Package ‘metagene’

October 9, 2015

Version 2.0.0

Date 2014-05-05

Title A package to produce metagene plots

Author Charles Joly Beauparlant <charles.joly-beauparlant@crchul.ulaval.ca>, Fabien Claude Lamaze <fabien.lamaze.1@ulaval.ca>, Rawane Samb <rawane.samb.1@ulaval.ca>, Astrid Louise Deschenes <Astrid-Louise.Deschenes@crchudequebec.ulaval.ca> and Arnaud Droit <arnaud.droit@crchuq.ulaval.ca>.

Author@R c(person(``Charles'', ``Joly Beauparlant'', email=``charles.joly-beauparlant@crchul.ulaval.ca''), person(``Fabien Claude'', ``Lamaze'', email=``fabien.lamaze.1@ulaval.ca''), person(``Rawane'', ``Samb'', email=``rawane.samb.1@ulaval.ca''), person(``Astrid Louise'',``Deschenes'', email=``Astrid-Louise.Deschenes@crchudequebec.ulaval.ca'') person(``Arnaud'', ``Droit'', email=``arnaud.droit@crchuq.ulaval.ca''))

Maintainer Charles Joly Beauparlant <charles.joly-beauparlant@crchul.ulaval.ca>

Description This package produces metagene plots to compare the behavior of DNA-interacting proteins at selected groups of genes/features. Bam files are used to increase the resolution. Multiple combination of group of bam files and/or group of genomic regions can be compared in a single analysis. Bootstraping analysis is used to compare the groups and locate regions with statistically different enrichment profiles.

biocViews ChIPSeq, Genetics, MultipleComparison, Coverage, Alignment

License Artistic-2.0 | file LICENSE

BugReports https://github.com/CharlesJB/metagene/issues

VignetteBuilder knitr

Depends R6 (>= 2.0), GenomicRanges, BiocParallel

Imports rtracklayer, gplots, biomaRt, tools, GenomicAlignments, ggplot2, muStat, Rsamtools

Suggests RUnit, BiocGenerics, knitr

NeedsCompilation no

R topics documented:

`Bam_Handler` *A class to manage BAM files.*

Description

This class will allow to load, convert and normalize alignments and regions files/data.

Usage

```
Bam_Handler
```

Format

A BAM manager

Value

`Bam_Handler$new` returns a `Bam_Handler` object which contains coverage related information for every BAM files.

Constructor

```
bh <- Bam_Handler$new(bam_files, cores = SerialParam())
```

bam_files A vector of BAM filenames. The BAM files must be indexed. i.e.: if a file is named file.bam, there must be a file named file.bam.bai in the same directory.

cores The number of cores available to parallelize the analysis. Either a positive integer or a `BiocParallelParam`. Default: `SerialParam()`.

`Bam_Handler$new` returns a `Bam_Handler` object that contains and manages BAM files. Coverage related information as alignment count can be obtain by using this object.

Methods

```
bh$get_aligned_count(bam_file)
```

bam_file The name of the BAM file.

```
bh$get_rpm_coefficient(bam_file)
```

bam_file The name of the BAM file.

```
bh$index_bam_files(bam_files)
```

bam_files A `vector` of BAM filenames.

```
bh$get_bam_files()
```

```
bh$get_normalized_coverage(bam_file, regions)
```

bam_file The name of the BAM file.

regions A not empty `GRanges` object.

Examples

```
bh <- metagene:::Bam_Handler$new(bam_files=get_demo_bam_files())
 bh$get_aligned_count(metagene:::get_demo_bam_files()[1])
```

`getGenes` *Fetch the annotation of all genes.*

Description

This function will fetch the positions of all known coding genes for a given specie. Currently supported species are: “mouse”, “human” (default).

Usage

```
getGenes(
  specie="human")
```

Arguments

`specie`	human: Homo sapiens (default) / mouse: Mus musculus

Details

This function will fetch all the ensembl_gene_id for a given specie ("human" or "mouse").

Value

`getGenes` return a `GRanges` object with a feature metadata that correspond to the ensembl_gene_id.

