

# DiffCircaPipeline R Shiny tutorial

August 31, 2022

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Preliminaries</b>	<b>2</b>
2.1	Citing DiffCircaPipeline . . . . .	2
2.2	Installation . . . . .	2
2.3	The welcome page . . . . .	3
2.4	Abbreviation terms and notations . . . . .	4
<b>3</b>	<b>Data Input and Result Output</b>	<b>4</b>
3.1	Data preparation . . . . .	4
3.2	Data input . . . . .	4
3.3	Study output . . . . .	6
3.4	Study summary . . . . .	6
3.5	Continue from existing study . . . . .	6
<b>4</b>	<b>Joint Rhythmicity</b>	<b>6</b>
4.1	Analyses and results . . . . .	6
4.2	Parameters . . . . .	8
<b>5</b>	<b>Differential rhythmicity analysis</b>	<b>9</b>
5.1	Differential Rhythm Parameter . . . . .	9
5.2	Differential Rhythm Fitness . . . . .	10
<b>6</b>	<b>Visualization tools</b>	<b>10</b>
6.1	Scatter Plots . . . . .	10
6.2	Heatmaps . . . . .	12
6.3	Radar plot . . . . .	12
6.4	Circos phase difference plot . . . . .	13
<b>7</b>	<b>Single-group analysis</b>	<b>14</b>

## 1 Introduction

DiffCircaPipeline is a differential rhythmicity analysis framework. It is able to evaluate multifaceted characteristics of differential rhythmic biomarkers with biological reasoning and statistical control of false positives. We developed the DiffCircaPipeline Shiny app to perform the pipeline systematically, allowing extensive visualization and interactive exploration of the differential rhythmicity findings. The pipeline applies to various omics data types, e.g., transcriptomic, proteomic, and methylation data. This manual uses transcriptomic data as an example and calls the biological features genes. For a detailed description and statistical validation of the analysis, please see [cite preprint here](#) and supplementary material.

Although DiffCircaPipeline is designed for differential rhythmicity analysis, the Shiny also performs single-group rhythmicity analysis. Throughout the manual Section 3 to 6, we will focus on illustrating the two-group differential rhythmicity analysis and leave the single-group analysis to Section 7.

Figure 1 gives the workflow of the pipeline. And below are the four modules of the Shiny:

- **Data Input and Result Output:** In this module, users input preprocessed gene expression data and time data. Users will also select an output directory for the analysis result.
- **Joint Rhythmicity:** This module corresponds to step (a) in Figure 1. It categorizes the genes into four types of joint rhythmicity: 1) rhythmic only in group I (RhyI); 2) rhythmic only in group II (RhyII); 3) rhythmic in both groups (RhyBoth); 4) arrhythmic in both groups (Arrhy). This step prepares genes for biological meaningful differential rhythmicity tests.
- **Differential Rhythmicity Analysis:** This module corresponds to step (b) in Figure 1. Two differential rhythmicity (DR) tests are performed for prepared gene categories. For genes that are rhythmic in more than one group, we perform the DR fitness test for  $\Delta R^2$ , and for genes that are rhythmic in both groups, we perform DR parameter tests for  $\Delta A$ ,  $\Delta \phi$  or  $\Delta M$ .
- **Visualization:** We provide four visualization tools. 1) The **scatter plot** shows a single gene's expression pattern across time with the fitted cosinor curve. 2) The **heatmap** plots the gene expression intensity across time for selected genes. 3) The **radar plot** summarizes the phase distribution of rhythmic genes in each group. 4) **Circos phase difference plot** visualize the differential phase result.

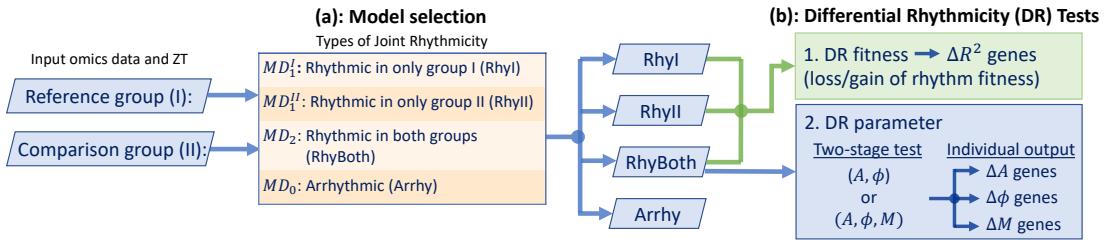


Figure 1: DiffCircaPipeline Workflow.

