Package 'tailtools'

May 20, 2016

Title Tail Tools R component

Description R component of the Tail Tools pipeline for analysing PAT-Seq datasets.

Author Paul Harrison [aut],
Michael See [ctb]

Maintainer Paul Harrison <paul.harrison@monash.edu>

URL https://github.com/Victorian-Bioinformatics-Consortium/tail-tools

Version 0.1

License LGPL-2.1

Depends BiocGenerics, S4Vectors, edgeR, limma, rtracklayer, GenomicRanges, ggbio, grid-Base, dplyr, readr, purrr, tidyr, stringr, assertthat, ggplot2, seriation, shiny, DT, nesoni, varistran

Suggests GOstats, org.Sc.sgd.db, org.Hs.eg.db, org.Ce.eg.db, org.Mm.eg.db

RoxygenNote 5.0.1

R topics documented:

tailtools-package *Tail Tools R library*

Description

This is the R library for Tail Tools, and is mostly to do with presenting the output of the Tail Tools python-based pipeline.

end_shift *End shift statistics, generic function*

Description

End shift statistics, generic function

Usage

```
end_shift(counts, peak_info, condition, group = NULL,
  gene_info_columns = c("gene", "product", "biotype"), ci = 0.95, fdr = T,
  edger = T, limma = T, min_reads = 10, title = "End-shift test",
  fdr_max_permute = 1000)
```

Arguments

peak_info	should be a data frame containing columns: id, position, strand (+/-1), parent. position is position in chromosome of transcription stop site. strand should be the strand of the *gene*, if including antisense features.
condition	is a logical vector splitting samples into control and experimental groups.
group	if given is a factor splitting samples into batches, if there is a batch effect.
gene_info_columns	Columns of peak_info to retain in per-gene output.
ci	Confidence interval. A value closer to 1 will more heavily demote genes with low read counts.
fdr	Perform permutation based FDR?
edger	Perform edgeR differential exon usage test. May need to disable if too few replicates.
limma	Perform limma differential exon usage test. May need to disable if no replicates.
min_reads	Discard genes with lower than this average number of reads per sample.
fdr_max_permute	If there are more than this many possible permutations, just use this many randomly sampled distinct permutations (but certainly including the true permutation).

end_shift_pipeline *End shift statistics from tail-tools pipeline output.*

Description

End shift statistics from tail-tools pipeline output.

Usage

```
end_shift_pipeline(path, condition, group = NULL, ci = 0.95, fdr = T,
  edger = T, limma = T, antisense = T, colliders = T, non_utr = T,
  min_reads = 10, title = "End-shift test", fdr_max_permute = 1000)
```

Arguments

fdr	Produce permutation based q values, can be very slow. Samples with NA in condition will be omitted.

end_shift_rnaseq *End-shift test for RNA-Seq data*

Description

End-shift test for RNA-Seq data

Usage

```
end_shift_rnaseq(samples, utrs, extended_utrs = NULL, exons = NULL,
  ci = 0.95, min_min_reads = 20)
```

Arguments

samples	should be a data frame with columns: name = name of sample, condition = logical vector with NA for samples to ignore, bigwig filenames giving depth of coverage cover_fwd, cover_rev.
utrs	a GRanges of 3' UTRS. Should have metadata columns: ID, Name, gene_id, gene, description.
extended_utrs	if given tries extending 3' UTRS beyond the annotated end point. A GRanges of validly extended 3' UTRS, in the same order as utrs. Suggestion is UTRs extended up to 10kb but not into the following gene on the same strand.
exons	if given, restricts testing to exonic regions. A GRanges of transcript exons, which should have metadata column Parent corresponding to the ID in the utrs parameter.

load_genome_features *Load features needed by plot_genome().*

Description

Load features needed by plot_genome().

Usage

```
load_genome_features(reference_dir)
```

plot_genome *Produce a depth of coverage plot of a set of samples.*

Description

Produce a depth of coverage plot of a set of samples.

