Package 'epidecodeR'

November 25, 2026

Type Package

Title epidecodeR: a functional exploration tool for epigenetic and epitranscriptomic regulation

Version 1.22.0

BugReports https://github.com/kandarpRJ/epidecodeR/issues

URL https://github.com/kandarpRJ/epidecodeR, https://epidecoder.shinyapps.io/shinyapp

Depends R (>= 3.1.0)

Imports EnvStats, ggplot2, rtracklayer, GenomicRanges, IRanges, rstatix, ggpubr, methods, stats, utils, dplyr

biocViews DifferentialExpression, GeneRegulation, HistoneModification, FunctionalPrediction, Transcription, GeneExpression, Epitranscriptomics, Epigenetics, FunctionalGenomics, SystemsBiology, Transcriptomics, ChipOnChip

Description epidecodeR is a package capable of analysing impact of degree of DNA/RNA epigenetic chemical modifications on dysregulation of genes or proteins. This package integrates chemical modification data generated from a host of epigenomic or epitranscriptomic techniques such as ChIP-seq, ATAC-seq, m6A-seq, etc. and dysregulated gene lists in the form of differential gene expression, ribosome occupancy or differential protein translation and identify impact of dysregulation of genes caused due to varying degrees of chemical modifications associated with the genes. epidecodeR generates cumulative distribution function (CDF) plots showing shifts in trend of overall log2FC between genes divided into groups based on the degree of modification associated with the genes. The tool also tests for significance of difference in log2FC between groups of genes.

License GPL-3

Encoding UTF-8

LazyData false

RoxygenNote 7.1.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

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

git_branch RELEASE_3_24

git_last_commit 98300f6

git_last_commit_date 2026-11-23

Repository Bioconductor 3.24

Date/Publication 2026-11-24

Author Kandarp Joshi [aut, cre],
Dan Ohtan Wang [aut]

Maintainer Kandarp Joshi <kandarpbioinfo@gmail.com>

Contents

epidecodeR 2
epidecodeR-class 3
get_empirical_table 4
get_eventcounts 4
get_grpcounts 5
get_grptables 6
get_signtest 6
get_theoretical_table 7
makeplot 7
plottingfunc 8
plot_test 9

Index **10**

`epidecodeR` *Analysis function for generating epidecodeR object. This function distributes dysregulated genes into user defined groups and calculates cumulative probabilities and ANOVA test statistics for significance testing in difference of log2FC means between groups of dysregulated genes*

Description

Analysis function for generating epidecodeR object. This function distributes dysregulated genes into user defined groups and calculates cumulative probabilities and ANOVA test statistics for significance testing in difference of log2FC means between groups of dysregulated genes

Usage

```
epidecodeR(events, deg, gtf_file, id_type, boundaries, pval, param, ints)
```

Arguments

`events`	(char) - Name of events file. This can be a txt file with two columns: 1) id & 2) counts of events in the gene. Optionally, users can provide a 3+ column .bed file. The count of events per gene in fourth column are calculated to determine degree of events per gene; Default NULL
`deg`	(char) - Name of dysregulated genes file. This file is a three column file consisting of column 1: id (Make sure ID type matches between events and deg); column 2) log2foldchange; 3) P value of signficance of fold change; Default NULL

gtf_file	(char) - Name of compressed gtf file. Use gtf file if .bed file used as events input and users wish to count events per gene from bed file by comparing coordinates in bed to gene coordinates in gtf to assign events to genes. Note: For coordinates overlapping to multiple features in gtf, only one feature is assigned to the coordinate, which is choosen arbitrarily; Default NULL
id_type	(char) - Name of id type used to count events per gene. ID type must match between events and DEG file. For example, if 'gene_name' is used as ID type in DEG file, same ID type must be used to assign coordinates to genes. In case the DEG list contains two ID types merged e.g. 'ENSMUSG00000035299.16|Mid1' users can give merge as parameter for id_type; Default gene_name
boundaries	(numeric) - Number of base pairs to include within boundries of genes for event counting. This option adds # of bases to start and end of coordinates of the genes to include promotor regions within gene for overlap with bed files and event counting; Default 0
pval	(numeric) - P value cut-off of dysregulated genes in DEG file to be considered for distribution into groups. Default: 0.05
param	(numeric) - Defines the number and size of groups of dysregulated genes. Allowed values are param = 1 [0 events: 1+ events]; param = 2 [0 events: 1-N event: (N+1)+ event]; param = 3 [0 events; 1 event; 2-N events; (N+1)+ events]; N is user defined limit of the group provided using ints parameter
ints	(vector) - A vector of intervals defining limits of the degree of group for param = 2 and param = 3. e.g. c(1, 4) or c(2, 5): For param = 2, Default :c(1,4) and for param = 3, Default: c(2,5)