## 2 Preliminaries

### 2.1 Citing DiffCircaPipeline

- cite preprint here

### 2.2 Installation

#### Requirement:

- R  $\geq 4.0.0$
- R studio
- Shiny  $\geq 1.0.0$

#### Note:

- R version higher than 4.0 is recommended. If you are using R 3.5 or lower, you may encounter errors in installing dependencies.

#### How to install the software:

```

1 rm(list=ls())
2 #Download R Shiny from github and extract it, you will see a folder called "RShiny-main".
3 #Set the working directory to be the path to the folder "RShiny-main" (excluding "/RShiny-main" from the directory).
4 path_to_DiffCircaPipelineRShiny = "/THE_DIRECTORY_TO_THE_FOLDER_CONTAINING_RShiny-main" (1)
5 setwd(path_to_DiffCircaPipelineRShiny)
6 shiny::runApp('RShiny-main', port=9987, launch.browser=T)
7

```

Figure 2: How to run the Shiny from R.

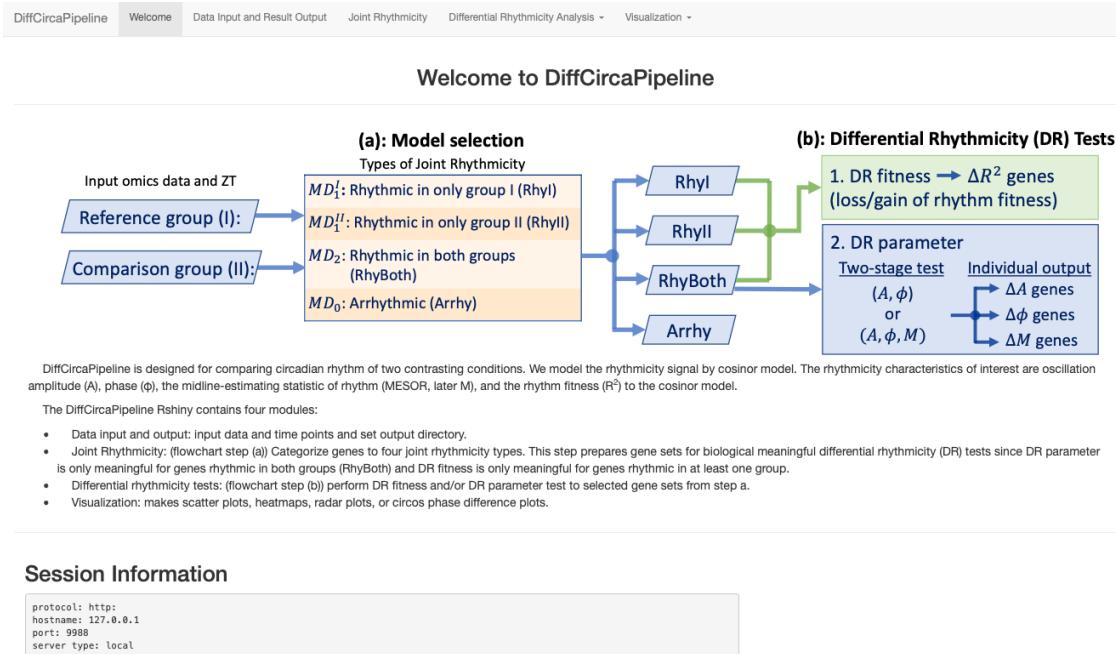


Figure 3: Welcome page

- Install the DiffCircaPipeline R package and its dependencies following the instructions <https://github.com/DiffCircaPipeline>
- Download the DiffCircaPipeline Shiny project at <https://github.com/DiffCircaPipeline/Rshiny> by clicking on “code” – > “Download ZIP”, then extract the ZIP file to a local folder. You should get a folder called “RShiny-main”.
- You will see an “app.R” file in the “RShiny-main” folder. Open it in R studio.
- In “app.R”, change the “/THE\_DIRECTORY\_TO\_THE\_FOLDER\_CONTAINING\_RShiny-main” to the directory of the folder “RShiny-main” (excluding “RShiny-main” from the directory) (Figure 2(1)).
- Click on the “Run App” bottom (Figure 2(2)), and the DiffCircaPipeline Shiny will open in your default browser.

### 2.3 The welcome page

After the Shiny opens, you will see the welcome page (Figure 3). The welcome pages displays the workflow with a simple introduction. At the top of the page are the four functional modules as introduced. At the bottom of the page you can find the session information.

