Package ‘DriverGenePathway’

May 5, 2023

Type Package

Title DriverGenePathway: Search significant genes triggering cancer using different hypothesis testing methods

Version 0.99.0

Date 2018-07-20

Maintainer Zeyuan Wang <1154094410@mail.dlut.edu.cn>

Description This R package identifies significant genes or gene sets with somatic mutation data. It provides various methods, including preprocessing for input raw files, and 3 different specific gene detecting functions, i.e. 2D projection, sigGenes, and AWRMP. Specifically, it could search significant genes with different hypothesis testing methods, such as binomial distribution test, beta binomial distribution test, Fisher combined P-value test, likelihood ratio test and convolution test. To find a driver gene set, it implements the AWRMP method that models an optimal submatrix function with the coverage and exclusivity, and solves it with genetic algorithm.

Depends R (>= 3.5.0)

Suggests RUnit,
knitr,
rmarkdown,
testthat

BiocViews Coverage, DriverMutation, SomaticMutation

Imports data.table,
utils,
dplyr,
stringr,
BSgenome,
plyr,
ggplot2,
pheatmap,
VennDiagram

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

VignetteBuilder knitr

R topics documented:

---

backgroundMutationRate

Calculate background mutation rates

Description

This function uses the data output by the preprocessing function in order to get background mutation rate. It uses the method of bagel. The result will be then used in varified specific gene detecting methods.

Usage

```
backgroundMutationRate(
  maf = NULL,
  coverage = NULL,
  covariate = NULL,
  original_mutation_rate = 1.2e-06,
  max_neighbors = 50,
  ref_genome = NULL,
  category_num = NULL,
  output_file = TRUE,
  quiet = FALSE
)
```

Arguments

`maf` Mutation Annotation Format, can be whether an R data frame or the path of a txt file.

`coverage` Coverage file, can be whether an R data frame or the path of a txt file.

`covariate` Covariate file, can be whether an R data frame or the path of a txt file.

`original_mutation_rate`
The default background mutation rate is 1.2e-6, which is used to screen a subset of genes through binomial test.

`max_neighbors` Max number of gene neighbors used to calculate the background mutation rate of a specified gene.

`ref_genome` Reference chromosome, either a folder name or a installed BSgenome package. Default NULL, tries to auto-detect from installed genomes.

`category_num` Number of mutation categories, default 4. If the number is 0, categories should exist.

`output_file` Determine whether to export the categories, categorised maf and coverage as txt files.

`quiet` Whether to show notes during processing.

Details

This function first selects a subset of genes whose mutation rates are significantly greater than original mutation rate according to binomial test, then calculates background mutation rates for these genes.

Value

The output BMR_out is an input of the DriverGenes function.

Examples

```
laml_maf <- system.file("extdata", "tcga_laml.maf", package = "DriverGenePathway")
coverage <- system.file("extdata", "coverage.rda", package = "DriverGenePathway")
covariate <- system.file("extdata", "gene.covariates.txt", package = "DriverGenePathway")
load(coverage)
bmr_result <- backgroundMutationRate(laml_maf, coverage, covariate, original_mutation_rate = 1.2e-6,
max_neighbors = 50, ref_genome = NULL, category_num = 1, output_file = TRUE, quiet = FALSE)
```

`covariatePreprocess` *Preprocess covariate file*

Description

This function guarantees that the covariate file is available for DriverGenePathway. Two kinds of covariate input are feasible. It can be whether a txt file or an R data frame. The function returns the preprocessed covariate. An txt file of the preprocessed covariate is optional to be exported.

Usage

```
covariatePreprocess(
  covariate = NULL,
  maf = NULL,
  output_file = TRUE,
  quiet = FALSE
)
```

Arguments

covariate	Covariate file, can be whether an R data frame or the path of a txt file.
maf	Mutation Annotation Format, can be whether an R data frame or the path of a txt file.
output_file	Determine whether to export the preprocessed maf as a txt file.
quiet	Whether to show notes during processing.

Value

The preprocessed covariate file.

