 Sequencing-based platforms

2.2.1 10x Genomics Visium

10x Genomics Visium

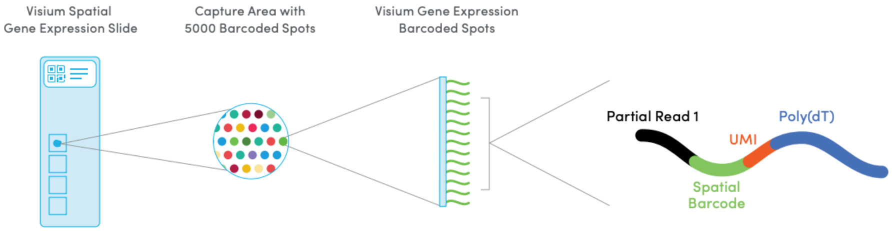


Figure 2.1: Schematic illustrating the 10x Genomics Visium platform. Source: 10x Genomics Visium

2.2.2 10x Genomics Visium HD

10x Genomics Visium HD

2.2.3 Curio Seeker

Curio Seeker

2.3 Molecule-based platforms

2.3.1 10x Genomics Xenium

10x Genomics

2.3.2 Vizgen MERSCOPE

Vizgen

2.3.3 NanoString CosMx

NanoString

References

3 Bioconductor data classes

3.1 Overview

3.2 SpatialExperiment class

SpatialExperiment

SingleCellExperiment

Bioconductor vignette

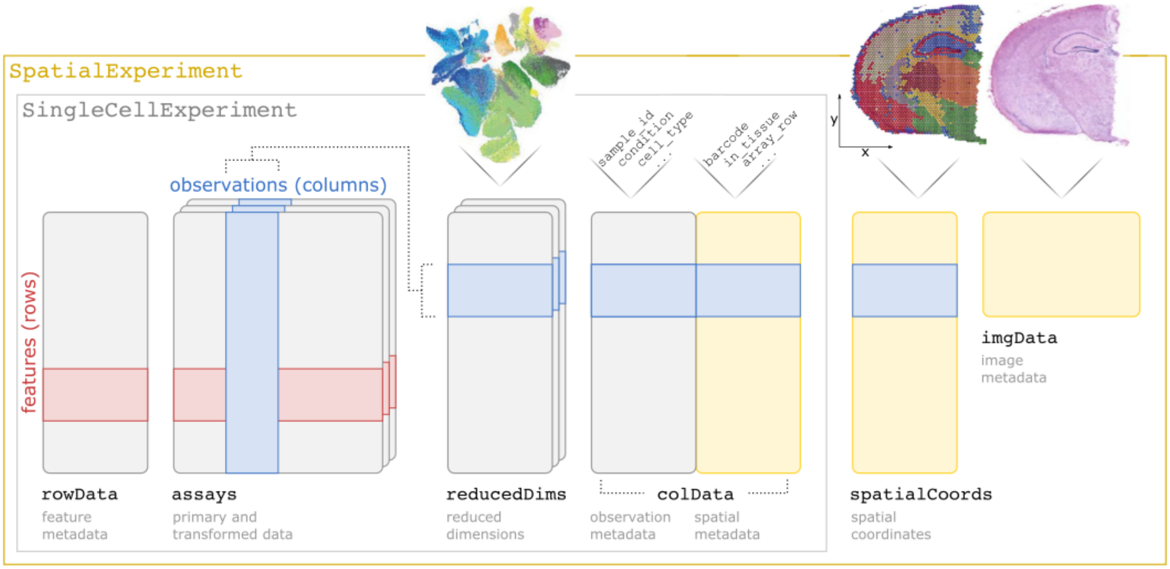


Figure 3.1: Overview of the `SpatialExperiment` data class for storing and manipulating spatial transcriptomics datasets within the Bioconductor framework.

3.3 Molecule-based data

3.3.1 MoleculeExperiment

Bioconductor package

3.3.2 SpatialFeatureExperiment

Bioconductor package

References

Part II

Analysis steps

4 Analysis steps

3

4.1 Save data objects for re-use in later chapters

4.1.1 Human DLPFC dataset

```
# LOAD DATA

library(SpatialExperiment)
library(STexampleData)
spe <- Visium_humanDLPFC()

# save object
library(here)
# if (!dir.exists(here("outputs"))) dir.create(here("outputs"))
# saveRDS(spe, file = here("outputs/spe_load.rds"))
saveRDS(spe, file = "spe_load.rds")
```

```
# QUALITY CONTROL (QC)

library(scater)
# subset to keep only spots over tissue
spe <- spe[, colData(spe)$in_tissue == 1]
# identify mitochondrial genes
is_mito <- grepl("(^MT-)|(^mt-)", rowData(spe)$gene_name)
# calculate per-spot QC metrics
spe <- addPerCellQC(spe, subsets = list(mito = is_mito))
# select QC thresholds
qc_lib_size <- colData(spe)$sum < 600
qc_detected <- colData(spe)$detected < 400
qc_mito <- colData(spe)$subsets_mito_percent > 28
qc_cell_count <- colData(spe)$cell_count > 10
# combined set of discarded spots
discard <- qc_lib_size | qc_detected | qc_mito | qc_cell_count
colData(spe)$discard <- discard
# filter low-quality spots
spe <- spe[, !colData(spe)$discard]

# save object
# saveRDS(spe, file = here("outputs/spe_qc.rds"))
saveRDS(spe, file = "spe_qc.rds")
```

```
# NORMALIZATION

library(scran)
# calculate logcounts using library size factors
spe <- logNormCounts(spe)

# save object
# saveRDS(spe, file = here("outputs/spe_logcounts.rds"))
saveRDS(spe, file = "spe_logcounts.rds")
```

```
# FEATURE SELECTION

# remove mitochondrial genes
spe <- spe[!is_mito, ]
# fit mean-variance relationship
dec <- modelGeneVar(spe)
# select top HVGs
top_hvgs <- getTopHVGs(dec, prop = 0.1)
```

```
# save object
# saveRDS(spe, file = here("outputs/spe_hvgs.rds"))
# saveRDS(top_hvgs, file = here("outputs/top_hvgs.rds"))
saveRDS(spe, file = "spe_hvgs.rds")
saveRDS(top_hvgs, file = "top_hvgs.rds")
```

```
# DIMENSIONALITY REDUCTION

# compute PCA
set.seed(123)
spe <- runPCA(spe, subset_row = top_hvgs)
# compute UMAP on top 50 PCs
set.seed(123)
spe <- runUMAP(spe, dimred = "PCA")
# update column names
colnames(reducedDim(spe, "UMAP")) <- paste0("UMAP", 1:2)

# save object
# saveRDS(spe, file = here("outputs/spe_reduceddims.rds"))
saveRDS(spe, file = "spe_reduceddims.rds")
```

```
# CLUSTERING

# graph-based clustering
set.seed(123)
k <- 10
g <- buildSNNGraph(spe, k = k, use.dimred = "PCA")
g_walk <- igraph::cluster_walktrap(g)
clus <- g_walk$membership
colLabels(spe) <- factor(clus)

# save object
# saveRDS(spe, file = here("outputs/spe_cluster.rds"))
saveRDS(spe, file = "spe_cluster.rds")
```

