

# Package ‘DiscoRhythm’

October 16, 2019

**Title** Interactive Workflow for Discovering Rhythmicity in Biological Data

**Version** 1.0.0

**Description** Set of functions for estimation of cyclical characteristics, such as period, phase, amplitude, and statistical significance in large temporal datasets. Supporting functions are available for quality control, dimensionality reduction, spectral analysis, and analysis of experimental replicates. Contains a R Shiny web interface to execute all workflow steps.

**Depends** R (>= 3.6.0)

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.1.1

**VignetteBuilder** knitr

**Imports** matrixTests, matrixStats, MetaCycle (>= 1.2.0), data.table, ggplot2, ggExtra, dplyr, broom, shiny, shinyBS, shinycssloaders, shinydashboard, shinyjs, BiocStyle, rmarkdown, knitr, kableExtra, magick, VennDiagram, UpSetR, heatmaply, viridis, plotly, DT, gridExtra, methods, stats, SummarizedExperiment, BiocGenerics, S4Vectors

**Suggests** testthat

**SystemRequirements** To generate html reports pandoc (<http://pandoc.org/installing.html>) is required.

**biocViews** Software, TimeCourse, QualityControl, Visualization, GUI, PrincipalComponent

**URL** <https://github.com/matthewcarlucci/DiscoRhythm>

**BugReports** <https://github.com/matthewcarlucci/DiscoRhythm/issues>

**git\_url** <https://git.bioconductor.org/packages/DiscoRhythm>

**git\_branch** RELEASE\_3\_9

**git\_last\_commit** 6986d78

**git\_last\_commit\_date** 2019-05-02

**Date/Publication** 2019-10-15

**Author** Matthew Carlucci [aut, cre],  
 Algimantas Kriščiūnas [aut],  
 Haohan Li [aut],  
 Povilas Gibas [aut],  
 Karolis Koncevičius [aut],  
 Art Petronis [aut],  
 Gabriel Oh [aut]

**Maintainer** Matthew Carlucci <Matthew.Carlucci@camh.ca>

## R topics documented:

discoApp . . . . .	2
discoBatch . . . . .	3
discoCheckInput . . . . .	4
discoDesignSummary . . . . .	5
discoDFtoSE . . . . .	6
discoGetSimu . . . . .	7
discoODAexclusionMatrix . . . . .	7
discoODAid2name . . . . .	8
discoParseMeta . . . . .	9
discoPCA . . . . .	9
discoPeriodDetection . . . . .	10
discoQC . . . . .	11
<b>Index</b>	<b>13</b>

---

discoApp

*Launch the DiscoRhythm Shiny Application*

---

### Description

This launches the web interface to DiscoRhythm containing all analysis tools. The vignette contains details on usage.

### Usage

```
discoApp(ncores = 1, port = 3838)
```

### Arguments

`ncores` numeric, number of cores to use for parallelized tasks. Currently, only used in oscillation detection function `discoODAs`.

`port` numeric, port to run the shiny application on. Sets `shiny.port` option.

### Value

Nothing is returned by this function.

**Examples**

```
## Not run:
discoApp()

## End(Not run)
```

---

discoBatch

*Core DiscoRhythm Workflow*


---

**Description**

Execute the DiscoRhythm workflow with one command to obtain the results of oscillation detection (discoODAs) and optionally generate an html report with data visualizations from an Rmarkdown template. See the DiscoRhythm vignette for more details on the analysis procedures.

**Usage**

```
discoBatch(indata, report = NULL, outdata = TRUE, ncores = 1,
  timeType = "circular", main_per = 24, cor_threshold = 3,
  cor_method = "pearson", cor_threshType = "sd", pca_threshold = 3,
  pca_scale = TRUE, pca_pcToCut = paste0("PC", seq_len(4)),
  aov_method = "None", aov_pcut = 0.05, aov_Fcut = 0,
  avg_method = "Median", osc_method = NULL, osc_period = 24)
```

**Arguments**

indata	SummarizedExperiment or data.frame, see the vignette for the specific formats expected for each of these input types. discoParseMeta.
report	character, if !is.null(report) an html report with
outdata	logical, whether to return the final discoODAs (note if run with is.null(report) discoBatch will return nothing).
ncores	numeric, number of cores to use for parallelized tasks. Currently, only used in oscillation detection function discoODAs.
timeType	character, nature of the sample times provided (one of "circular" or "linear").
main_per	numeric, the length of the main hypothesized period (e.g. 24hr for circadian experiments). Used in discoPeriodDetection.
cor_threshold	numeric, threshold used in inter-sample correlation analysis for outlier detection. Either in units of correlation coefficient or standard deviations from the mean (see cor_threshType).
cor_method	character, which correlation method to use for outlier removal (see cor for more details).
cor_threshType	character, one of "sd" or "value" indicating whether cor_threshold should be set by absolute correlation coefficient or by standard deviations from the mean of all samples.
pca_threshold	numeric, the number of standard deviations to set as the threshold for outlier detection in PCA outlier removal.
pca_scale	logical, whether to scale the data prior to PCA.

