Package 'ScreenBEAM2'

August 8, 2019

Type Package

Title A novel meta-
analysis algorithm for functional genomics screens via Bayesian hierarchical modeling, Version 2

Version 0.1.0

Author@R person('Xinge', 'Wang', email = 'wang.xinge@outlook.com')

Maintainer@R person('Xinge', 'Wang', email = 'wang.xinge@outlook.com')

Description ScreenBEAM2 is the upgraded version of ScreenBEAM1. It can perform functional genomics screening from FASTQ file. It is a trim-free, allowing number of mismatch, BLAT-depend pipeline. Downstream analysis uses Bayesian hierarchical modeling, removed bias caused by the unbalanced number of sh/sgRNA targeting the same gene.

License GPL(>=2)

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1.9000

R topics documented:

ScreenBEAM.check.lib 1
ScreenBEAM.dir.create 2
ScreenBEAM.gene.level 3
ScreenBEAM.mapping.QC 4
ScreenBEAM.raw.count 5
ScreenBEAM.rna.level 5
ScreenBEAM.trim.helper 6

Index **8**

`ScreenBEAM.check.lib` *Check sh/sgRNA library and Remove duplicated/substring sh/sgRNA*

Description

`ScreenBEAM.check.lib` is a pre-processing function for `ScreenBEAM.raw.count`. It requires the file path of the library file (.csv file, with 1st column to be RNAid, 2nd column RNA sequence, 3rd column target gene name). It will remove the duplicated/substring sh/sgRNA and store it into a list.

Usage

```
ScreenBEAM.check.lib(lib.path)
```

Arguments

lib.path	character, the file path of the library. Should be a csv file. The 1st column is RNAid, 2nd column RNA sequence and 3rd column target gene name.

Value

Return a list object, containing whole library information after removed duplicates; The most command sh/sgRNA length; Total sh/sgRNA number; All the sh/sgRNA names; Each sh/sgRNA length; Removed sh/sgRNA information.

ScreenBEAM.dir.create *Manipulation of Working Directories for ScreenBEAM2 analysis*

Description

ScreenBEAM.dir.create is used to help users create an organized working directory for the Functional Genomics Screening Projects. It creates a hierarchcial working directory and returns a list contains this directory information.

Usage

```
ScreenBEAM.dir.create(project_main_dir = NULL, lib_name = NULL,
  DATE = TRUE)
```

Arguments

lib_name	character, name of the sh/sgRNA library. Each main working directory can contain multiple library sub-project folder.
DATE	logical, if TRUE, current date information will be added to the sub-project folder name. Default is TRUE.
projecet_main_dir	character, name or absolute path of the main working directory.

Details

This function needs users to define the main working directory and the library's name. It creates a main working directory with a subdirectory of the project(each library is a project). It also automatically creates 4 subfolders (fastq, metadata, library and output) within the project folder. fastq/, storing unzipped fastq files of each sample; metadata/, storing the experiment design, describe detailed information of each sample; library/, storing sh/sgRNA library csv file and its fasta file (for detail, please check demo). output/ folder contains 3 subfolder to store intermediate results generated by ScreenBEAM2 analysis. They are mapping/, QC/ and DR/.

Inside mapping/ there are 3 subfolders as well. mapping/Step1_unique_fast, mapping/Step2_blat and mapping/Step3_raw_count. mapping/Step1_unique_fast stores unique fasta read from each sample's fastq file. mapping/Step2_blat stores blat result when using unique fasta file and library fasta file. mapping/Step3_raw_count stores 3 raw count table created later in the pipeline. This function also returns a list object (analysis.par in the demo) with directory information wrapped

inside. This list is an essential for ScreenBEAM2 analysis, all the important intermediate data generated later will be wrapped inside.

QC/ is used to store quality control analysis of the mapping result. Including library check and mapping statistics.

DR/ is used to store the Bayesian hierarchical meta analysis result.

Value

Return a list object, containing all the paths of the created folder.

