Package 'GenomicOZone'

April 15, 2020

Type Package

Title Delineate outstanding genomic zones of differential gene activity

biocViews Software, GeneExpression, Transcription, DifferentialExpression, FunctionalPrediction, GeneRegulation, BiomedicalInformatics, CellBiology, FunctionalGenomics, Genetics, SystemsBiology, Transcriptomics, Clustering, Regression, RNASeq, Annotation, Visualization, Sequencing, Coverage, DifferentialMethylation, GenomicVariation, StructuralVariation

Version 1.0.0

Author Hua Zhong, Mingzhou Song

Maintainer Hua Zhong<zh9118@gmail.com>, Mingzhou Song <joemsong@cs.nmsu.edu>

Description

The package clusters gene activity along chromosome into zones, detects differential zones as outstanding, and visualizes maps of outstanding zones across the genome. The method guarantees cluster optimality, linear runtime to sample size, and reproducibility. It enables new characterization of effects due to genome reorganization, structural variation, and epigenome alteration.

License LGPL (>=3)

Encoding UTF-8

NeedsCompilation no

Depends R (>= 3.6), Ckmeans.1d.dp (>= 4.3.0), GenomicRanges, biomaRt, ggplot2

Suggests readxl, GEOquery, knitr, rmarkdown

Imports grDevices, stats, utils, plyr, gridExtra, sjstats, parallel, ggbio, S4Vectors, IRanges, GenomeInfoDb, Rdpack

RdMacros Rdpack

LazyData true

VignetteBuilder knitr

RoxygenNote 6.1.1

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

git_branch RELEASE_3_10

git_last_commit 5cd2322

git_last_commit_date 2019-10-29

Date/Publication 2020-04-14

R topics documented:

`extract_outputs` *Extract annotation of genes, zones, outstanding zones, and expression of zones*

Description

Extract informtation from the output of `GenomicOZone` function, including a gene annotation object, a zone annotation object, an outstanding zone annotation object, or a zone activity matrix, respectively. The activity of genes without annotation is appended at the bottom of the zone activity matrix.

Usage

```
extract_genes(GOZ.ds)

extract_zones(GOZ.ds)

extract_outstanding_zones(
              GOZ.ds,
              alpha = 0.05,
              min.effect.size = 0.8)

extract_zone_expression(GOZ.ds)
```

Arguments

`GOZ.ds`	a object returned from the `GenomicOZone` function.
`alpha`	a cutoff for adjusted p-values. Default to 0.05.
`min.effect.size`	the minimum effect size required for an outstanding zone. The effect size for ANOVA ranging from 0 to 1 is calculated by R package **sjstats** (Lüdecke 2019). Default to 0.8.

Details

These functions take the input of an object created by `GOZDataSet` and processed by `GenomicOZone`. The functions access the object and fetch the results. The function `extract_zone_expression` offers the zone activity matrix. The activity of a zone is the total activity of genes within the zone for each sample. The activity of genes without annotation is included as last rows in the zone activity matrix.

Value

The first three functions return an object of GRanges class (Lawrence et al. 2013) for all genes, all zones and outstanding genomic zones only. The gene GRanges object includes genome annotation and the zones where the genes belong. The zone GRanges object includes zone positions and p-values of differential zone analysis. Outstanding genomic zones are a subset of all zones that satisfy required p-value and effect size.

References

Lawrence M, Huber W, Pages H, Aboyoun P, Carlson M, Gentleman R, Morgan MT, Carey VJ (2013). "Software for computing and annotating genomic ranges." *PLoS computational biology*, **9**(8), e1003118.

Lüdecke D (2019). *sjstats: Statistical Functions for Regression Models (Version 0.17.5)*. doi: 10.5281/zenodo.1284472, https://CRAN.R-project.org/package=sjstats.

See Also

See GOZDataSet for how to create the input object before outstanding genomic zone analysis. The object must contain information obtained from outstanding zone analysis function GenomicOZone.