	A	B	C	D	
1		gl_1	gl_2	gl_3	
2	gene 1	3.52177541	3.86417835	0.99206354	
3	gene 2	2.20553068	5.25917028	4.25886047	
4	gene 3	4.68594942	4.57831335	3.85096082	
5	gene 4	3.37010449	2.28365178	4.49345588	
6	gene 5	4.44208612	5.23400559	5.80917006	
7	gene 6	4.2904513	4.08503541	4.268396	
8	gene 7	3.48614817	2.89674326	2.75593529	
9	gene 8	5.91884376	1.94546264	2.66498601	
10	gene 9	4.85115528	5.38985704	4.26244542	
11	gene 10	3.80605849	5.46672679	5.74446228	

	A	B	C
1		SampleID	time
2	1 gl_1	12.5498731	
3	2 gl_2	6.31356119	
4	3 gl_3	9.37201223	
5	4 gl_4	8.45972512	
6	5 gl_5	4.66092657	
7	6 gl_6	8.88161545	
8	7 gl_7	19.1047759	
9	8 gl_8	7.22662064	
10	9 gl_9	13.2379342	
11	10 gl_10	9.18831342	

A

B

Figure 4: Data input format example. A. Format for expression data input. B. Format for time data input.

## 2.4 Abbreviation terms and notations

- *A*: amplitude
- phase: used interchangeably with peak time or  $\phi$
- *M*: midline-estimating statistic of rhythm (MESOR)
- TOJR: type of joint rhythmicity
- DR: differential rhythmicity

## 3 Data Input and Result Output

This section introduces the preparation of the data and how to input the data into the module.

### 3.1 Data preparation

DiffCircaPipeline requires **omics expression data** and **time data** for each group. The expression data have genes in rows and samples in columns. Figure 4 shows examples for format of the two inputs. The first column is gene names, and the first row is sample IDs. The time data must contain the following two columns: sample IDs that match the column names in the expression data, and time for recorded time for each sample. The headers must be “SampleID” and “time” respectively.

The expression data should be preprocessed before input. For RNA-seq data, the common preprocessing procedure includes filtering out the low-expressed transcripts, normalization between samples, and log transformation to achieve a Gaussian-like distribution.

### 3.2 Data input

Figure 5 shows the user interface (UI) of the Data Input and Result Output module. To start a new two-group differential rhythmicity analysis, select “No and upload data” in Figure 5(1) and “Rhythmicity Comparison of two groups” in Figure 5(2). Then users will see input areas for group I data and group II data separately (Figure 5(3) is the input area for group I data). The group I will be used as the reference group. By clicking the “Browse...” button, users can browse the local folders for prepared input data. Users should also specify group labels for both groups. The group labels will be used for result outputs and plots annotations. In this manual, we illustrate with the Case study 2 in [cite preprint](#): group I is labeled “Control” and group II is labeled “SCZ”.

DiffCircaPipeline   Welcome   Data Input and Result Output   Joint Rhythmicity   Differential Rhythmicity Analysis ▾   Visualization ▾

### ?Upload data

Continue from existing analysis?

Yes and choose directory (1)

No and upload data

Choose study type:

Rhythmicity comparison of two groups (2)

Rhythmicity analysis of a single group (no differential rhythmicity analysis)

Group I label: (3)  
Control

Upload group I gene expression data file (.csv)  
Browse... control\_data.csv  
Upload complete

Upload group I time of expression file (.csv)  
Browse... control\_meta.csv  
Upload complete

Group II label:  
SCZ

Upload group II gene expression data file (.csv)  
Browse... SCZ\_data.csv  
Upload complete

Upload group II time of expression file (.csv)  
Browse... SCZ\_meta.csv  
Upload complete

Output directory: (4)  
~/Library/CloudStorage/OneDrive-UniversityofPit  
...

Study name: (5)  
SCZvsControl

Save

### Study summary

group	nSample	nGene
Control	46	14496
SCZ	46	14496

### Distribution of sampling time

(6)  
(7)  
 Study SCZvsControl is successfully uploaded to /SCZvsControl

Figure 5: UI of module Data Input and Result Output.

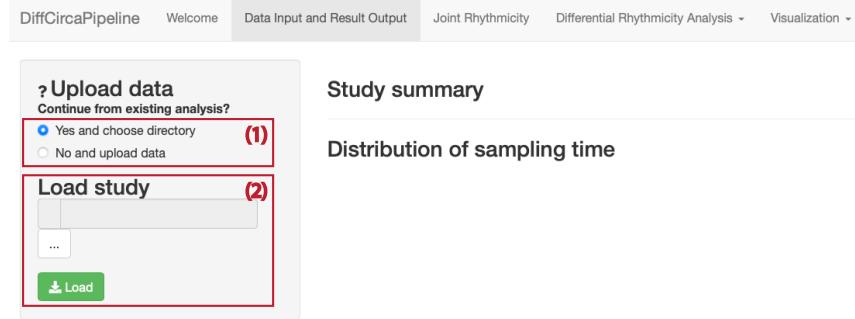


Figure 6: UI of loading an existing analysis.

