Package ‘dpeak’

October 16, 2022

Type Package

Title dPeak (Deconvolution of Peaks in ChIP-seq Analysis)

Version 1.8.0

Depends R (>= 4.0.0), methods, stats, utils, graphics, Rcpp

Imports MASS, IRanges, BSgenome, grDevices, parallel

Suggests BSgenome.Ecoli.NCBI.20080805

LinkingTo Rcpp

SystemRequirements GNU make, meme, fimo

Date 2020-02-25

Author Dongjun Chung, Carter Allen

Maintainer Dongjun Chung <dongjun.chung@gmail.com>

Description dPeak is a statistical framework for the high resolution identification of protein-DNA interaction sites using PET and SET ChIP-Seq and ChIP-exo data. It provides computationally efficient and user friendly interface to process ChIP-seq and ChIP-exo data, implement exploratory analysis, fit dPeak model, and export list of predicted binding sites for downstream analysis.

License GPL (>= 2)

LazyData FALSE

NeedsCompilation yes

biocViews ChIPSeq, Genetics, Sequencing, Software, Transcription

BugReports https://github.com/dongjunchung/dpeak/issues

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

git_branch RELEASE_3_15

git_last_commit 0ae2693

git_last_commit_date 2022-04-26

Date/Publication 2022-10-16

R topics documented:

dpeak-package 2
DpeakData-class 3
dpeakFit 5
DpeakFit-class 6
dpeakMotif 8
DpeakMotif-class 9
dpeakRead 10
exampleData 12
exportPeakList 12
exportPlot 13
printEmpty 14

Index **16**

`dpeak-package` *dPeak (Deconvolution of Peaks in ChIP-seq Analysis)*

Description

This package provides functions for fitting dPeak, a statistical framework to deconvolve ChIP-seq peaks.

Details

Package:	dpeak
Type:	Package
Version:	2.0.1
Date:	2014-09-15
License:	GPL (>= 2)
LazyLoad:	yes

This package contains two main classes, `DpeakData` and `DpeakFit`, which represent dPeak data and deconvolution model fit, respectively. This package contains two main methods, `dpeakRead` and `dpeakFit`. `dpeakRead` method imports peak list and aligned read file and construct `DpeakData` class object. `dpeakFit` method fits deconvolution model using `DpeakData` class object and constructs `DpeakFit` class object. `DpeakFit` class object can be exported as text files and can be used for the downstream analysis.

Author(s)

Dongjun Chung

Maintainer: Dongjun Chung <chungdon@stat.wisc.edu>

See Also

dpeakRead, dpeakFit, exportPeakList, DpeakData, DpeakFit.

Examples

```
## Not run:
# work flow for PET data

dataPET <- dpeakRead( peakfile="examplePeak.txt", readfile="examplePETRead.txt",
    fileFormat="eland_result", PET=TRUE )
dataPET
exportPlot( dataPET, filename="exPETplot.pdf" )

fitPET <- dpeakFit( dataPET )
fitPET
exportPlot( fitPET, filename="exPETResult.pdf", plotType="fit" )
exportPlot( fitPET, filename="exPETGOF.pdf", plotType="GOF" )

# work flow for SET data

dataSET <- dpeakRead( peakfile="examplePeak.txt", readfile="exampleSETRead.txt",
    fileFormat="eland_result", PET=FALSE, fragLen=150 )
dataSET
exportPlot( dataSET, filename="exSETplot_combined.pdf", strand=FALSE )
exportPlot( dataSET, filename="exSETplot_strand_1.pdf",
    strand=TRUE, extension=1, smoothing=TRUE )
exportPlot( dataSET, filename="exSETplot_strand_150.pdf",
    strand=TRUE, extension=150, smoothing=FALSE )

fitSET <- dpeakFit( dataSET )
fitSET
exportPlot( fitSET, filename="exSETResult_combined.pdf",
    plotType="fit", strand=FALSE )
exportPlot( fitSET, filename="exSETResult_strand_1.pdf",
    plotType="fit", strand=TRUE, extension=1, smoothing=TRUE )
exportPlot( fitSET, filename="exSETResult_strand_150.pdf",
    plotType="fit", strand=TRUE, extension=150, smoothing=FALSE )
exportPlot( fitSET, filename="exSETGOF.pdf", plotType="GOF" )

# (common for both PET and SET data)

exportPeakList( fitSET, type="txt", filename="result.txt" )
exportPeakList( fitSET, type="bed", filename="result.bed" )
exportPeakList( fitSET, type="gff", filename="result.gff" )

## End(Not run)
```