Author(s)

Charles Joly Beauparlant <Charles.Joly-Beauparlant@crchul.ulaval.ca>

Examples

```
## Not run: knownGenes <- getGenes("human")
```

`getGenesBiomart` *Fetch the annotation of all genes from biomart.*

Description

This function will fetch the positions of all known coding genes for a given specie. Currently supported species are: “mouse”, “human” (default).

This function was used to create external datasets for the `getGenes` function.

Usage

```
getGenesBiomart(
  specie="human")
```

Arguments

`specie`	human: Homo sapiens (default) / mouse: Mus musculus

Details

Using `biomaRt` package, this function will fetch all the ensembl_gene_id for a given specie ("human" or "mouse").

Value

`getGenesBiomart` return a `GRanges` object with a feature metadata that corresponds to emsembl_gene_id.

Author(s)

Charles Joly Beauparlant <Charles.Joly-Beauparlant@crchul.ulaval.ca>

Examples

```
## Not run: knownGenes <- getGenesBiomart("human")
```

`get_demo_bam_files` *Get BAM filenames for demo*

Description

Get BAM filenames for demo

Usage

```
get_demo_bam_files()
```

Value

A vector of BAM filenames

Examples

```
bam_files <- get_demo_bam_files()
```

`get_demo_regions` *Get regions filenames for demo*

Description

Get regions filenames for demo

Usage

```
get_demo_regions()
```

Value

A vector of regions filenames

Examples

```
regions <- get_demo_regions()
```

`metagene` *A class to manage metagene analysis.*

Description

This class will allow to load, convert and normalize alignments and regions files/data. Once the data is ready, the user can then chose to produce metagene plots on the data (or a subset of the data).

Usage

```
metagene
```

Format

A metagene experiment manager

Value

`metagene$new` returns a `metagene` object which contains the normalized coverage values for every regions and for every BAM files.

Constructor

```
 mg <- metagene$new(regions, bam_files, padding_size = 0,                      cores = SerialParam()
```

regions Either a `vector` of BED filenames, a `GRanges` object or a `GRangesList` object.

bam_files A `vector` of BAM filenames. The BAM files must be indexed. i.e.: if a file is named file.bam, there must be a file named file.bam.bai in the same directory.

padding_size The regions will be extended on each side by the value of this parameter. The padding_size must be a non-negative integer. Default = 0.

cores The number of cores available to parallelize the analysis. Either a positive integer or a `BiocParallelParam`. Default: `SerialParam()`.

verbose Print progression of the analysis. A logical constant. Default: `FALSE`.

`metagene$new` returns a `metagene` object that contains the coverages for every bam files in the regions from the `regions` param.

Methods

```
 df <- mg$plot(design = NULL, regions_group = NULL,           bin_size = 100, alpha = 0.05, sample_cou
```

design A `data.frame` that describe to experiment to plot. The first column must be the existing BAM filenames. The other columns (at least one is required) represent how the files should be grouped. All those colums should be in numeric format. All the files in the same group will be combined before doing the statistical analysis. For each column, there will be one line with its ribon in the metagene plot. At least one file should be selected (not zero). Default = NULL.

0: Do not use file. 1: File is input. 2: File is control.

region_group A `list` or a `vector` of region names to include in the analysis. If `NULL`, all the regions used when creating the `metagene` object will be used. Default: `NULL`

bin_size The size of bin to use before calculating the statistics. larger `bin_size` will reduce the calculation time and produce smoother curves. Default = 100.

alpha The condifence interval (CI) to represent with a ribbon. Must be a value between 0 and 1. With a value of 0.05, the ribbon will represent 95 Default = 0.05.

sample_count The number of draw to do during bootstrap analysis. Default: 1000.

```
mg$export(bam_file, region, file)
```

bam_file The name of the bam file to export.

region The name of the region to export.

file The name of the ouput file.

```
mg$heatmap(region, bam_file, bin_size)
```

region The name of the region to export.

bam_file The name of the bam file to export.

bin_size The size of the bin to produce before creating heatmap.

Examples

```
regions <- get_demo_regions()
 bam_files <- get_demo_bam_files()
 mg <- metagene$new(regions, bam_files)
 ## Not run:
 df <- metagene$plot()
## End(Not run)
```

Index