### 3.3 Study output

Users need to select an output directory in Figure 5(4) and specify a study name in Figure 5(5). Immediately after users click the “Save” bottom, the output folder with the study name will be created in the selected directory. And a notification of a successfully save will show up in the corner of the panel (Figure 5(5)).

For example, if the directory selected is “MyOut”, and the study name is set to “MyStudy” then all outputs will be saved to “MyOut/MyStudy/” and all the intermediate results will be save to “MyOut/MyStudy/save/”.

### 3.4 Study summary

After data input, a study summary will automatically show on the right panel (Figure 5(6)). The table on the top summarizes the number of samples and number of genes in each group, and the histograms at the bottom show the distribution of time for both groups.

### 3.5 Continue from existing study

The Shiny offers the option to load an existing study and continue to perform more analyses. To do so, select “Yes and choose directory” as in Figure 6(1) and select the folder of saved study (e.g., “MyOut/MyStudy/”, if continuing from the example in Section 3.3), then click the “Load” button in Figure 6(2).

## 4 Joint Rhythmicity

The module Joint Rhythmicity categorize the genes into four types of joint rhythmicity (TOJR):

- rhythmic only in group I (RhyI)
- rhythmic only in group II (RhyII)
- rhythmic in both groups (RhyBoth)
- arrhythmic in both groups (Arrhy).

This step ensures that in the next module, the differential rhythmicity analyses will only be performed for genes that are biologically meaningful for the tests.

Users can perform multiple tests with varied parameters, and the corresponding numbers of genes in each TOJR category by different cutoffs will be summarized.

### 4.1 Analyses and results

By clicking the “Start analysis” button in Figure 7(1), the Shiny performs rhythmicity analysis for the two groups and save the results in the output directory with group labels specified in Section 3.2 (see Figure 7(5) for example outputs). These outputs contain the estimates from fitting the cosinor model:

**A**

**B**

**Parameter estimates**

Summary of TOJR Parameter estimates

method	cutoff	RhyI	RhyII	RhyBoth	Arrhy
1 Sidak_FS	p-value < 0.05&Agt;0	268	252	12	13964
2 Sidak_FS	q-value < 0.05&Agt;0	0	0	1	14495
3 Sidak_FS	p-value < 0.05&Agt;0.25	165	155	8	14168

Showing 1 to 3 of 3 entries

Search:

Previous  Next

(3)

(1)

(2)

(5)

(6)

Figure 7: Joint Rhythmicity module. A. UI. B. output.

- $M$ : midline-estimating statistic of rhythm (MESOR),
- $A$ : amplitude,
- phase:  $\phi$  in the cosinor model in [cite preprint](#), equivalent to peak time
- pvalue: p-value of rhythmicity
- qvalue: adjusted p-value for multiple comparisons
- sigma:  $\sigma$ , square root of the variance estimate
- $R^2$ : coefficient of determination, which is the proportion of variance explained by the circadian pattern

Separate outputs of the TOJR will be saved with the corresponding parameters (Figure 7(6)). These outputs contain the TOJR assignment to each gene. At the same time, a summary of the numbers of genes in each TOJR category will be displayed in Figure 7(3), where the parameter estimation results can also be found by clicking the “Parameter estimates” tab.

This initializing analysis will create two outputs and summary records by both the chosen p-value cutoff and the FDR cutoff. Users can perform additional TOJR analysis using a different cutoff and method in Figure 7(2). Then new TOJR output and summary will be added.

## 4.2 Parameters

This section introduces what each parameter does in Figure 7(1).

- Only two parameters impact the cosinor estimation:
  - “Period” determines the rhythm period (Period = 24 for circadian rhythm)
  - checking the box for “Output confidence interval (CI)” will add confidence interval estimates for  $A$ ,  $M$  and phase
- The following three parameters set thresholds for determining TOJR:
  - Amplitude cutoff: genes will be filtered by their estimated  $A$ . Genes with  $A$  smaller than the cutoff will be considered arrhythmic no matter the p-value.
  - Critical value: p-value cutoff.
  - FDR: cutoff of p-values adjusted for multiple comparison.
- There are four methods available for joint rhythmicity categorization:
  - Sidak\_FS/Sidak\_BS: selective sequential model selection with Sidak adjusted cutoff. This method sequentially selects the rhythmic groups for each gene and decides the number of rhythmic groups using ForwardStop (Sidak\_FS) or BasicStop (Sidak\_BS) (Fithian et al. (2015)).
  - VDA (Venn diagram analysis): This method compares  $p_1$  and  $p_2$  with the given cutoff separately to get lists of genes that are rhythmic in each group. Then the overlap of the two gene lists are considered RhyBoth.
  - AWFisher (Adaptively weighted Fisher’s method). This method calculates a combined p-value from  $p_1$  and  $p_2$  indicating the existence of rhythm in at least one of the groups. At the same time, weights (0 or 1) are estimated for the two groups to indicate joint rhythmicity (i.e., the weight is (1, 1) for RhyBoth genes and (1, 0) for RhyI genes) (Huo et al. (2020)).
- Number of cores for computation: increasing the number of cores will speed up the computation. This argument only works with MacOS. This same argument will also be used for the “Differential Rhythmicity Analysis” module and will not be explained further.