`ScreenBEAM.gene.level` *GENE level meta-analysis of high-throughput Functional Genomics Screening analysis*

Description

`ScreenBEAM.gene.level` takes a ".tsv" table file with samples raw count number, user needs to assign the control sample's name and case sample's name vector. It will do a gene-level meta-analysis of screening data. User also need to assign value to `rna.size` to the most common number of sh/sgRNAs targeting one gene. This function will remove bia caused by unbalanced number of sh/sgRNAs targeting the same gene.

Usage

```
ScreenBEAM.gene.level(input.file, control.samples, case.samples,
  control.groupname = "control", case.groupname = "treatment",
  gene.columnId = 2, data.type = c("microarray", "NGS"),
  do.normalization = TRUE, total = 1e+06, filterLowCount = FALSE,
  filterBy = "control", count.cutoff = 4, nitt = 15000,
  burnin = 5000, thin = 10, rna.size = 6, sample.rna.time = 100,
  method = "Bayesian", pooling = "partial", ...)
```

Arguments

`input.file,`	character, the file path of ".tsv" table with raw count number for each sample (column) and sh/sgRNA (row).
`control.samples,`	vector of character, the control samples names, this information should match the column names of input.file. For example, c(T0_A, T0_B).
`case.samples,`	vector of character, the case samples names, this information should match the column names of input.file. For example, c(T16_A, T16_B).
`control.groupname,`	character, the name of your control samples' group name. For example, T0.
`case.groupname,`	character, the name of your case samples' group name. For example, T16.
`gene.columnId,`	integer, the number of the column which store the gene name. By default, 2.
`data.type,`	character, can either be "microarray" or "NGS". Default is "microarray".

`do.normalization`, logical, if TRUE, a scaled normalization will be performed for each sample. To quantify this scale, user need to assign a value to `total`. Default is TRUE.

`total`, integer, need to be larger than the colSums of the raw count table. Default is 1e6.

`filterLowCount`, logical, if TRUE, will remove rnas with low count, based on `filterBy` and `count.cutoff`. Default is TRUE.

`filterBy`, logical, if TRUE, will remove rnas with low count based on "control" samples or "case" samples. Default is "control".

`count.cutoff`, integer, the threshold of removing low count. Default is 4.

`nitt`, integer, the number of MCMC iterations. Default is 15000.

`burnin`, integer, burnin. Default is 5000.

`thin`, integer, thinning interval. Default is 10.

`rna.size`, integer, the most common number of sh/sgRNAs targetting the same gene. Default is 6.

`sample.rna.time`, integer, to remove unbalance number of sh/sgRNAs targetting the same gene. `rna.size` of sh/sgRNAs will be sampled, the sampling time.

`method`, character, estimation model. Either "Bayesian" or "MLE". Default is "Bayesian".

`pooling`, character, pooling method. Either "full" or "partial". Default is "partial".

Value

Return a data frame containing all the gene-level meta-analysis statistical values. Including geneID, log2FC, z value, p value, FDR and so on.

`ScreenBEAM.mapping.QC` *Create an HTML report to perform quality control of library, mapping status*

Description

`ScreenBEAM.mapping.QC` takes the master `analysis.par` list, which contains all the key data through ScreenBEAM meta-analysis. It will call Rmarkdown file to create an HTML report of library quality control (show the deleted duplicated/substring sh/sgRNA table). Percentage of mapping. Suggested `rna.size`, which is the most common number of sh/sgRNAs targeting the same gene. A barplot at the end shows the suggested number of mismatch for user to rescure the most reads.

Usage

```
ScreenBEAM.mapping.QC(analysis.par, QC.Rmd.path = system.file("Rmd",
  "mapping_QC_report.Rmd", package = "ScreenBEAM2"))
```

Arguments

`analysis.par`, the master list, containing unique reads files path, blat result files path.

`QC.Rmd.path`, character, the path where user put the Rmarkdown file of "mapping_QC_report". Default, it will call the one in ScreenBEAM2 package.

Value

It will create an Rmarkdown file and an HTML QC report in the folder of `analsis.par$out.dir.output.QC`.