Examples

```
# Create an object of GOZ.ds
data <- matrix(c(1,5,2,6,5,1,6,2), ncol = 2, byrow = TRUE)
rownames(data) <- paste("Gene", 1:4, sep='')
colnames(data) <- paste("Sample", c(1:2), sep='')

colData <- data.frame(Sample_name = paste("Sample", c(1:2), sep=''),
                      Condition = c("Cancer", "Normal"))

design <- ~ Condition

rowData.GRanges <- GRanges(seqnames = Rle(rep("chr1", 4)),
                           ranges = IRanges(start = c(1,2,3,4), end = c(5,6,7,8)))
names(rowData.GRanges) <- paste("Gene", 1:4, sep='')

ks <- c(2)
names(ks) <- "chr1"

GOZ.ds <- GOZDataSet(data, colData, design,
                     rowData.GRanges = rowData.GRanges,
                     ks = ks)
####

# Run outstanding zone analysis
GOZ.ds <- GenomicOZone(GOZ.ds)
####

# Extract output in various formats
Gene.GRanges <- extract_genes(GOZ.ds)
head(Gene.GRanges)

Zone.GRanges <- extract_zones(GOZ.ds)
head(Zone.GRanges)
```

```
OZone.GRanges <- extract_outstanding_zones(
                              GOZ.ds,
                              alpha = 0.05,
                              min.effect.size = 0.8)
head(OZone.GRanges)

Zone.exp.mat <- extract_zone_expression(GOZ.ds)
head(Zone.exp.mat)
```

`generate_plots` *Generate plots of genome, chromosomes and zones.*

Description

Generate the plot from the processed GenomicOZone dataset object, including genome plots, chromosome plots and zone plots.

Usage

```
plot_genome(GOZ.ds, plot.file,
            alpha = 0.05, min.effect.size = 0.8,
            plot.width = NULL, plot.height = NULL)

plot_chromosomes(GOZ.ds, plot.file,
            alpha = 0.05, min.effect.size = 0.8,
            plot.width = NULL, plot.height = NULL)

plot_zones(GOZ.ds, plot.file,
            alpha = 0.05, min.effect.size = 0.8,
            log.exp = TRUE, plot.all.zones = FALSE)
```

Arguments

`GOZ.ds`	a GenomicOZong dataset object after running `GenomicOZone` function.
`plot.file`	a output file name. The file type is "pdf".
`alpha`	a cutoff for selecting adjuted p-values.
`min.effect.size`	the minimum effect size required for an outstanding zone. The effect size for ANOVA ranging from 0 to 1 is calculated by R package **sjstats** (Lüdecke 2019). Default to 0.8.
`plot.width`	a numerical number to specify the width of page in the plot. Using `NULL` will automatically determine the page width. Default is `NULL`.
`plot.height`	a numerical number to specify the height of page in the plot. Using `NULL` will automatically determine the page height. Default is `NULL`.
`log.exp`	a logical indicating whether to use log-scaled activity in the plot.
`plot.all.zones`	a logical indicating whether to plot all zones into the file. If `FALSE`, only outstanding genomic zones will be plotted.

Details

The three functions plot visualizations of the genome, chromosomes and zones. The R packages `ggplot2` (Wickham 2016) and `ggbio` (Yin et al. 2012) are used to generate the plots.

The function `plot_genome` plots the genome-wide overviews with marked significant differential zones.

The function `plot_chromosomes` plots the chromosome-wide heatmap of normalized and linearized activity between sorted zones and samples, visualizing the zones with significant ones marked.

The function `plot_zones` plots the line chart and box-plot of the activity of the genes within each significant zone, visualizing gene activity changes over sample conditions.

Value

The function takes an input of a object, which has been created by `GOZDataSet` and and processed by `GenomicOZone`. The functions accesse the object and generate visualizations. See `GOZDataSet` for how to create the input object. See `GenomicOZone` for how to process the input object and perform the analysis.

References

Lüdecke D (2019). *sjstats: Statistical Functions for Regression Models (Version 0.17.5)*. doi: 10.5281/zenodo.1284472, https://CRAN.R-project.org/package=sjstats.

Wickham H (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. ISBN 978-3-319-24277-4, https://ggplot2.tidyverse.org.

Yin T, Cook D, Lawrence M (2012). "ggbio: an R package for extending the grammar of graphics for genomic data." *Genome biology*, **13**(8), R77.

Examples

```
# Create an example of GOZ.ds
data <- matrix(c(1,5,2,6,5,1,6,2), ncol = 2, byrow = TRUE)
rownames(data) <- paste("Gene", 1:4, sep='')
colnames(data) <- paste("Sample", c(1:2), sep='')

colData <- data.frame(Sample_name = paste("Sample", c(1:2), sep=''),
                      Condition = c("Cancer", "Normal"))

design <- ~ Condition

rowData.GRanges <- GRanges(seqnames = Rle(rep("chr1", 4)),
                           ranges = IRanges(start = c(1,2,3,4), end = c(5,6,7,8)))
names(rowData.GRanges) <- paste("Gene", 1:4, sep='')

ks <- c(2)
names(ks) <- "chr1"

GOZ.ds <- GOZDataSet(data, colData, design,
                     rowData.GRanges = rowData.GRanges,
                     ks = ks)
####

# Run the zoing process
GOZ.ds <- GenomicOZone(GOZ.ds)
```

```
####

# Generate plots
plot_genome(GOZ.ds, plot.file = "Test_genome.pdf",
            plot.width = 15, plot.height = 4)

plot_chromosomes(GOZ.ds, plot.file = "Test_chromosome.pdf",
                 plot.width = 20, plot.height = 4)

plot_zones(GOZ.ds, plot.file = "Test_zone.pdf",
           plot.all.zones = FALSE)
```

