Package 'MACPET'

October 16, 2018

Type Package

Title Model based analysis for paired-end data

Version 1.0.0

Date 2018-05-01

Author Ioannis Vardaxis

Maintainer Ioannis Vardaxis <ioannis.vardaxis@ntnu.no>

Description The MACPET package can be used for binding site analysis for ChIA-PET data. MACPET reads ChIA-PET data in BAM or SAM format and separates the data into Self-ligated, Intra- and Inter-chromosomal PETs. Furthermore, MACPET breaks the genome into regions and applies 2D mixture models for identifying candidate peaks/binding sites using skewed generalized students-t distributions (SGT). It then uses a local poisson model for finding significant binding sites. MACPET is mainly written in C++, and it supports the BiocParallel package.

License GPL-3

Encoding UTF-8

LazyData true

ByteCompile true

RoxygenNote 6.0.1

biocViews Software, DNA3DStructure, PeakDetection, StatisticalMethod, Clustering, Classification, HiC

Depends R (>= 3.5), InteractionSet (>= 1.7.6)

Imports intervals (>= 0.15.1), plyr (>= 1.8.4), Rsamtools (>= 1.31.3), stats (>= 3.5.0), utils (>= 3.5.0), methods (>= 3.5.0), GenomicRanges (>= 1.31.20), S4Vectors (>= 0.17.32), IRanges (>= 2.13.26), GenomeInfoDb (>= 1.15.5), gtools (>= 3.5.0), GenomicAlignments (>= 1.15.12), knitr (>= 1.19), Rcpp (>= 0.12.15), rtracklayer (>= 1.39.9), BiocParallel (>= 1.13.1), Rbowtie (>= 1.19.1), GEOquery (>= 2.47.17), Biostrings (>= 2.47.9), ShortRead (>= 1.37.1), rbamtools (>= 2.16.6), futile.logger (>= 1.4.3)

Suggests ggplot2 (>= 2.2.1), igraph (>= 1.1.2), rmarkdown (>= 1.8), reshape2 (>= 1.4.3), BiocStyle (>= 2.7.8)

VignetteBuilder knitr

LinkingTo Rcpp

SystemRequirements C++11

Collate 'ALLData.R' 'AllClasses.R' 'AnalysisStatistics.R'
'ConvertToPE_BAM.R' 'ConvertToPSelf-methods.R' 'InputChecks.R'
'MACPETUlt.R' 'MACPET_pkg.R' 'PeaksToGRanges-methods.R'
'PeaksToNarrowPeak-methods.R' 'RcppExports.R'
'Stage_0_FilteringLinkersFunctions.R'
'Stage_1_MappingFunctions.R'
'Stage_2_PETClassificationFunctions.R'
'Stage_3_PeakFinderFunctions.R' 'TagsToGInteractions-methods.R'
'exportPeaks-methods.R' 'plot-methods.R' 'summary-methods.R'

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

git_branch RELEASE_3_7

git_last_commit 86a3ef8

git_last_commit_date 2018-04-30

Date/Publication 2018-10-15

R topics documented:

AnalysisStatistics *Count Statistics for ChIA-PET.*

Description

AnalysisStatistics prints and saves count statistics for the current inputs of the peak-calling analysis.

Usage

```
AnalysisStatistics(x.self, x.intra = NULL, x.inter = NULL,
  file.out = NULL, threshold = 1e-05, savedir = NULL)
```

Arguments

x.self	An object of class PSelf or PSFit.
x.intra	An object of class PIntra (optional).
x.inter	An object of class PInter (optional).
file.out	A string with the name of the output to be saved to savedir. If NULL the function will only print the output.
threshold	A numeric indicating the FDR cut-off, used when class(x.self)=PSFit. If NULL, no threshold is applied.
savedir	A string with the directory to save the ouput file. If NULL then the function will only print the output.

Value

Based on the inputs, AnalysisStatistics prints the total Self-ligated, Intra- and Inter-chromosomal PETs, as well as the total regions, total candidate peaks and total significant peaks (if threshold!=NULL and class(x.self)=PSFit). If file.out and savedir are not NULL then it also saves the output to a csv file in savedir.

Author(s)

Ioannis Vardaxis, <ioannis.vardaxis@ntnu.no>

References

Vardaxis I, Drabløs F, Rye M and Lindqvist BH (2018). *MACPET: Model-based Analysis for ChIA-PET*. To be published.

See Also

PSelf, PSFit,PIntra, PInter

Examples

```
#Create a temporary test forder, or anywhere you want:
savedir=file.path(tempdir(),'MACPETtest')
dir.create(savedir)#where you will save the results

#load Inter-chromosomal data:
load(system.file('extdata', 'MACPET_pinterData.rda', package = 'MACPET'))
class(MACPET_pinterData)

#load Intra-chromosomal data:
load(system.file('extdata', 'MACPET_pintraData.rda', package = 'MACPET'))
class(MACPET_pintraData)

#load Self-ligated data: (class=PSelf)
load(system.file('extdata', 'MACPET_pselfData.rda', package = 'MACPET'))
class(MACPET_pselfData)

#Print analysis:
AnalysisStatistics(x.self=MACPET_pselfData,
                   x.intra=MACPET_pintraData,
                   x.inter=MACPET_pinterData,
                   file.out='AnalysisStats',
                   savedir=savedir)

#################################################################
#load Self-ligated data: (class=PSFit)
load(system.file('extdata', 'MACPET_psfitData.rda', package = 'MACPET'))
class(MACPET_psfitData)

#Print analysis:
AnalysisStatistics(x.self=MACPET_psfitData,
                   x.intra=MACPET_pintraData,
                   x.inter=MACPET_pinterData,
                   file.out='AnalysisStats',
                   savedir=savedir,
                   threshold=1e-5)

#-----delete test directory:
unlink(savedir,recursive=TRUE)
```

ConvertToPE_BAM *Convert two BAM files into one paired-end BAM file.*

Description

Stage 2 in MACPETUlt needs a paired-end BAM file to run. This can be created in Stage 1 using the usable_1 and usable_2 fastq files created in Stage 0. However the user might have two single-end BAM files already created but not paired (by filtering with another way than that in Stage 0 or mapping using another algorithm than that in Stage 1) and only needs to run Stages 2 and 3 in MACPETUlt. ConvertToPE_BAM can be used on the two BAM files for pairing them, and the resulted paired-end BAM file can then be used in Stage 2 in MACPETUlt.

Usage

```
ConvertToPE_BAM(S1_AnalysisDir = "", SA_prefix = "MACPET",
  S1_BAMStream = 2e+06, S1_image = TRUE, S1_genome = "hg19",
  BAM_file_1 = "", BAM_file_2 = "", S1_makeSam = FALSE)
```

Arguments

`S1_AnalysisDir` The directory where the resulted paired-end BAM file will be saved.

`SA_prefix` see `MACPETUlt`.

`S1_BAMStream` see `MACPETUlt`.

`S1_image` see `MACPETUlt`.

`S1_genome` see `MACPETUlt`.

`BAM_file_1` The directory of the BAM file with the first reads. Their Qnames have to end with /1.

`BAM_file_2` The directory of the BAM file with the second reads. Their Qnames have to end with /2.