pca_pcToCut	character, names of which PCs to use for outlier detection (e.g. "PC1", "PC2" etc.).
aov_method	character, method to use for ANOVA. One of: "Equal Variance", "Welch", or "None".
aov_pcut	numeric, p-value cutoff used to select rows with statistically significant signal-to-noise.
aov_Fcut	numeric, F-statistic cutoff used to select rows with high signal-to-noise based on magnitude.
avg_method	character, method for averaging technical replicates. One of: "Median", "Mean", "Random", or "None".
osc_method	character, vector of oscillation detection algorithms to apply to the data. Methods that are detminded to be innappropraite for the experimental design (using the discoODAexclusionMatrix) will be ignored. If <code>is.null(osc_method)</code> all suitable methods will be executed.
osc_period	numeric, a fixed period to use for oscillation detection using all methods.

**Value**

returns the results of discoODAs

**See Also**

discoODAs, discoRepAnalysis, discoPeriodDetection, discoPCAoutliers, discoInterCorOurliers

**Examples**

```

indata <- discoGetSimu()

# Batch execute (on demo data) to generate a DiscoRhythm_report.html report.
# Returns the results of discoODAs
discoODAs <- discoBatch(indata,
  report="DiscoRhythm_report.html",
  osc_method="CS")

```

---

discoCheckInput

*Import Data for DiscoRhythm Analysis*

---

**Description**

Performs various checks and cleaning operations on the input data.

**Usage**

```
discoCheckInput(se, n_min_sample = 3)
```

**Arguments**

se	SummarizedExperiment, the main data object used by DiscoRhythm expected to contain se\$ID, se\$ReplicateID, se\$Time sample metadata and non-null row-names. See the vignette for more details.
n_min_sample	numeric value specifying minimal number of samples needed to perform analysis.

**Details**

Rows containing NA's or all constant values are removed. If matrix values are character it will be attempted to convert them to numeric. If input is not a matrix it will be converted using `as.matrix()`. User will be warned if row IDs contain duplicate entries.

**Value**

SummarizedExperiment checked for errors and modified as needed

**Examples**

```
se <- discoGetSimu(TRUE)
se_clean <- discoCheckInput(se)
```

---

`discoDesignSummary`      *Summarize the experimental design*

---

**Description**

Using sample times and biological sample IDs, constructs a summary table of the number of total samples at each timepoint and additionally summarizes the number of replicates for each biological sample.

**Usage**

```
discoDesignSummary(Metadata)
```

**Arguments**

`Metadata`      data.frame of sample data, usually generated by using `discoParseMeta` on the column names of the `Maindata` data.frame. If `is.null(Metadata)` and `Maindata` is provided as input, `Metadata` will be generated from `Maindata`.

**Value**

A table where the first row summarizes the number of datapoints for each timepoint and other cells indicate the number of technical replicates for a given biological sample.

**See Also**

`discoParseMeta`

**Examples**

```
# import example data
Metadata <- SummarizedExperiment::colData(discoGetSimu(TRUE))
# Summarize the experiment design
discoDesignSummary(Metadata)
```

discoDFtoSE

*Data formatting for DiscoRhythm***Description**

Functions to import a data.frame (from the format expected by the web application discoApp()) as a SummarizedExperiment object or to export a SummarizedExperiment for use with the web application.

**Usage**

```
discoDFtoSE(Maindata, Metadata = NULL, shinySession = NULL)
```

```
discoSEtoDF(se)
```

**Arguments**

Maindata	data.frame with the first column containing row IDs and all subsequent columns containing experimental values. Columns should follow the expected naming format described in the vignette.
Metadata	data.frame of sample data, usually generated by using discoParseMeta on the column names of the Maindata data.frame. If is.null(Metadata) and Maindata is provided as input, Metadata will be generated from Maindata.
shinySession	shiny session object for use only by the DiscoRhythm shiny app discoApp() to update the axis labels using the time value prefix.
se	SummarizedExperiment, the main data object used by DiscoRhythm expected to contain se\$ID, se\$ReplicateID, se\$Time sample metadata and non-null row-names. See the vignette for more details.