DpeakData-class *Class "DpeakData"*

Description

This class represents dPeak data.

Objects from the Class

Objects can be created by calls of the form `new("DpeakData", ...)`.

Slots

`fragSet`: Object of class `"list"`, representing list of fragments for each peak.

`PET`: Object of class `"logical"`, representing whether it is paired-end tag (PET) or single-end tag (SET) data.

`fragLenTable`: Object of class `"table"`, representing distribution of fragment length when `PET=TRUE`.

`aveFragLen`: Object of class `"numeric"`, representing average fragment length when `PET=FALSE`.

`Fratio`: Object of class `"numeric"`, representing proportion of forward reads when `PET=FALSE`.

`stackedFragment`: Object of class `"list"`, representing number of fragments aligning to each genomic position.

`peakChr`: Object of class `"character"`, representing a vector of chromosome of each peak.

`peakStart`: Object of class `"numeric"`, representing a vector of start position of each peak.

`peakEnd`: Object of class `"numeric"`, representing a vector of end position of each peak.

`emptyList`: Object of class `"character"`, representing a vector of peak regions without reads.

Methods

dpeakFit `signature(object = "DpeakData")`: fit the deconvolution model.

exportPlot `signature(x = "BinData", y = "missing", filename=NULL, strand=FALSE, extension=1, smoothing=FALSE )`: provide exploratory plots of fragments or reads in each peak region. Plots are exported to a PDF file (its file name is specified in `filename`). Options `strand`, `extension`, and `smoothing` are supported only for SET data. If `strand=TRUE`, reads are plotted in a strand-specific manner, where reads are extended to `extension` from its 5' end. If `smoothing=TRUE`, a smoothed plot (using the smoothing spline) is provided. If `strand=FALSE`, strand information is ignored.

printEmpty `signature(x = "DpeakData")`: provide the data frame of peak regions without reads.

show `signature(object = "DpeakData")`: provide brief summary of the object.

Author(s)

Dongjun Chung

See Also

`dpeakRead`, `dpeakFit`.

Examples

```
data(exampleData)
exampleData
```

dpeakFit *Fit dPeak model*

Description

Fit a deconvolution model.

Usage

```
dpeakFit( object, ... )
## S4 method for signature 'DpeakData'
dpeakFit( object,
objectMotif=NULL, estDeltaSigma="common", init="localmax",
nTop=100, lbDelta = 25, lbSigma = 25,
   psize=21, maxComp=5, pConst=0.2, nCore=1, verbose=FALSE, iterInit=50, iterMain=25, epsilon=1e-6 )
```

Arguments

object	Object of class DpeakData, dPeak data imported using method dpeakRead.
objectMotif	Object of class DpeakMotif, motif data generated using method dpeakMotif. If incorporated, locations of binding events are initialized using motif information.
estDeltaSigma	Approach to estimate delta and sigma parameters for SET data. Possible values are either "common" (estimate single delta and sigma parameters that are used for all peaks) or "separate" (estimate delta and sigma parameters for each peak separately). Default is "common". Not relevant when PET data is used.
init	Approach to initialize locations of binding events. Possible values are "localmax" and "uniform". Default is "localmax".
nTop	Number of candidate regions used to estimate common delta and sigma estimates. Relevant only when estDeltaSigma="common".
lbDelta	Lower bound for delta parameter.
lbSigma	Lower bound for sigma parameter.
psize	Approximate size of the binding protein of interest.
maxComp	Maximum possible number of binding events in each peak region.
pConst	Value to determine the plateau in the BIC curve. Should be a value larger than zero and smaller than one.
nCore	Number of CPUs to be used when parallel computing is utilized.
verbose	Use verbose mode? Possible values are either TRUE (use) or FALSE (do not use).
iterInit	Iteration number for initial estimation of binding sites.
iterMain	Iteration number for main estimation of binding sites.
epsilon	Criterion to stop iteration for binding site estimation.
...	Other parameters to be passed through to generic dpeakFit.