References

5 Load data

5.1 Overview

3

Visium Data Preprocessing

STexampleData

5.2 Dataset

5.3 Load data

STexampleData

```
library(SpatialExperiment)
library(STexampleData)

# load object
spe <- Visium_humanDLPFC()
```

5.4 SpatialExperiment object

3

```
# check object
spe
## class: SpatialExperiment
## dim: 33538 4992
## metadata(0):
## assays(1): counts
## rownames(33538): ENSG00000243485 ENSG00000237613 ... ENSG00000277475
##   ENSG00000268674
## rowData names(3): gene_id gene_name feature_type
## colnames(4992): AAACAACGAATAGTTC-1 AAACAAGTATCTCCCA-1 ...
##   TTGTTTGTATTACACG-1 TTGTTTGTGTAAATTC-1
## colData names(8): barcode_id sample_id ... reference cell_count
## reducedDimNames(0):
## mainExpName: NULL
## altExpNames(0):
## spatialCoords names(2) : pxl_col_in_fullres pxl_row_in_fullres
## imgData names(4): sample_id image_id data scaleFactor

# number of genes (rows) and spots (columns)
dim(spe)
## [1] 33538  4992

# names of 'assays'
assayNames(spe)
## [1] "counts"

# row (gene) data
head(rowData(spe))
## DataFrame with 6 rows and 3 columns
##                           gene_id   gene_name    feature_type
##                       <character> <character>     <character>
##  ENSG00000243485 ENSG00000243485 MIR1302-2HG Gene Expression
##  ENSG00000237613 ENSG00000237613     FAM138A Gene Expression
##  ENSG00000186092 ENSG00000186092       OR4F5 Gene Expression
##  ENSG00000238009 ENSG00000238009  AL627309.1 Gene Expression
##  ENSG00000239945 ENSG00000239945  AL627309.3 Gene Expression
##  ENSG00000239906 ENSG00000239906  AL627309.2 Gene Expression

# column (spot) data
head(colData(spe))
##  DataFrame with 6 rows and 8 columns
##                              barcode_id     sample_id in_tissue array_row
##                             <character>   <character> <integer> <integer>
##  AAACAACGAATAGTTC-1 AAACAACGAATAGTTC-1 sample_151673         0         0
##  AAACAAGTATCTCCCA-1 AAACAAGTATCTCCCA-1 sample_151673         1        50
##  AAACAATCTACTAGCA-1 AAACAATCTACTAGCA-1 sample_151673         1         3
##  AAACACCAATAACTGC-1 AAACACCAATAACTGC-1 sample_151673         1        59
##  AAACAGAGCGACTCCT-1 AAACAGAGCGACTCCT-1 sample_151673         1        14
##  AAACAGCTTTCAGAAG-1 AAACAGCTTTCAGAAG-1 sample_151673         1        43
##                     array_col ground_truth   reference cell_count
##                     <integer>  <character> <character>  <integer>
##  AAACAACGAATAGTTC-1        16           NA          NA         NA
##  AAACAAGTATCTCCCA-1       102       Layer3      Layer3          6
##  AAACAATCTACTAGCA-1        43       Layer1      Layer1         16
##  AAACACCAATAACTGC-1        19           WM          WM          5
##  AAACAGAGCGACTCCT-1        94       Layer3      Layer3          2
##  AAACAGCTTTCAGAAG-1         9       Layer5      Layer5          4

# spatial coordinates
head(spatialCoords(spe))
##                     pxl_col_in_fullres pxl_row_in_fullres
##  AAACAACGAATAGTTC-1               3913               2435
##  AAACAAGTATCTCCCA-1               9791               8468
##  AAACAATCTACTAGCA-1               5769               2807
##  AAACACCAATAACTGC-1               4068               9505
##  AAACAGAGCGACTCCT-1               9271               4151
##  AAACAGCTTTCAGAAG-1               3393               7583

# image data
imgData(spe)
##  DataFrame with 2 rows and 4 columns
##        sample_id    image_id   data scaleFactor
##      <character> <character> <list>   <numeric>
##   1 sample_151673      lowres   ####   0.0450045
##   2 sample_151673       hires   ####   0.1500150
```

5.5 Build object

SpatialExperiment

```
# create data
n_genes <- 200
n_spots <- 100

counts <- matrix(0, nrow = n_genes, ncol = n_spots)

row_data <- DataFrame(
  gene_name = paste0("gene", sprintf("%03d", seq_len(n_genes)))
)

col_data <- DataFrame(
  sample_id = rep("sample01", n_spots)
)

spatial_coords <- matrix(0, nrow = n_spots, ncol = 2)
colnames(spatial_coords) <- c("x", "y")

# create SpatialExperiment object
spe <- SpatialExperiment(
  assays = list(counts = counts),
  colData = col_data,
  rowData = row_data,
  spatialCoords = spatial_coords
)
```

5.6 Molecule-based data

3

References

6 Quality control

6.1 Overview

-
-
-
-

6.2 Load data from previous steps

4

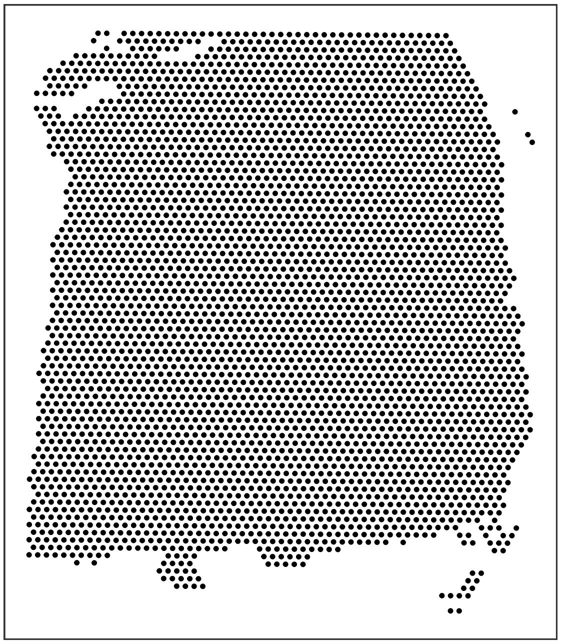
```
library(SpatialExperiment)
library(here)
# spe <- readRDS(here("outputs/spe_load.rds"))
spe <- readRDS("spe_load.rds")
```