**A**

Differential Parameter tests

Number of cores for computation: 1

Parameters to test: phase

Critical value: 0.05

DR parameter analysis (button)

Summary of TOJR

method	cutoff	RhyI	RhyII	RhyBoth	Arrhy	DRParameter	DRFitness
1 Sidak_FS	p-value < 0.05&A>0	268	252	12	13964	selected	selected
2 Sidak_FS	q-value < 0.05&A>0	0	0	1	14495	N	N
3 Sidak_FS	p-value < 0.05&A>0.25	165	155	8	14168	N	N

Showing 1 to 3 of 3 entries

Search: [ ]

Previous [ ] Next [ ]

**B**

Differential fitness test

Number of cores for computation: 1

Select method for differential R<sup>2</sup> test: likelihood ratio test (Default)

DR fitness analysis (button)

Summary of TOJR

method	cutoff	RhyI	RhyII	RhyBoth	Arrhy	DRParameter	DRFitness
1 Sidak_FS	p-value < 0.05&A>0	268	252	12	13964	selected	selected
2 Sidak_FS	q-value < 0.05&A>0	0	0	1	14495	N	N
3 Sidak_FS	p-value < 0.05&A>0.25	165	155	8	14168	N	N

Showing 1 to 3 of 3 entries

Search: [ ]

Previous [ ] Next [ ]

**C**

SCZvsControl

Name

- DRfitness\_Sidak\_FS\_p-value\_0.05\_LR.csv (7)
- DRparameter\_Sidak\_FS\_p-value\_0.05\_phase.csv (6)

Figure 8: Module Differential Rhythmicity Analysis. A. UI for differential rhythm fitness test. B. UI for differential rhythm parameter test. C. Example output from this module.

## 5 Differential rhythmicity analysis

This module performs the two differential rhythmicity (DR) tests. The DR fitness test will be performed for genes that are rhythmic in at least one group, and the DR parameter test will be performed for genes that are rhythmic in both groups.

### 5.1 Differential Rhythm Parameter

Firstly, users select one TOJR cutoff they want to use for the analyses from “Summary of TOJR” table (Figure 8(2)). The cutoff selected will be shaded.

Then users select which parameter they want to test in Figure 8(1) “Parameters to test”. The Shiny provides the option of testing the three parameters ( $M$ ,  $A$ , and phase) individually or two two-stage differential parameter tests with global null hypotheses: 1) there is no difference in  $A$  or phase; 2) there is no difference in  $A$ , phase, or  $M$ . If users are only interested in individual parameter change (e.g.  $M$ ,  $A$ , or phase), they should choose the corresponding individual test. However, when more than one parameter is of interest, the

two-stage test should be performed instead of multiple individual tests to avoid multiplicity. The first-stage test conducts a global hypothesis test with multiple parameters, thus controlling the global type I error. If the first-stage null is rejected, the second-stage test performs post hoc individual parameter tests.

After the analysis is finished, the result will be displayed in Figure 8(3) and saved in the output directory (e.g. Figure 8(6)). The exact p-values and q-values will be output for individual tests. For the two-stage tests, the output will contain additional p-values and q-values for the first-stage global test and the post hoc results with the given “Critical value” (Figure 8(1)). For example, a gene has “post.hoc.A.By.p” being “TRUE” means that the first stage global null is rejected by the p-value cutoff and then the individual differential A p-value has rejected the post hoc null.

## 5.2 Differential Rhythm Fitness

Similarly, users must select a TOJR cutoff for the differential fitness test. Three methods are available for the differential fitness test in Figure 8(4) “Select methods for differential  $R^2$  test”. The default likelihood ratio test is recommended for the fastest speed and better statistical performance.

The permutation test and the bootstrap test are both sampling-based. They are both non-parametric, thus might be a better choice when the underlying distribution is non-Gaussian. For sampling-based methods, a large number of samplings is required for better accuracy, which results in a long computing time ( $>1$  hour). As a result, running the analysis with the R package *DiffCircaPipeline* on a device with multiple cores is recommended.

