Package: tfboot (via r-universe)

September 12, 2024

Title Bootstrapping and statistical analysis for TFBS-disrupting SNPs

Version 1.0.1

Description Bootstrap motifbreakR results for statistical analysis on TFBS-disrupting SNPs in upstream regions of a set of genes of interest.

License file LICENSE

URL https://github.com/colossal-compsci/tfboot, https://colossal-compsci.github.io/tfboot/

BugReports https://github.com/colossal-compsci/tfboot/issues

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.0

Imports dplyr, GenomicFeatures, GenomicRanges, ggplot2, MatrixGenerics, MotifDb, plyranges, purrr, rlang, S4Vectors, tibble, tidyr, VariantAnnotation

Suggests motifbreakR, BSgenome.Ggallus.UCSC.galGal6, TxDb.Ggallus.UCSC.galGal6.refGene, org.Gg.eg.db, AnnotationDbi, knitr, rmarkdown

VignetteBuilder knitr

Depends R (>= 4.2.0)

LazyData true

Language en-US

Repository https://stephenturner.r-universe.dev

RemoteUrl https://github.com/colossal-compsci/tfboot

RemoteRef HEAD

RemoteSha 32e2518007524a55b56166d33859f290fd9c21d0

Contents

`get_upstream` *Get upstream intervals*

Description

Get upstream intervals from a GRanges object.

Usage

```
get_upstream(gr, width = 5000L)
```

Arguments

`gr`	A GenomicRanges object.
`width`	Width of the upstream interval. Default 5000.

Value

A GRanges object with intervals of width specified by `width` upstream of the start position of the `gr` input GRanges object.

Examples

```
gr <- data.frame(seqnames = "chr1",
                 strand = c("+", "-"),
                 start = c(42001, 67001),
                 width = c(1000, 1000),
                 gene = c("A", "B")) |>
  plyranges::as_granges()
gr
get_upstream(gr)
```

get_upstream_snps *Get SNPs in upstream regions*

Description

Get SNPs upstream of gene regions. Input SNPs read in with read_vcf and an appropriate TxDb object.

Usage

```
get_upstream_snps(snps, txdb, level = "genes", ...)
```

Arguments

snps	SNPs read in with read_vcf.
txdb	A GenomicFeatures::TxDb object with transcript annotations for the organism of interest. Must match the organism specified in the bsgenome argument of read_vcf.
level	Currently only "genes" and "transcripts" are supported, which run GenomicFeatures::genes and GenomicFeatures::transcripts, respectively.
...	Additional arguments passed to get_upstream (e.g., width=).

Value

A GRanges object containing SNPs in regions upstream of intervals in the specified TxDb. See get_upstream.

Examples

```
## Not run:
library(BSgenome.Ggallus.UCSC.galGal6)
library(TxDb.Ggallus.UCSC.galGal6.refGene)
bs <- BSgenome.Ggallus.UCSC.galGal6
tx <- TxDb.Ggallus.UCSC.galGal6.refGene
vcf_file <- system.file("extdata", "galGal6-chr33.vcf.gz", package="tfboot", mustWork = TRUE)
snps <- read_vcf(vcf_file, BSgenome.Ggallus.UCSC.galGal6)
upstreamsnps <- get_upstream_snps(snps=snps, txdb=tx, level="genes")
upstreamsnps

## End(Not run)
```

mb_bootstats *Bootstrap statistics*

Description

Get statistics (p-values) on your gene set's motifbreakR results compared to the bootstrapped empirical null distribution.

Usage

```
mb_bootstats(mbsmry, mbboot)
```

Arguments

mbsmry	Results from running mb_summarize on motifbreakR results from your gene set of interest.
mbboot	Results from running mb_bootstrap on a full background of all genes. Typically this should be performed once, with results read in from a file.

Value

A tibble with each metric from your bootstrap resampling (see mb_summarize), and p-value comparing the actual value of your gene set (`stat`) against the empirical null distribution (`bootdist`).

Examples

```
data(vignettedata)
mbres <- vignettedata$mbres
mball <- vignettedata$mball
mbsmry <- mb_summarize(mbres)
mbsmry
set.seed(42)
mbboot <- mb_bootstrap(mball, ngenes=5, boots = 100)
mbboot
mb_bootstats(mbsmry, mbboot)
```

mb_bootstrap *Bootstrap motifbreakR results*

Description

Bootstrap motifbreakR results. Takes motifbreakR results as a tibble (from mb_to_tibble), draws `boots` random samples of `ngenes` genes, and returns as a list (1) a wide tibble with results for each bootstrap, and (2) another tibble with the distribution of each metric in a listcol. See examples.

Usage

```
mb_bootstrap(mbtibble, ngenes, boots = 100, key_col = "gene_id")
```

Arguments

mbtibble	A tibble of motifbreakR results from mb_to_tibble.
ngenes	The number of genes to sample with each bootstrap resample.
boots	The number of bootstrap resamples.
key_col	The name of the column used to key the txdb. Default gene_id. May be transcript_id or otherwise if you use a different value of level in get_upstream_snps.

Details

Typically, you want to run this function on a full set of all genes to create an empirical null distribution. You should run motifbreakR *once* on all genes, save this as an RData object, and read it in during bootstrap resampling. See vignettes.

Value

A list of two tibbles. See description and examples.