`S1_makeSam` see `MACPETUlt`.

Details

The BAM files `BAM_file_1` and `BAM_file_2` do not need to be sorted, but their Qnames have to end with /1 and /2 respectively. Furthermore, the BAM files have to include the header section.

Value

A paired-end BAM file named `SA_prefix_MACPET_Paired_end.bam` and its index, saved in `S1_AnalysisDir`.

Author(s)

Ioannis Vardaxis, `<ioannis.vardaxis@ntnu.no>`

References

Vardaxis I, Drabløs F, Rye M and Lindqvist BH (2018). *MACPET: Model-based Analysis for ChIA-PET*. To be published.

See Also

`MACPETUlt`, `SampleChIAPETDataRead_1.bam`, `SampleChIAPETDataRead_2.bam`

Examples

```
requireNamespace('ggplot2')

#Create a temporary forder, or anywhere you want:
S1_AnalysisDir=file.path(tempdir(),'MACPETtest')
dir.create(S1_AnalysisDir)#where you will save the results

#directories of the BAM files:
BAM_file_1=system.file('extdata', 'SampleChIAPETDataRead_1.bam', package = 'MACPET')
BAM_file_2=system.file('extdata', 'SampleChIAPETDataRead_2.bam', package = 'MACPET')
SA_prefix='MACPET'
```

```
#convert to paired-end BAM:
ConvertToPE_BAM(S1_AnalysisDir=S1_AnalysisDir,
                SA_prefix=SA_prefix,
                S1_BAMStream=2000000,
                S1_image=TRUE,
                S1_genome='hg19',
                BAM_file_1=BAM_file_1,
                BAM_file_2=BAM_file_2)

#test if the resulted BAM is paired-end:
PairedBAM=file.path(S1_AnalysisDir,paste(SA_prefix,'_Paired_end.bam',sep=''))
Rsamtools::testPairedEndBam(file = PairedBAM, index = PairedBAM)

bamfile = Rsamtools::BamFile(file = PairedBAM,asMates = TRUE)
GenomicAlignments::readGAlignmentPairs(file = bamfile,use.names = FALSE,
                                       with.which_label = FALSE,
                                       strandMode = 1)

#-----delete test directory:
unlink(S1_AnalysisDir,recursive=TRUE)
```

ConvertToPSelf *Convert GInteraction object to PSelf object*

Description

ConvertToPSelf converts a GInteractions object to class to PSelf object.

Usage

```
ConvertToPSelf(object, ...)

## Default S3 method:
ConvertToPSelf(object, ...)

## S3 method for class 'GInteractions'
ConvertToPSelf(object, S2_BlackList, SA_prefix,
  S2_AnalysisDir, ...)
```

Arguments

object	An object of GInteractions class.
...	not used.
S2_BlackList	See MACPETUlt.
SA_prefix	See MACPETUlt.
S2_AnalysisDir	The directory in which the object will be saved.

Details

MACPETUlt at State 2 separates the Inter-chromosomal, Intra-chromosomal and Self-ligated PETs by taking the paired-end BAM/SAM file as input. However the user might only have Self-ligated data available and already separated from the Inter/Intra-chromosomal PETs. ConvertToPSelf can then be used in the Self-ligated data to convert a GInteractions object containing only the Self-ligated PETs to a PSelf class for further analysis in Stage 3. The object will be saved in the S2_AnalysisDir directory with the name SA_prefix_pselfData. Note that if S2_BlackList==TRUE then the GInteractions object given as input has to include the genome name in the seqinfo slot. Also, the sequences lengths are mandatory in the seqinfo slot since they are used in stage 3 of the analysis.

Value

An object of class PSelf.

Author(s)

Ioannis Vardaxis, <ioannis.vardaxis@ntnu.no>

References

Vardaxis I, Drabløs F, Rye M and Lindqvist BH (2018). *MACPET: Model-based Analysis for ChIA-PET*. To be published.

See Also

PSelf

Examples

```
#load Self-ligated data: (class=PSelf)
load(system.file('extdata', 'MACPET_pselfData.rda', package = 'MACPET'))
class(MACPET_pselfData)

object=MACPET_pselfData
#--remove information and convert to GInteractions:
S4Vectors::metadata(object)=list(NULL)
class(object)='GInteractions'
#----input parameters
S2_BlackList=TRUE
SA_prefix='MACPET'
S2_AnalysisDir=file.path(tempdir(),'MACPETtest')
if(!dir.exists(S2_AnalysisDir)) dir.create(S2_AnalysisDir)

ConvertToPSelf(object=object,
                      S2_BlackList=S2_BlackList,
                      SA_prefix=SA_prefix,
                      S2_AnalysisDir=S2_AnalysisDir)
#load object:
rm(MACPET_pselfData)#old object
load(file.path(S2_AnalysisDir,'MACPET_pselfData'))
class(MACPET_pselfData)
#-----delete test directory:
unlink(S2_AnalysisDir,recursive=TRUE)
```

exportPeaks *Exports peaks to csv file*

Description

exportPeaks is an S3 method for the PSFit class. It exports peak information to a csv file in a given directory.

Usage

```
exportPeaks(object, ...)

## Default S3 method:
exportPeaks(object, ...)

## S3 method for class 'PSFit'
exportPeaks(object, file.out, savedir, threshold = NULL, ...)
```

Arguments

object	An object of PSFit class.
...	(not used).
file.out	A string with the name of the output to be saved to savedir.
savedir	A string with the directory to save the output.
threshold	A numeric indicating the FDR cut-off used for subseting significant peaks. If NULL all the peaks are returned.

Value

For PSFit class: a csv file named after the value of file.out with all the information about the peaks found by the MACPETUlt function, plus comments which explain the column names.

Author(s)

Ioannis Vardaxis, <ioannis.vardaxis@ntnu.no>

References

Vardaxis I, Drabløs F, Rye M and Lindqvist BH (2018). *MACPET: Model-based Analysis for ChIA-PET*. To be published.

See Also

PSFit

Examples

```
#Create a temporary forder, or anywhere you want:
savedir=file.path(tempdir(),'MACPETtest')
dir.create(savedir)#where you will save the results

#load Self-ligated data: (class=PSFit)
load(system.file('extdata', 'MACPET_psfitData.rda', package = 'MACPET'))
class(MACPET_psfitData)
exportPeaks(object=MACPET_psfitData,file.out='Peaks',threshold=1e-5,savedir=savedir)

#-----delete test directory:
unlink(savedir,recursive=TRUE)
```

MACPET *An R-package for binding site analysis of ChIA-PET data.*

Description

The MACPET package can be used for general analysis of paired-end (PET) data like ChIA-PET. MACPET currently implements the following four stages: Linker filtering (stage 0), mapping to the reference genome (stage 1), PET classification (stage 2) and peak-calling (stage 3). All of the MACPET stages can be run at once, or separately. In stage 0, MACPET identifies the linkers in the fastq files and classifies the reads as usable, chimeric or ambiguous. Usable reads are considered in the subsequent stages. In stage 1, MACPET maps the usable reads to the reference genome using `bowtie` and produces a paired-end BAM file. This BAM file is further used in stage 2 to classify the PETs as self-ligated/intra- or inter-chromosomal. Self-ligated PETs are used in stage 3 for the identification of significant peaks. In stage 3, MACPET segments the genome into regions and applies 2D mixture models for identifying candidate peaks using skewed generalized students-t distributions (SGT). It then uses a local poisson model for finding significant binding sites. MACPET is mainly written in C++, and it supports the BiocParallel package.