6.3 Plot data

ggspavis

```
library(ggspavis)
```

```
# plot spatial coordinates (spots)
plotSpots(spe)
```



6.4 Calculate QC metrics

```
library(scater)
```

```
# subset to keep only spots over tissue
spe <- spe[, colData(spe)$in_tissue == 1]
dim(spe)
##  [1] 33538  3639
```

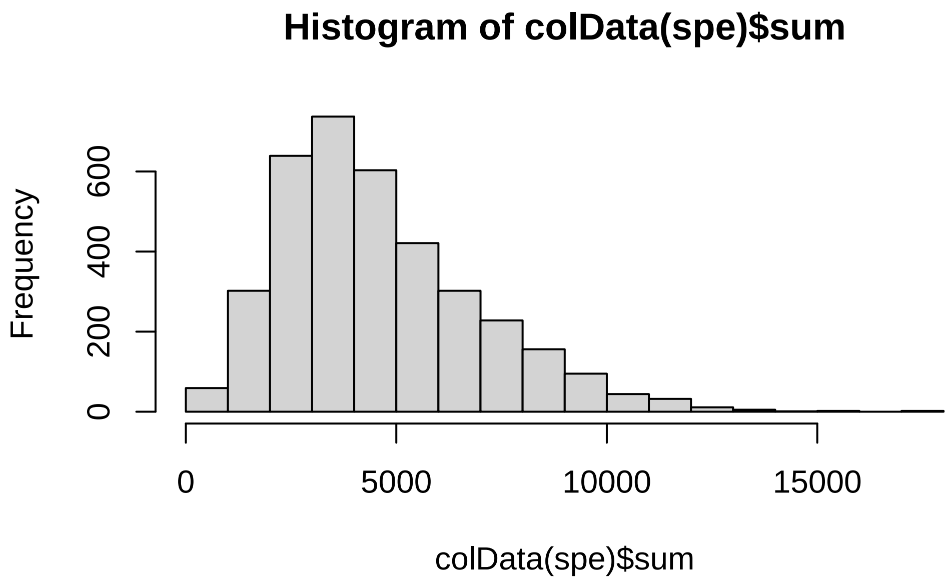
```
# identify mitochondrial genes
is_mito <- grepl("(^MT-)|(^mt-)", rowData(spe)$gene_name)
table(is_mito)
##  is_mito
##  FALSE  TRUE
##  33525    13
rowData(spe)$gene_name[is_mito]
##   [1] "MT-ND1"  "MT-ND2"  "MT-CO1"  "MT-CO2"  "MT-ATP8" "MT-ATP6" "MT-CO3"
##   [8] "MT-ND3"  "MT-ND4L" "MT-ND4"  "MT-ND5"  "MT-ND6"  "MT-CYB"

# calculate per-spot QC metrics and store in colData
spe <- addPerCellQC(spe, subsets = list(mito = is_mito))
head(colData(spe))
##  DataFrame with 6 rows and 14 columns
##                             barcode_id     sample_id in_tissue array_row
##                            <character>   <character> <integer> <integer>
##  AAACAAGTATCTCCCA-1 AAACAAGTATCTCCCA-1 sample_151673         1        50
##  AAACAATCTACTAGCA-1 AAACAATCTACTAGCA-1 sample_151673         1         3
##  AAACACCAATAACTGC-1 AAACACCAATAACTGC-1 sample_151673         1        59
##  AAACAGAGCGACTCCT-1 AAACAGAGCGACTCCT-1 sample_151673         1        14
##  AAACAGCTTTCAGAAG-1 AAACAGCTTTCAGAAG-1 sample_151673         1        43
##  AAACAGGGTCTATATT-1 AAACAGGGTCTATATT-1 sample_151673         1        47
##                     array_col ground_truth   reference cell_count       sum
##                     <integer>  <character> <character>  <integer> <numeric>
##  AAACAAGTATCTCCCA-1       102       Layer3      Layer3          6      8458
##  AAACAATCTACTAGCA-1        43       Layer1      Layer1         16      1667
##  AAACACCAATAACTGC-1        19           WM          WM          5      3769
##  AAACAGAGCGACTCCT-1        94       Layer3      Layer3          2      5433
##  AAACAGCTTTCAGAAG-1         9       Layer5      Layer5          4      4278
##  AAACAGGGTCTATATT-1        13       Layer6      Layer6          6      4004
##                      detected subsets_mito_sum subsets_mito_detected
##                     <numeric>        <numeric>             <numeric>
##  AAACAAGTATCTCCCA-1      3586             1407                    13
##  AAACAATCTACTAGCA-1      1150              204                    11
##  AAACACCAATAACTGC-1      1960              430                    13
##  AAACAGAGCGACTCCT-1      2424             1316                    13
##  AAACAGCTTTCAGAAG-1      2264              651                    12
##  AAACAGGGTCTATATT-1      2178              621                    13
##                     subsets_mito_percent     total
##                                <numeric> <numeric>
##  AAACAAGTATCTCCCA-1              16.6351      8458
##  AAACAATCTACTAGCA-1              12.2376      1667
##  AAACACCAATAACTGC-1              11.4089      3769
##  AAACAGAGCGACTCCT-1              24.2223      5433
##  AAACAGCTTTCAGAAG-1              15.2174      4278
##  AAACAGGGTCTATATT-1              15.5095      4004
```