The differential fitness result will be displayed in Figure 8(5) and saved in the output directory (e.g. Figure 8(7)). The “Summary of TOJR” table will also be updated to show if a DR analysis is performed with each TOJR cutoff: “selected” means that the result is now active in the environment for the “Visualization” module; “Y” (Yes) means the analysis is performed and saved but not active; “N” (No) means that there is no DR analysis performed with the cutoff.

# 6 Visualization tools

There are four visualization tools implemented in the Shiny: 1) The scatter plot shows a single gene’s expression pattern across time and fitted cosinor curve. 2) The heatmap plots the gene expression intensity across time for selected genes. 3) The radar plot summarizes the phase distribution of rhythmic genes in each group. 4) Circos phase difference plot visualize the differential phase result.

## 6.1 Scatter Plots

Users have two ways to select what genes to make scatter plots. The first way is to select a gene based on results from previous analyses. These results include parameter estimates from module *Joint Rhythmicity*, and DR parameter and DR fitness result from module *Differential Rhythmicity Analysis* (Figure 9(2)). For example, after users click the “DR fitness result” tab, the active DR fitness result will be displayed (Figure 9(3)). The scatter plots will be produced immediately after clicking a row in the table. Figure 9(5) shows the scatter plot of the gene with most significant difference in  $R^2$ , which is found by sorting the table by “p.R2” (Figure 9(4)).

The second way is to input a self-defined gene list by choosing the “Self-defined gene list” tab in Figure 9(2). Then the UI in Figure 9(6) will appear. Users either input the gene list by typing the gene names in the box, one at a line, or copying and pasting a column of gene names from an excel spreadsheet. Then users need to click the “Plot self-defined gene list” bottom to make the plots.

The first way only allows plotting one gene at a time, while the second allows plotting multiple genes simultaneously. To save the plots to PDF files, users need to input a file name (usually the name of the gene being plotted), and click “Export”. For two-group analysis, the plot width is recommended to be two times of the plot height (default = 4).

DiffCircaPipeline   Welcome   Data Input and Result Output   Joint Rhythmicity   Differential Rhythmicity Analysis   Visualization ▾

### Export to PDF file

file name: Selected Gene

plot width: 14

plot height: 7

**► Export**

(1)

Self-defined gene list

Paste a gene list:

Input a gene list (a gene at a line) or copy and paste from an excel spreadsheet

**► Plot self-defined gene list**

(6)

Select a gene from:

Parameter estimates   DR parameter result   DR fitness result   **Self-defined gene list**

Show 10 entries

gname	delta.R2	p.R2	q.R2
ENSG00000185220	0.256	0	0
ENSG00000204175	-0.307	0.001	0.333
ENSG00000143387	0.244	0.004	0.424
ENSG00000131885	0.244	0.005	0.424
ENSG00000214331	-0.27	0.005	0.424
ENSG00000001714	-0.26	0.006	0.424
ENSG00000147099	0.234	0.006	0.424
ENSG00000120254	-0.25	0.008	0.424
ENSG00000253797	0.226	0.008	0.424
ENSG00000155008	0.223	0.009	0.424

Showing 1 to 10 of 532 entries

Previous   1   2   3   4   ...   54   Next (3)

(2)

The scatter plot

Control ENSG00000185220  
p=0.9026, R<sup>2</sup>= 0.01, A= 0.02, phase= 13.8 sigma= 0.21

SCZ ENSG00000185220  
p=0.0064, R<sup>2</sup>= 0.26, A= 0.18, phase= 22.9 sigma= 0.22

(5)

Figure 9: UI for making scatter plots in module Visualization.

