Package ‘ChIPseeker’

October 7, 2014

Type Package

Title ChIPseeker for ChIP peak Annotation, Comparison, and Visualization

Version 1.0.11

Author Guangchuang Yu

Maintainer Guangchuang Yu `<guangchuangyu@gmail.com>`

Description This package implements functions to retrieve the nearest genes around the peak, annotate genomic region of the peak, statstical methods for estimate the significance of overlap among ChIP peak data sets, and incorporate GEO database for user to compare the own dataset with those deposited in database. The comparison can be used to infer cooperative regulation and thus can be used to generate hypotheses. Several visualization functions are implemented to summarize the coverage of the peak experiment, average profile and heatmap of peaks binding to TSS regions, genomic annotation, distance to TSS, and overlap of peaks or genes.

Depends R (>= 2.10)

Imports AnnotationDbi, BiocGenerics, IRanges, GenomicFeatures,GenomicRanges, ggplot2, gplots, grDevices, gtools, Matrix,plyr, RColorBrewer, rtracklayer,TxDb.Hsapiens.UCSC.hg19.knownGene

Suggests clusterProfiler, ReactomePA, DOSE, GOSemSim, org.Hs.eg.db,knitr

VignetteBuilder knitr

License Artistic-2.0

biocViews Annotation, ChIPSeq, Software, Visualization,MultipleComparison

R topics documented:

`ChIPseeker-package` *ChIP-SEQ Annotation, Visualization and Comparison*

Description

This package is designed for chip-seq data analysis

Details

Package:	ChIPseeker
Type:	Package
Version:	1.0.10
Date:	2-01-2014
biocViews:	ChIPSeq, Annotation, Software
Depends:	
Imports:	methods, ggplot2
Suggests:	clusterProfiler, GOSemSim
License:	Artistic-2.0

Author(s)

Guangchuang Yu

Maintainer: Guangchuang Yu <`guangchuangyu@gmail.com`>

`addGeneAnno` *addGeneAnno*

Description

add gene annotation, symbol, gene name etc.

Usage

```
addGeneAnno(annoDb, geneID, type)
```

Arguments

`annoDb`	annotation package
`geneID`	query geneID
`type`	gene ID type

Value

data.frame

Author(s)

G Yu

`annotatePeak` *annotatePeak*

Description

Annotate peaks

Usage

```
annotatePeak(peak, tssRegion = c(-3000, 3000), as = "GRanges",
  TranscriptDb = NULL, level = "transcript",
  assignGenomicAnnotation = TRUE, annoDb = NULL, addFlankGeneInfo = FALSE,
  flankDistance = 5000, verbose = TRUE)
```

Arguments

`peak`	peak file or GRanges object
`tssRegion`	Region Range of TSS
`as`	one of "data.frame", "GRanges" and "txt"
`TranscriptDb`	TranscriptDb object
`level`	one of transcript and gene
`assignGenomicAnnotation`	logical, assign peak genomic annotation or not
`annoDb`	annotation package
`addFlankGeneInfo`	logical, add flanking gene information from the peaks
`flankDistance`	distance of flanking sequence
`verbose`	print message or not

Value

data.frame or GRanges object with columns of:

all columns provided by input.

annotation: genomic feature of the peak, for instance if the peak is located in 5'UTR, it will annotated by 5'UTR. Possible annotation is Promoter-TSS, Exon, 5' UTR, 3' UTR, Intron, and Intergenic.

geneChr: Chromosome of the nearest gene

geneStart: gene start

geneEnd: gene end

geneLength: gene length

geneStrand: gene strand

geneId: entrezgene ID

distanceToTSS: distance from peak to gene TSS

if annoDb is provided, extra column will be included:

ENSEMBL: ensembl ID of the nearest gene

SYMBOL: gene symbol

GENENAME: full gene name

Author(s)

G Yu

See Also

`plotAnnoBar` `plotAnnoPie` `plotDistToTSS`

Examples

```
require(TxDb.Hsapiens.UCSC.hg19.knownGene)
txdb <- TxDb.Hsapiens.UCSC.hg19.knownGene
peakfile <- system.file("extdata", "sample_peaks.txt", package="ChIPseeker")
peakAnno <- annotatePeak(peakfile, tssRegion=c(-3000, 100), as="GRanges", TranscriptDb=txdb)
head(peakAnno)
```