MACPET main function

`MACPETUlt` runs the whole analysis at once.

MACPET classes

`PSelf` S4 class for Self-ligated PETs.

`PSFit` S4 class for Self-ligated PETs after peak-calling.

`PInter` S4 class for Inter-chromosomal PETs.

`PIntra` S4 class for Intra-chromosomal PETs.

MACPET methods

`plot` Method for plotting different objects.

`summary` Method for summarizing different objects.

`TagsToGInteractions` Method for converting Tags to `GInteractions` class.

`PeaksToGRanges` Method for converting peaks to `GRanges` class.

`exportPeaks` Method for exporting peaks in cvs file format.

ConvertToPSelf Method for converting a GInteractions class of Self-ligated PETs to object of PSelf class.

PeaksToNarrowPeak Method for converting peaks to narrowPeak (BED) format for use in interaction analysis using the MANGO algorithm.

MACPET supplementary functions

ConvertToPE_BAM Function for converting two separate BAM files into one paired-end BAM file.

AnalysisStatistics Prints summary of multiple objects.

MACPET sample data

SampleChIAPETData.bam Sample ChIA-PET data.

SampleChIAPETDataRead_1.bam First reads from the sample ChIA-PET data.

SampleChIAPETDataRead_2.bam Second reads from the sample ChIA-PET data.

MACPET_pinterData.rda Sample PInter data.

MACPET_pintraData.rda Sample PIntra data.

MACPET_pselfData.rda Sample PSelf data.

MACPET_psfitData.rda Sample PSFit data.

Author(s)

Ioannis Vardaxis, <ioannis.vardaxis@ntnu.no>

References

Vardaxis I, Drabløs F, Rye M and Lindqvist BH (2018). *MACPET: Model-based Analysis for ChIA-PET*. To be published.

MACPETUlt *Paired-end Tag (PET) Analysis Function.*

Description

MACPETUlt is used for running analysis based on paired-end DNA data, including stages for linker removal, mapping to the reference genome, PET classification and binding site identification.

Usage

```
MACPETUlt(SA_AnalysisDir = "", SA_stages = c(0:3), SA_prefix = "MACPET",
  S0_fastq1 = "", S0_fastq2 = "", S0_LinkerA = "GTTGGATAAG",
  S0_LinkerB = "GTTGGAATGT", S0_MinReadLength = 18, S0_MaxReadLength = 50,
  S0_LinkerOccurence = 0, S0_image = TRUE, S0_fastqStream = 2e+06,
  S1_fastq1_usable_dir = "", S1_fastq2_usable_dir = "", S1_image = TRUE,
  S1_BAMStream = 2e+06, S1_makeSam = TRUE, S1_genome = "hg19",
  S1_RbowtieIndexBuild = FALSE, S1_RbowtieIndexDir = "",
  S1_RbowtieIndexPrefix = "", S1_RbowtieRefDir = "",
  S2_PairedEndBAMpath = "", S2_image = TRUE, S2_BlackList = TRUE,
  S3_fileSelfDir = "", S3_image = TRUE, S3_method = "BH")
```

Arguments

`SA_AnalysisDir` A directory were all the ouput is to be saved. This parameter is mandatory for every stage.

`SA_stages` Numeric vector or integer (if stages are run separately). This parameter is mandatory for every stage (see details).

`SA_prefix` A string which is going to be used as prefix for the outputs (default: 'MACPET'). This parameter is mandatory for every stage.

`S0_fastq1` A string with the directory of the 5-end fastq (or fastq.gz) file. This parameter is mandatory if Stage 0 is run (see details).

`S0_fastq2` A string with the directory of the 3-end fastq (or fastq.gz) file. This parameter is mandatory if Stage 0 is run (see details).

`S0_LinkerA` A string with the first linker sequence (default 'GTTGGATAAG'). This parameter is mandatory if Stage 0 is run (see details).

`S0_LinkerB` A string with the second linker sequence (default 'GTTGGAATGT'). This parameter is mandatory if Stage 0 is run (see details).

`S0_MinReadLength` A positive integer with the minimum read length after linker trimming (default: 18). This parameter is mandatory if Stage 0 is run (see details).

`S0_MaxReadLength` A positive integer with the maximum read length after linker trimming (default: 50). This parameter is mandatory if Stage 0 is run (see details).

`S0_LinkerOccurence` One of the following: 0, 1, 2, 3, 4. This parameter defines the linker-occurence mode (see details). Default 0.

`S0_image` Logical, indicating if a pie-chart image for the fastq files classification will be produced (default=TRUE). This parameter is mandatory if Stage 0 is run.

`S0_fastqStream` Positive integer for total lines of fastq files to be loaded in R (best to leave it at default because it might cause memory crash). This parameter is mandatory if Stage 0 is run.

`S1_fastq1_usable_dir` String with the directory of the 5-end usable fastq (or fastq.gz) files. This parameter might not be mandatory (see details).

`S1_fastq2_usable_dir` String with the directory of the 3-end usable fastq (or fastq.gz) files. This parameter might not be mandatory (see details).

`S1_image` Logical indicating if images for the mapping percentage and the pairing percentage will be produced (default=TRUE). This parameter is mandatory if Stage 1 is run.

`S1_BAMStream` Positive integer for the total number of bam file lines to be loaded in R in a loop for pairing (best to leave it at default because it might cause memory crash). This parameter is mandatory if Stage 1 is run.

`S1_makeSam` Logical indicating whether the resulted paired-end BAM file will be splitted to two SAM files (one for each read). The output SAM files can be used as input in the MANGO algorithm (default=TRUE). Note, that the user has to remove the SAM header before running MANGO. This parameter is mandatory if Stage 1 is run.

`S1_genome` String with the genome to be used in the bam file header (default='hg19'). This parameter is mandatory if Stage 1 is run (see details).

`S1_RbowtieIndexBuild`
Logical indicating whether you want to build the bowtie index or not (default=FALSE). This parameter is mandatory if Stage 1 is run (see details).

`S1_RbowtieIndexDir`
String with the directory of the bowtie index (if `S1_RbowtieIndexBuild==FALSE`) or with the directory where the bowtie index will be saved (if `S1_RbowtieIndexBuild==TRUE`). This parameter is mandatory if Stage 1 is run (see details).

`S1_RbowtieIndexPrefix`
String with the prefix for the bowtie indeces in `S1_RbowtieIndexDir` (see details). This parameter is mandatory if Stage 1 is run (see details).

`S1_RbowtieRefDir`
A vector with the directories of the .fa files, used if `S1_RbowtieIndexBuild==TRUE`. This parameter is mandatory if Stage 1 is run and `S1_RbowtieIndexBuild==TRUE` (see details).

`S2_PairedEndBAMpath`
A string with the directory of the paired-end bam file (or paired-end sam file). This parameter might not be mandatory (see details).