Examples

```
covariate <- system.file("extdata", "gene.covariates.txt", package = "DriverGenePathway")
laml_maf <- system.file("extdata", "tcga_laml.maf", package = "DriverGenePathway")
covariate <- covariatePreprocess(covariate, laml_maf)
```

coveragePreprocess *Preprocess coverage file*

Description

This function guarantees that the coverage file is available for DriverGenePathway. Two kinds of coverage input are feasible. It can be whether a txt file or an R data frame. The function returns the preprocessed coverage. An txt file of the preprocessed coverage is optional to be exported.

Usage

```
coveragePreprocess(
  coverage = NULL,
  maf = NULL,
  output_file = TRUE,
  quiet = FALSE
)
```

Arguments

coverage	Coverage file, can be whether an R data frame or the path of a txt file.
maf	Mutation Annotation Format, can be whether an R data frame or the path of a txt file.
output_file	Determine whether to export the preprocessed maf as a txt file.
quiet	Whether to show notes during processing.

Value

The preprocessed coverage file.

Examples

```
coverage <- system.file("extdata", "coverage.rda", package = "DriverGenePathway")
load(coverage)
laml_maf <- system.file("extdata", "tcga_laml.maf", package = "DriverGenePathway")
coverage <- coveragePreprocess(coverage, laml_maf)
```

---

driverGenes *Identify significante genes with hypothetical test methods.*

Description

This function Applies different hypothetical test methods to identify significant genes. It outputs the genes and their q-values.

Usage

```
driverGenes(
  bmr = NULL,
  p_class = "all",
  maf = NULL,
  coverage = NULL,
  covariate = NULL,
  original_mutation_rate = 1.2e-06,
  max_neighbors = 50,
  ref_genome = NULL,
  category_num = NULL,
  output_file = TRUE,
  filter = TRUE,
  sigThreshold = 0.1,
  quiet = FALSE
)
```

Arguments

`bmr` A list output by the backgroundMutationRate function.

`p_class` Hypothetical test methods. "betaBinomial" represents beta binomial test; "fisherBinomial" represents Fisher combined P-value test; "likelihoodRatio" represents likelihood ratio test; "convolution" represents convolution test; "projection" represents 2D projection method; "all" represents applying all above methods.

`maf` Mutation Annotation Format, can be whether an R data frame or the path of a txt file.

`coverage` Coverage file, can be whether an R data frame or the path of a txt file.

`covariate` Covariate file, can be whether an R data frame or the path of a txt file.

`original_mutation_rate` The default background mutation rate is 1.2e-6, which is used to screen a subset of genes through binomial test.

`max_neighbors` Max number of gene neighbors used to calculate the background mutation rate of a specified gene.

`ref_genome` Reference chromosome, either a folder name or a installed BSgenome package. Default NULL, tries to auto-detect from installed genomes.

`category_num` Number of mutation categories, default 4. If the number is 0, categories should exist.

`output_file` Whether to export the results as csv files and pngs.

`filter` Whether to filter genes by q-values.

`sigThreshold` The threshhold of q-value to judge if the gene is significant

`quiet` Whether to show notes during processing.

Details

This function searches the significant genes using different hypothetical test methods, including beta binomial test, Fisher combined P-value test, likelihood ratio test, convolution test and 2D projection method. There are two ways to run driverGenes. Since driverGenes can be treated as the next step of the backgroundMutationRate function, the output of backgroundMutationRate can one form of input. The other way is by inputting the maf, coverage and covariate files, and driverGenes automaticaly runs the backgroundMutationRate procedure. The output of backgroundMutationRate corresponds to the argument "bmr".