Details

Parallel computing can be utilized for faster computation if `parallel` package is installed. Users can change the number of CPUs to be used by changing the argument `nCore`.

Value

Construct `DpeakFit` class object.

Author(s)

Dongjun Chung

See Also

`dpeakRead`, `DpeakFit`.

Examples

```
data(exampleData)
exampleFit <- dpeakFit(exampleData, maxComp = 5)
```

`DpeakFit-class` *Class "DpeakFit"*

Description

This class represents deconvolution model fit.

Objects from the Class

Objects can be created by calls of the form `new("DpeakFit", ...)`.

Slots

`fits`: Object of class `"list"`, representing list of deconvolution fits of all possible models for each peak.

`optFit`: Object of class `"list"`, representing list of fits of the optimal model for each peak.

`optMu`: Object of class `"list"`, representing list of binding sites of the optimal model for each peak.

`optPi`: Object of class `"list"`, representing list of relative strengths of the optimal model for each peak.

`optPi0`: Object of class `"list"`, representing list of background strengths of the optimal model for each peak.

`optGamma`: Object of class `"list"`, representing list of background proportion of the optimal model for each peak.

`optDelta`: Object of class `"list"`, representing list of read shift of the optimal model for each peak when `PET=FALSE`.

`optSigma`: Object of class `"list"`, representing list of read standard deviation of the optimal model for each peak when `PET=FALSE`.

`bicVec`: Object of class `"list"`, representing list of BIC values for each peak.

`aicVec`: Object of class `"list"`, representing list of AIC values for each peak.

`fragSet`: Object of class `"list"`, representing list of fragments for each peak.

`PET`: Object of class `"logical"`, representing whether it is paired-end tag (PET) or single-end tag (SET) data.

`fragLenTable`: Object of class `"table"`, representing distribution of fragment length when `PET=TRUE`.

`aveFragLen`: Object of class `"numeric"`, representing average fragment length when `PET=FALSE`.

`Fratio`: Object of class `"numeric"`, representing proportion of forward reads when `PET=FALSE`.

`stackedFragment`: Object of class `"list"`, representing number of fragments aligning to each genomic position.

`peakChr`: Object of class `"character"`, representing a vector of chromosome of each peak.

`peakStart`: Object of class `"numeric"`, representing a vector of start position of each peak.

`peakEnd`: Object of class `"numeric"`, representing a vector of end position of each peak.

`estDeltaSigma`: Object of class `"character"`, representing the approach to estimate delta and sigma parameters for SET data.

`nTop`: Object of class `"numeric"`, representing the number of candidate regions used to estimate common delta and sigma estimates.

`lbDelta`: Object of class `"numeric"`, representing a lower bound for the delta parameter.

`lbSigma`: Object of class `"numeric"`, representing a lower bound for the sigma parameter.

`psize`: Object of class `"numeric"`, representing approximate size of the binding protein of interest.

`maxComp`: Object of class `"numeric"`, representing maximum possible number of binding events in each peak region.

`pConst`: Object of class `"numeric"`, representing value to determine the plateau in the BIC curve.

`iterInit`: Object of class `"numeric"`, representing iteration number for initial estimation of binding sites.

`iterMain`: Object of class `"numeric"`, representing iteration number for main estimation of binding sites.

