BasicSTARRseq: a R/Bioconductor package for basic peak calling on STARR-seq data

Annika Bürger

November 24, 2026

1 Introduction

Transcriptional enhancers in genomic DNA sequences are important for gene regulation. The STARR-seq method was presented by [1] to identify enhancers on large scale. See also [2] for an discussion of that method.

The STARR-seq data consists of two sequencing datasets of the same targets in a specific genome. The *input* sequences show which regions where tested for enhancers. The *STARR-seq* sequences then show significant enriched peaks i.e. a lot more sequences in one region than in the input where enhancers in the genomic DNA are. So the approach pursued in [1] is to call *peak* every region in which there is a lot more (significant in a binomial model) STARR-seq signal than input signal and propose an enhancer at that very same position. Enhancers then are called weak or strong dependent of there degree of enrichment in comparison to input.

1.1 Loading the package

After installation, the package can be loaded into R by typing

```
> library(BasicSTARRseq)
```

into the R console.

1.2 Provided functionality

BasicSTARRseq requires the R-packages GenomicAlignments and GenomicRanges.

This package provides the following basic functions for the analysis of STARR-seq data:

- `getPeaks`: Performs basic peak calling on STARR-seq data based on a method introduced in [1].

In addition to the examples presented in this vignette, more detailed information on the functions' parameters are presented in the corresponding manual pages.

2 Data preparation and bioinformatics software

The main input format for BasicSTARRseq's main function is the binary alignment / map (BAM) format. Any alignment software can be used to map the raw experimental data to the corresponding reference genome. For example BWA or bowtie. For required filtering steps can then used SAMtools and BEDtools. For the analysis of STARR-seq data, two fragment libraries are required. The input library and the STARR-seq library. In comparison of these two the data gets meaningful.

3 Peak calling in STARR-seq data

3.1 Initialization of a STARRseqData object

There are two constructors available for creating a STARRseqData object. After alligning the sequencing data with for example bwa or bowtie one can specify the bam-file names and the type of experiment (i.e. paired-end or single-end), and load the data with the constructor function `STARRseqData`. Here a small paired-end example dataset[1]:

[1] small extraction of S2 *Drosophila Melanogaster* dataset published in [1], aligned with bowtie

```
> starrseqFileName <- system.file("extdata", "smallSTARR.bam",
+                                 package="BasicSTARRseq")
> inputFileName <- system.file("extdata", "smallInput.bam",
+                              package="BasicSTARRseq")
> STARRseqData(sample=starrseqFileName, control=inputFileName,
+              pairedEnd=TRUE)

STARRseqData object with 3019 STARR-seq fragments and 2612 input fragments
```

For single-end data, set the argument `pairedEnd` to `FALSE`:

```
> STARRseqData(sample=starrseqFileName, control=inputFileName,
+              pairedEnd=FALSE)

STARRseqData object with 6063 STARR-seq fragments and 5520 input fragments
```

If you have already loaded the aligned data into the `GRanges` format in R, a `STARRseqData` object can directly be created.

```
> starrseqGRanges <- granges(readGAlignmentPairs(starrseqFileName))
> inputGRanges <- granges(readGAlignmentPairs(inputFileName))
> data <- STARRseqData(sample=starrseqGRanges, control=inputGRanges)
> data

STARRseqData object with 3019 STARR-seq fragments and 2612 input fragments
```

3.2 Peak calling on STARRseqData object

Then if you have a valid `STARRseqData` object, it is possible to call peaks on the data via:

```
> peaks <- getPeaks(data)
> peaks

GRanges object with 3 ranges and 4 metadata columns:
      seqnames          ranges strand | sampleCov controlCov        pVal
         <Rle>       <IRanges>  <Rle> | <integer>  <numeric>   <numeric>
  [1]    chr2R 5836524-5837023      * |        14         32 1.60012e-08
  [2]    chr2R 5837035-5837534      * |        50         71 5.52777e-40
  [3]    chr2R 5842700-5843199      * |        16         81 4.74226e-05
      enrichment
       <numeric>
  [1]    2.83901
  [2]    7.06757
  [3]    1.53216
  -------
  seqinfo: 6 sequences from an unspecified genome
```