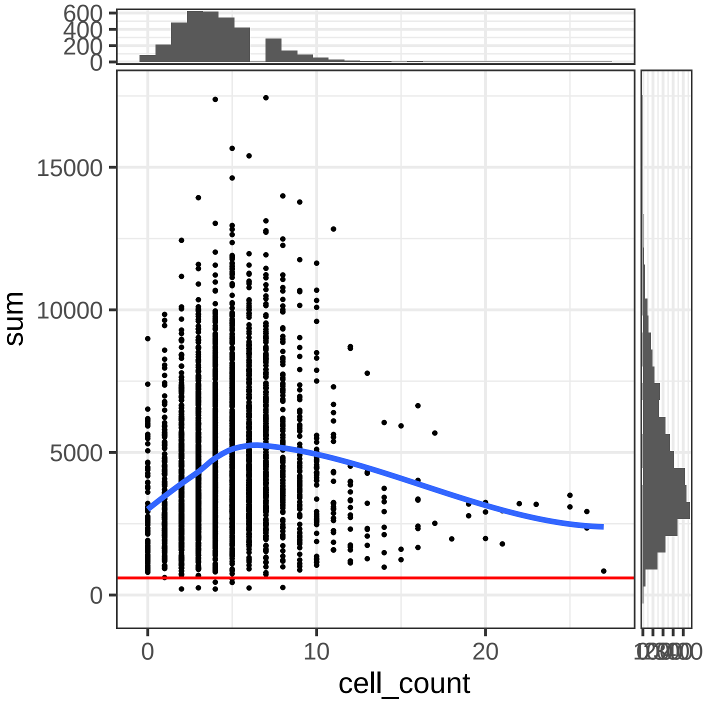
6.5 Selecting thresholds

6.5.1 Library size

```
# histogram of library sizes
hist(colData(spe)$sum, breaks = 20)
```



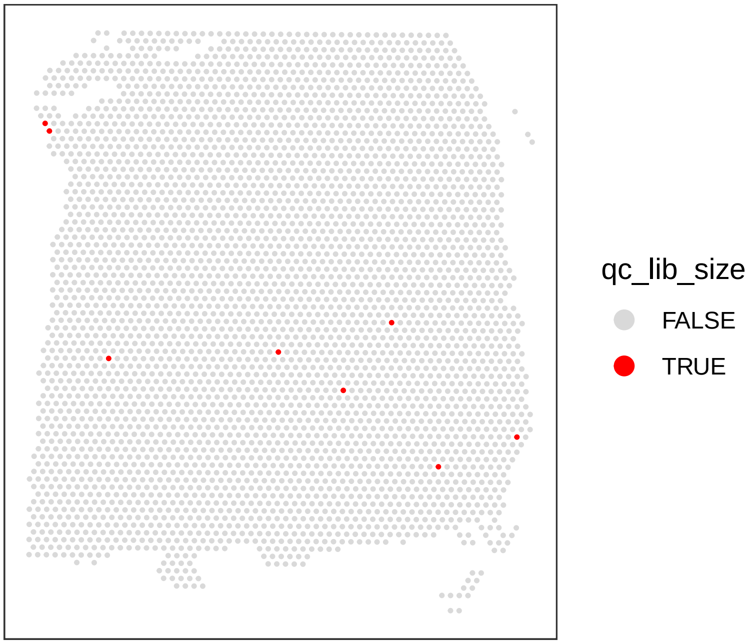
```
# plot library size vs. number of cells per spot
plotSpotQC(spe, plot_type = "scatter",
           x_metric = "cell_count", y_metric = "sum",
           y_threshold = 600)
## `geom_smooth()` using formula = 'y ~ x'
## `stat_xsidebin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_ysidebin()` using `bins = 30`. Pick better value with `binwidth`.
```



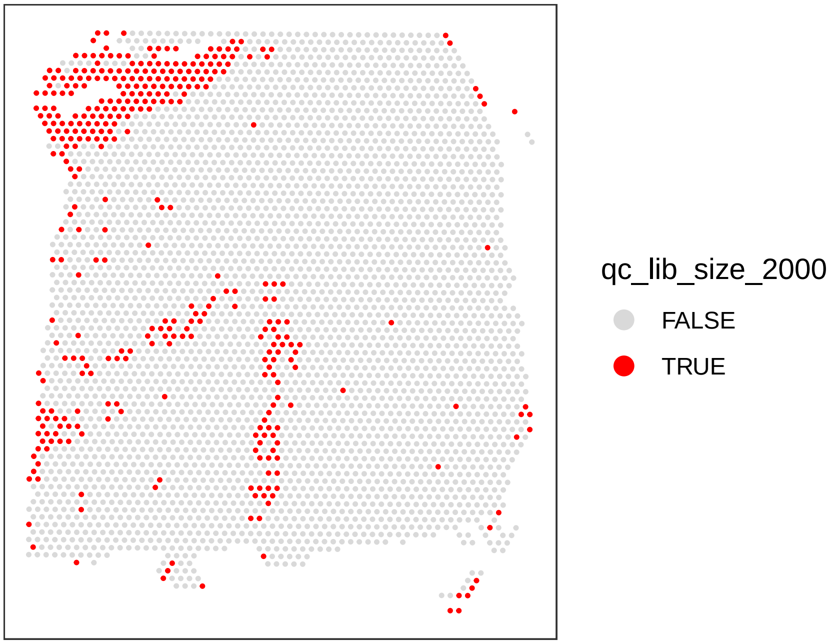
```
# select QC threshold for library size
qc_lib_size <- colData(spe)$sum < 600
table(qc_lib_size)
## qc_lib_size
## FALSE  TRUE
##  3631     8

colData(spe)$qc_lib_size <- qc_lib_size
```

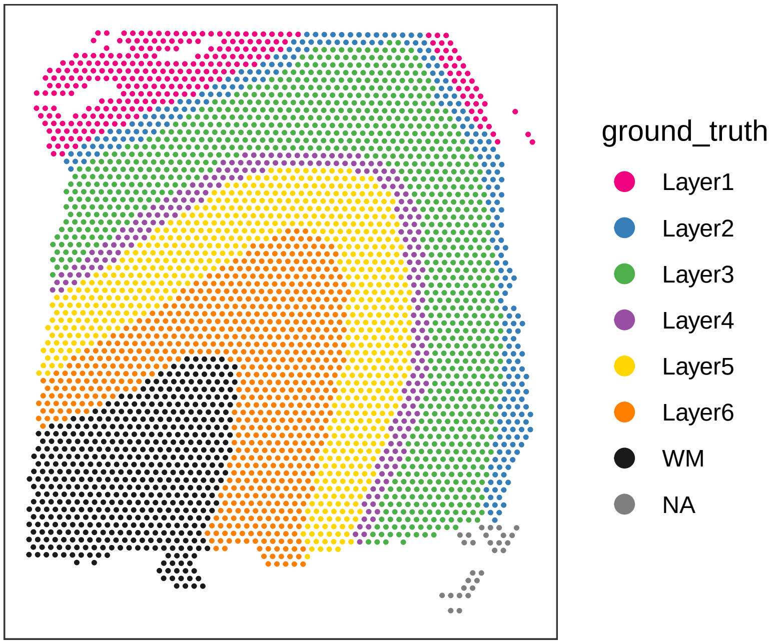
```
# check spatial pattern of discarded spots
plotSpotQC(spe, plot_type = "spot",
           annotate = "qc_lib_size")
```



```r
# check spatial pattern of discarded spots if threshold is too high
qc_lib_size_2000 <- colData(spe)$sum < 2000
colData(spe)$qc_lib_size_2000 <- qc_lib_size_2000
plotSpotQC(spe, plot_type = "spot",
           annotate = "qc_lib_size_2000")
```

