Package 'csmFinder'

April 9, 2019

Type Package

Title Find cell-subset methylation (CSM) region in single-cell methylomes or bulk methylomes

Version 0.1.0

Author Liduo Yin, Xiaowei Wu, Jianlin He, Yanting Luo

Maintainer The package maintainer <yinliduo@big.ac.cn>

Description This package is used for identifying putative CSM loci from methylation datasets generated by single cells or bulk tissue. For single-cell bisulfite sequencing datasets, a beta mixture model is used for divide the single-cell into two subsets with hypo and hypermethylation states in candidate CSM regions. For regular bisulfite sequencing datasets, a Nonparametric Bayesian clustering is used to identify the 4-CpG segments with biplolar methylation patterns.

License What license is it under?

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

R topics documented:

`bismark2segment` *Process bismark report file to 4-CpG segments information for pCSM loci identification*

Description

This function is used for processing data into the format that could be recognised by beta mixture model or nonparametric Bayesian clustering algorithm to identify pCSM loci.

Usage

```
bismark2segment(files,file_type="regular",split_by_chrom=FALSE)
```

Arguments

`files`	File or files with CpG methylation information in each sequenced read generated by bismark_extractor in ".gz" compressed format. Note that only one filename with full path is needed for regular methylation dataset and a vector containning the filename with full path of each single-cell is needed for processing single-cell datasets.
`file_type`	Type of input dataset with "regular" denotes the regular methylation data and "single-cell" denotes single-cell mathylation data.
`split_by_chrom`	Logical; Used for single-cell datasets when the number of cells is huge. Note that by setting split_by_chrom=TRUE, a list will be returned with each elements denotes the input of beta mixture model for one chromsome

Value

`segment`	A matrix or a list containning the 4CpG segments infromation for pCSM loci identification.

Note

loading and processing the CpG index may need several minutes

Examples

```
file <- paste(system.file(package='csmFinder'),'extdata/bulk_CpG_extract_file/chr19.demo.dataset.gz',sep='/
segment <- bismark2segment(file)

############################

file_dir <- paste(system.file(package='csmFinder'),"extdata/single_cell_CpG_extract_file",sep='/')
file_list <- paste(file_dir,list.files(file_dir),sep='/')
segment <- bismark2segment(files=file_list,file_type="single-cell")
```

`calculate_ml_in_csm` *Calculate methylation level in pCSM loci*

Description

Calculation average methylation level of pCSM loci

Usage

```
calculate_ml_in_csm(csm_bed,methy_profile)
```

Arguments

`csm_bed`	The coordinate of pCSM loci in genome with "bed" format
`methy_profile`	Methylation profile of the sample

Value

A matrix with each row denotes one pCSM loci, the first collumn denotes the methylation level and the second collumd denotes the number of CpG loci in such pCSM loci.

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (x)
{
  }
```

`co_methylation_step1` *The first step of co-methylation analysis*

Description

The first step of the co-methylation analysis, including kmeans analysis to group pCSM loci into three clusters, i.e. hypo/mid/hyper-methylation cluster, and, for each kmeans cluster, the network topology analysis function in WGCNA package is called to pick the soft-thresholding power.

Usage

```
co_methylation_step1(csm_ml_matrix,plot=FALSE)
```

Arguments

`csm_ml_matrix`	methylation profile of pCSM loci in each sample
`plot`	Logical; determine whether to produce the figure with methylation level of pCSM loci in each kmeans cluster

Value

A list the following two components:

profile	the methylation profile of pCSM loci
modult_id	the label tells that which co-methylation module the pCSM loci belong to

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (x)
{
  }
```

---

co_methylation_step2 *The second step of co-methylation analysis*

Description

The second step of the co-methylation analysis, i.e. co-methylation analysis in each kmeans cluster based on WGCNA package.

Usage

```
co_methylation_step2(kmeans_data,softPower_list,plot=FALSE)
```

Arguments

kmeans_data	a kmeans object including the data and cluster information
softPower_list	a numeric vector contains 3 soft-thresholding power for WGCNA analysis in each kmeans cluster
plot	Logical; determine whether to produce the figures with methylation level of pCSM loci in each WGCNA cluster

Value

An object of "kmeans" with cluster information.