`epsilon`: Object of class `"numeric"`, representing criterion to stop iteration for binding site estimation.

Methods

exportPlot `signature(x = "DpeakFit", y = "missing", filename=NULL, plotType="fit", strand=FALSE, extension=1, smoothing=FALSE, threshold=0.10, nsimul=10000, seed=12345, nCore=8)`: draw plots of deconvolution results if `plotType="fit"`, goodness of fit (GOF) plots if `plotType="GOF"`, or plots of Bayesian information criterion (BIC) and Akaike information criterion (AIC) curves if `plotType="BIC"`. Plots are exported to a PDF file (its file name is specified in `filename`). In deconvolution result plots, binding sites with strength larger than `threshold`

are drawn in dark blue and other binding sites are drawn in light blue. When `plotType="fit"`, options `strand`, `extension`, and `smoothing` are supported for SET data. If `strand=TRUE`, reads are plotted in a strand-specific manner, where reads are extended to `extension` from its 5' end. If `smoothing=TRUE`, a smoothed plot (using the smoothing spline) is provided. If `strand=FALSE`, strand information is ignored. For the GOF plots, `nsimul` fragments are simulated from the fitted model (`seed` indicates random seed; `nCore` CPUs are used for parallel computing).

show `signature(object = "DpeakFit")`: provide brief summary of the object.

Author(s)

Dongjun Chung

See Also

`dpeakFit`, `exportPeakList`.

Examples

```
data(exampleData)
exampleFit <- dpeakFit(exampleData, maxComp = 5)
```

`dpeakMotif` *Implement de novo motif analysis based on the peak list*

Description

Implement de novo motif analysis based on the peak list, using MEME and FIMO.

Usage

```
dpeakMotif( peakfile=NULL, refGenome=NULL, flanking=100,
memeArgument="-dna -mod zoops -nmotifs 1 -minw 10 -maxw 20 -revcomp -maxsize 1000000000",
fimoArgument="-max-stored-scores 100000000 -motif-pseudo 0.000001",
    tempDir=NULL )
```

Arguments

`peakfile`	File name of the peak list.
`refGenome`	`BSgenome` class object to extract sequences.
`flanking`	Flanking length.
`memeArgument`	Parameters for `MEME`.
`fimoArgument`	Parameters for `FIMO`.
`tempDir`	Directory of temporary files for sequence extraction, MEME, and FIMO.

Details

The first three columns of the peak list file (specifed as `peakfile`) are assumed to be chromosome, start and end positions of each peak region. There should be no header in the peak list file.

`refGenome` is a `BSgenome` class object and assumed to already be available in the R environment.

Value

Construct `DpeakMotif` class object.

Author(s)

Dongjun Chung

See Also

`dpeakFit`, `DpeakMotif`.

Examples

```
## Not run:
library(BSgenome.Ecoli.NCBI.20080805)
resultMotif <- dpeakMotif( peakfile="examplePeak.txt", refGenome=Ecoli )

## End(Not run)
```

`DpeakMotif-class` *Class "DpeakMotif"*

Description

This class represents dPeak data.

Objects from the Class

Objects can be created by calls of the form `new("DpeakMotif", ...)`.

Slots

`motif`: Object of class `"character"`, representing a vector of motifs.

`locMotif`: Object of class `"list"`, representing list of locations of motifs in candidate regions.

`peakChr`: Object of class `"character"`, representing a vector of chromosome of each peak.

`peakStart`: Object of class `"numeric"`, representing a vector of start position of each peak.

`peakEnd`: Object of class `"numeric"`, representing a vector of end position of each peak.

Methods

dpeakFit `signature(object = "DpeakMotif")`: fit the deconvolution model.

show `signature(object = "DpeakMotif")`: provide brief summary of the object.

Author(s)

Dongjun Chung

See Also

`dpeakMotif`, `dpeakFit`.