Usage

```
plot_genome(features, samples, pos, marks = c())
```

plot_patseq_heatmap *Printable heatmap grob*

Usage

```
plot_patseq_heatmap(matf1, matf2, gmatf, clusterby = 1,
  sample_labels = NULL, sample_labels2 = NULL, feature_labels = NULL,
  gene_labels = NULL, product_labels = NULL, row_ord = 1)
```

Arguments

matf1	Data frame of Tail length
matf2	Data frame of Counts (genewise expression) (Should already be normalised)
gmatf	Data frame of annotation data
clusterby	Cluster columns by tail length or expression (defaults to None)
sample_labels	Sample labels
sample_labels2	Sample labels (second plot)
feature_labels	Feature labels
gene_labels	Gene labels
product_labels	Product labels
row_ord	Order rows by tail length or expression

Details

Produces a print-able heatmap grob

`shiny_end_shift` *Show a report about an end shift analysis.*

Description

Show a report about an end shift analysis.

Usage

```
shiny_end_shift(result)
```

Arguments

`result`	output from end_shift()

`shiny_end_shift_pipeline` *Perform end shift analyses on pipeline output and report results.*

Description

Perform end shift analyses on pipeline output and report results.

Usage

```
shiny_end_shift_pipeline(tests, cache_prefix = "cache_")
```

Arguments

`tests`	Named list of lists giving parameters to end_shift_pipeline. Each name should be unique.

shiny_end_shift_rnaseq_multiple *Interface to multiple RNA-Seq end-shift tests.*

Description

Interface to multiple RNA-Seq end-shift tests.

Usage

```
shiny_end_shift_rnaseq_multiple(samples, tests, references,
  cache_prefix = "cache", prefix = "", title = "RNA-Seq end-shift test")
```

Arguments

samples	A data frame of samples. Should have columns: name = name of sample, bigwig files: cover_fwd, cover_rev, span_fwd, span_rev.
references	A named character vector of reference directories created with "tail-tools make-rnaseq-reference:".
conditions	A list of condition vectors.

shiny_patseq_heatmap *Produces detailed heatmap*

Usage

```
shiny_patseq_heatmap(datfr, sample_labels = NULL, sample_labels2 = NULL,
  feature_labels = NULL, prefix = "", species = NULL)
```

Arguments

datfr	List of dataframes Takes a read.grouped.table() as input or a list of four dataframes (more data frames are ok but it only uses these): Counts - Genewise counts of expression. Tail - Mean tail length Tail_counts - Number of poly-A tails counted. Annotation - Information regarding the annotation information Gene name, chromosome, gene product, biotype etc...
sample_labels	Sample labels
sample_labels2	Sample labels (second plot)
feature_labels	Feature labels
prefix	Prefix for plot
species	Species of the data. Currently supports Human (Hs), Saccharomyces cerevisiae (Sc), Caenorhabditis elegans (Ce), Mus musculus (Mm) Now optional. Not entering this disables GO term analysis

Details

Workhorse function for this package.

Value

Returns a composable shiny app object

`shiny_patseq_heatmap_inner` *Integrates heatmap into shiny*

Usage

```
shiny_patseq_heatmap_inner(callback, width = 500, height = 500,
  dlname = "plot", prefix = "", selin, rorder, spp, hgc = 0.05,
  otype = 1, goenabl = TRUE)
```

Arguments

`callback`	Heatmap Grob from plot_patseq_heatmap
`width`	Default width of the heatmap grob when the shiny app loads
`height`	Default height of the heatmap grob when the shiny app loads
`dlname`	Default filename when downloading the heatmap image
`prefix`	Prefix
`selin`	Selection of genes from sh_hmap_detailed
`rorder`	Row order from sh_hmap_detailed
`spp`	Species of the data (Hs, Sc, Ce, Mm)
`goenabl`	Whether GO Term analysis is enabled. TRUE by default. This might need changing in future

Details

Shiny wrapper for sh_hmap_detailed Uses gridBase to produce brush for interactivity with plot

---

`shiny_repat` *Shiny report for REPAT results*

Description

Shiny report for REPAT results

Usage

```
shiny_repat(filename, normalizing_gene = NULL, title = "3'TAP results")
```

shiny_tailtools_report *Shiny report based on pipeline output*

Description

Shiny report based on pipeline output

Usage

```
shiny_tailtools_report(path, species = NULL)
```

Arguments

path	Directory containing pipeline output.
species	Species for tail heatmap. Currently supports Human ("Hs"), Saccharomyces cerevisiae ("Sc"), Caenorhabditis elegans ("Ce"), Mus musculus ("Mm")

Index