Package 'derfinderHelper'

November 11, 2026

Type Package

Title derfinder helper package

Version 1.46.0

Date 2021-08-05

Depends R(>= 3.2.2)

Imports IRanges (>= 1.99.27), Matrix, methods, S4Vectors (>= 0.2.2)

Suggests sessioninfo, knitr (>= 1.6), BiocStyle (>= 2.5.19), RefManageR, rmarkdown (>= 0.3.3), testthat, covr

VignetteBuilder knitr

Description Helper package for speeding up the derfinder package when using multiple cores. This package is particularly useful when using BiocParallel and it helps reduce the time spent loading the full derfinder package when running the F-statistics calculation in parallel.

License Artistic-2.0

LazyData false

URL https://github.com/leekgroup/derfinderHelper

BugReports https://support.bioconductor.org/t/derfinderHelper

biocViews DifferentialExpression, Sequencing, RNASeq, Software, ImmunoOncology

RoxygenNote 7.2.3

Encoding UTF-8

Roxygen list(markdown = TRUE)

git_url https://git.bioconductor.org/packages/derfinderHelper

git_branch RELEASE_3_23

git_last_commit 0595ac3

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Leonardo Collado-Torres [aut, cre] (ORCID:
<https://orcid.org/0000-0003-2140-308X>),
Andrew E. Jaffe [aut] (ORCID: <https://orcid.org/0000-0001-6886-1454>),
Jeffrey T. Leek [aut, ths] (ORCID:
<https://orcid.org/0000-0002-2873-2671>)

Maintainer Leonardo Collado-Torres <lcolladotor@gmail.com>

Contents

derfinderHelper-package 2
fstats.apply 3

Index **6**

`derfinderHelper-package`
derfinderHelper: derfinder helper package

Description

Helper package for speeding up the derfinder package when using multiple cores. This package is particularly useful when using BiocParallel and it helps reduce the time spent loading the full derfinder package when running the F-statistics calculation in parallel.

Author(s)

Maintainer: Leonardo Collado-Torres <lcolladotor@gmail.com> (ORCID)

Authors:

- Andrew E. Jaffe <andrew.jaffe@libd.org> (ORCID)
- Jeffrey T. Leek <jtleek@gmail.com> (ORCID) [thesis advisor]

See Also

Useful links:

- https://github.com/leekgroup/derfinderHelper
- Report bugs at https://support.bioconductor.org/t/derfinderHelper

fstats.apply *Calculate F-statistics per base by extracting chunks from a DataFrame*

Description

Extract chunks from a DataFrame and get the F-statistics on the rows of `data`, comparing the models `mod` (alternative) and `mod0` (null).

Usage

```
fstats.apply(
  index = Rle(TRUE, nrow(data)),
  data,
  mod,
  mod0,
  adjustF = 0,
  lowMemDir = NULL,
  method = "Matrix",
  scalefac = 32
)
```

Arguments

`index`	An index (logical Rle is the best for saving memory) indicating which rows of the DataFrame to use.
`data`	The DataFrame containing the coverage information. Normally stored in `coveragePrep$coverageProce` from `derfinder::preprocessCoverage`. Could also be the full data from `derfinder::loadCoverage`.
`mod`	The design matrix for the alternative model. Should be m by p where p is the number of covariates (normally also including the intercept).
`mod0`	The design matrix for the null model. Should be m by p_0.
`adjustF`	A single value to adjust that is added in the denominator of the F-stat calculation. Useful when the Residual Sum of Squares of the alternative model is very small.
`lowMemDir`	The directory where the processed chunks are saved when using `derfinder::preprocessCoverage` with a specified `lowMemDir`.
`method`	Has to be either 'Matrix' (default), 'Rle' or 'regular'. See details.
`scalefac`	The scaling factor used in `derfinder::preprocessCoverage`. It is only used when `method='Matrix'`.

Details

If `lowMemDir` is specified then `index` is expected to specify the chunk number.

fstats.apply has three different implemenations which are controlled by the `method` parameter. `method='regular'` coerces the data to a standard 'matrix' object. `method='Matrix'` coerces the data to a sparseMatrix which reduces the required memory. This method is only usable when the

projection matrices have row sums equal to 0. Note that these row sums are not exactly 0 due to how the computer works, thus leading to very small numerical differences in the F-statistics calculated versus `method='regular'`. Finally, `method='Rle'` calculates the F-statistics using the Rle compressed data without coercing it to other types of objects, thus using less memory that the other methods. However, it's speed is affected by the number of samples (n) as the current implementation requires n (n + 1) operations, so it's only recommended for small data sets. `method='Rle'` does result in small numerical differences versus `method='regular'`.

Overall `method='Matrix'` is faster than the other options and requires less memory than `method='regular'`. With tiny example data sets, `method='Matrix'` can be slower than `method='regular'` because the coercion step is slower.

In derfinder versions <= 0.0.62, `method='regular'` was the only option available.

Value

A numeric Rle with the F-statistics per base for the chunk in question.

Author(s)

Leonardo Collado-Torres, Jeff Leek

Examples

```
## Create some toy data
library("IRanges")
toyData <- DataFrame(
    "sample1" = Rle(sample(0:10, 1000, TRUE)),
    "sample2" = Rle(sample(0:10, 1000, TRUE)),
    "sample3" = Rle(sample(0:10, 1000, TRUE)),
    "sample4" = Rle(sample(0:10, 1000, TRUE))
)

## Create the model matrices
group <- c("A", "A", "B", "B")
mod.toy <- model.matrix(~group)
mod0.toy <- model.matrix(~ 0 + rep(1, 4))

## Get the F-statistics
fstats <- fstats.apply(
    data = toyData, mod = mod.toy, mod0 = mod0.toy,
    scalefac = 1
)


## Example with data from derfinder package
## Not run:
## Load the data
library("derfinder")

## Create the model matrices
mod <- model.matrix(~ genomeInfo$pop)
mod0 <- model.matrix(~ 0 + rep(1, nrow(genomeInfo)))
```

```
## Run the function
system.time(fstats.Matrix <- fstats.apply(
    data = genomeData$coverage, mod = mod,
    mod0 = mod0, method = "Matrix", scalefac = 1
))
fstats.Matrix

## Compare methods
system.time(fstats.regular <- fstats.apply(
    data = genomeData$coverage,
    mod = mod, mod0 = mod0, method = "regular"
))
system.time(fstats.Rle <- fstats.apply(
    data = genomeData$coverage, mod = mod,
    mod0 = mod0, method = "Rle"
))

## Small numerical differences can occur
summary(fstats.regular - fstats.Matrix)
summary(fstats.regular - fstats.Rle)

## You can make the effect negligible by appropriately rounding
## findRegions(cutoff) so the DERs will be the same regardless of the method
## used.

## Extra comparison, although the method to compare against is 'regular'
summary(fstats.Rle - fstats.Matrix)

## End(Not run)
```

Index

∗ **internal**
derfinderHelper-package, 2

derfinderHelper
(derfinderHelper-package), 2
derfinderHelper-package, 2

fstats.apply, *3*, 3

sparseMatrix, *3*