`downloadGEObedFiles` *downloadGEObedFiles*

Description

download all BED files of a particular genome version

Usage

```
downloadGEObedFiles(genome, destDir = getwd())
```

Arguments

`genome`	genome version
`destDir`	destination folder

Author(s)

G Yu

`downloadGSMbedFiles` *downloadGSMbedFiles*

Description

download BED supplementary files of a list of GSM accession numbers

Usage

```
downloadGSMbedFiles(GSM, destDir = getwd())
```

Arguments

`GSM`	GSM accession numbers
`destDir`	destination folder

Author(s)

G Yu

`enrichAnnoOverlap` *enrichAnnoOverlap*

Description

calcuate overlap significant of ChIP experiments based on their nearest gene annotation

Usage

```
enrichAnnoOverlap(queryPeak, targetPeak, TranscriptDb = NULL,
  pAdjustMethod = "BH", chainFile = NULL)
```

Arguments

`queryPeak`	query bed file
`targetPeak`	target bed file(s) or folder containing bed files
`TranscriptDb`	TranscriptDb
`pAdjustMethod`	pvalue adjustment method
`chainFile`	chain file for liftOver

Value

data.frame

Author(s)

G Yu

`enrichPeakOverlap` *enrichPeakOverlap*

Description

calculate overlap significant of ChIP experiments based on the genome coordinations

Usage

```
enrichPeakOverlap(queryPeak, targetPeak, TranscriptDb = NULL,
  pAdjustMethod = "BH", nShuffle = 1000, chainFile = NULL)
```

Arguments

`queryPeak`	query bed file
`targetPeak`	target bed file(s) or folder that containing bed files
`TranscriptDb`	TranscriptDb
`pAdjustMethod`	pvalue adjustment method
`nShuffle`	shuffle numbers
`chainFile`	chain file for liftOver

Value

data.frame

Author(s)

G Yu

`getGenomicAnnotation` *getGenomicAnnotation*

Description

get Genomic Annotation of peaks

Usage

```
getGenomicAnnotation(peaks, distance, tssRegion = c(-3000, 3000),
  TranscriptDb)
```

Arguments

`peaks`	peaks in GRanges object
`distance`	distance of peak to TSS
`tssRegion`	tssRegion, default is -3kb to +3kb
`TranscriptDb`	TranscriptDb object

Value

character vector

Author(s)

G Yu

`getGEOgenomeVersion` *getGEOgenomeVersion*

Description

get genome version statistics collecting from GEO ChIPseq data

Usage

```
getGEOgenomeVersion()
```

Value

data.frame

Author(s)

G Yu

`getGEOInfo` *getGEOInfo*

Description

get subset of GEO information by genome version keyword

Usage

```
getGEOInfo(genome, simplify = TRUE)
```

Arguments

`genome`	genome version
`simplify`	simplify result or not

Value

data.frame

Author(s)

G Yu

`getGEOspecies` *getGEOspecies*

Description

accessing species statistics collecting from GEO database

Usage

```
getGEOspecies()
```

Value

data.frame

Author(s)

G Yu

`getNearestFeatureIndicesAndDistances` *getNearestFeatureIndicesAndDistances*

Description

get index of features that closest to peak and calculate distance

Usage

```
getNearestFeatureIndicesAndDistances(peaks, features)
```

Arguments

`peaks`	peak in GRanges
`features`	features in GRanges

Value

list

Author(s)

G Yu

`getPromoters` *getPromoters*

Description

prepare the promoter regions

Usage

```
getPromoters(TranscriptDb = NULL, upstream = 1000, downstream = 1000,
  by = "gene")
```

Arguments

`TranscriptDb`	TranscriptDb
`upstream`	upstream from TSS site
`downstream`	downstream from TSS site
`by`	one of gene or transcript

Value

GRanges object

`getSampleFiles` *getSampleFiles*

Description

get filenames of sample files

Usage

```
getSampleFiles()
```

Value

list of file names

Author(s)

G Yu

`getTagMatrix` *getTagMatrix*

Description

calculate the tag matrix

Usage

```
getTagMatrix(peak, weightCol, windows)
```

Arguments

`peak`	peak file or GRanges object
`weightCol`	column name of weight, default is NULL
`windows`	a collection of region with equal size, eg. promoter region.