**Value**

A SummarizedExperiment object with colData containing sample metadata.

A DiscoRhythm format Maindata data.frame.

**Examples**

```
df <- discoGetSimu()
se <- discoDFtoSE(df)

df <- discoSEtoDF(se)
```

---

`discoGetSimu`*Read in the DiscoRhythm Simulated dataset*

---

**Description**

A convenience function to get the simulated circadian transcriptomic system data file used in DiscoRhythm for various demonstrations and tests.

**Usage**

```
discoGetSimu(as_se = FALSE)
```

**Arguments**

`as_se` logical, indicates if example data should be returned as a SummarizedExperiment or data.frame.

**Value**

The simulated demo dataset used in the DiscoRhythm web application as a data.frame or SummarizedExperiment.

**Examples**

```
indata <- discoGetSimu()
```

---

`discoODAexclusionMatrix`*Algorithm Exclusion Matrix*

---

**Description**

A small matrix indicating which algorithms should be excluded given certain experimental designs and data types.

**Usage**

```
discoODAexclusionMatrix
```

**Format**

An object of class `matrix` with 4 rows and 7 columns.

## Examples

```
# Code used to generate discoODAexclusionMatrix

itemNames <- c(
  "missing_value",
  "with_bio_replicate",
  "non_integer_interval",
  "uneven_interval",
  "circular_t",
  "invalidPeriod",
  "invalidJTKperiod"
)

# Creating requirements matrix, first assuming all methods are valid
# Then applying exclusion criteria of MetaCycle plus CS criteria
mat <- matrix(TRUE, nrow = 4, ncol = length(itemNames))
rownames(mat) <- c("CS", "JTK", "LS", "ARS")
colnames(mat) <- itemNames

# Exclusion criteria from MetaCycle v1.1, i.e. can algorithm handle XXX
mat[c("ARS", "JTK"), c("non_integer_interval", "uneven_interval")] <- FALSE
mat["ARS", "with_bio_replicate"] <- FALSE
mat["ARS", "missing_value"] <- FALSE
mat["JTK", "invalidJTKperiod"] <- FALSE

# Additional exclusion criteria
mat["ARS", "circular_t"] <- FALSE
mat["CS", "missing_value"] <- FALSE
mat[c("CS", "JTK", "ARS", "LS"), "invalidPeriod"] <- FALSE

discoODAexclusionMatrix <- mat
```

---

discoODAid2name

*Mapping Identifiers to Full Names*

---

## Description

A small named vector mapping oscillation detection algorithm names to a convenient identifier.

## Usage

```
discoODAid2name
```

## Format

A named vector, length 4

**names(discoODAid2name)** Identifier

**as.vector(discoODAid2name)** Full names



---

discoParseMeta                      *Generate Experiment Metadata*

---

### Description

Parses the sample metadata from a vector of sample names (often column names of a Maindata format data.frame).

### Usage

```
discoParseMeta(samplenames, shinySession = NULL)
```

### Arguments

**samplenames**      character, a list of sample names following the DiscoRhythm naming convention (<prefix><Time>\_<UniqueID>\_<ReplicateID>).

**shinySession**      shiny session object for use only by the DiscoRhythm shiny app discoApp() to update the axis labels using the time value prefix.

### Details

The regular expression used to obtain metadata is `"^([[:alpha:]]*)(\\-[0-9]+[\\.]?[0-9]*)\\_? ([[:alnum:]]*)\\_?([[:alnum:]]*)$"`

Where each () will be used to construct the final metadata data.frame

### Value

a data.frame containing 3 columns of metadata. ID = unique sample identity. Time = sample collection time. ReplicateID = Identifier where Time + ReplicateID indicates a biological sample ID.

### Examples

```
discoParseMeta(c("CT24_AD_1", "CT24_AS_1", "CT24_AE_2", "CT24_AW_2",
"CT26_AB_1", "CT26_AC_1", "CT26_BB_2", "CT26_BC_2"))
```

---

discoPCA                              *Perform PCA*

---

### Description

Calculates PCA results from prcomp with error handling and outputs suitable for the DiscoRhythm workflow.

### Usage

```
discoPCA(se, scale = TRUE, npcs = 10)
```

**Arguments**

se	SummarizedExperiment, the main data object used by DiscoRhythm expected to contain se\$ID, se\$ReplicateID, se\$Time sample metadata and non-null row-names. See the vignette for more details.
scale	logical, whether or not to scale the data prior to PCA, see <a href="#">prcomp</a> for more details.
npcs	numeric, maximum number of principal components to return.