Value

An epidecodeR object containing tables of theoretical and empirical cumulative probabilities of the log2FC (quantiles), tables of genes distributed into user defined groups, counts of genes per user defined groups, table of one-way ANOVA significance testing for difference in mean log2FC of groups of genes

Examples

```
events<-system.file("extdata", "NOMO-1_ref_peaks.bed", package="epidecodeR")
deg<-system.file("extdata", "FTOi.txt", package="epidecodeR")
epiobj<-epidecodeR(events=events,deg=deg,pval=0.05,param=3,ints=c(2,4))
```

---

epidecodeR-class *epidecodeR object - a S4 class object*

Description

epidecodeR object - a S4 class object

Slots

`t` data.frame.

`e` data.frame.

`eventcounts` numeric.

grptables list.

grpcounts integer.

sign.test data.frame.

`get_empirical_table` *get_empirical_table method*

Description

get_empirical_table method

Usage

```
get_empirical_table(object)

## S4 method for signature 'epidecodeR'
get_empirical_table(object)
```

Arguments

object	epidecodeR object

Value

empirical_table

Examples

```
events<-system.file("extdata", "NOMO-1_ref_peaks.bed", package="epidecodeR")
deg<-system.file("extdata", "FTOi.txt", package="epidecodeR")
epiobj<-epidecodeR(events=events,deg=deg,pval=0.05,param=3,ints=c(2,4))
get_empirical_table(epiobj)
```

`get_eventcounts` *get_eventcounts method*

Description

get_eventcounts method

Usage

```
get_eventcounts(object)

## S4 method for signature 'epidecodeR'
get_eventcounts(object)
```

Arguments

object	epidecodeR object

Value

eventcounts

Examples

```
events<-system.file("extdata", "NOMO-1_ref_peaks.bed", package="epidecodeR")
deg<-system.file("extdata", "FTOi.txt", package="epidecodeR")
epiobj<-epidecodeR(events=events,deg=deg,pval=0.05,param=3,ints=c(2,4))
get_eventcounts(epiobj)
```

`get_grpcounts` *get_grpcounts method*

Description

get_grpcounts method

Usage

```
get_grpcounts(object)

## S4 method for signature 'epidecodeR'
get_grpcounts(object)
```

Arguments

`object`	epidecodeR object

Value

grpcounts

Examples

```
events<-system.file("extdata", "NOMO-1_ref_peaks.bed", package="epidecodeR")
deg<-system.file("extdata", "FTOi.txt", package="epidecodeR")
epiobj<-epidecodeR(events=events,deg=deg,pval=0.05,param=3,ints=c(2,4))
get_grpcounts(epiobj)
```

get_grptables *get_grptables method*

Description

get_grptables method

Usage

```
get_grptables(object)

## S4 method for signature 'epidecodeR'
get_grptables(object)
```

Arguments

object	epidecodeR object

Value

grptables

Examples

```
events<-system.file("extdata", "NOMO-1_ref_peaks.bed", package="epidecodeR")
deg<-system.file("extdata", "FTOi.txt", package="epidecodeR")
epiobj<-epidecodeR(events=events,deg=deg,pval=0.05,param=3,ints=c(2,4))
get_grptables(epiobj)
```

get_signtest *get_signtest method*

Description

get_signtest method

Usage

```
get_signtest(object)

## S4 method for signature 'epidecodeR'
get_signtest(object)
```

Arguments

object	epidecodeR object

Value

signtest table

Examples

```
events<-system.file("extdata", "NOMO-1_ref_peaks.bed", package="epidecodeR")
deg<-system.file("extdata", "FTOi.txt", package="epidecodeR")
epiobj<-epidecodeR(events=events,deg=deg,pval=0.05,param=3,ints=c(2,4))
get_signtest(epiobj)
```

