Package 'rmspc'

April 26, 2024

Type Package

Title Multiple Sample Peak Calling

Version 1.9.0

Imports processx, BiocManager, rtracklayer, stats, tools, methods, GenomicRanges, stringr

Suggests knitr, rmarkdown, BiocStyle, testthat (>= 3.0.0)

VignetteBuilder knitr

Description The rmspc package runs MSPC (Multiple Sample Peak Calling) software using R. The analysis of ChIP-seq samples outputs a number of enriched regions (commonly known as ``peaks''), each indicating a protein-DNA interaction or a specific chromatin modification. When replicate samples are analyzed, overlapping peaks are expected. This repeated evidence can therefore be used to locally lower the minimum significance required to accept a peak. MSPC uses combined evidence from replicated experiments to evaluate peak calling output, rescuing peaks, and reduce false positives. It takes any number of replicates as input and improves sensitivity and specificity of peak calling on each, and identifies consensus regions between the input samples.

biocViews ChIPSeq, Sequencing, ChipOnChip, DataImport, RNASeq

License GPL-3

Encoding UTF-8

URL https://genometric.github.io/MSPC/

BugReports https://github.com/Genometric/MSPC/issues

SystemRequirements .NET 6.0

RoxygenNote 7.1.1

Config/testthat/edition 3

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

git_branch devel

git_last_commit 75871f6

git_last_commit_date 2023-10-24

Repository Bioconductor 3.19

Date/Publication 2024-04-25

Author Vahid Jalili [aut],
Marzia Angela Cremona [aut],
Fernando Palluzzi [aut],
Meriem Bahda [aut, cre]

Maintainer Meriem Bahda <meriembahda@gmail.com>

Contents

`rmspc-package` *rmspc: Multiple Sample Peak Calling*

Description

The rmspc package runs MSPC (Multiple Sample Peak Calling) software using R. The analysis of ChIP-seq samples outputs a number of enriched regions (commonly known as "peaks"), each indicating a protein-DNA interaction or a specific chromatin modification. When replicate samples are analyzed, overlapping peaks are expected. This repeated evidence can therefore be used to locally lower the minimum significance required to accept a peak. MSPC uses combined evidence from replicated experiments to evaluate peak calling output, rescuing peaks, and reduce false positives. It takes any number of replicates as input and improves sensitivity and specificity of peak calling on each, and identifies consensus regions between the input samples.

Author(s)

Maintainer: Meriem Bahda <meriembahda@gmail.com>

Authors:

- Vahid Jalili <jalili.vahid@gmail.com>
- Marzia Angela Cremona <marzia.cremona@fsa.ulaval.ca>
- Fernando Palluzzi <fernando.palluzzi@gmail.com>

See Also

Useful links:

- https://genometric.github.io/MSPC/
- Report bugs at https://github.com/Genometric/MSPC/issues

mspc *Multiple Sample Peak Calling*

Description

MSPC comparatively evaluates ChIP-seq peaks and combines the statistical significance of repeated evidences, with the aim of lowering the minimum significance required to rescue weak peaks; hence reducing false-negatives.

Usage

```
mspc(
  input,
  replicateType,
  stringencyThreshold,
  weakThreshold,
  gamma = NULL,
  c = NULL,
  alpha = NULL,
  multipleIntersections = NULL,
  degreeOfParallelism = NULL,
  inputParserConfiguration = NULL,
  outputPath = NULL,
  GRanges = FALSE,
  keep = NULL,
  directoryGRangesInput = NULL,
  mspcPath = NULL
)
```

Arguments

input	Character vector or GRanges list. The input argument is a character vector when the data the user wants to use as a sample is in a BED file format. In this case, the data should be a tab-delimited file in BED format for each replicate, each containing enriched regions (aka peaks) called with a permissive p-value threshold. The input argument will then be a character vector, each element of the vector will have the file path of the BED file the user wants to use. The input argument is a GRanges list when the user wants to use multiple GRanges objects as samples. This arguments is required. More information in Details
replicateType	Character string. This argument defines the replicate type. Possible values of the argument : 'Bio','Biological', 'Tec', 'Technical'. This arguments is required . More information in Details.
stringencyThreshold	Double. A threshold on p-values, where peaks with p-value lower than this threshold, are considered stringent.This arguments is required
weakThreshold	Double. A threshold on p-values, such that peaks with p-value between this and stringency threshold, are considered weak peaks. This arguments is required

`gamma` Double. The combined stringency threshold. Peaks with combined p-value below this threshold are confirmed. Default value is the value of the argument stringencyThreshold. This arguments is optional.

`c` Integer or character string. the minimum number of overlapping peaks required before the MSPC program combines their p-value. The value of c can be given in absolute (e.g., c = 2 will require at least 2 samples) or percentage of input samples (e.g., c = 50 will require at least 50 Default value is 1. This arguments is optional. More information in Details.

`alpha` Double. The threshold for Benjamini-Hochberg multiple testing correction. Default value is 0.05. This arguments is optional.

`multipleIntersections`
Character string. When multiple peaks from a sample overlap with a given peak, this argument defines which of the peaks to be considered: the one with lowest p-value, or the one with highest p-value? Possible value of the argument are either 'Lowest' or 'Highest'. Default value is 'Lowest'. This arguments is optional.

`degreeOfParallelism`
Integer. The number of parallel threads the MSPC program can utilize simultaneously when processing data. Default value is the number of logical processors on the current machine. This arguments is optional.

`inputParserConfiguration`
File path. The path to a JSON file containing the configuration for the input BED file parser. This is an optional argument and it has no default value.