`S2_image` Logical indicating whether images for the s elf-ligated/intra-chromosomal cut-off as well as pie-charts for the PET classification will be produced (default=TRUE). This parameter is mandatory if Stage 1 is run.

`S2_BlackList` Logical indicating whether black-listed regions will be removed from the data based on the `S1_genome` parameter (see details). Alternatively a `GRanges` object with the user specified regions. This parameter is mandatory if Stage 2 is run.

`S3_fileSelfDir` A string with the directory of the of the object of class `PSelf`. This parameter might not be mandatory (see details).

`S3_image` Logical indicating whether images for the binding site's FDR, sizes of the binding sites, sizes of binding site's upstream/downstream peaks will be created. This parameter is mandatory if Stage 3 is run.

`S3_method` String with the FDR method used for finding p-values of significant peaks in the data. See `p.adjust.methods` (default= 'BH'). This parameter is mandatory if Stage 3 is run.

Details

Every stage has parameters associated with it. Parameters with prefix `SA` correspond to all stages, `S0` to Stage 0, `S1` to Stage 1 etc. Parameters with `SA` prefix are mandatory for every stage.

If `SA_stages` parameter is given as vector, then the vector has to be continuous, that is for example c(0:3) or c(2:3), not c(0,2,3). In general the best practice is to run all the stages at once.

The fastq files in `S0_fastq1` and `S0_fastq2` have to be of same length and be sorted by their ID. Furthermore, the IDs in `S0_fastq1` have to end with /1 and the ones in `S0_fastq2` with /2, representing the 5- and 3-end tags respectively. In other words, for the same line in `S0_fastq1` and `S0_fastq2`, their IDs have to be identical, except form their suffixes /1 and /2 respectively. Moreover, the "/" symbol can be replaced with any other symbol, this will not cause any problems.

`S0_LinkerOccurence` parameter defines the linker-occurence mode and separates the usable from the ambiguous PETs. PETs with both reads including linkers are not affeted by `S0_LinkerOccurence`. Also, reads which do not meet the `S0_MaxReadLength`/`S0_MinReadLength` lengths, are moved to ambiguous anyway. The four values of `S0_LinkerOccurence` are:

`Mode 0`: Both reads have to include a linker in order to be checked as usable or chimeric, if they dont, they are moved to ambiguous.

Mode 1: If read 1 is not matching any linker, but read 2 does, then the PET will be moved to usable.

Mode 2: If read 2 is not matching any linker, but read 1 does, then the PET will be moved to usable.

Mode 3: If any of the reads does not match any linker then the PET they will be moved to usable.

Mode 4: If both reads do not match any of the linkers, then the PET will be moved to usable.

S0_MaxReadLength has to be greater than S0_MinReadLength. The user should leave those two at default unless the PET data is produced by tagmentation.

S1_fastq1_usable_dir and S1_fastq2_usable_dir are not mandatory if Stage 0 is run right before Stage 1 (SA_stages=c(0,1)). Those two are only mandatory if Stage 1 is run separately. Then those parameters assume to have the usable reads only. The same fastq specifications apply as those for S0_fastq1 and S0_fastq2.

The parameter S1_genome is very important. First the genome name given in S1_genome should be the same as the one used for building the bowtie index for mapping. This parameter will add an 'AS' column to the paired-end bam file with the genome information. In Stage 2, this header will be used for identifying which kind on black-listed regions to use if S2_BlackList==TRUE.

If S1_RbowtieIndexBuild==FALSE then the bowtie index is assumed to be already built and saved in S1_RbowtieIndexDir. Then the S1_RbowtieIndexDir folder should include the following files: S1_RbowtieIndexPrefix.1.ebwt, S1_RbowtieIndexPrefix.2.ebwt, S1_RbowtieIndexPrefix.3.ebwt, S1_RbowtieIndexPrefix.4.ebwt, S1_RbowtieIndexPrefix.rev.1.ebwt and S1_RbowtieIndexPrefix.rev.2.ebw or with .ebwtl. Where S1_RbowtieIndexPrefix is also given as input.

If S1_RbowtieIndexBuild==TRUE then the bowtie index will be build using the bowtie_build function. This function will need the .fa files which should be given as input in the S1_RbowtieRefDir vector. This is a character vector with the directories of the .fa files to use. The output index will be saved in S1_RbowtieIndexDir. if S1_RbowtieIndexBuild==FALSE then S1_RbowtieRefDir can be an empty string.

The parameter S2_PairedEndBAMpath has to be specified only if Stage 2 is run without running Stage 1 right before (SA_Stages=c(2) or c(2,3), not c(1,2) or c(0,1,2) for example). If this is the case, the S2_PairedEndBAMpath has to be the path to the BAM/SAM paired-end file. The file has to include the header with the 'SN', 'LN' and 'AS' columns. Moreover the mate flags of the file have to be correct and also the duplicated PETs must be flagged too. Stage 2 will upload the whole data in R using readGAlignmentPairs function with flags isDuplicate=FALSE and isPaired=TRUE. So if duplicated PETs are not flagged, they will be used in the analysis. If the previous stages are run in sequence, then S2_PairedEndBAMpath will be overwritten with the newly created BAM file, which will have the correct flags.

If S2_BlackList==TRUE then which genome black-list is going to be used is decided by the 'AS' column in the S2_PairedEndBAMpath file, which is specified by the S1_genome if Stage 1 is also run. The black-listed regions cover the following genomes: 'hg19', 'ce10', 'dm3', 'hg38', 'mm9', 'mm10'. If the 'AS' header column is missing from the S2_PairedEndBAMpath file, or if the S1_genome is not matching any of the above named genomes, then a warning will be produced saying that no black-listed regions will be removed. Alternatily, the user can provide its own black-listed regions as a GRanges object.

The parameter S3_fileSelfDir is not mandatory if the stages are run in sequence, if Stage 2 is run right before stage 3. If this is the case then S3_fileSelfDir will be overwritten with the data produced in Stage 2. If Stage 3 is run separately, then S3_fileSelfDir has to be provided. It should be a PSelf object and both the name of the object in the directory and the one uploaded in R should be SA_prefix_pselfData.

Value

All outputs are saved at the SA_AnalysisDir. The output depents of the stages run:

Stage 0: (outputs saved in a folder named `S0_results` in `SA_AnalysisDir`) `SA_prefix_usable_1.fastq.gz`: fastq.gz files with the usable 5-end tags. To be used in Stage 1.

`SA_prefix_usable_2.fastq.gz`: fastq.gz files with the usable 3-end tags. To be used in Stage 1.

`SA_prefix_chimeric_1.fastq.gz`: fastq.gz files with the chimeric 5-end tags.

`SA_prefix_chimeric_2.fastq.gz`: fastq.gz files with the chimeric 3-end tags.

`SA_prefix_ambiguous_1.fastq.gz`: fastq.gz files with the ambiguous 5-end tags.

`SA_prefix_ambiguous_2.fastq.gz`: fastq.gz files with the ambiguous 3-end tags.

`SA_prefix_stage_0_image.jpg`: Pie chart image with the split of two fastq files used as input (if `S0_image==TRUE`).

Stage 1: (outputs saved in a folder named `S1_results` in `SA_AnalysisDir`) `SA_prefix_usable_1.sam`: sam file with the mapped 5-end reads (if `S1_makeSam==FALSE`).