Examples

```
## Not run:
library(BSgenome.Ecoli.NCBI.20080805)
resultMotif <- dpeakMotif( peakfile="vignettes/examplePeak.txt", refGenome=Ecoli )

## End(Not run)
```

---

dpeakRead — *Import peak list and aligned read files*

Description

Import and process peak list and aligned read files.

Usage

```
dpeakRead( peakfile=NULL, readfile=NULL, fileFormat="eland_result",
   PET=FALSE, fragLen=200, parallel=FALSE, nCore=1, tempDir=NULL, perl = "perl" )
```

Arguments

`peakfile`	File name of the peak list.
`readfile`	Name of the aligned read file.
`fileFormat`	Format of the aligned read file to be processed. For single-end tag (SET) ChIP-seq data, `dpeakRead` permits the following aligned read file formats: `"eland_result"` (Eland result), `"eland_extended"` (Eland extended), `"eland_export"` (Eland export), `"bowtie"` (default Bowtie), `"sam"` (SAM), and `"bed"` (BED). For paired-end tag (PET) ChIP-seq data, `dpeakRead` permits only `"eland_result"` (Eland result format).
`PET`	Is the aligned read file paired-end tag (PET)? Possible values are either `TRUE` (PET) or `FALSE` (SET). Default is `FALSE` (SET).
`fragLen`	Average fragment length. Default is 200. Not relevant when `PET=TRUE`.

parallel	Utilize multiple CPUs for parallel computing using "parallel" package? Possible values are TRUE (use "parallel") or FALSE (not use "parallel"). Default is FALSE (not use "parallel").
nCore	Number of CPUs when parallel computing is utilized.
tempDir	Directory to store temporary files. If tempDir=NULL, dpeakRead() will use the temporary directory used by R.
perl	Name of the perl executable to be called. Default is "perl".

Details

The first three columns of the peak list file (specifed as peakfile) are assumed to be chromosome, start and end positions of each peak region. There should be no header in the peak list file.

When the data contains multiple chromosomes, parallel computing can be utilized for faster preprocessing if parallel=TRUE and parallel package is installed. nCore determines number of CPUs used for parallel computing.

Value

Construct DpeakData class object.

Author(s)

Dongjun Chung

See Also

dpeakFit, DpeakData.

Examples

```
# PET data

# dataPET <- dpeakRead( peakfile="examplePeak.txt", readfile="examplePETRead.txt",
#    fileFormat="eland_result", PET=TRUE )

# SET data

# dataSET <- dpeakRead( peakfile="examplePeak.txt", readfile="exampleSETRead.txt",
#    fileFormat="eland_result", PET=FALSE, fragLen=150 )

data(exampleData)
```

exampleData — *E. coli ChIP-seq Dataset*

Description

This is an example E. coli ChIP-seq dataset.

Usage

```
data(exampleData)
```

Format

`DpeakData` class object containing aligned reads for a peak.

Examples

```
data(exampleData)
exampleData
```

exportPeakList — *Export deconvolution results to text files*

Description

Export deconvolution results to text files in TXT, BED, or GFF file formats.

Usage

```
exportPeakList(object, ...)
## S4 method for signature 'DpeakFit'
exportPeakList( object, type=NA, filename=NA, ... )
```

Arguments

`object`	Object of class `DpeakFit`, deconvolution model fits obtained using the method `dpeakFit`.
`type`	Format of the exported file. Possible values are `"txt"`, `"bed"`, and `"gff"`. See Details.
`filename`	Name of the exported file.
`...`	Other parameters to be passed through to generic `exportPeakList`.

Details

Columns of TXT file format (`type="txt"`) include chromosome, binding site, relative binding strength in each peak region, and the peak region that each binding event belongs to. `type="bed"` and `type="gff"` export deconvolution results in standard BED and GFF file formats, respectively, where score is the relative binding strength multiplied by 1000. The feature of GFF file and the name of BED file are the peak region that each binding event belongs to.

