BiocBook.tidyomics

Table of contents

Welcome

Package: BiocBook.tidyomics **Authors:** Jacques Serizay [aut, cre] **Compiled:** 2024-07-26 **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 landing page of the `BiocBook` entitled ….

This book introduces the reader to ….

Docker image

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

ghcr.io/js2264/biocbook.tidyomics

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/js2264/biocbook.tidyomics: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/js2264/biocbook.tidyomics:devel
```

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

Session info

Click to expand

Preamble

Part I

Fundamentals concepts

1 Manipulating genomic interval data

1.1 Importing GRanges from files

```
library(GenomicRanges)

library(rtracklayer)

bedf <- system.file('extdata', 'S288C-borders.bed', package = 'Bioc2024tidyWorkshop', mustWo

import(bedf)
```

```
library(tidyverse)

tib <- read_tsv(bedf, col_names = FALSE)

tib

library(plyranges)

gr <- as_granges(tib, seqnames = X1, start = X2, end = X3)

gr
```

tidy evaluation

1.2 Manipulating GRanges with tidy verbs

a number of tidy operations

-
-
-
-
-

```
gr |>
    mutate(score = runif(n())) |>
    filter(score > 0.2) |>
    mutate(round_score = round(score, digits = 1)) |>
    group_by(round_score) |>
    summarize(mean = mean(score))
```

```
gr |>
    mutate(
        seqnames = factor('XVI', levels(seqnames)),
        width = 1,
        strand = rep(c('-', '+'), n()/2)
    )
```

```
gr |>
    anchor_center() |>
    stretch(extend = -1000) |>
    shift_upstream(250) |>
    flank_upstream(100)
```

2 Manipulating genomic interaction data

2.1 What are GInteractions?

2.1.1 Creating a GInteractions object from scratch

```
library(InteractionSet)

gr1 <- GRanges("I:10-50")

gr2 <- GRanges("I:100-110")

GInteractions(anchor1 = gr1, anchor2 = gr2)
```

```
GInteractions(anchor1 = c(1, 2, 3), anchor2 = c(1, 4, 5), regions = gr)
```

2.1.2 Importing genomic interaction data from files

```r
bedpef <- system.file('extdata', 'S288C-loops.bedpe', package = 'Bioc2024tidyWorkshop', mustW

tib <- read_tsv(bedpef, col_names = FALSE)

tib

library(plyinteractions)

gi <- tib |>
    as_ginteractions(
        seqnames1 = X1, start1 = X2, end1 = X3,
        seqnames2 = X4, start2 = X5, end2 = X6
    )

gi
```

2.2 Manipulating `GInteractions` the tidy way

2.2.1 Moving anchors around

```r
gi |>
    mutate(
        seqnames1 = factor('XVI', levels(seqnames1)),
        strand1 = '+',
        start2 = end1,
        width2 = width1 + 100,
        score = runif(length(gi)),
        is_cis = ifelse(seqnames1 == seqnames2, TRUE, FALSE)
    )
```

2.2.2 Filtering interactions

```
gi |> filter(seqnames1 == 'I')

gi |> filter(seqnames2 == 'I')

gi |>
    mutate(score = runif(length(gi))) |>
    filter(seqnames2 == 'I', score > 0.2)
```

2.2.3 Overlapping anchors

```
centros <- system.file('extdata', 'col', package = 'Bioc2024tidyWorkshop', mustWork = TRUE)
    read_tsv() |>
    as_granges(seqnames = seqID) |>
    anchor_center() |>
    stretch(20000)

gi |>
    join_overlap_left(centros) |>
    filter(!is.na(patternName))
```

```
gi |>
    pin_anchors1() |>
    join_overlap_left(centros) |>
    filter(!is.na(patternName))

gi |>
    pin_anchors2() |>
    join_overlap_left(centros) |>
    filter(!is.na(patternName))
```

2.3 Real-world use case: computing a P(s)

2.3.1 Importing data from pairs file

```
pairsf <- system.file('extdata', 'mESCs.pairs.gz', package = 'Bioc2024tidyWorkshop', mustWorl

pairs <- read_tsv(pairsf, col_names = FALSE, comment = "#") |>
    set_names(c(
        "ID", "seqnames1", "start1", "seqnames2", "start2", "strand1", "strand2"
    )) |>
    as_ginteractions(end1 = start1, end2 = start2, keep.extra.columns = TRUE)
```

2.3.2 Counting interactions by strands

```
df <- pairs |>
    add_pairdist() |>
    filter(pairdist < 2000) |>
    group_by(strand1, strand2, pairdist) |>
    count()

ggplot(df, aes(x = pairdist, y = n, col = interaction(strand1, strand2))) +
    geom_smooth() +
    scale_y_log10()
```

2.3.3 Plot P(s)

```
x <- 1.1^(1:200-1)
lmc <- coef(lm(c(1,1161443398)~c(x[1], x[200])))
bins_breaks <- unique(round(lmc[2]*x + lmc[1]))
bins_widths <- lead(bins_breaks) - bins_breaks