`SA_prefix_usable_2.sam`: sam file with the mapped 3-end reads (if `S1_makeSam==FALSE`).

`SA_prefix_Paired_end.bam`: paired-end bam file with the mapped PETs. To be used in Stage 2

`SA_prefix_Paired_end.bam.bai`: .bai file for `SA_prefix_Paired_end.bam`. To be used in Stage 2.

`SA_prefix_stage_1_p1_image.jpg`: Pie-chart for the mapped/unmapped reads from `SA_prefix_usable_1.sa` `SA_prefix_usable_2.sam` (if `S1_image==TRUE`).

`SA_prefix_stage_1_p2_image.jpg`: Pie-chart for the paired/unpaired reads of `SA_prefix_Paired_end.bam` (if `S1_image==TRUE`).

Stage 2: (outputs saved in a folder named `S2_results` in `SA_AnalysisDir`) `SA_prefix_pselfData`: An object of `PSelf` class with the Self-ligated PETs. To be used in Stage 3.

`SA_prefix_pintraData`: An object of `PIntra` class with the Intra-chromosomal PETs.

`SA_prefix_pinterData`: An object of `PInter` class with the Inter-chromosomal PETs.

`SA_prefix_stage_2_p1_image.jpg`: Pie-chart reliable/dublicated/black-listed PETs of `SA_prefix_Paired_en` (if `S2_image==TRUE`).

`SA_prefix_stage_2_p2_image.jpg`: Histogram with the self-ligated/intra-chromosomal cut-off for `SA_prefix_Paired_end.bam` (if `S2_image==TRUE`).

`SA_prefix_stage_2_p3_image.jpg`: Pie-chart for the self-ligated/intra-chromosomal/inter-chromosomal PETs of `SA_prefix_Paired_end.bam` (if `S2_image==TRUE`).

Stage 3: (outputs saved in a folder named `S3_results` in `SA_AnalysisDir`) `SA_prefix_psfitData`: An object of `PSFit` class with the peak information.

`SA_prefix_stage_3_p1_image.jpg`: Sizes of the upstream vs downstream peaks of each binding site given the binding site's FDR (if `S3_image==TRUE`).

`SA_prefix_stage_3_p2_image.jpg`: FDR of the binding sites. The horizontal red line is at FDR=0.05 (if `S3_image==TRUE`).

`SA_prefix_stage_3_p3_image.jpg`: Comparison of binding site sizes given their FDR (if `S3_image==TRUE`).

`SA_prefix_stage_3_p3_image.jpg`: FDR for the upstream/donwstream peaks of the binding sites given the binding sites FDR (if `S3_image==TRUE`).

Stage 0:3 : All the above outputs. Furthermore, a log file named `SA_prefix_analysis.log` is always created in `SA_AnalysisDir` with information about the process.

Stages description

`MACPETUlt` runs a complete or partial analysis for PET data, depending on the stages of the analysis the user wants to run. The stages of the analysis are the following:

Stage 0: Linker identification stage: This stage uses the two fastq files for the 5- and 3-end tags and identifies which tags contain any of the linkers. Based on the linker combinations it classifies the PETs as usable (linkers A/A or B/B), chimeric (linkers A/B or B/A) and ambiguous (linkers non/A, non/B, A/non, B/non unless chosen otherwise by `S0_LinkerOccurence`, or be smaller/bigger than the `S0_MinReadLength`/`S0_MaxReadLength` after the linker removal, respectively). Only usable PETs are considered in the subsequent steps.

Stage 1: PET mapping stage: This stage uses the usable PETs identified by stage 0. It maps them separately to the reference genome using the `bowtie` function with no mismatch per read, and keeps the uniquely mapped reads only. It then maps the unmapped reads to the reference genome with at most one mismatch and keeps the uniquely mapped reads. Uniquely mapped reads with zero or one mismatch are then merged and paired, their duplicates are marked and a paired-end bam file is created which is used in State 2.

Stage 2: PET classification stage: This stage takes the BAM paired-end file from stage 1 and classifies the PETs as: Inter-chromosomal PETs (which connect two different chromosomes), Intra-chromosomal PETs (which connect regions of the same chromosome) and Self-ligated PETs (which are used for binding site analysis). Self-ligated PETs are used for finding the protein binding sites (peaks), while Intra- and Inter-chromosomal are used for interactions between the peaks. The algorithm uses the elbow-method to seperate the Self-ligated from the Intra-chromosomal population. Note that loading the data into R might take a while depending on the size of the data.

Stage 3: Peak calling stage: This stage uses the Self-ligated PETs and it runs the EM algorithm to find clusters which represent candidate peaks/binding sites in 2 dimentional space using skewed generalized students-t distributions (SGT). After the peak-calling analysis is done, the algorithm assesses the significance of the candidate peaks using a local Poisson model.

Parallel

All stages can be run in parallel using the `register` function. The user has to register a parallel backhead before starting the function.

Author(s)

Ioannis Vardaxis, `<ioannis.vardaxis@ntnu.no>`

References

Vardaxis I, Drabløs F, Rye M and Lindqvist BH (2018). *MACPET: Model-based Analysis for ChIA-PET*. To be published.

Consortium EP (2012) *An integrated encyclopedia of DNA elements in the human genome.*. Nature, 489(7414), pp. 57–74. `http://dx.doi.org/10.1038/nature11247`.

See Also

PSelf, PIntra, PInter,`summary`, `AnalysisStatistics`, `plot` `BiocParallel`,`ConvertToPSelf`, `exportPeaks`,`TagsToGIn`
`PeaksToGRanges`, `PeaksToNarrowPeak`, `ConvertToPE_BAM`

Examples

```
#Create a temporary forder, or anywhere you want:
SA_AnalysisDir=file.path(tempdir(),'MACPETtest')
dir.create(SA_AnalysisDir)#where you will save the results
#give directory of the BAM file:
```

```
S2_PairedEndBAMpath=system.file('extdata', 'SampleChIAPETData.bam', package = 'MACPET')

#give prefix name:
SA_prefix='MACPET'

#parallel backhead can be created using the BiocParallel package
#parallel backhead can be created using the BiocParallel package
#requireNamespace('BiocParallel')
#snow <- BiocParallel::SnowParam(workers = 4, type = 'SOCK', progressbar=FALSE)
#BiocParallel::register(snow, default=TRUE)

#-run for the whole binding site analysis:
MACPETUlt(SA_AnalysisDir=SA_AnalysisDir,
          SA_stages=c(2:3),
          SA_prefix=SA_prefix,
          S2_PairedEndBAMpath=S2_PairedEndBAMpath,
          S2_image=TRUE,
          S2_BlackList=TRUE,
          S3_image=TRUE)



#load results:
SelfObject=paste(SA_prefix,'_pselfData',sep='')
load(file.path(SA_AnalysisDir,'S2_results',SelfObject))
SelfObject=get(SelfObject)
class(SelfObject) # see methods for this class

IntraObject=paste(SA_prefix,'_pintraData',sep='')
load(file.path(SA_AnalysisDir,'S2_results',IntraObject))
IntraObject=get(IntraObject)
class(IntraObject) # see methods for this class

InterObject=paste(SA_prefix,'_pinterData',sep='')
load(file.path(SA_AnalysisDir,'S2_results',InterObject))
InterObject=get(InterObject)
class(InterObject) # see methods for this class

SelfFitObject=paste(SA_prefix,'_psfitData',sep='')
load(file.path(SA_AnalysisDir,'S3_results',SelfFitObject))
SelfFitObject=get(SelfFitObject)
class(SelfFitObject) # see methods for this class

#-----delete test directory:
unlink(SA_AnalysisDir,recursive=TRUE)
```

MACPET_pinterData.rda *Inter-chromosomal PETs from ChIA-PET data*

Description

Inter-chromosomal PETs data from ESR1 ChIA-PET subset data on human MCF-7.