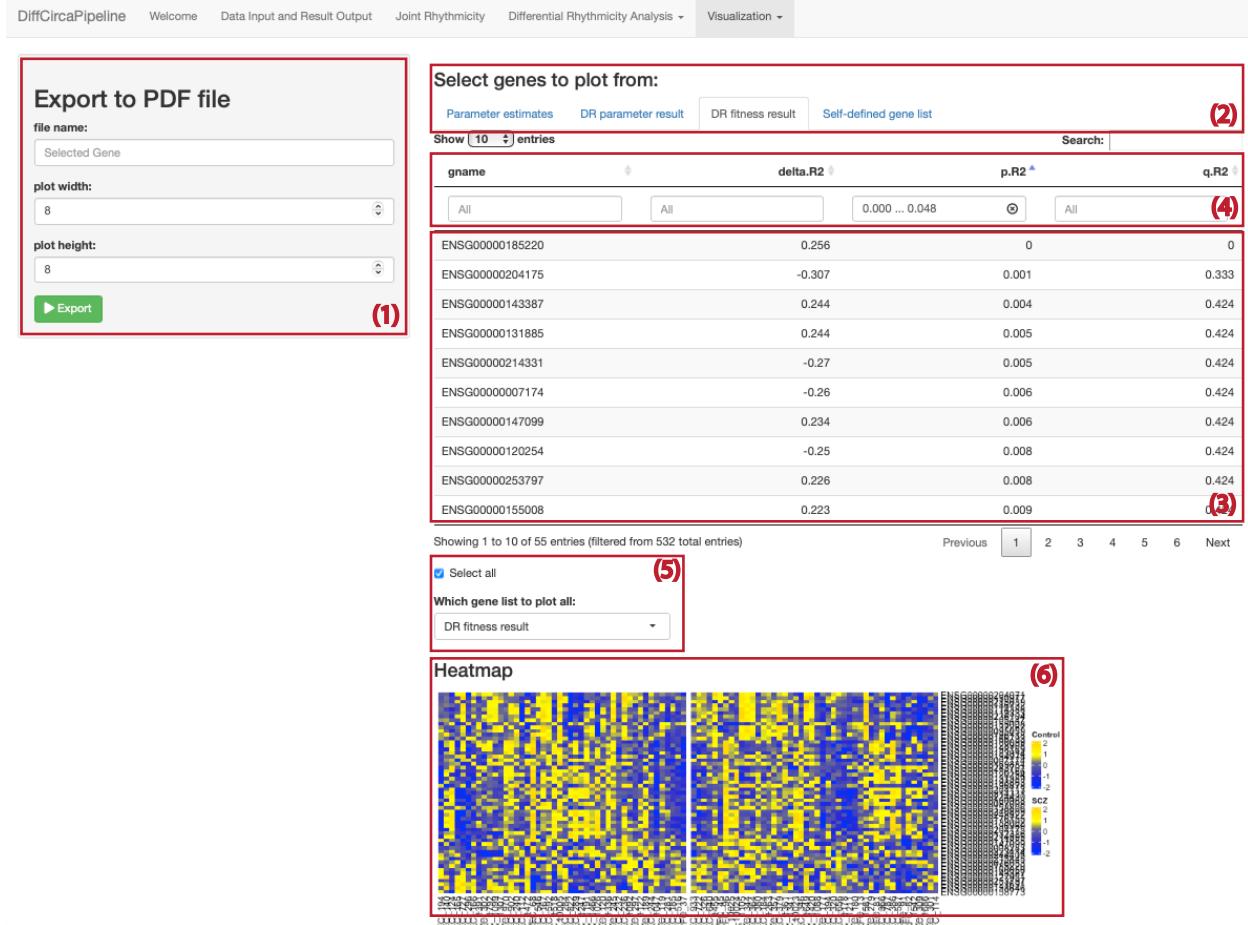


Figure 10: UI for making heatmaps in module Visualization.

## 6.2 Heatmaps

In the heatmaps, each column is a sample and each row is a gene. Columns are ordered by time of the samples and rows are ordered by phase of the genes in group I.

Users need to select genes they want to plot. Similar to Section 6.1, users can either select genes from the results or input a self-defined gene list. Section 6.1 explains how to input a self-defined gene list, thus we will focus on the former approach in this section. After users select what result to use in Figure 10(2), genes can be selected by clicking the corresponding rows in Figure 10(3). However, selecting the genes one by one is inefficient, so we also allow users to plot all the genes shown in the table. First filter genes by adding filtering criteria in Figure 10(4), then check the “Select all” in Figure 10(5) and select the table to be plotted.

Finally, the heatmap can be exported to PDF file in Figure 10(1).

## 6.3 Radar plot

The radar plot shows the phase distribution of rhythmic genes. Two types of plots are available for this purpose: the default “Radar plot” makes a spider web chart (Figure 11(4)), and the “Circos histogram” makes histograms of phase ((5)). The arguments for the two plots are similar, so we only illustrate the “Radar plot” here.

Before making the plot, users need to select a TOJR cutoff from the “Summary of TOJR” table. The rhythmic genes are then identified using the chosen cutoff. By default, all rhythmic genes in each group will be plotted, i.e., RhyI plus RhyBoth genes for group I. Users can choose to plot only the RhyBoth genes by