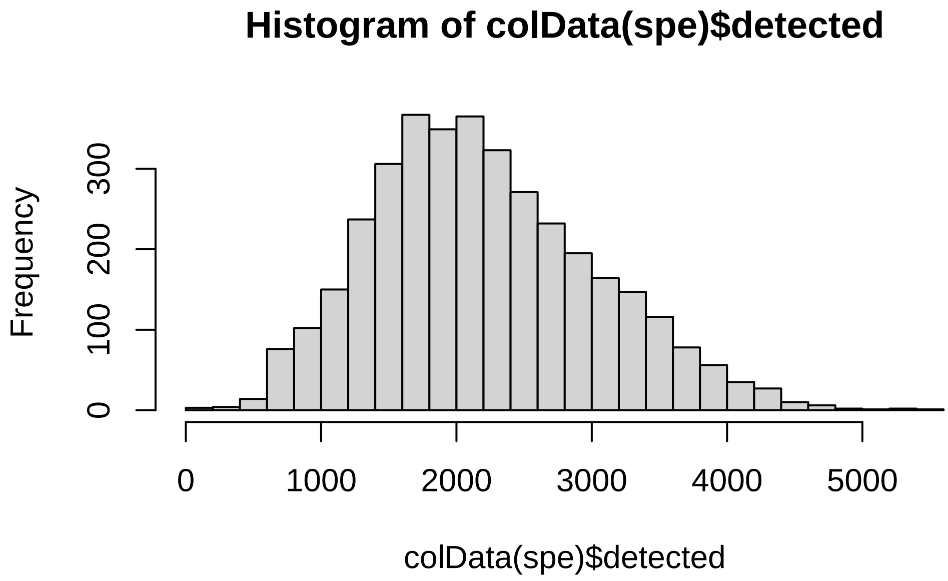


```
# plot ground truth (manually annotated) layers
plotSpots(spe, annotate = "ground_truth",
          pal = "libd_layer_colors")
```

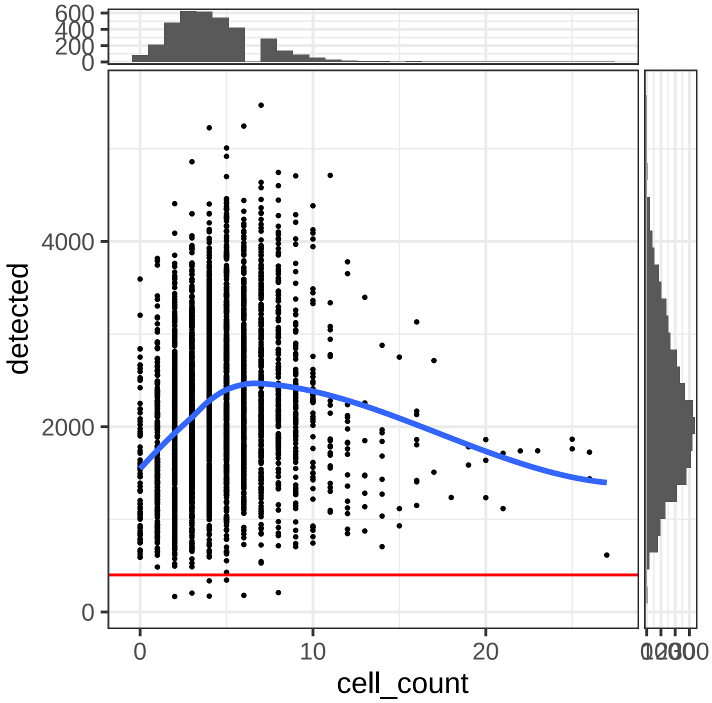


6.5.2 Number of expressed features

```r
# histogram of numbers of expressed genes
hist(colData(spe)$detected, breaks = 20)
```



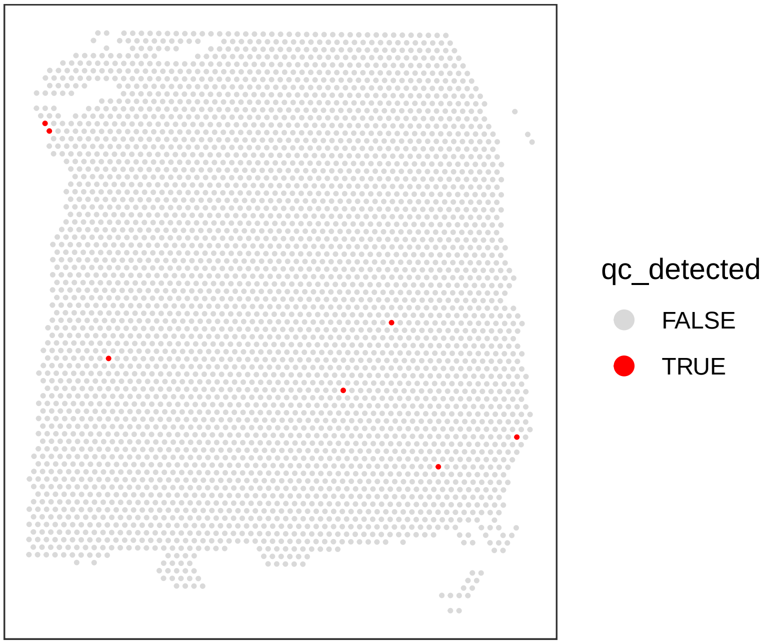
```
# plot number of expressed genes vs. number of cells per spot
plotSpotQC(spe, plot_type = "scatter",
           x_metric = "cell_count", y_metric = "detected",
           y_threshold = 400)
## `geom_smooth()` using formula = 'y ~ x'
## `stat_xsidebin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_ysidebin()` using `bins = 30`. Pick better value with `binwidth`.
```