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (x)
{
  }
```

csmFinder *Find cell-subset methylation region in genome*

Description

Find cell-subset methylation segments from methylation datasets generated by single cells or bulk tissue. For single-cell bisulfite sequencing datasets, a beta mixture model is used for divide the single-cell into two subsets with hypo and hyper-methylation states in candidate CSM segments. For regular bisulfite sequencing datasets, a Nonparametric Bayesian clustering is used to identify the 4-CpG segments with biplolar methylation patterns.

Usage

```
csmFinder(candidate,data_type='regular',depth=10,distance=0.3,pval=0.05,thread=1)
```

Arguments

candidate	the candidate segments used for CSM identification
data_type	"regular" and "single-cell" denotes regular datasets and single-cell datasets, respectively
depth	number of reads (for regular datasets)or cells (for single-cell datasets) covered the candidate segments
distance	methylation difference between hypo and hyper-methylated cells subsets or reads
pval	significance of the differnece between hypo and hyper-methylated cells subsets or reads
thread	number threads used to identify candidate pCSM segment

Value

For single-cell dataset, the output is in the same format with the output of beta mixture model(https://github.com/Evan-Evans/Beta-Mixture-Model). For regular methylation datasets, the output is a matrix contains the methylation difference between hypo and hyper-methylated reads, and its significance.

References

Wu, X., et al., 2015, Nonparametric Bayesian clustering to detect bipolar methylated genomic loci, BMC Bioinformatics, 16.

Luo, Y., et al., 2018, Integrative single-cell omics analyses reveal epigenetic heterogeneity in mouse embryonic stem cells, PLoS computational biology, 14, e1006034

Examples

```
pCSM_segment <- csmFinder(candidate,data_type='regular')
pCSM_segment2 <- csmFinder(candidate2,data_type='single-cell',depth=5)
```

`extract_eigen` *Extract eigen-pCSM loci from each co-methylation module*

Description

PCA analysis is performed and the loci with the largest loadings in PC1 will be picked as eigen-pCSM lcoi

Usage

```
extract_eigen(csm.ml ,all_label , number_of_eig )
```

Arguments

`csm.ml`	The methylation profile all pCSM loci
`all_label`	A character vector containning the module id that each pCSM loci belongs to
`number_of_eig`	Number of eigen-pCSM loci need to be extracted

Value

`methy_prof`	methylation profile of eigen-pCSM
`nmf.input.label`	The module id for each eigen-pCSM loci

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (x)
{
  }
```

`find_candidate` *Find pCSM candidate segments*

Description

Determine the pCSM candidate segments statify depth cuttoff, and for single-cell datasets, candidate are determined as the segments covered by both methylated cell and unmethylated cell. For regular datasets, candidates are determined as the segments with totally methylated read and unmethylated read.

Usage

```
find_candidate(segment,depth=10,thread=1,data_type='regular')
```

Arguments

segment	matrix with segment information
depth	number of reads (for regular datasets)or cells (for single-cell datasets) covered the candidate segments
thread	number threads used to identify candidate pCSM segment
data_type	"regular" and "single-cell" denotes regular datasets and single-cell datasets, respectively

Examples

```
candidate <- find_candidate(segment)
candidate2 <- find_candidate(segment2,data_type="single-cell")
```

---

merge_segment *Merge overlapped 4-CpG segments into pCSM regions*

Description

This function is used to convert the 4-CpG segments into pCSM loci/region

Usage

```
merge_segment(pCSM_segment,data_type="regular")
```

Arguments

pCSM_segment	the segments determined as the pCSM
data_type	"regular" and "single-cell" denotes regular datasets and single-cell datasets, respectively

Examples

```
pcsm_loci <- merge_segment(pcsm)
pcsm_loci2 <- merge_segment(pcsm2,data_type="single-cell")
```

---

run.beta.mixture.model *beta mixure model to decect pCSM loci*

Description

This function is used for identifying the pCSM loci from single-cell methylomes. Briefly, a beta mixture model is utilized to divide the single-cells with hyper and hypo-methylation state into different cell subsets in a given CSM candidate region and determine the significance.

Usage

```
run.beta.mixture.model(candidate,thread=1,is.candidate=rep(1,nrow(candidate)))
```

Arguments

candidate	A matrix containning the candidate pCSM loci with the format produced by find.candidate.
thread	Number of thread used to be tun beta mixture model.
is.candidate	A numeric vector containning the candidate information for the input loci, with 1 denotes candidate and 0 denotes non-condidate loci.

Value

beta.mixture.output	The matrix of beta mixture model output
is.csm	A nuneric vector containning the index of input loci determined as pCSM loci

References

https://github.com/Evan-Evans/Beta-Mixture-Model

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (x)
{
  }
```

Index