Format

rda object of PInter class.

Details

`MACPET_pinterData` is produced by the `MACPETUlt` function at Stage 2 and it contains the Inter-chromosomal PETs of the sample data.

Author(s)

Main data creators Yijun Ruan, GIS, 2012-05-24

MACPET_pinterData.rda creator Ioannis Vardaxis, <`ioannis.vardaxis@ntnu.no`>

See Also

`SampleChIAPETData.bam`, `PInter`

`MACPET_pintraData.rda` *Intra-chromosomal PETs from ChIA-PET data*

Description

Intra-chromosomal PETs data from ESR1 ChIA-PET subset data on human MCF-7.

Format

rda object of `PIntra` class.

Details

`MACPET_pintraData` is produced by the `MACPETUlt` function at Stage 2 and it contains the Intra-chromosomal PETs of the sample data.

Author(s)

Main data creators Yijun Ruan, GIS, 2012-05-24

MACPET_pintraData.rda creator Ioannis Vardaxis, <`ioannis.vardaxis@ntnu.no`>

See Also

`SampleChIAPETData.bam`, `PIntra`

MACPET_pselfData.rda *Self-ligated PETs from ChIA-PET data*

Description

Self-ligated PETs data from ESR1 ChIA-PET subset data on human MCF-7.

Format

rda object of PSelf class.

Details

MACPET_pselfData is produced by the MACPETUlt function at Stage 2 and it contains the Self-ligated PETs of the sample data.

Author(s)

Main data creators Yijun Ruan, GIS, 2012-05-24

MACPET_pselfData.rda creator Ioannis Vardaxis, <ioannis.vardaxis@ntnu.no>

See Also

SampleChIAPETData.bam, PSelf

MACPET_psfitData.rda *Self-ligated PETs from ChIA-PET data*

Description

Self-ligated PETs data from ESR1 ChIA-PET subset data on human MCF-7.

Format

rda object of PSFit class.

Details

MACPET_psfitData is produced by the MACPETUlt function at Stage 3 and it contains the self-ligated PETs of the sample data after calling for candidate peaks.

Author(s)

Main data creators Yijun Ruan, GIS, 2012-05-24

MACPET_psfitData.rda creator Ioannis Vardaxis, <ioannis.vardaxis@ntnu.no>

See Also

SampleChIAPETData.bam, PSFit

PeaksToGRanges *Convert peaks to GRanges object*

Description

PeaksToGRanges converts peaks of an object of PSFit class to GRanges object.

Usage

```
PeaksToGRanges(object, ...)

## Default S3 method:
PeaksToGRanges(object, ...)

## S3 method for class 'PSFit'
PeaksToGRanges(object, threshold = NULL, ...)
```

Arguments

object	An object of class PSFit.
...	Further arguments to be passed to PeaksToGRanges (not used).
threshold	A numeric with the FDR cut-off threshold used to take a subset of significant peaks. If threshold=NULL then all the peaks are returned.

Details

PeaksToGRanges converts peak information into a GRanges object. Each row in the GRanges object represents a peak with ’CIQ.Up.start’ and ’CIQ.Down.end’ as start and end coordinates, respectively (see PSFit) Metadata will also include information for the total PETs, the p-value and the FDR of each peak.

Value

For PSFit class, a GRanges object created by the estimated peak information including metadata columns for the total PETs, the p-value and the FDR of each peak.

Author(s)

Ioannis Vardaxis, <ioannis.vardaxis@ntnu.no>

References

Vardaxis I, Drabløs F, Rye M and Lindqvist BH (2018). *MACPET: Model-based Analysis for ChIA-PET*. To be published.

See Also

PSFit, PeaksToNarrowPeak

Examples

```
#load Self-ligated data: (class=PSFit)
load(system.file('extdata', 'MACPET_psfitData.rda', package = 'MACPET'))
class(MACPET_psfitData)
PeaksToGRanges(object=MACPET_psfitData,threshold=1e-5)
```

`PeaksToNarrowPeak` *Convert Peaks to narrowPeak (BED) object.*

Description

`PeaksToNarrowPeak` converts peaks of an object of `PSFit` class to `narrowPeak` object. The object is saved in a user specified directory and can be used in the MANGO or MICC algorithms for interaction analysis.

Usage

```
PeaksToNarrowPeak(object, ...)

## Default S3 method:
PeaksToNarrowPeak(object, ...)

## S3 method for class 'PSFit'
PeaksToNarrowPeak(object, threshold = NULL, savedir, file.out,
  ...)
```

Arguments

`object`	An object of class `PSFit`.
`...`	Further arguments to be passed to `PeaksToNarrowPeak` (not used).
`threshold`	A numeric with the FDR cut-off threshold used to take a subset of significant peaks. If `threshold=NULL` then all the peaks are returned.
`savedir`	A string with the directory to save the ouput file.
`file.out`	A string with the name of the output to be saved to `savedir`.

Details

Each Peak in the narrowPeak object is represented by an interval starting from the ’CIQ.Up.start’ estimated variable to its ’CIQ.Down.end’ (see `PSFit`). Close Peaks in genomic distance are NOT merged by the `PeaksToNarrowPeak` function. However the user can specify a distance window for merging in the MANGO or MICC algorithms. Note also that MANGO and MICC find a self-ligated cut-off by itself which is usually very different than that found by MACPET. We suggest that the user overwrites MANGOS’s or MICC’s cut-off with that of MACPET.

Value

A `narrowPeak` object named after the value of `file.out` and saved in the `savedir`.

Author(s)

Ioannis Vardaxis, <ioannis.vardaxis@ntnu.no>

References

Vardaxis I, Drabløs F, Rye M and Lindqvist BH (2018). *MACPET: Model-based Analysis for ChIA-PET*. To be published.

See Also

PSFit

Examples

```
#Create a temporary forder, or anywhere you want:
savedir=file.path(tempdir(),'MACPETtest')
dir.create(savedir)#where you will save the results
file.out='MACPET_peaks.narrowPeak'

#load Self-ligated data: (class=PSFit)
load(system.file('extdata', 'MACPET_psfitData.rda', package = 'MACPET'))
class(MACPET_psfitData)
PeaksToNarrowPeak(object=MACPET_psfitData,threshold=1e-5,file.out=file.out,savedir=savedir)

#-----delete test directory:
unlink(savedir,recursive=TRUE)
```

PInter-class *PInter S4 Class*

Description

PInter class in a S4 class which inherits from the GInteractions class and it contains Inter-chromosomal data.

Details

PInter class is created by the MACPETUlt function at Stage 2.