Value

Export deconvolution results to text files

Author(s)

Dongjun Chung

See Also

`dpeakFit`, `DpeakFit`.

Examples

```
data(exampleData)
fit <- dpeakFit(exampleData)
exportPeakList( fit, type="txt", filename="result.txt" )
exportPeakList( fit, type="bed", filename="result.bed" )
exportPeakList( fit, type="gff", filename="result.gff" )
```

`exportPlot` *Export plots to pdf files.*

Description

Exports the plots of estimated binding sites (`plotType="fit"`) or the goodness of fit (GOF) plots (`plotType="GOF"`) to a PDF file.

Usage

```
exportPlot(x, y, ...)
```

Arguments

`x`	Object of class `DpeakFit`,
`y`	Name of file to export to.
`...`	Other parameters to be passed through to generic `exportPlot`.

Details

Exports the plots of estimated binding sites (`plotType="fit"`) or the goodness of fit (GOF) plots (`plotType="GOF"`) to a PDF file. Its file name needs to be specified in the `filename` argument. In both of these plots, estimated binding sites or simulated fragments are superimposed on the plots of reads (or fragments) aligned to each position. For SET data, if `plotType="fit"` and `strand=TRUE`, reads will be plotted in a strand-specific manner, where each read is extended to `extension` from its 5' end. If `smoothing=TRUE`, a smoothed plot (using the smoothing spline) is provided. Unsmoothed plot is provided by default.

Value

Export plots to files

Author(s)

Dongjun Chung

Examples

```
data(exampleData)
exampleFit <- dpeakFit( exampleData, maxComp=5)
exportPlot( exampleFit, filename="exampleResult_combined.pdf" )
```

---

printEmpty *Return the peak regions without any reads.*

Description

Return the data frame of the peak regions without any reads.

Usage

```
printEmpty( object, ... )
## S4 method for signature 'DpeakData'
printEmpty( object, ... )
```

Arguments

`object`	Object of class `dpeakData`, dPeak data obtained using the method `dpeakRead`.
`...`	Other parameters to be passed through to generic `printEmpty`.

Value

Return the peak regions without any reads

Author(s)

Dongjun Chung

See Also

dpeakRead, DpeakData.

Examples

```
data(exampleData)
printEmpty(exampleData)
```

Index

∗ **classes**
DpeakData-class, 3
DpeakFit-class, 6
DpeakMotif-class, 9
∗ **datasets**
exampleData, 12
∗ **methods**
dpeakFit, 5
dpeakMotif, 8
dpeakRead, 10
exportPeakList, 12
exportPlot, 13
printEmpty, 14
∗ **models**
dpeakFit, 5
dpeakMotif, 8
dpeakRead, 10
exportPeakList, 12
exportPlot, 13
printEmpty, 14
∗ **package**
dpeak-package, 2

dpeak *(dpeak-package)*, 2
dpeak-package, 2
DpeakData, *3*, *11*, *15*
DpeakData-class, 3
DpeakFit, *3*, *6*, *13*
dpeakFit, *3, 4*, 5, *8–11*, *13*
dpeakFit,DpeakData-method *(dpeakFit)*, 5
DpeakFit-class, 6
DpeakMotif, *9*
dpeakMotif, 8, *10*
DpeakMotif-class, 9
dpeakRead, *3, 4*, *6*, 10, *15*

exampleData, 12
exportPeakList, *3*, *8*, 12
exportPeakList,DpeakFit-method *(exportPeakList)*, 12
exportPlot, 13
exportPlot,DpeakData,missing-method *(DpeakData-class)*, 3
exportPlot,DpeakFit,missing-method *(DpeakFit-class)*, 6

printEmpty, 14
printEmpty,DpeakData-method *(printEmpty)*, 14

show,DpeakData-method *(DpeakData-class)*, 3
show,DpeakFit-method *(DpeakFit-class)*, 6
show,DpeakMotif-method *(DpeakMotif-class)*, 9