---

get_theoretical_table *get_theoretical_table method*

Description

get_theoretical_table method

Usage

```
get_theoretical_table(object)

## S4 method for signature 'epidecodeR'
get_theoretical_table(object)
```

Arguments

object	epidecodeR object

Value

theoretical_table

Examples

```
events<-system.file("extdata", "NOMO-1_ref_peaks.bed", package="epidecodeR")
deg<-system.file("extdata", "FTOi.txt", package="epidecodeR")
epiobj<-epidecodeR(events=events,deg=deg,pval=0.05,param=3,ints=c(2,4))
get_theoretical_table(epiobj)
```

---

makeplot *Generate CDF plot using epidecodeR object generated using epidecodeR function*

Description

Generate CDF plot using epidecodeR object generated using epidecodeR function

Usage

```
makeplot(obj, type, lim, title, xlab, ylab)
```

Arguments

obj	epidecodeR object - epidecodeR object generated using epidecodeR function
type	char - Type of CDF plot to generate; Accepted values 't': theoretical CDF plot; 'e': empirical CDF plot; 'both': Creates both theoretical and empirical plots. Default: both
lim	vector - Upper and lower limits of log2FC for X-axis
title	char - Title of the plot
xlab	char - X-axis label
ylab	char - Y-axis label

Value

A CDF plot

Examples

```
events<-system.file("extdata","NOMO-1_ref_peaks.bed", package="epidecodeR")
deg<-system.file("extdata", "FTOi.txt", package="epidecodeR")
epiobj<-epidecodeR(events=events,deg=deg,pval=0.05,param=3,ints=c(2,4))
makeplot(epiobj, lim = c(-10,10), xlab = "log2FC")
```

---

plottingfunc *Title*

Description

Title

Usage

```
plottingfunc(objdf, title, lim, xlab, ylab)
```

Arguments

objdf	Data frame of the object from epidecodeR object
title	Title of the plot
lim	Limits of x-axis
xlab	X-axis name
ylab	Y-axis name

Value

returns a generic ggplot to be filled by makeplot function

Examples

```
## Not run:
events<-system.file("extdata","NOMO-1_ref_peaks.bed", package="epidecodeR")
deg<-system.file("extdata", "FTOi.txt", package="epidecodeR")
epiobj<-epidecodeR(events=events,deg=deg,pval=0.05,param=3,ints=c(2,4))
plottingfunc(objdf=get_theoretical_table(epiobj), title="title",
lim=c(-2,2), xlab="xlab", ylab="ylab")

## End(Not run)
```

`plot_test` *Generates boxplot of distribution of log2FC of dysregulated genes and adjusted P value of signficance test of difference in mean log2FC between groups computed using one-way ANOVA test*

Description

Generates boxplot of distribution of log2FC of dysregulated genes and adjusted P value of signficance test of difference in mean log2FC between groups computed using one-way ANOVA test

Usage

```
plot_test(obj, title, ylab)
```

Arguments

`obj`	epidecodeR object - epidecodeR object generated using epidecodeR function
`title`	Title of the plot
`ylab`	Y-axis label

Value

Boxplot of distribution of log2FC of dysregulated genes between groups

Examples

```
events<-system.file("extdata","NOMO-1_ref_peaks.bed",package="epidecodeR")
deg<-system.file("extdata", "FTOi.txt", package="epidecodeR")
epiobj<-epidecodeR(events=events,deg=deg,pval=0.05,param=3,ints=c(2,4))
plot_test(epiobj,title="log2FC distribution based on m6A degree",ylab="log2FC")
```

Index

∗ **internal**
plottingfunc, 8

epidecodeR, 2
epidecodeR-class, 3

get_empirical_table, 4
get_empirical_table,epidecodeR-method
*(*get_empirical_table*)*, 4
get_eventcounts, 4
get_eventcounts,epidecodeR-method
*(*get_eventcounts*)*, 4
get_grpcounts, 5
get_grpcounts,epidecodeR-method
*(*get_grpcounts*)*, 5
get_grptables, 6
get_grptables,epidecodeR-method
*(*get_grptables*)*, 6
get_signtest, 6
get_signtest,epidecodeR-method
*(*get_signtest*)*, 6
get_theoretical_table, 7
get_theoretical_table,epidecodeR-method
*(*get_theoretical_table*)*, 7

makeplot, 7

plot_test, 9
plottingfunc, 8