`outputPath` Directory path. This argument sets the path in which analysis results should be persisted. This is an optional argument. More information in Details.

`GRanges` Logical. Determines whether or not the mspc function should import the files, created while running the mspc function, as GRanges objects into the R environment. The default value is FALSE. However, when the keep argument is set to FALSE, the value of the argument GRanges is set to TRUE, and the value given by the user to the GRanges argument is ignored.

`keep` Logical. This argument determines whether the mspc function should keep or delete all the files generated while running the MSPC program.This is an optional argument. More information in Details.

`directoryGRangesInput`
Folder path. When the input argument is a GRanges list, the mspc function exports it as multiple BED files. The directoryGRangesInput argument specifies the directory where these BED files should be stored. More information in Details.

`mspcPath` File path. The MSPC program that the rmspc package uses can be installed from the official Github page of the MSPC program. If the users wishes to use the version of the MSPC program he installs, he can specify the installation path of the mspc dll program installed using the mspcPath argument. The default value is NULL. If no value is given to this argument, the MSPC program used is the one included in the rmspc package.

Details

input :

The input can either be BED files or a GRanges list. Only one type of inputs is supported by the mspc function at a time, ie, the user can either give all the inputs as names of BED files, or all the inputs as GRanges objects. Therefore, the user cannot give an input argument that contains BED files and GRanges objects.

replicateType:

Samples could be biological or technical replicates. MSPC differentiates between the two replicate types based on the fact that less variations between technical replicates is expected compared to biological replicates.

c :

It sets the minimum number of overlapping peaks required before MSPC combines their p-value. For example, given three replicates (rep1, rep2 and rep3), if c = 3, a peak on rep1 must overlap with at least two peaks, one from rep2 and one from rep3, before MSPC combines their p-value; otherwise, MSPC discards the peak. If c = 2, a peak on rep1 must overlap with at least one peak from either rep2 or rep3, before MSPC combines their p-values; otherwise MSPC discards the peak.

outputPath:

When the mspc function is called, it creates a set of files in the directory specified by the argument outputPath. If no value is given to this argument, a folder is created in the current working directory, under the name "session_ + <Timestamp>". If a folder name is given to the argument outputPath, a folder under the name specified is created in the current working directory. If a given folder name already exists, and is not empty, the MSPC program will append _n where n is an integer until no duplicate is found in which analysis results should be persisted.

keep :

When the mspc function is called, it creates a set of files in the user's computer. The user can choose to keep or not the files created. When the argument keep is set to FALSE, all the files are created in a temporary folder, which is deleted after the R session is closed. When the argument keep is set to TRUE, the files are created in the folder specified by the argument outputPath. The default value of the argument keep is defined as follows : if the input argument is a GRanges object, the default value of the keep argument is FALSE. if the input argument is a character vector of the file path of input BED files, the default value of the keep argument is TRUE.

directoryGRangesInput :

The default value is the current working directory. It is important to note that when the argument keep is set to FALSE, the value of this argument is set to a temporary folder. If the input argument is a character vector of BED files names, the argument directoryGRangesInput is ignored.

Output of the mspc function :

When the value of the argument keep is set to FALSE, the argument GRanges is automatically set to TRUE. all the files are created in a temporary folder, which is deleted after the R session is closed. The files created are also imported to the R environment as GRanges objects.

In this case, the function mspc returns the following :

1. status

2. GRangesObjects

When the value of the argument GRanges is set to FALSE and the argument keep is set to TRUE, no GRanges object will be imported to the R environment. In this case, the function mspc returns the following :

1. status

2. filesCreated.

About the files generated by mspc :

As previously mentioned, when the mspc function is called, it creates a set of files. These files are listed in the object filesCreated, returned by mspc.

The files created are :

1. A log file that contains the execution log : This file contains the information, debugging messages, and exceptions that occurred during the execution.

2. Consensus peaks in standard BED and MSPC format.

3. One folder per each replicates that contains BED files, containing stringent, weak, background, confirmed, discarded, true-positive, and false-positive peaks.

Value

The mspc function prints the results of the MSPC program.

The mspc function also creates a set of files in the directory specified by the argument outputPath.

The function can return the following :

1. status : Integer. The exit status of running the mspc function. A zero exit status means the function ran successfully.

2. filesCreated : List of character vectors. The names of the files generated while running the mspc function.

3. GrangesObjects : GRanges list. All the files generated while running the mspc function are imported as GRanges objects, and are combined in a GRanges list.

It is important to note that the mspc function does not always return these 3 elements.

The output of the function depends on the arguments keep and GRanges given to the mspc function.

More information regarding the output in Details.

Examples

```
#Providing input as BED files :

path <- system.file("extdata", package="rmspc")
input1 <- paste0(path, "/rep1.bed")
input2 <- paste0(path, "/rep2.bed")
input <- c(input1, input2)
results <- mspc(input = input, replicateType = "Technical",
                stringencyThreshold = 1e-8,
                weakThreshold = 1e-4, gamma = 1e-8,
                keep = FALSE,GRanges = TRUE,
                multipleIntersections = "Lowest",
                c = 2,alpha = 0.05)


#Providing input as a GRanges list :
```

```
library(GenomicRanges)
library(rtracklayer)
GR1 <- import(input1, format="bed")
GR2 <- import(input2, format="bed")
GR <- GRangesList("GR1"=GR1, "GR2"=GR2)
results <- mspc(
input = GR, replicateType = "Biological",
stringencyThreshold = 1e-8, weakThreshold = 1e-4,
gamma =  1e-8, GRanges = TRUE, keep = FALSE,
multipleIntersections = "Highest",
c = 2,alpha = 0.05)
```

Index