

BestPracticesSTBiocBook

Table of contents

Welcome	4
Docker image	5
RStudio Server	6
Session info	7
1 Introduction	10
1.1 Overview	10
1.2 Contents	10
1.3 Scope and who this book is for	10
1.4 Bioconductor	11
1.5 Additional resources	11
1.6 Contributions	12
References	12
2 Spatial transcriptomics	13
2.1 Overview	13
2.2 Sequencing-based platforms	13
2.2.1 10x Genomics Visium	14
2.2.2 10x Genomics Visium HD	15
2.2.3 Curio Seeker	15
2.3 Molecule-based platforms	15
2.3.1 10x Genomics Xenium	15
2.3.2 Vizgen MERSCOPE	16
2.3.3 NanoString CosMx	16
References	16
3 Bioconductor data classes	17
3.1 Overview	17
3.2 SpatialExperiment class	17
3.3 Molecule-based data	18
3.3.1 MoleculeExperiment	18
3.3.2 SpatialFeatureExperiment	18
References	19

4	Analysis steps	20
4.1	Save data objects for re-use in later chapters	20
4.1.1	Human DLPFC dataset	20

Welcome

Package: BestPracticesSTBiocBook **Authors:** First Last [aut, cre] **Compiled:** 2024-09-25
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

```
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

```
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

 Click to expand

1 Introduction

1.1 Overview

[Bioconductor](#)

1.2 Contents

-
-
-
-

1.3 Scope and who this book is for

[Preprocessing](#)

[Visium Data](#)

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

of the Year 2020

Method

2.2 Sequencing-based platforms

2.2.1 10x Genomics Visium

10x Genomics Visium

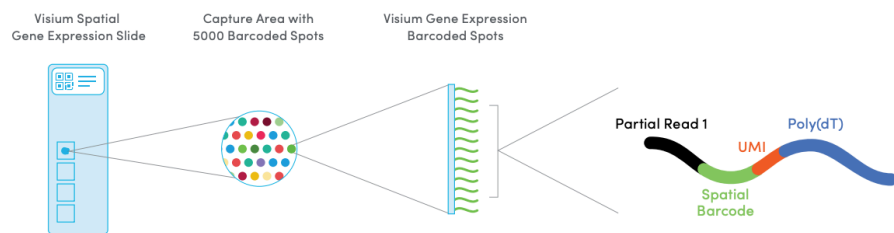


Figure 2.1: Schematic illustrating the 10x Genomics Visium platform. Source: [10x Genomics Visium](#)

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](#)

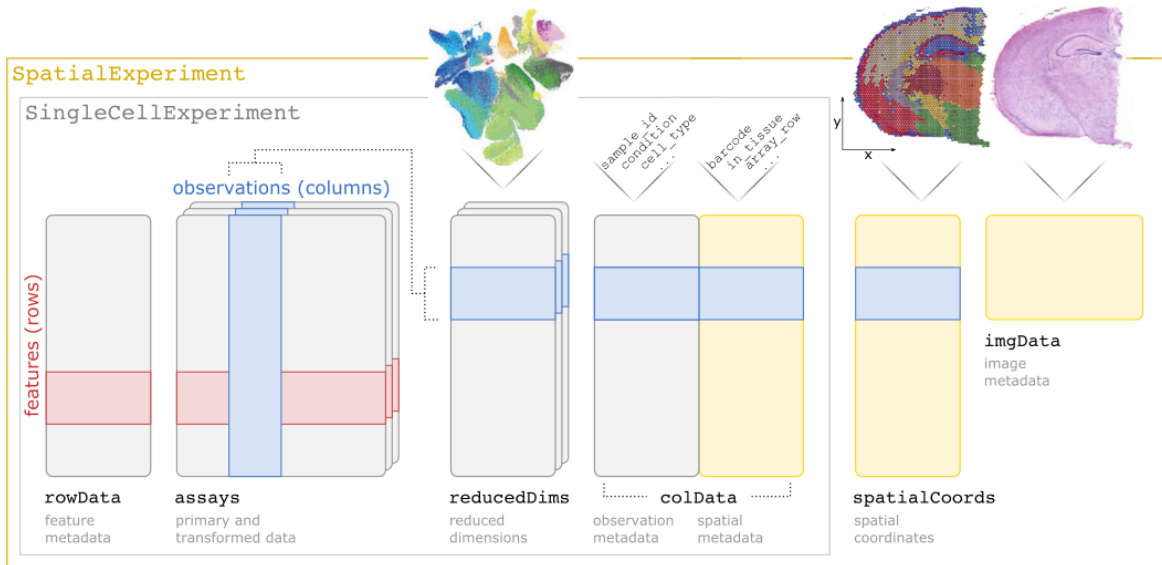


Figure 3.1: Overview of the `SpatialExperiment` data class for storing and manipulating spatial transcriptomics datasets within the Bioconductor framework.

3.3 Molecule-based data

3.3.1 MoleculeExperiment

[Bioconductor package](#)

3.3.2 SpatialFeatureExperiment

[Bioconductor package](#)

References

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 = here("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 = here("spe_qc.rds"))

```

```

# NORMALIZATION

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

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

```