# Bin distances
df <- pairs |>
```

```r
    add_pairdist(colname = 's') |>
    mutate(
        binned_s = bins_breaks[as.numeric(cut(s, bins_breaks))],
        bin_width = bins_widths[as.numeric(cut(s, bins_breaks))]
    ) |>
    group_by(binned_s, bin_width) |>
    count(name = "n") |>
    as_tibble() |>
    mutate(Ps = n / sum(n) / bin_width)

ggplot(df, aes(x = binned_s, y = Ps)) +
    geom_line() +
    scale_y_log10() +
    scale_x_log10() +
    annotation_logticks() +
    labs(x = "Genomic distance", y = "Contact frequency") +
    theme_bw()
```

Resources

Session info

Click to expand

3 Manipulating summarized experiment data

Resources

Session info

Click to expand

Part II

Specific omics

4 Manipulating transcriptomic data

Resources

Session info

Click to expand

5 Manipulating epigenomics data

5.1 Introduction to `tidyCoverage`

5.1.1 `CoverageExperiment` and `AggregatedCoverage` class

```
library(tidyCoverage)

data(ce)

data(ac)

ce

ac
```

5.1.2 Creating a `CoverageExperiment` object from tracks and features

```
tracks <- BigWigFileList(c(
    mnase = system.file("extdata", "MNase.bw", package = "tidyCoverage"),
    cohesin = system.file("extdata", "Scc1.bw", package = "tidyCoverage")
))
features <- GRangesList(
    TSSs = system.file("extdata", "TSSs.bed", package = "tidyCoverage") |> import() |> sampl
    TTSs = system.file("extdata", "TTSs.bed", package = "Bioc2024tidyWorkshop") |> import()
)
```

```r
ce2 <- CoverageExperiment(
    tracks = tracks,
    features = features,
    width = 2000,
    ignore.strand = FALSE
)

ce2
```

```r
colData(ce2)

rowData(ce2)

assay(ce2, 'coverage')

class(assay(ce2, 'coverage')['TSSs', 'mnase'])

class(assay(ce2, 'coverage')[['TSSs', 'mnase']])

dim(assay(ce2, 'coverage')[['TSSs', 'mnase']])
```

5.1.3 Tidy coverage? That's right!

```r
library(tidySummarizedExperiment)

ce2

ce2 |> filter(features == 'TSSs')

ce2 |> slice(2)

ce2 |> select(features, n)
```

5.1.4 `expand()` or `aggregate()`

```
tib <- expand(ce2)

tib
```

```
ac2 <- aggregate(ce2)

ac2
```

5.1.5 Visualizing aggregated coverage

-
-

```
CoverageExperiment(tracks, GRanges("II:1-100000"), window = 100) |>
    expand() |>
    ggplot() +
    geom_coverage() +
    facet_grid(track ~ features, scales = "free") +
    labs(x = 'chrV', y = 'Signal coverage')

ggplot(ac2) +
    geom_aggrcoverage() +
    facet_grid(track ~ features, scales = "free") +
    labs(x = 'Distance from genomic features', y = 'Signal coverage')
```

5.2 Real-world use case: studying epigenomic landscape of reulatory elements

5.2.1 Fetch coverage data from ENCODE

```
library(AnnotationHub)
ah <- AnnotationHub()
ids <- c('AH32207', 'AH35187')
names(ids) <- c('DNAse', 'H3K4me3')
bws <- lapply(ids, function(.x) ah[[.x]] |> resource()) |> BigWigFileList()
names(bws) <- names(ids)
```

5.2.2 Plotting coverage data over several loci

```
ce3 <- CoverageExperiment(
    bws,
    list(
        ccno = GRanges("chr5:55220001-55235000"),
        mcidas = GRanges("chr5:55159001-55174000")
    ),
    window = 50
)
expand(ce3) |>
    mutate(coverage = scales::oob_squish(coverage, c(0, 10))) |>
    ggplot() +
    geom_coverage(aes(fill = track), unit = 'Mb') +
    facet_grid(track~features, scales = 'free')
```

5.2.3 Import DNase peaks from ENCODE

```
features <- list(DNase = ah[['AH30077']] |> filter(zScore > 100) |> sample(1000))
```

5.2.4 Generating coverage aggregates and heatmaps over DNAse peak

```r
ce4 <- CoverageExperiment(bws, features, width = 2000, window = 10)
```

```r
aggregate(ce4) |>
    ggplot(aes(x = coord, y = mean)) +
    geom_aggrcoverage(aes(col = track)) +
    facet_wrap(~track) +
    labs(x = 'Distance from DNAse peak', y = 'Signal')
```

Resources

Session info

Click to expand

6 Manipulating single-cell data

Resources

Session info

Click to expand

7 Manipulating spatial single-cell data

Resources

Session info

Click to expand

8 Manipulating mass cytometry data

Resources

Session info

Click to expand

Part III

Additional resources

9 Helper packages

-
-
-

Resources

Session info

Click to expand

10 Future directions

-
-
-

Resources

Session info

Click to expand