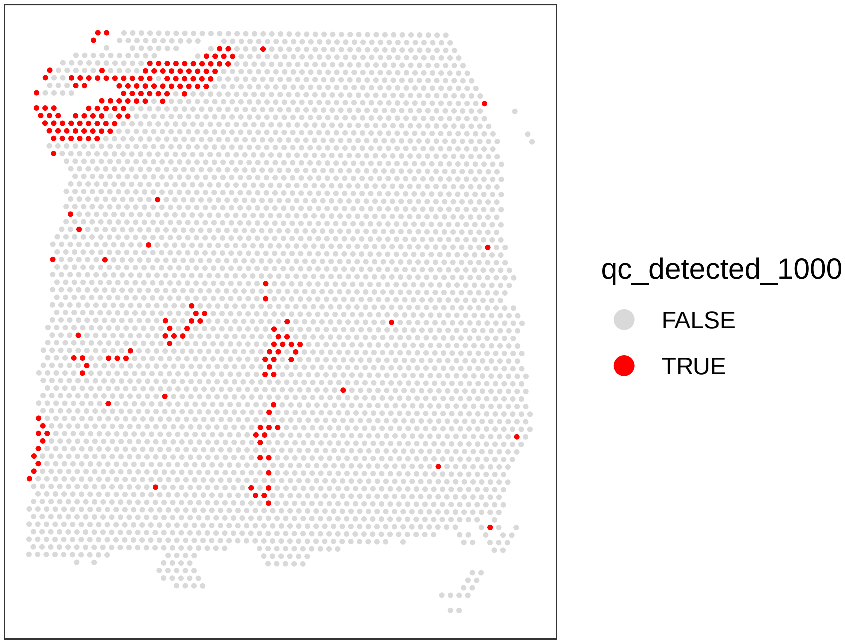
```
# select QC threshold for number of expressed genes
qc_detected <- colData(spe)$detected < 400
table(qc_detected)
## qc_detected
## FALSE  TRUE
##  3632     7

colData(spe)$qc_detected <- qc_detected
```

```
# check spatial pattern of discarded spots
plotSpotQC(spe, plot_type = "spot",
           annotate = "qc_detected")
```

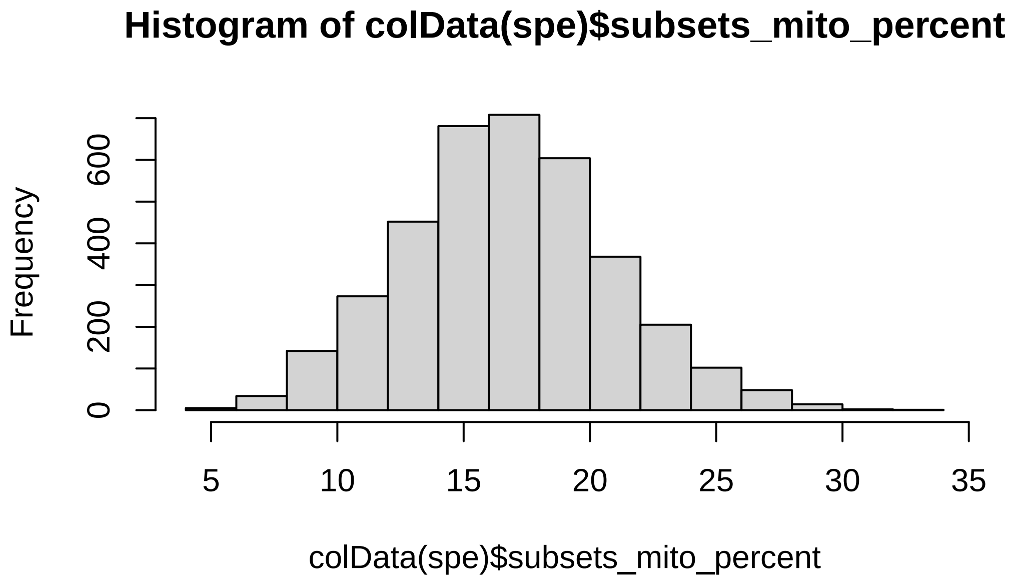


```r
# check spatial pattern of discarded spots if threshold is too high
qc_detected_1000 <- colData(spe)$detected < 1000
colData(spe)$qc_detected_1000 <- qc_detected_1000
plotSpotQC(spe, plot_type = "spot",
           annotate = "qc_detected_1000")
```

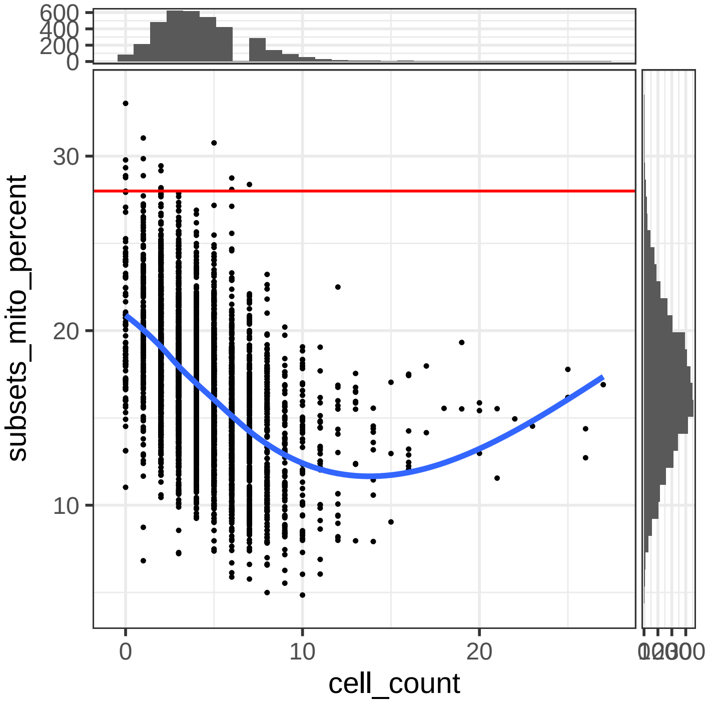


6.5.3 Proportion of mitochondrial reads

```
# histogram of mitochondrial read proportions
hist(colData(spe)$subsets_mito_percent, breaks = 20)
```