`ScreenBEAM.raw.count` *Create raw count table for NGS data*

Description

`ScreenBEAM.raw.count` takes a list object, which contains all the paths of library information, unique fasta read path information and blat result path information. This function calculates basic statistics for the FASTQ raw reads, checks library quality, calculates mapping statistics and also collecting raw count numbe from blat result tables. This function can be run directly, or run step by step from its source code.

Usage

```
ScreenBEAM.raw.count(analysis.par)
```

Arguments

`analysis.par`	list, the master list storing all the information along ScreenBEAM2 analysis.

Value

Return an updated list object, wrapping the raw count table information inside.

`ScreenBEAM.rna.level` *RNA level meta-analysis of high-throughput Functional Genomics Screening analysis*

Description

`ScreenBEAM.rna.level` takes a ".tsv" table file with samples raw count number, user needs to assign the control sample's name and case sample's name vector. It will do a rna-level meta-analysis of screening data.

Usage

```
ScreenBEAM.rna.level(input.file, control.samples, case.samples,
  control.groupname, case.groupname, gene.columnId = 2,
  filterLowCount = TRUE, filterBy = "control", count.cutoff = 16,
  do.normalization = TRUE, total = 1e+06, do.log2 = TRUE,
  pooling = "full", family = gaussian,
  estimation.method = "Bayesian")
```

Arguments

`input.file,`	character, the file path of ".tsv" table with raw count number for each sample (column) and sh/sgRNA (row).
`control.samples,`	vector of character, the control samples names, this information should match the column names of input.file. For example, c(T0_A, T0_B).
`case.samples,`	vector of character, the case samples names, this information should match the column names of input.file. For example, c(T16_A, T16_B).
`control.groupname,`	character, the name of your control samples' group name. For example, T0.
`case.groupname,`	character, the name of your case samples' group name. For example, T16.
`gene.columnId,`	integer, the number of the column which store the gene name. By default, 2.
`filterLowCount,`	logical, if TRUE, will remove rnas with low count, based on `filterBy` and `count.cutoff`. Default is TRUE.
`filterBy,`	logical, if TRUE, will remove rnas with low count based on "control" samples or "case" samples. Default is "control".
`count.cutoff,`	integer, the threshold of removing low count. Default is 16.
`do.normalization,`	logical, if TRUE, a scaled normalization will be performed for each sample. To quantify this scale, user need to assign a value to `total`. Default is TRUE.
`total,`	integer, need to be larger than the colSums of the raw count table. Default is 1e6.
`do.log2,`	logical, if TRUE, the expression data will be log2 transformed.
`pooling,`	character, pooling method. Either "full" or "partial". Default is "full".
`family,`	function, default if gaussian.
`estimation.method,`	character, estimation model. Either "Bayesian" or "MLE". Default is "Bayesian".

Value

Return a data frame containing all the rna-level meta-analysis statistical values. Including sh/sgRNA ID, log2FC, p.value and so on.

`ScreenBEAM.trim.helper`

Create a pdf plot to visualize the trim location.

Description

`ScreenBEAM.trim.helper` is a helper function for users who wants to know the most common trim place in their FASTQ reads. This function requires the linux command `seqtk`. Please make sure you have it before calling this function.

Usage

```
ScreenBEAM.trim.helper(fastq.path, sample_num = 10000, pdf.file)
```

Arguments

fastq.path,	character, the file path of one FASTQ reads file. Unzipped.
sample_num,	integer, the random sample number of FASTQ reads when creating the barplot. Default is 10000.
pdf.file,	character, the file path to store the output pdf plot.

Value

It will create a PDF barplot file showing A,T,C,G and N's percentage at each base position. User can pick the trim location based on the plot.

Index

ScreenBEAM.check.lib, 1
ScreenBEAM.dir.create, 2
ScreenBEAM.gene.level, 3
ScreenBEAM.mapping.QC, 4
ScreenBEAM.raw.count, 5
ScreenBEAM.rna.level, 5
ScreenBEAM.trim.helper, 6