**Value**

output from [prcomp](#) with an added table summary

**Examples**

```
se <- discoGetSimu(TRUE)
pca <- discoPCA(se)
```

---

discoPeriodDetection *Detect dataset-wide fits to multiple periodicities*

---

**Description**

Detect dataset-wide fits to multiple periodicities

**Usage**

```
discoPeriodDetection(se, timeType = c("linear", "circular"),
  main_per = 24, test_periods = NULL)
```

**Arguments**

se	SummarizedExperiment, the main data object used by DiscoRhythm expected to contain se\$ID, se\$ReplicateID, se\$Time sample metadata and non-null row-names. See the vignette for more details.
timeType	character, time is either reported as "linear" or "circular" on some base-cycle (ex. time of day). This determines the periods that will be tested for.
main_per	numeric, if timeType=="circular" main_per indicates the period of the base-cycle where sampling times are derived.
test_periods	numeric, a vector of the periods to test. if timeType=="linear" and length(test_periods)==2 it will be assumed to be a range of periods to test over.

**Value**

A data.frame of Rsquared values for each period, for each row of Maindata.

**Examples**

```
se <- discoGetSimu(TRUE)

# Detect periods
rsqs <- discoPeriodDetection(se)
```

discoQC

*Quality Control for DiscoRhythm***Description**

Functions for executing outlier detection and row filtering procedures prior to rhythmicity analysis.

**Usage**

```
discoPCAoutliers(se, threshold = 3, scale = TRUE,
  pcToCut = seq_len(4))

discoInterCorOutliers(se, cor_method = c("pearson", "kendall",
  "spearman"), threshold = 3, thresh_type = c("sd", "value"))

discoRepAnalysis(se, aov_method = c("Equal Variance", "Welch", "None"),
  aov_pcut = 0.05, aov_Fcut = 0, avg_method = c("Median", "Mean",
  "Random", "None"))
```

**Arguments**

se	SummarizedExperiment, the main data object used by DiscoRhythm expected to contain se\$ID, se\$ReplicateID, se\$Time sample metadata and non-null row-names. See the vignette for more details.
threshold	numeric, a threshold determining which samples are outliers (for discoInterCorOutliers, in units of thresh_type, for discoPCAoutliers in units of standard deviations).
scale	logical, whether or not to scale the data prior to PCA, see <a href="#">prcomp</a> for more details.
pcToCut	numeric, which PCs to use for outlier detection. It is recommended to select the first X PCs based on which PCs explain a significant amount of variance in the data.
cor_method	character, method of pairwise correlation (see <a href="#">cor</a> 's "method" argument for all options).
thresh_type	character indicating threshold type (either standard deviations below the mean, or an absolute correlation value). One of: "sd" or "value".
aov_method	character, method to use for ANOVA. One of: "Equal Variance", "Welch", or "None".
aov_pcut	numeric, p-value cutoff used to select rows with statistically significant signal-to-noise.
aov_Fcut	numeric, F-statistic cutoff used to select rows with high signal-to-noise based on magnitude.
avg_method	character, method for averaging technical replicates. One of: "Median", "Mean", "Random", or "None".

**Value**

list containing PCA results and the detected outliers

A list of 3 objects: 1) outliers - named logical indicating if the sample is an outlier 2) meanCor - mean of all pairwise correlations for a given sample 3) corMat - Matrix of all pairwise correlation values

**Examples**

```
se <- discoGetSimu(TRUE)
PCARes <- discoPCAoutliers(se)

CorRes <- discoInterCorOutliers(se)

ANOVARes <- discoRepAnalysis(se)
```

# Index

## \*Topic **datasets**

discoODAexclusionMatrix, [7](#)  
discoODAid2name, [8](#)

cor, [3](#), [11](#)

discoApp, [2](#)

discoBatch, [3](#)

discoCheckInput, [4](#)

discoDesignSummary, [5](#)

discoDFtoSE, [6](#)

discoGetSimu, [7](#)

discoInterCorOutliers (discoQC), [11](#)

discoODAexclusionMatrix, [7](#)

discoODAid2name, [8](#)

discoParseMeta, [9](#)

discoPCA, [9](#)

discoPCAoutliers (discoQC), [11](#)

discoPeriodDetection, [10](#)

discoQC, [11](#)

discoRepAnalysis (discoQC), [11](#)

discoSEtoDF (discoDFtoSE), [6](#)

prcomp, [10](#), [11](#)