

# Package ‘epidecodeR’

November 20, 2024

**Type** Package

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

**Version** 1.14.0

**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\_20

**git\_last\_commit** 2e8d50f

**git\_last\_commit\_date** 2024-10-29

**Repository** Bioconductor 3.20

**Date/Publication** 2024-11-20

**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

<b>Index</b>	<b>10</b>
--------------	-----------

---

epidecodeR	<i>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</i>
------------	--

---

## 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 significance of fold change; Default NULL

<code>gtf_file</code>	(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 chosen arbitrarily; Default NULL
<code>id_type</code>	(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 <code>id_type</code> ; Default <code>gene_name</code>
<code>boundaries</code>	(numeric) - Number of base pairs to include within boundaries 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
<code>pval</code>	(numeric) - P value cut-off of dysregulated genes in DEG file to be considered for distribution into groups. Default: 0.05
<code>param</code>	(numeric) - Defines the number and size of groups of dysregulated genes. Allowed values are <code>param = 1</code> [0 events: 1+ events]; <code>param = 2</code> [0 events: 1-N event: (N+1)+ event]; <code>param = 3</code> [0 events; 1 event; 2-N events; (N+1)+ events]; N is user defined limit of the group provided using <code>ints</code> parameter
<code>ints</code>	(vector) - A vector of intervals defining limits of the degree of group for <code>param = 2</code> and <code>param = 3</code> . e.g. <code>c(1, 4)</code> or <code>c(2, 5)</code> ; For <code>param = 2</code> , Default: <code>c(1,4)</code> and for <code>param = 3</code> , Default: <code>c(2,5)</code>

**Value**

An `epidecodeR` object containing tables of theoretical and empirical cumulative probabilities of the  $\log_2FC$  (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  $\log_2FC$  of groups of genes

**Examples**

```
events<-system.file("extdata", "NOMO-1_ref_peaks.bed", package="epidecodeR")
deg<-system.file("extdata", "FT0i.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", "FT0i.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", "NOM0-1_ref_peaks.bed", package="epidecodeR")
deg<-system.file("extdata", "FT0i.txt", package="epidecodeR")
epiobj<-epidecodeR(events=events,deg=deg,pval=0.05,param=3,ints=c(2,4))
get_eventcounts(epiobj)
```

---

get_grpcounts	<i>get_grpcounts method</i>
---------------	-----------------------------

---

**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", "NOM0-1_ref_peaks.bed", package="epidecodeR")
deg<-system.file("extdata", "FT0i.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", "NOM0-1_ref_peaks.bed", package="epidecodeR")
deg<-system.file("extdata", "FT0i.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="epicodeR")
deg<-system.file("extdata", "FT0i.txt", package="epicodeR")
epiobj<-epicodeR(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 'epicodeR'
get_theoretical_table(object)
```

**Arguments**

object            epicodeR object

**Value**

theoretical\_table

**Examples**

```
events<-system.file("extdata", "NOMO-1_ref_peaks.bed", package="epicodeR")
deg<-system.file("extdata", "FT0i.txt", package="epicodeR")
epiobj<-epicodeR(events=events,deg=deg,pval=0.05,param=3,ints=c(2,4))
get_theoretical_table(epiobj)
```

---

makeplot	<i>Generate CDF plot using epicodeR object generated using epicodeR function</i>
----------	--

---

**Description**

Generate CDF plot using epicodeR object generated using epicodeR 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", "FT0i.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", "FT0i.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	<i>Generates boxplot of distribution of log2FC of dysregulated genes and adjusted P value of significance test of difference in mean log2FC between groups computed using one-way ANOVA test</i>
-----------	--

---

## Description

Generates boxplot of distribution of log2FC of dysregulated genes and adjusted P value of significance 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", "FT0i.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