Author(s)

Ioannis Vardaxis, <ioannis.vardaxis@ntnu.no>

References

Vardaxis I, Drabløs F, Rye M and Lindqvist BH (2018). *MACPET: Model-based Analysis for ChIA-PET*. To be published.

See Also

AnalysisStatistics, plot, summary, MACPETUlt

PIntra-class *PIntra S4 Class*

Description

PIntra class in a S4 class which inherits from the GInteractions class and it contains Intra-chromosomal data.

Details

PIntra class is created by the MACPETUlt function at Stage 2.

Author(s)

Ioannis Vardaxis, <ioannis.vardaxis@ntnu.no>

References

Vardaxis I, Drabløs F, Rye M and Lindqvist BH (2018). *MACPET: Model-based Analysis for ChIA-PET*. To be published.

See Also

AnalysisStatistics, plot, summary, MACPETUlt

plot *plot methods for MACPET classes*

Description

Different plot methods for the classes in the MACPET package.

Usage

```
## S3 method for class 'PInter'
plot(x, ...)

## S3 method for class 'PIntra'
plot(x, ...)

## S3 method for class 'PSelf'
plot(x, ...)

## S3 method for class 'PSFit'
plot(x, kind, RegIndex = NULL, threshold = NULL, ...)
```

Arguments

x An object of correct class used to create different plots.

... further arguments to be passed in the plot functions.

kind A string with one of the following arguments. Note that if a region visualization is plotted, the vertical lines represent peak-summits.

PETcounts For a bar-plot of the PET-counts in each chromosome.

RegionCounts For a bar-plot for the region counts in each chromosome.

PeakCounts For a bar-plot for the Peak-counts in each chromosome.

RegionPETs For a ggplot for a visualization of the PETs in a region.

RegionTags For a ggplot for a visualization of the tags in a region. The tags are classified by stream (upper/lower)

PeakPETs For a ggplot for a visualization of the PETs in a region. The PETs are classified by the peak they belong to.

PeakTags For a ggplot for a visualization of the tags in a region. The tags are classified by the peak they belong to.

SigPETCounts For a bar-plot with the significant PET-counts in each chromosome.

SigRegionCounts For a bar-plot with the significant region-counts in each chromosome.

SigPeakCounts For a bar-plot with the significant peak-counts in each chromosome.

RegIndex an integer indicating which region to plot (1 means the biggest in terms of total PETs.)

threshold The FDR cut-off when plotting the total significant peaks for each chromosome in the data.

Value

For the PInter class: A network plot. Each node is a chromosome with size proportional to the total PETs of the corresponding chromosome. Edges connect chromosomes which have common PETs, where the thickness of an edge is proportional on the total number of PETs connecting the two chromosomes.

For the PIntra class: A bar-plot. Each bar represents the total number of Intra-chromosomal PETs for each chromosome in the data.

For the PSelf class: A bar-plot. Each bar represents the total number of Self-ligated PETs for each chromosome in the data.

For the PSFit class: Different plots depenting on the kind argument.

Author(s)

Ioannis Vardaxis, <ioannis.vardaxis@ntnu.no>

References

Vardaxis I, Drabløs F, Rye M and Lindqvist BH (2018). *MACPET: Model-based Analysis for ChIA-PET*. To be published.

See Also

PSelf, PSFit PInter,PIntra

Examples

```
#load Inter-chromosomal data:
load(system.file('extdata', 'MACPET_pinterData.rda', package = 'MACPET'))
class(MACPET_pinterData)
requireNamespace('igraph')
plot(MACPET_pinterData)
#load Intra-chromosomal data:
load(system.file('extdata', 'MACPET_pintraData.rda', package = 'MACPET'))
class(MACPET_pintraData)
requireNamespace('ggplot2')
plot(MACPET_pintraData)
#load Self-ligated data:
load(system.file('extdata', 'MACPET_pselfData.rda', package = 'MACPET'))
class(MACPET_pselfData)
requireNamespace('ggplot2')
plot(MACPET_pselfData)
#load Self-ligated data:
load(system.file('extdata', 'MACPET_psfitData.rda', package = 'MACPET'))
class(MACPET_psfitData)
requireNamespace('ggplot2')
plot(MACPET_psfitData,kind='PETcounts')
plot(MACPET_psfitData,kind='PeakCounts')
plot(MACPET_psfitData,kind='PeakPETs',RegIndex=1)
plot(MACPET_psfitData,kind='PeakTags',RegIndex=1)
```

PSelf-class *PSelf S4 Class*

Description

PSelf class in a S4 class which inherits from the GInteractions class and it contains Self-ligated PETs from ChIA-PET experiment. Furthermore it also contains the following in the metadata field:

- Self_info A data.frame with the count statistics for the total PETs in each chromosome.
- SLmean The mean size of the PETs in the data.
- MaxSize Maximum size of self-ligated PETs.
- MinSize Minimum size of self-ligated PETs.

Details

PSelf class is created by the MACPETUlt function at Stage 2.

Author(s)

Ioannis Vardaxis, <ioannis.vardaxis@ntnu.no>

References

Vardaxis I, Drabløs F, Rye M and Lindqvist BH (2018). *MACPET: Model-based Analysis for ChIA-PET*. To be published.

See Also

AnalysisStatistics, plot, summary, MACPETUlt, ConvertToPSelf

PSFit-class *PSFit S4 Class*

Description

PSFit class in a S4 class which inherits from the GInteractions class and it contains Self-ligated PETs from ChIA-PET experiment and information about the genome of the data. Furthermore it also contains the following in the metadata field:

Self_info Counts statistics for the total PETs, total regions and total Peaks in each chromosome.

SLmean The mean size of the PETs in the data.

MaxSize Maximum size of self-ligated PETs.

MinSize Minimum size of self-ligated PETs.

Classification.Info A matrix with information for the Data-row ID, region ID and Peak ID (0 represent noise) of each peak in the data.

Peaks.Info Information for each peak found by the peak-calling algorithm:

- Chrom The chromosome which the peak belongs to.
- Region The region which the peak belongs to.
- Peak The peak ID (a region might have more than one peaks).
- Pets Total PETs in the peak.
- Peak.Summit Summit of the peak.
- Up.Summit Summit of the left-stream PETs.
- Down.Summit Summit of the right-stream PETs.
- CIQ.Up.start Start of the 95 Quantile confidence interval for the left-stream PETs.
- CIQ.Up.end End of the 95 Quantile confidence interval for the left-stream PETs.
- CIQ.Up.size Size of the 95 Quantile confidence interval for the left-stream PETs.
- CIQ.Down.start Start of the 95 Quantile confidence interval for the right-stream PETs.
- CIQ.Down.end End of the 95 Quantile confidence interval for the right-stream PETs.
- CIQ.Down.size Size of the 95 Quantile confidence interval for the right-stream PETs.
- CIQ.Peak.size Size of the Peak based on the interval (CIQ.Up.start,CIQ.Down.end).
- sdx The standard deviation of the upstream PETs.
- lambdax The skewness of the upstream PETs.
- sdy The standard deviation of the downstream PETs.
- lambday The skewness of the downstream PETs.
- lambdaUp The expected number of PETs in the left-stream Peak region by random chance.
- FoldEnrichUp Fold enrichment for the left-stream Peak region.
- p.valueUp p-value for the left-stream Peak region.
- lambdaDown The expected number of PETs in the right-stream Peak region by random chance.
- FoldEnrichDown Fold enrichment for the right-stream Peak region.
- p.valueDown p-value for the right-stream Peak region.
- p.value p-value for the Peak (p.valueUp*p.valueDown).
- FDRUp FDR correction for the left-stream Peak region.
- FDRDown FDR correction for the right-stream Peak region.
- FDR FDR correction for the Peak.