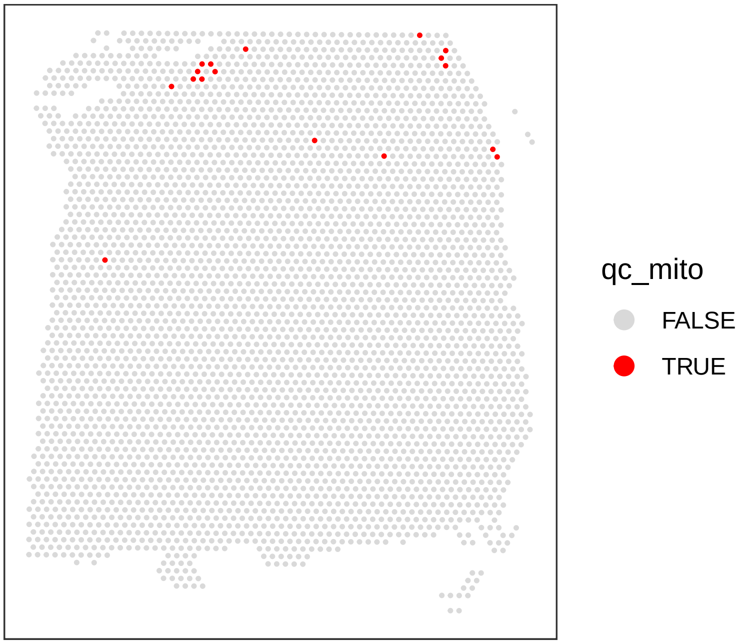
```
# plot mitochondrial read proportion vs. number of cells per spot
plotSpotQC(spe, plot_type = "scatter",
           x_metric = "cell_count", y_metric = "subsets_mito_percent",
           y_threshold = 28)
## `geom_smooth()` using formula = 'y ~ x'
## `stat_xsidebin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_ysidebin()` using `bins = 30`. Pick better value with `binwidth`.
```



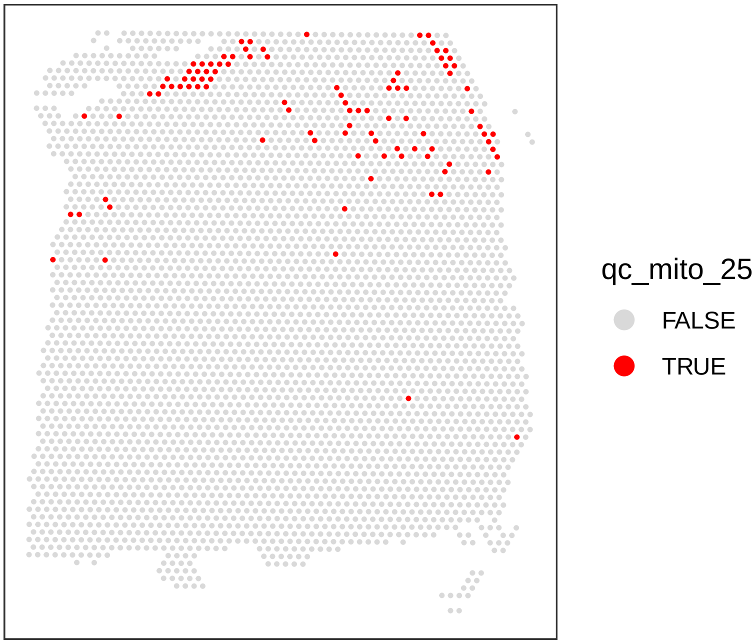
```
# select QC threshold for mitochondrial read proportion
qc_mito <- colData(spe)$subsets_mito_percent > 28
table(qc_mito)
## qc_mito
## FALSE  TRUE
##  3622    17

colData(spe)$qc_mito <- qc_mito
```

```
# check spatial pattern of discarded spots
plotSpotQC(spe, plot_type = "spot",
           annotate = "qc_mito")
```

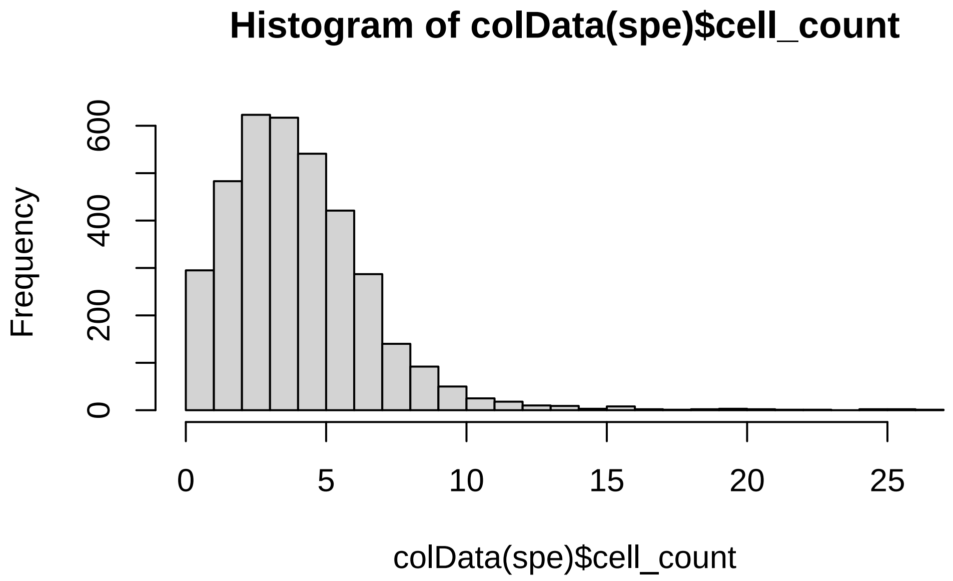


```
# check spatial pattern of discarded spots if threshold is too high
qc_mito_25 <- colData(spe)$subsets_mito_percent > 25
colData(spe)$qc_mito_25 <- qc_mito_25
plotSpotQC(spe, plot_type = "spot",
           annotate = "qc_mito_25")
```