GenomicOZone *Delineate outstanding genomic zones*

Description

Delineate outstanding genomic zones along chromosomes such that genes within an outstanding zone have consistent activity patterns that are different across samples.

Usage

```
GenomicOZone(GOZ.ds)
```

Arguments

GOZ.ds	an object created by function GOZDataSet.

Details

This is the most important function of the package. It integrates genome annotation, gene activity matrix preprocessing, chromosome clustering, and differential zone analysis.

Genome annotation can be specified either by the user, or obtained from the R package **biomaRt** (Smedley et al. 2015) to access ensembl annotation databases (Zerbino et al. 2017).

The function calls the weighted univariate clustering method (Wang and Song 2011) implemented in the **Ckmeans.1d.dp** package. If ks is specified, a fixed number of zones at each chromosome will be delineated. If ks is NULL, an optimal number of clusters at each chromosome will be determined by Bayesian information criterion.

The function also conducts differential zone analysis by using one-way ANOVA (Chambers et al. 1992) based on gene ranks. Given p-value cutoff alpha and effect size threshold min.effect.size, outstanding genomic zones will be selected.

Advanced differential zone analysis such as generalized linear modeling can be performed on the zone activity matrix using third-party software. The zone activity matrix can be generated by auxiliary functions.

Value

an object which is the input object attached with intermediate and final results. Results can be accessed by calling several functions in extract_outputs. The results can be visualized by calling the functions in generate_plots.

References

Chambers JM, Hastie TJ, others (1992). "Statistical models in S." In volume 251, chapter 5. Wadsworth & Brooks/Cole Advanced Books & Software Pacific Grove, CA.

Smedley D, Haider S, Durinck S, Pandini L, Provero P, Allen J, Arnaiz O, Awedh MH, Baldock R, Barbiera G, others (2015). "The BioMart community portal: an innovative alternative to large, centralized data repositories." *Nucleic acids research*, **43**(W1), W589–W598.

Wang H, Song M (2011). "Ckmeans. 1d. dp: optimal k-means clustering in one dimension by dynamic programming." *The R journal*, **3**(2), 29.

Zerbino DR, Achuthan P, Akanni W, Amode MR, Barrell D, Bhai J, Billis K, Cummins C, Gall A, Girón CG, others (2017). "Ensembl 2018." *Nucleic acids research*, **46**(D1), D754–D761.

See Also

See `GOZDataSet` for how to create the input list. See `extract_outputs` and `generate_plots` for how to access the results and generate visualizations.

Examples

```
# Create an example of GOZ.ds
data <- matrix(c(1,5,2,6,5,1,6,2), ncol = 2, byrow = TRUE)
rownames(data) <- paste("Gene", 1:4, sep='')
colnames(data) <- paste("Sample", c(1:2), sep='')

colData <- data.frame(Sample_name = paste("Sample", c(1:2), sep=''),
                      Condition = c("Cancer", "Normal"))

design <- ~ Condition

rowData.GRanges <- GRanges(seqnames = Rle(rep("chr1", 4)),
                           ranges = IRanges(start = c(1,2,3,4), end = c(5,6,7,8)))
names(rowData.GRanges) <- paste("Gene", 1:4, sep='')

ks <- c(2)
names(ks) <- "chr1"

GOZ.ds <- GOZDataSet(data, colData, design,
                     rowData.GRanges = rowData.GRanges,
                     ks = ks)
####

# Run the zoing process
GOZ.ds <- GenomicOZone(GOZ.ds)
####
```

`GOZDataSet` *Create an object for outstanding genomic zone analysis*

Description

The function prepares an object for outstanding genomic zone analysis. It integrates data, annotation, and analysis parameters into the object and performs additional check on data integrity.