Value

tagMatrix

`info` *Information Datasets*

Description

ucsc genome version, precalcuated data and gsm information

`overlap` *overlap*

Description

calculate the overlap matrix, which is useful for vennplot

Usage

```
overlap(Sets)
```

Arguments

`Sets`	a list of objects

Value

data.frame

Author(s)

G Yu

`peakHeatmap` *peakHeatmap*

Description

plot the heatmap of peaks align to flank sequences of TSS

Usage

```
peakHeatmap(peak, weightCol = NULL, TranscriptDb = NULL, upstream = 1000,
  downstream = 1000, xlab = "", ylab = "", title = NULL, color = NULL,
  verbose = TRUE)
```

Arguments

`peak`	peak file or GRanges object
`weightCol`	column name of weight
`TranscriptDb`	TranscriptDb object
`upstream`	upstream position
`downstream`	downstream position
`xlab`	xlab
`ylab`	ylab
`title`	title
`color`	color
`verbose`	print message or not

Value

figure

Author(s)

G Yu

`plotAnnoBar` *plotAnnoBar*

Description

plot feature distribution based on their chromosome region

Usage

```
plotAnnoBar(peakAnno, title = "Feature Distribution", xlab = "",
  ylab = "Percentage(%)")
```

Arguments

`peakAnno`	peakAnno in data.frame
`title`	plot title
`xlab`	xlab
`ylab`	ylab

Details

plot chromosome region features

Value

bar plot that summarize genomic features of peaks

Author(s)

G Yu

See Also

`annotatePeak` `plotAnnoPie`

Examples

```
require(TxDb.Hsapiens.UCSC.hg19.knownGene)
txdb <- TxDb.Hsapiens.UCSC.hg19.knownGene
peakfile <- system.file("extdata", "sample_peaks.txt", package="ChIPseeker")
peakAnno <- annotatePeak(peakfile, TranscriptDb=txdb)
plotAnnoBar(peakAnno)
```

`plotAnnoPie` *plotAnnoPie*

Description

pieplot from peak genomic annotation

Usage

```
plotAnnoPie(peakAnno, ndigit = 2, cex = 0.9, col = NA,
  legend.position = "rightside", pie3D = FALSE, ...)
```

Arguments

`peakAnno`	peakAnno
`ndigit`	number of digit to round
`cex`	label cex
`col`	color
`legend.position`	topright or other.
`pie3D`	plot in 3D or not
`...`	extra parameter

Value

pie plot of peak genomic feature annotation

Author(s)

G Yu

See Also

`annotatePeak` `plotAnnoBar`

Examples

```
## example not run
## require(TxDb.Hsapiens.UCSC.hg19.knownGene)
## txdb <- TxDb.Hsapiens.UCSC.hg19.knownGene
## peakfile <- system.file("extdata", "sample_peaks.txt", package="chipseeker")
## peakAnno <- annotatePeak(peakfile, TranscriptDb=txdb)
## plotAnnoPie(peakAnno)
```

`plotAvgProf` *plotAvgProf*

Description

plot the profile of peaks

Usage

```
plotAvgProf(tagMatrix, xlim, xlab = "Genomic Region (5->3)",
  ylab = "Read Count Frequency")
```

Arguments

`tagMatrix`	tagMatrix or a list of tagMatrix
`xlim`	xlim
`xlab`	x label
`ylab`	y label

Value

ggplot object

Author(s)

G Yu

`plotAvgProf2` *plotAvgProf*

Description

plot the profile of peaks that align to flank sequences of TSS

Usage

```
plotAvgProf2(peak, weightCol = NULL, TranscriptDb = NULL, upstream = 1000,
  downstream = 1000, xlab = "Genomic Region (5->3)",
  ylab = "Read Count Frequency", verbose = TRUE)
```

Arguments

`peak`	peak file or GRanges object
`weightCol`	column name of weight
`TranscriptDb`	TranscriptDb object
`upstream`	upstream position
`downstream`	downstream position
`xlab`	xlab
`ylab`	ylab
`verbose`	print message or not