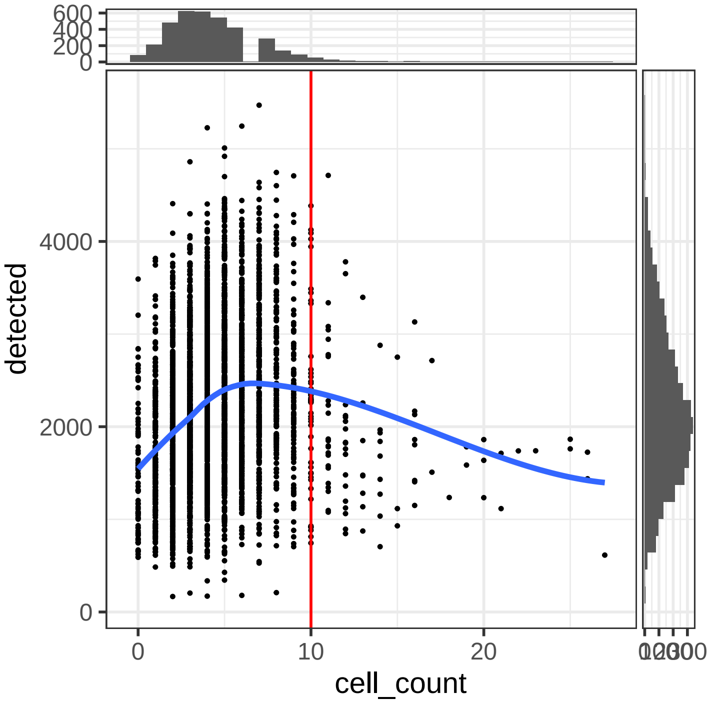
6.5.4 Number of cells per spot

```
# histogram of cell counts
hist(colData(spe)$cell_count, breaks = 20)
```



```
# distribution of cells per spot
tbl_cells_per_spot <- table(colData(spe)$cell_count)
```

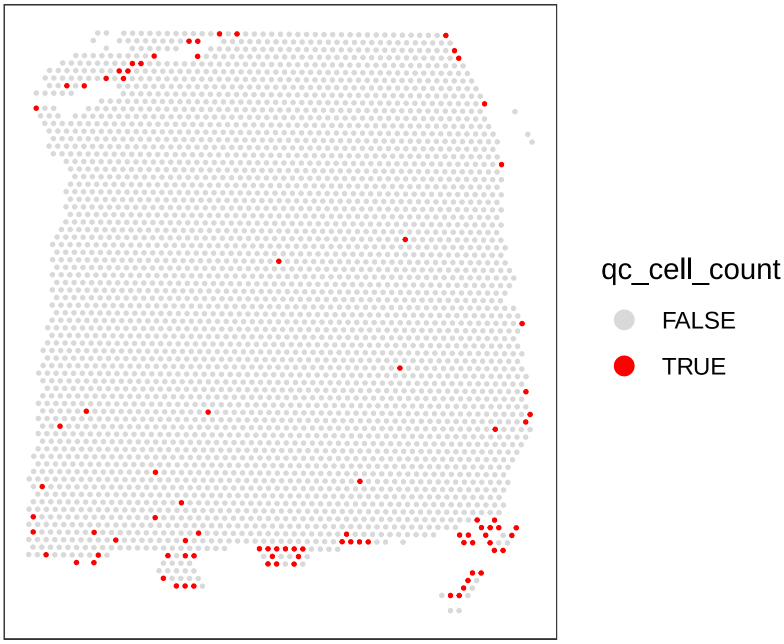
```
# plot number of expressed genes vs. number of cells per spot
plotSpotQC(spe, plot_type = "scatter",
           x_metric = "cell_count", y_metric = "detected",
           x_threshold = 10)
## `geom_smooth()` using formula = 'y ~ x'
## `stat_xsidebin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_ysidebin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
# select QC threshold for number of cells per spot
qc_cell_count <- colData(spe)$cell_count > 10
table(qc_cell_count)
## qc_cell_count
## FALSE  TRUE
##  3549    90

colData(spe)$qc_cell_count <- qc_cell_count
```

```
# check spatial pattern of discarded spots
plotSpotQC(spe, plot_type = "spot",
           annotate = "qc_cell_count")
```



6.5.5 Remove low-quality spots

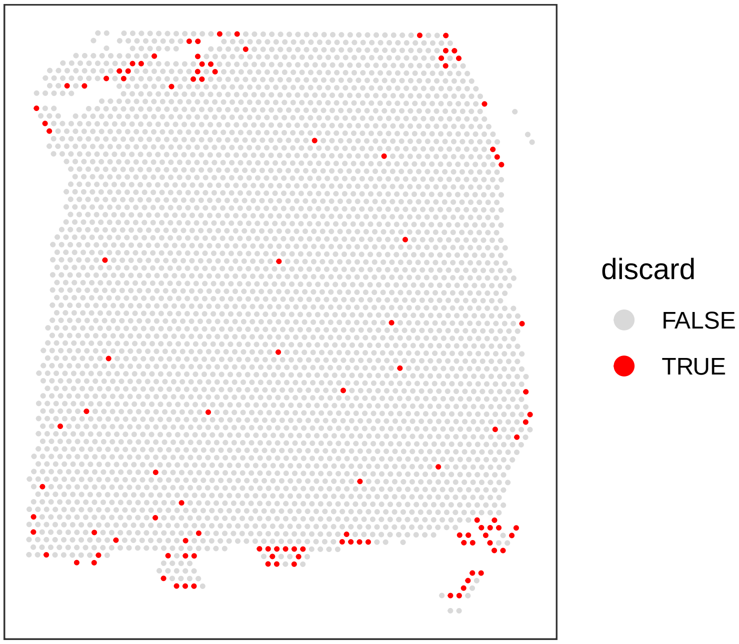
```
# number of discarded spots for each metric
apply(cbind(qc_lib_size, qc_detected, qc_mito, qc_cell_count), 2, sum)
##    qc_lib_size   qc_detected       qc_mito qc_cell_count 
##              8             7            17            90

# combined set of discarded spots
discard <- qc_lib_size | qc_detected | qc_mito | qc_cell_count
table(discard)
##  discard
```

```
## 
## FALSE  TRUE 
##  3524   115

# store in object
colData(spe)$discard <- discard
```

```
# check spatial pattern of combined set of discarded spots
plotSpotQC(spe, plot_type = "spot", 
           annotate = "discard")
```



```
# remove combined set of low-quality spots
spe <- spe[, !colData(spe)$discard]
dim(spe)
## [1] 33538  3524
```

6.6 Zero-cell and single-cell spots

```
# distribution of cells per spot
tbl_cells_per_spot[1:13]
##
##    0   1   2   3   4   5   6   7   8   9  10  11  12
##   84 211 483 623 617 541 421 287 140  92  50  25  18

# as proportions
prop_cells_per_spot <- round(tbl_cells_per_spot / sum(tbl_cells_per_spot), 2)
prop_cells_per_spot[1:13]
##
##    0    1    2    3    4    5    6    7    8    9   10   11   12
## 0.02 0.06 0.13 0.17 0.17 0.15 0.12 0.08 0.04 0.03 0.01 0.01 0.00
```

6.7 Quality control at gene level

References