Usage

```
GOZDataSet(data, colData, design,
           clustering.method = "1C",
           rowData.GRanges = NULL,
           ks = NULL,
           genome = NULL,
           ensembl.mirror = "www",
           gene.ID.type = NULL,
           ncores = 1)
```

Arguments

data	a numerical matrix of gene activity data. Rows represent genes. Columns represent samples. The activity data can be gene expression, methylation beta value, copy number variation segment mean, or other gene-based omic data. Row and column names of the matrix must be specified.
colData	a dataframe of sample information. The first column must be sample names, corresponding to columns in the data matrix. Each additional column must contain at least two experimental conditions necessary for differential zone analysis.
design	a one-sided formula with only right-hand side variables. Only one variable is supported in this version. The formula describes on which variable in colData to apply differential zone analysis.
clustering.method	a chracter string. An option to choose either using "1C" to accumulate all channels of weight into one channel, or using "MC" to allow multi channel of weight in the clustering. Default is "1D".
rowData.GRanges	an optional genome annotation of GRanges class. Rows of rowData.GRanges correspond to rows of the data matrix. Row names of rowData.GRanges must be consistent with row names of data. Their orders are not necessarily the same. Only annotated genes in data will be used in genomic zone analysis. One of rowData.GRanges and genome must be specified.
ks	an optional numerical vector to specify the number of zones to divide each chromosome into. The names of the ks vector must be chromosome names. It is only used with user-specified rowData.GRanges. The seqlevels of rowData.GRanges must have a corresponding name in ks. If not specified, an optimal k value (1~400) will be determined for each chromosome. 400 is equivalent to cluster the longest human chromosome into zones averaging wider than 1 million base pairs. Default is NULL.
genome	an optional value of character type to select a genome from **biomaRt**. Available genomes can be found in the "version" column of the available ensembl datasets from **biomaRt** database by calling listDatasets(useMart("ensembl")). One of rowData.GRanges and genome must be specified.
ensembl.mirror	an optional Ensembl mirror server to connect to. It is used only when genome is not NULL. The options are "www", "uswest", "useast" and "asia". Default is "www".
gene.ID.type	an optional value of character type to specify a gene ID type. Options are "hgnc_symbol", "mgi_symbol", "ensembl_gene_id" and "ensembl_transcript_id". Only these four types are allowed in this version. This parameter only works with user-specified genome. If unspecified, all four types would be evaluated to choose the best one.

ncores an optional integer to specify the number of cores to use parallely in outstanding genomic zone analysis. Default is 1.

Details

The function collects all the input information, checks requirement completeness and integrates the inputs into a list, in preparation for function GenomicOZone to perform outstanding zone analysis.

A genome annotation parameter of GRanges class (Lawrence et al. 2013) or a genome version must be assigned by the user. The annotation is used to sort genes by their genomic coordinates. The genome parameter is for function GenomicOZone to obtain genome annotation from the R package **biomaRt** (Smedley et al. 2015) to access Ensembl annotation databases (Zerbino et al. 2017). Using rowData.GRanges is recommended over using genome.

Value

A list object with all relevant information for oustanding genomic zone analysis. It will be expanded by further analysis.

References

Lawrence M, Huber W, Pages H, Aboyoun P, Carlson M, Gentleman R, Morgan MT, Carey VJ (2013). "Software for computing and annotating genomic ranges." *PLoS computational biology*, **9**(8), e1003118.

Smedley D, Haider S, Durinck S, Pandini L, Provero P, Allen J, Arnaiz O, Awedh MH, Baldock R, Barbiera G, others (2015). "The BioMart community portal: an innovative alternative to large, centralized data repositories." *Nucleic acids research*, **43**(W1), W589–W598.

Zerbino DR, Achuthan P, Akanni W, Amode MR, Barrell D, Bhai J, Billis K, Cummins C, Gall A, Girón CG, others (2017). "Ensembl 2018." *Nucleic acids research*, **46**(D1), D754–D761.

Examples

```
data <- matrix(c(1,5,2,6,5,1,6,2), ncol = 2, byrow = TRUE)
rownames(data) <- paste("Gene", 1:4, sep='')
colnames(data) <- paste("Sample", c(1:2), sep='')

colData <- data.frame(Sample_name = paste("Sample", c(1:2), sep=''),
                      Condition = c("Cancer", "Normal"))

design <- ~ Condition

rowData.GRanges <- GRanges(seqnames = Rle(rep("chr1", 4)),
                           ranges = IRanges(start = c(1,2,3,4), end = c(5,6,7,8)))
names(rowData.GRanges) <- paste("Gene", 1:4, sep='')

ks <- c(2)
names(ks) <- "chr1"

GOZ.ds <- GOZDataSet(data, colData, design,
                     rowData.GRanges = rowData.GRanges,
                     ks = ks)
```

Index