Value

Signifiant genes

Examples

```
laml_maf <- system.file("extdata", "tcga_laml.maf", package = "DriverGenePathway")
coverage <- system.file("extdata", "coverage.rda", package = "DriverGenePathway")
covariate <- system.file("extdata", "gene.covariates.txt", package = "DriverGenePathway")
load(coverage)
bmr_result <- backgroundMutationRate(laml_maf, coverage, covariate, original_mutation_rate = 1.2e-6,
                                max_neighbors = 50, ref_genome = NULL, category_num = 1,
                                    output_file = TRUE, quiet = FALSE)
driverGenes(bmr_result, p_class = "betaBinomial", output_file = TRUE, filter = TRUE,
            sigThreshold = 0.1, quiet = FALSE)

driverGenes(p_class = "betaBinomial", maf = laml_maf, coverage = coverage, covariate = covariate,
original_mutation_rate = 1.2e-6, max_neighbors = 50, ref_genome = NULL, category_num = 1,
output_file = TRUE, filter = TRUE, sigThreshold = 0.1, quiet = FALSE)
```

`driverPathway` *Search driver pathway using de novo method*

Description

This function searches driver pathway using de novo method based on mutual exclusivity and coverage It outputs the driver pathway and its p-value as a txt file.

Usage

```
driverPathway(
  mutation_matrix = NULL,
  maf = NULL,
  threshold = 5,
  analyse_genes = NULL,
  exclude_genes = NULL,
  analyse_patients = NULL,
  exclude_patients = NULL,
  background_gene_filter = FALSE,
  coverage = NULL,
  covariate = NULL,
  original_mutation_rate = 1.2e-06,
  ref_genome = NULL,
  category_num = NULL,
  driver_size = 3,
  pop_size = 200,
  iters = 200,
  permut_time = 500,
  output_file = TRUE,
  quiet = FALSE
)
```

Arguments

`mutation_matrix`
Matrix of mutations occur in which patient and which gene.

`maf` Mutation Annotation Format, can be whether an R data frame or the path of a txt file.

`threshold` Minimum mutation count for genes to be analysed, default to be 5.

`analyse_genes` Genes to be analysed.

`exclude_genes` Genes excluded from analysis.

`analyse_patients`
Patrients to be analysed.

`exclude_patients`
Patrients excluded from analysis.

`background_gene_filter`
Exclude genes having fewer mutations than expected using a binomial test.

`coverage` Coverage file, can be whether an R data frame or the path of a txt file.

`covariate` Covariate file, can be whether an R data frame or the path of a txt file.

`original_mutation_rate`
The default background mutation rate is 1.2e-6, which is used to screen a subset of genes through binomial test.

`ref_genome` Reference chromosome, either a folder name or a installed BSgenome package. Default NULL, tries to auto-detect from installed genomes.

`category_num` Number of mutation categories, default 4. If the number is 0, categories should exist.

`driver_size` Size of output driver gene set, default to be 3.

pop_size	Population size of GA, default to be 200.
iters	Time of iteration of GA, default to be 500.
permut_time	Time of Permutation test, default to be 1000.
output_file	Determine whether to export the preprocessed maf as a txt file.
quiet	Whether to show notes during processing.

Examples

```
mutation_matrix <- system.file("extdata", "sample_mutation_matrix.rda",
package = "DriverGenePathway")
load(mutation_matrix)
driverPathway(mutation_matrix, driver_size = 3, pop_size = 30, iters = 100, permut_time = 50)

laml_maf <- system.file("extdata", "tcga_laml.maf", package = "DriverGenePathway")
driverPathway(maf = laml_maf, driver_size = 3, pop_size = 30, iters = 100, permut_time = 50)
```

mafCategoryAssign *Assign categories for maf and coverage*

Description

The mutation categories generated from 192 basic types to designated number of types. The reference chromosome is necessary if the number of categories is set more than one. The chromosome can either be name of a folder containing txt files or name of the installed BSgenome package. Example: BSgenome.Hsapiens.UCSC.hg19. The preprocess of maf and coverage is integrated in this function.

Usage

```
mafCategoryAssign(
  maf = NULL,
  coverage = NULL,
  ref_genome = NULL,
  category_num = NULL,
  output_file = TRUE,
  quiet = FALSE
)
```

Arguments

maf	Mutation Annotation Format, can be whether an R data frame or the path of a txt file.
coverage	Coverage file, can be whether an R data frame or the path of a txt file.
ref_genome	Reference chromosome, either a folder name or a installed BSgenome package. Default NULL, tries to auto-detect from installed genomes.
category_num	Number of mutation categories, default 4. If the number is 0, categories should exist.
output_file	Determine whether to export the categories, categorised maf and coverage as txt files.
quiet	Whether to show notes during processing.