Value

ggplot object

Author(s)

G Yu

`plotChrCov` *plotChrCov*

Description

plot the Peak Regions over Chromosomes

Usage

```
plotChrCov(peak, weightCol = NULL, xlab = "Chromosome Size (bp)",
  ylab = "", title = "ChIP Peaks over Chromosomes")
```

Arguments

`peak`	peak file or GRanges object
`weightCol`	weight column of peak
`xlab`	xlab
`ylab`	ylab
`title`	title

Value

ggplot2 object

Author(s)

G Yu

`plotDistToTSS` *plotDistToTSS*

Description

plot feature distribution based on the distances to the TSS

Usage

```
plotDistToTSS(peakAnno, distanceColumn = "distanceToTSS", xlab = "",
  ylab = "Binding sites (%) (5->3)",
  title = "Distribution of transcription factor-binding loci relative to TSS")
```

Arguments

`peakAnno`	peak annotation
`distanceColumn`	column name of the distance from peak to nearest gene
`xlab`	x label
`ylab`	y lable
`title`	figure title

Value

bar plot that summarize distance from peak to TSS of the nearest gene.

Author(s)

G Yu

See Also

`annotatePeak`

Examples

```
require(TxDb.Hsapiens.UCSC.hg19.knownGene)
txdb <- TxDb.Hsapiens.UCSC.hg19.knownGene
peakfile <- system.file("extdata", "sample_peaks.txt", package="ChIPseeker")
peakAnno <- annotatePeak(peakfile, TranscriptDb=txdb)
plotDistToTSS(peakAnno)
```

`readPeakFile` *readPeakFile*

Description

read peak file and store in data.frame or GRanges object

Usage

```
readPeakFile(peakfile, as = "GRanges")
```

Arguments

`peakfile`	peak file
`as`	output format, one of GRanges or data.frame

Value

peak information, in GRanges or data.frame object

Author(s)

G Yu

Examples

```
peakfile <- system.file("extdata", "sample_peaks.txt", package="ChIPseeker")
peak.gr <- readPeakFile(peakfile, as="GRanges")
peak.gr
```

`shuffle` *shuffle*

Description

shuffle the position of peak

Usage

```
shuffle(peak.gr, TranscriptDb)
```

Arguments

`peak.gr`	GRanges object
`TranscriptDb`	TranscriptDb

Value

GRanges object

Author(s)

G Yu

`tagHeatmap` *tagHeatmap*

Description

plot the heatmap of tagMatrix

Usage

```
tagHeatmap(tagMatrix, xlim, xlab = "", ylab = "", title = NULL,
  color = "red")
```

Arguments

`tagMatrix`	tagMatrix or a list of tagMatrix
`xlim`	xlim
`xlab`	xlab
`ylab`	ylab
`title`	title
`color`	color

Value

figure

Author(s)

G Yu

`vennplot` *vennplot*

Description

plot the overlap of a list of object

Usage

```
vennplot(Sets, by = "gplots")
```

Arguments

`Sets`	a list of object, can be vector or GRanges object
`by`	one of gplots or Vennerable

Value

venn plot that summarize the overlap of peaks from different experiments or gene annotation from different peak files.

Author(s)

G Yu

Examples

```
require(TxDb.Hsapiens.UCSC.hg19.knownGene)
txdb <- TxDb.Hsapiens.UCSC.hg19.knownGene
peakfile <- system.file("extdata", "sample_peaks.txt", package="ChIPseeker")
peakAnno <- annotatePeak(peakfile, TranscriptDb=txdb)
peakAnnoList <- lapply(1:3, function(i) peakAnno[sample(1:length(peakAnno), 100),])
names(peakAnnoList) <- paste("peak", 1:3, sep="_")
genes= lapply(peakAnnoList, function(i) unlist(i$geneId))
vennplot(genes)
```

`vennplot.peakfile` *vennplot.peakfile*

Description

vennplot for peak files

Usage

```
vennplot.peakfile(files, labels = NULL)
```

Arguments

`files`	peak files
`labels`	labels for peak files

Value

figure

Author(s)

G Yu

Index