Details

PSFit class is created by the MACPETUlt function at Stage 3.

Author(s)

Ioannis Vardaxis, <ioannis.vardaxis@ntnu.no>

References

Vardaxis I, Drabløs F, Rye M and Lindqvist BH (2018). *MACPET: Model-based Analysis for ChIA-PET*. To be published.

See Also

AnalysisStatistics,plot, summary, MACPETUlt, exportPeaks, PeaksToGRanges, TagsToGInteractions, PeaksToNarrowPeak

---

SampleChIAPETData.bam *Subset sample of ChIA-PET data*

Description

A subset of ChIA-PET data:

Target: ESR1

Biosample summary: Homo sapiens MCF-7

GEO: GSM970212

Format

A BAM file.

Author(s)

Yijun Ruan, GIS, 2012-05-24 (main data creators)

Source

https://www.encodeproject.org/experiments/ENCSR000BZZ/

References

Consortium EP (2012) *An integrated encyclopedia of DNA elements in the human genome.*. Nature, 489(7414), pp. 57–74. http://dx.doi.org/10.1038/nature11247.

See Also

MACPET_pinterData.rda, MACPET_pintraData.rda, MACPET_pselfData.rda, MACPET_psfitData.rda, SampleChIAPETDataRead_1.bam, SampleChIAPETDataRead_2.bam

SampleChIAPETDataRead_1.bam *First reads from a subset of ChIA-PET data in* SampleChIAPETData.bam

Description

First reads from a subset of ChIA-PET data in SampleChIAPETData.bam:

Target: ESR1

Biosample summary: Homo sapiens MCF-7

GEO: GSM970212

Format

A BAM file.

Author(s)

Yijun Ruan, GIS, 2012-05-24 (main data creators)

Source

https://www.encodeproject.org/experiments/ENCSR000BZZ/

References

Consortium EP (2012) *An integrated encyclopedia of DNA elements in the human genome.*. Nature, 489(7414), pp. 57–74. http://dx.doi.org/10.1038/nature11247.

See Also

SampleChIAPETData.bam, ConvertToPE_BAM

SampleChIAPETDataRead_2.bam *Second reads from a subset of ChIA-PET data in* SampleChIAPETData.bam

Description

Second reads from a subset of ChIA-PET data in SampleChIAPETData.bam:

Target: ESR1

Biosample summary: Homo sapiens MCF-7

GEO: GSM970212

Format

A BAM file.

Author(s)

Yijun Ruan, GIS, 2012-05-24 (main data creators)

Source

https://www.encodeproject.org/experiments/ENCSR000BZZ/

References

Consortium EP (2012) *An integrated encyclopedia of DNA elements in the human genome.*. Nature, 489(7414), pp. 57–74. http://dx.doi.org/10.1038/nature11247.

See Also

SampleChIAPETData.bam, ConvertToPE_BAM

---

summary *summary methods for the MACPET classes.*

Description

Different summary methods for the classes in the MACPET package.

Usage

```
## S3 method for class 'PSelf'
summary(object, ...)

## S3 method for class 'PSFit'
summary(object, threshold = NULL, ...)

## S3 method for class 'PIntra'
summary(object, heatmap = FALSE, ...)

## S3 method for class 'PInter'
summary(object, heatmap = FALSE, ...)
```

Arguments

object	An object of correct class used to create different summaries.
...	Further arguments to be passed to the summary function.
threshold	A numeric representing the FDR cutoff for summarizing singificant peaks, if NULL the summary is based on all the peaks found.
heatmap	TRUE or FALSE indicating whether the user wants to plot a heat-map plot for the Intra/Inter-chromosomal PET counts within chromosomes or between different chromosomes.

Value

A summary of the `object` and a heat-map plot depending on the class of the input.

Author(s)

Ioannis Vardaxis, <ioannis.vardaxis@ntnu.no>

References

Vardaxis I, Drabløs F, Rye M and Lindqvist BH (2018). *MACPET: Model-based Analysis for ChIA-PET*. To be published.

See Also

PSelf, PSFit, PInter,PIntra

Examples

```
#load Self-ligated data: (class=PSelf)
load(system.file('extdata', 'MACPET_pselfData.rda', package = 'MACPET'))
class(MACPET_pselfData)
summary(MACPET_pselfData)
#load Self-ligated data: (class=PSFit)
load(system.file('extdata', 'MACPET_psfitData.rda', package = 'MACPET'))
class(MACPET_psfitData)
summary(MACPET_psfitData)
summary(MACPET_psfitData,threshold=1e-5)
#load Intra-chromosomal data: (class=PIntra)
load(system.file('extdata', 'MACPET_pintraData.rda', package = 'MACPET'))
class(MACPET_pintraData)
summary(MACPET_pintraData)
requireNamespace('ggplot2')
requireNamespace('reshape2')
summary(MACPET_pintraData,heatmap=TRUE)#sample data, not good heatmap plot.
#load Inter-chromosomal data: (class=PInter)
load(system.file('extdata', 'MACPET_pinterData.rda', package = 'MACPET'))
class(MACPET_pinterData)
summary(MACPET_pinterData)
requireNamespace('ggplot2')
requireNamespace('reshape2')
summary(MACPET_pinterData,heatmap=TRUE)#sample data, not good heatmap plot.
```

TagsToGInteractions *Convert PETs to GInteractions object*

Description

TagsToGInteractions converts the PETs of an object of PSFit class to GInteractions object.

Usage

```
TagsToGInteractions(object, ...)

## Default S3 method:
TagsToGInteractions(object, ...)

## S3 method for class 'PSFit'
TagsToGInteractions(object, threshold = NULL, ...)
```

Arguments

object	An object of class PSFit.
...	(not used).
threshold	A numeric for the FDR threshold used to take a subset of significant peaks/binding sites. If threshold=NULL then all the peaks are returned.

Value

For PSFit class: A GInteractions object containing PETs from all the peaks found in the data (removing noisy and insignificant PETs). Furthermore, it also includes information about the binding sites which can be accessed via the metadata function.

Author(s)

Ioannis Vardaxis, <ioannis.vardaxis@ntnu.no>

References

Vardaxis I, Drabløs F, Rye M and Lindqvist BH (2018). *MACPET: Model-based Analysis for ChIA-PET*. To be published.

See Also

PSFit

Examples

```
#load Self-ligated data: (class=PSFit)
load(system.file('extdata', 'MACPET_psfitData.rda', package = 'MACPET'))
class(MACPET_psfitData)
object=TagsToGInteractions(object=MACPET_psfitData,threshold=1e-5)
object
S4Vectors::metadata(object)$Peaks.Info #peak/binding site information
```

Index