Value

Categorised maf and coverage.

Examples

```
laml_maf <- system.file("extdata", "tcga_laml.maf", package = "DriverGenePathway")
coverage <- system.file("extdata", "coverage.rda", package = "DriverGenePathway")
load(coverage)
Categ_result <- mafCategoryAssign(laml_maf, coverage, category_num = 1, output_file = TRUE,
quiet = FALSE)
```

---

mafPreprocess *Preprocess maf*

Description

This function guarantees that the maf is available for DriverGenePathway. Two kinds of maf input are feasible. It can be whether a txt file or an R data frame. The function returns the preprocessed maf. An txt file of the preprocessed maf is optional to be exported.

Usage

```
mafPreprocess(maf = NULL, output_file = TRUE, quiet = FALSE)
```

Arguments

`maf`	Mutation Annotation Format, can be whether an R data frame or the path of a txt file.
`output_file`	Determine whether to export the preprocessed maf as a txt file.
`quiet`	Whether to show notes during processing.

Value

The preprocessed maf.

Examples

```
laml_maf <- system.file("extdata", "tcga_laml.maf", package = "DriverGenePathway")
maf <- mafPreprocess(laml_maf)
```

mafToMatrix *Transform maf to mutation matrix*

Description

This function transforms maf files to mutation matrix in order to search driver pathways.

Usage

```
mafToMatrix(
  maf = NULL,
  threshold = 5,
  analyse_genes = NULL,
  exclude_genes = NULL,
  analyse_patients = NULL,
  exclude_patients = NULL,
  output_file = TRUE,
  quiet = FALSE
)
```

Arguments

maf	Mutation Annotation Format, can be whether an R data frame or the path of a txt file.
threshold	Minimum mutation count for genes to be analysed, default to be 5.
analyse_genes	Genes to be analysed.
exclude_genes	Genes excluded from analysis.
analyse_patients	Patrients to be analysed.
exclude_patients	Patrients excluded from analysis.
output_file	Determine whether to export the preprocessed maf as a txt file.
quiet	Whether to show notes during processing.

Examples

```
laml_maf <- system.file("extdata", "tcga_laml.maf", package = "DriverGenePathway")
mafToMatrix(maf = laml_maf)
```

preprocessAssignCateg *Assign categories for mutation types*

Description

For each of the 192 mutation types, assigning a corresponding category from number 1 to 4.

Usage

```
preprocessAssignCateg(K)
```

Arguments

`K`	Dataframe of previously detected mutation categories

`preprocessCollapseNn65to32` *Collapsing category number to 32*

Description

Collapsing previous category number from 65 to 32.

Usage

```
preprocessCollapseNn65to32(Y)
```

Arguments

`Y`	Dataframe of numbers of coverage and 4 mutation types of 65 categories

Details

This function collapses the previous category number from 65 to 32 by first removing the "any N" type, then fusing a specific trinucleotide and its opposite.

`preprocessFindMutCateg` *Finding mutation categories*

Description

Turning the number of categories from 32 to a reasonable expected value, through analysing the numbers of mutations and coverages.

Usage

```
preprocessFindMutCateg(Nn, P)
```

Arguments

`Nn`	The frame containing numbers of coverages and mutations for each of the 32 types
`P`	The frame containing the expected number of categs and the name of output mutation file

Details

Through analysing the mutation numbers of the 32 types previously defined, this function finds several reasonable mutation categories as categs. The default number of categs is 4.

preprocessGenerateCategContext65

Generating 65 mutation categories

Description

Generating 65 types of mutaion categories for subsequently classification and analysis.

Usage

```
preprocessGenerateCategContext65()
```

Details

Considering the contexts for each of the 4 bases(A,C,G,T), i.e. the total number of categories reaches to 65(64 types of trinucleotides and a supplement called "any N").

Index