Package ‘CREAM’

July 21, 2025

Type Package

Title Clustering of Genomic Regions Analysis Method

Version 1.1.1

Date 2018-05-30

Description Provides a new method for identification of clusters of genomic regions within chromosomes. Primarily, it is used for calling clusters of cis-regulatory elements (COREs). 'CREAM' uses genome-wide maps of genomic regions in the tissue or cell type of interest, such as those generated from chromatin-based assays including DNaseI, ATAC or ChIP-Seq. 'CREAM' considers proximity of the elements within chromosomes of a given sample to identify COREs in the following steps: 1) It identifies window size or the maximum allowed distance between the elements within each CORE, 2) It identifies number of elements which should be clustered as a CORE, 3) It calls COREs, 4) It filters the COREs with lowest order which does not pass the threshold considered in the approach.

License GPL (>= 3)

Imports stats, utils

Depends R (>= 3.3)

URL https://github.com/bhklab/CREAM

Suggests testthat

RoxygenNote 6.0.1

LazyData true

biocViews PeakDetection, FunctionalPrediction, BiomedicalInformatics, Clustering

BugReports https://github.com/bhklab/CREAM/issues

Encoding UTF-8

NeedsCompilation no

Author Seyed Ali Madani Tonekaboni [aut],
Victor Kofia [aut],
Mathieu Lupien [aut],
Benjamin Haibe-Kains [aut, cre]

Maintainer Benjamin Haibe-Kains <benjamin.haibe.kains@utoronto.ca>

Repository CRAN

Date/Publication 2018-06-06 09:07:56 UTC

Contents

CREAM 2
ElementRecog 3
PeakMinFilt 4
WindowSizeRecog 4
WindowVec 5

Index **7**

CREAM *CREAM is the main function for CORE identification*

Description

CREAM is the main function for CORE identification

Usage

```
CREAM(in_path, WScutoff = 1.5, MinLength = 1000, peakNumMin = 2)
```

Arguments

`in_path`	Path to the input file (The file inclusing the functional regions) Note. You have to make sure that there is no overlapping regions within the input file
`WScutoff`	Threshold used to identify WS within distribution of maximum distance between peaks for each order of CORE
`MinLength`	Criteria for the minimum number of functional regions in the input file
`peakNumMin`	Minimum number of peaks for CORE identification

Value

Bed file including the identified COREs

Examples

```
CREAM(system.file("extdata", "A549_Chr21.bed", package = "CREAM"),
MinLength = 1000, peakNumMin = 2)
```

ElementRecog — *ElementRecog is a function to identify COREs*

Description

ElementRecog is a function to identify COREs

Usage

```
ElementRecog(InputData, windowSize_Vec, peakNumMax, peakNumMin)
```

Arguments

`InputData`	The input data as a table including chromosome regions in which the first column is chromosome annotation, and second and third columns are start and ending positions.
`windowSize_Vec`	Vector of window sizes ordered based on order of CORE
`peakNumMax`	Maximum order of COREs (e.g. maximum number of peaks within COREs)
`peakNumMin`	Minimum order of COREs (e.g. minimum number of peaks within COREs)

Value

Identified COREs for the given input regions

Examples

```
InputData <- read.table(system.file("extdata", "A549_Chr21.bed",
package = "CREAM"), sep="\t")
colnames(InputData) <- c("chr", "start", "end")
MinLength <- 1000
if(nrow(InputData) < MinLength){
   stop(paste( "Number of functional regions is less than ", MinLength,
   ".", sep = "", collapse = ""))
}
peakNumMin <- 2
WScutoff <- 1.5
WindowVecFinal <- WindowVec(InputData, peakNumMin, WScutoff)
OutputList <- ElementRecog(InputData, WindowVecFinal,
(1+length(WindowVecFinal)), peakNumMin)
```

PeakMinFilt *PeakMinFilt is a function to filter the lowest Order of COREs which distance between functional regions is close to the corresponding Window Size*

Description

PeakMinFilt is a function to filter the lowest Order of COREs which distance between functional regions is close to the corresponding Window Size

Usage

```
PeakMinFilt(Clusters_init, WindowVecFinal)
```

Arguments

`Clusters_init`	Table of indetified COREs before filteration
`WindowVecFinal`	Vector of window sizes ordered based on order of CORE

Value

Minimum order of COREs

WindowSizeRecog *WindowSizeRecog is a function to specify window size for each order of COREs*

Description

WindowSizeRecog is a function to specify window size for each order of COREs

Usage

```
WindowSizeRecog(InputData, COREorder, WScutoff)
```

Arguments

`InputData`	The input data as a table including chromosome regions in which the first column is chromosome annotation, and second and third columns are start and ending positions.
`COREorder`	Order of the COREs which window size has to be determined for.
`WScutoff`	Threshold used to identify WS within distribution of maximum distance between peaks for each order of CORE

Value

Window size identified for each order of CORE

Examples

```
InputData <- read.table(system.file("extdata", "A549_Chr21.bed",
package = "CREAM"), sep="\t")
colnames(InputData) <- c("chr", "start", "end")
MinLength <- 1000
if(nrow(InputData) < MinLength){
   stop(paste( "Number of functional regions is less than ", MinLength,
   ".", sep = "", collapse = ""))
}
peakNumMin <- 2
WScutoff <- 1.5
WindowSize <- WindowSizeRecog(InputData, peakNumMin, WScutoff)
```

`WindowVec` *WindowVec is a function to specify window size for each order of COREs*

Description

WindowVec is a function to specify window size for each order of COREs

Usage

```
WindowVec(InputData, peakNumMin, WScutoff)
```

Arguments

`InputData` The input data as a table including chromosome regions in which the first column is chromosome annotation, and second and third columns are start and ending positions.

`peakNumMin` Minimum order of COREs

`WScutoff` Threshold used to identify WS within distribution of maximum distance between peaks for each order of CORE

Value

Vector of window sizes from order 2 up to maximum order of COREs

Examples

```
InputData <- read.table(system.file("extdata", "A549_Chr21.bed",
package = "CREAM"), sep="\t")
colnames(InputData) <- c("chr", "start", "end")
MinLength <- 1000
if(nrow(InputData) < MinLength){
   stop(paste( "Number of functional regions is less than ", MinLength,
   ".", sep = "", collapse = ""))
}
peakNumMin <- 2
WScutoff <- 1.5
WindowVecFinal <- WindowVec(InputData, peakNumMin, WScutoff)
```

Index

CREAM, 2

ElementRecog, 3

PeakMinFilt, 4

WindowSizeRecog, 4
WindowVec, 5