To further customize the peak calling, there are some arguments you can specify:

- `minQuantile` Which quantile of coverage height should be considered as peaks. Default value 0.9.
- `peakWidth` The width (in base pairs) that the peaks should have. Default value 500.
- `maxPval` The maximal p-value of peaks that is desired. Default value 0.001.
- `deduplicate` Wether the sequences should be deduplicated before calling peaks or not. Default value `TRUE`.
- `model` Which binomial model should be applied to calculate the p-values. Default value 1.

The default values are chosen corresponding to [1].

The peak calling works the following way: All genomic positions having a STARR-seq coverage over the quantile `minQuantile` are considered to be the center of a peak with width `peakWidth`. If then two ore more peaks overlap, the lower one is discarded. If then the binomial p-Value of the peak is higher than `maxPval` the peak is discarded as well.

The binomial model 1 for calculating the p-Value is:

- number of trials = total number of STARR-seq sequences,
- number of successes = STARR-seq coverage,
- estimated sucess probability in each trial = input coverage divided by total number of input sequences.

The binomial model 2 (this is used in [1]) for caculating the p-Value is:

- number of trials = STARR-seq coverage plus input coverage,
- number of successes = STARR-seq coverage,
- estimated success probability in each trial = total number of STARR-seq sequences divided by the sum of the total number of STARR-seq sequences and the total number of input sequences.

The `enrichment` of STARR-seq over input coverage is then calculated as follows:

$$\frac{\frac{\text{STARR-seq coverage of peak}}{\text{total number of STARR-seq sequences}}}{\frac{\text{input coverage of peak}}{\text{total number of input sequences}}}$$

The numinator and denuminator are corrected conservatively to the bounds of the 0.95 binomial confidence inverval corresponding to model 1.

References

1. *Genome-Wide Quantitative Enhancer Activity Maps Identified by STARR-seq* Arnold et al. Science. 2013 Mar 1;339(6123):1074-7. doi: 10.1126/science.1232542. Epub 2013 Jan 17.
2. *STARR-seq - Principles and Applications* Felix Muerdter, Lukasz M. Boryn, Cosmas D. Arnold. 2015 Sep;106(3):145-50. doi: 10.1016/j.ygeno.2015.06.001. Epub 2015 Jun 11.

4 Session Information

```
R version 4.6.1 (2026-06-24)
Platform: x86_64-pc-linux-gnu
Running under: Ubuntu 24.04.5 LTS

Matrix products: default
BLAS:   /home/biocbuild/bbs-3.24-bioc/R/lib/libRblas.so
LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.12.0  LAPACK version 3.12.0

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=en_GB              LC_COLLATE=C
 [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

time zone: America/New_York
tzcode source: system (glibc)

attached base packages:
[1] stats4    stats     graphics  grDevices utils     datasets  methods
[8] base

other attached packages:
 [1] BasicSTARRseq_1.42.0     GenomicAlignments_1.50.0 Rsamtools_2.30.0
 [4] Biostrings_2.82.0        XVector_0.54.0           GenomicRanges_1.66.0
 [7] Seqinfo_1.4.0            IRanges_2.48.0           S4Vectors_0.52.0
```

```
[10] BiocGenerics_0.60.0      generics_0.1.4

loaded via a namespace (and not attached):
 [1] codetools_0.2-20    xfun_0.62           parallel_4.6.1
 [4] knitr_1.52          bitops_1.1-0        cli_3.6.6
 [7] compiler_4.6.1      tools_4.6.1         evaluate_1.0.5
[10] otel_0.3.0          rlang_1.3.0         crayon_1.5.3
[13] cigarillo_1.4.0     BiocParallel_1.48.0
```