Examples

```
data(vignettedata)
vignettedata$mball
mb_bootstrap(vignettedata$mball, ngenes=5, boots=5)
```

---

mb_summarize *Summarize motifbreakR results*

Description

Summarizes motifbreakR results as a tibble from mb_to_tibble. See details.

Usage

```
mb_summarize(mbtibble, key_col = "gene_id")
```

Arguments

mbtibble	motifbreakR results summarized with mb_to_tibble.
key_col	The name of the column used to key the txdb. Default gene_id. May be transcript_id or otherwise if you use a different value of level in get_upstream_snps.

Details

Summarizes motifbreakR results. Returns a tibble with columns indicating:

1. `ngenes`: The number of genes in the SNP set.
2. `nsnps`: The number of SNPs total.
3. `nstrong`: The number of SNPs with a "strong" effect.
4. `alleleDiffAbsMean` The mean of the absolute values of the `alleleDiff` scores.
5. `alleleDiffAbsSum` The sum of the absolute values of the `alleleDiff` scores.
6. `alleleEffectSizeAbsMean` The mean of the absolute values of the `alleleEffectSize` scores.
7. `alleleEffectSizeAbsSum` The sum of the absolute values of the `alleleEffectSize` scores.

Value

A tibble. See description.

Examples

```
data(vignettedata)
vignettedata$mbres
mb_summarize(vignettedata$mbres)
```

`mb_to_tibble` *motifbreakR results to tibble*

Description

Make a compact tibble with only select columns from motifbreakR results GRanges objects.

Usage

```
mb_to_tibble(mb, key_col = "gene_id")
```

Arguments

`mb`	motifbreakR results GRanges object.
`key_col`	The name of the column used to key the txdb. Default `gene_id`. May be `transcript_id` or otherwise if you use a different value of `level` in get_upstream_snps.

Details

It's a good idea to run motifbreakR *once* on the background set of *all* genes and save this as an RData (or .rds) file, and read in when you need them.

Value

A tibble containing the key column (usually `gene_id`), and a select number of other columns needed for downstream statistical analysis.

plot_bootstats *Plot bootstrap distributions*

Description

Plot bootstrap distributions of motifbreakR results with your critical value highlighted by a vertical red line.

Usage

```
plot_bootstats(bootstats)
```

Arguments

bootstats	Output from running mb_bootstats on your results and a bootstrap resampling.

Value

A ggplot2 plot object. See description and examples.

Examples

```
data(vignettedata)
mbres <- vignettedata$mbres
mball <- vignettedata$mball
mbsmry <- mb_summarize(mbres)
set.seed(42)
mbboot <- mb_bootstrap(mball, ngenes=5, boots = 250)
bootstats <- mb_bootstats(mbsmry, mbboot)
plot_bootstats(bootstats)
```

read_vcf *Read in SNPs from a VCF*

Description

Helper function to read in SNP data from a VCF file.

Usage

```
read_vcf(file, bsgenome)
```

Arguments

file	File path to a VCF file. See details.
bsgenome	An object of class BSgenome for the species you are interrogating; see BSgenome::available.genomes for a list of species.

Details

Note that the VCF **must** be filtered to only contain variant sites (i.e., no `0/0`), or only homozygous alt sites if you choose (`0/1` or `1/1`). This can be accomplished with bcftools:

```
# Filter to any variant sites:
bcftools view -i 'GT="alt"' ...
# Filter to homozygous alt sites:
bcftools view -i 'GT="AA"' ...
```

Value

A GRanges object containing `SNP_id`, `REF`, and `ALT` columns.

Examples

```
## Not run:
library(BSgenome.Ggallus.UCSC.galGal6)
vcf_file <- system.file("extdata", "galGal6-chr33.vcf.gz", package="tfboot", mustWork = TRUE)
snps <- read_vcf(vcf_file, BSgenome.Ggallus.UCSC.galGal6)
snps

## End(Not run)
```

`split_gr_by_id` *Split GRanges by gene*

Description

Splits a GRanges object into a GRangesList by a column (typically `gene_id`). This function is deprecated and generally has no good use case. Originally written to split up a GRanges object into a list to iterate over using `furrr::future_map()`, but deprecated in favor of using built-in parallelization in motifbreakR.

Usage

```
split_gr_by_id(gr, key_col = "gene_id")
```

Arguments

`gr`	A GRanges object returned by get_upstream_snps.
`key_col`	The name of the column in `gr` to split by (default `gene_id`).

Value

A list of genomic ranges split by `split_col`.

Examples

```
## Not run:
gr <- data.frame(seqnames=rep(c("chr1", "chr2", "chr1", "chr3"), c(1, 3, 2, 4)),
           start=1:10,
           width=1,
           gene_id = rep(c("gene1", "gene2", "gene3", "gene4"), c(4, 2, 1, 3))) |>
  plyranges::as_granges()
gr
split_gr_by_id(gr, key_col="gene_id")

## End(Not run)
```

`vignettedata` *Vignette data*

Description

Pre-cooked data used by the vignette. This data is precomputed to speed vignette compilation time and is used throughout the examples. Contains a list of two objects:

- `mbres`: Results from running motifbreakR on five randomly selected genes, then run through mb_to_tibble.
- `mball`: Results from running motifbreakR on *all* genes, then run through mb_to_tibble.

Usage

```
vignettedata
```

Format

An object of class `list` of length 2.

Index