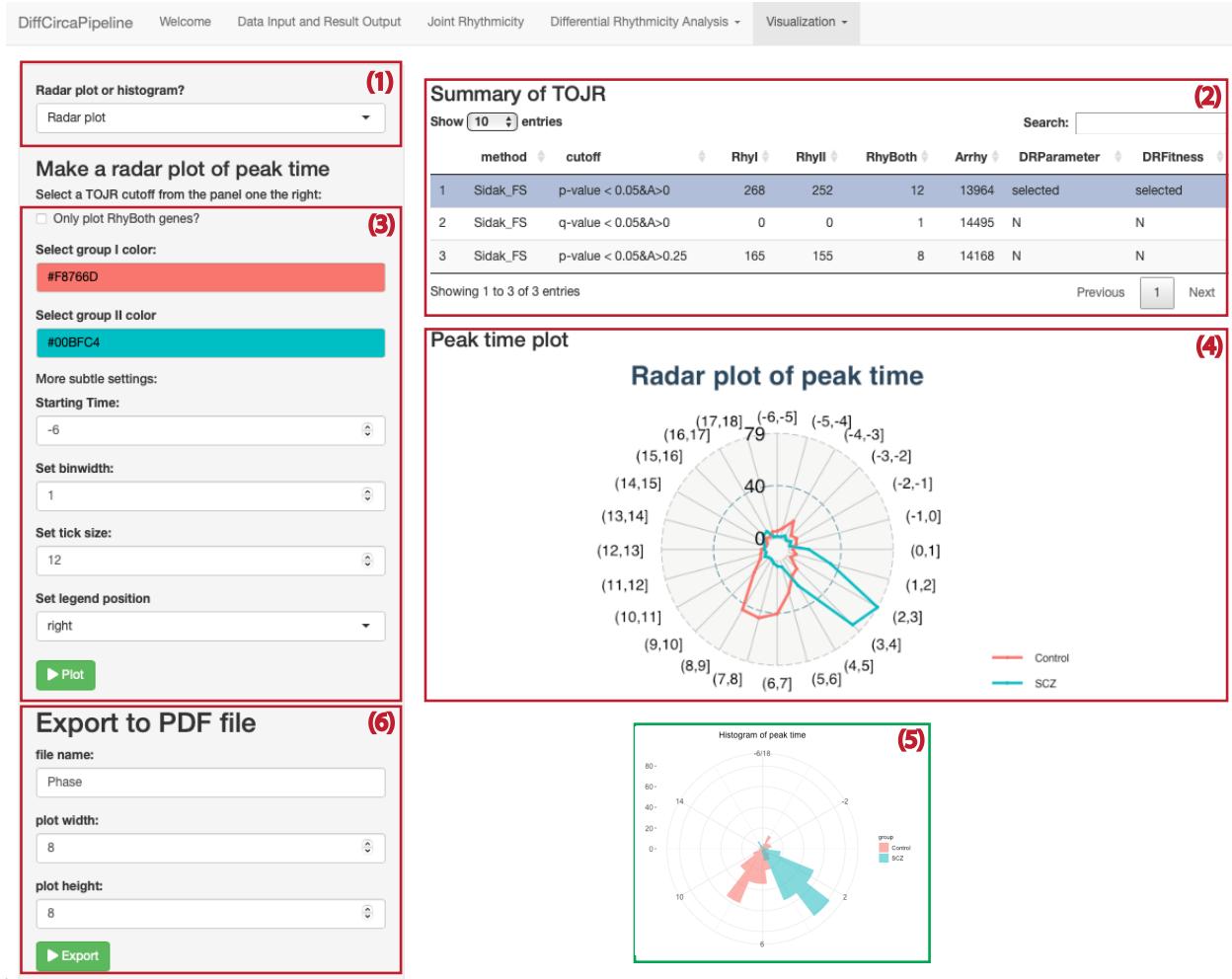


Figure 11: UI for making radar plots in module Visualization.

checking “Only plot RhyBoth genes?” in Figure 11(3). Users can change the color representation for each group by clicking on the colored boxes. More subtle settings include: the starting time (default is -6, which would be clock time 12 am if the input time is zeitgeber time); the binwidth, which controls the precision of the grid; the tick size, which controls the size of the axis annotation; the legend position, where “no legend” will turn off the legend.

## 6.4 Circos phase difference plot

The circos phase difference plot is only available for two-group differential rhythmicity analysis, because we need phase from both groups. First, users select a TOJR cutoff, by which RhyBoth genes will be plotted. Every dot in Figure 12(5) is a RhyBoth gene, the angular axis is the phase in group I and radial axis is the phase difference between group II and group I.

The “Color specification” panel consisting of Figure 12(2) and (3) will appear only if differential phase analysis has been performed, where (3) only appears when (2) is checked. The color specification will annotate the genes with different colors according to the differential phase result. Users select a cutoff to differentiate the genes on the top row of Figure 12(3) and specify colors in the bottom row.

More settings in Figure 12(4) configure the breaks for the axis and the reference lines:

- “Starting time” defines the lower limit of the angular axis;

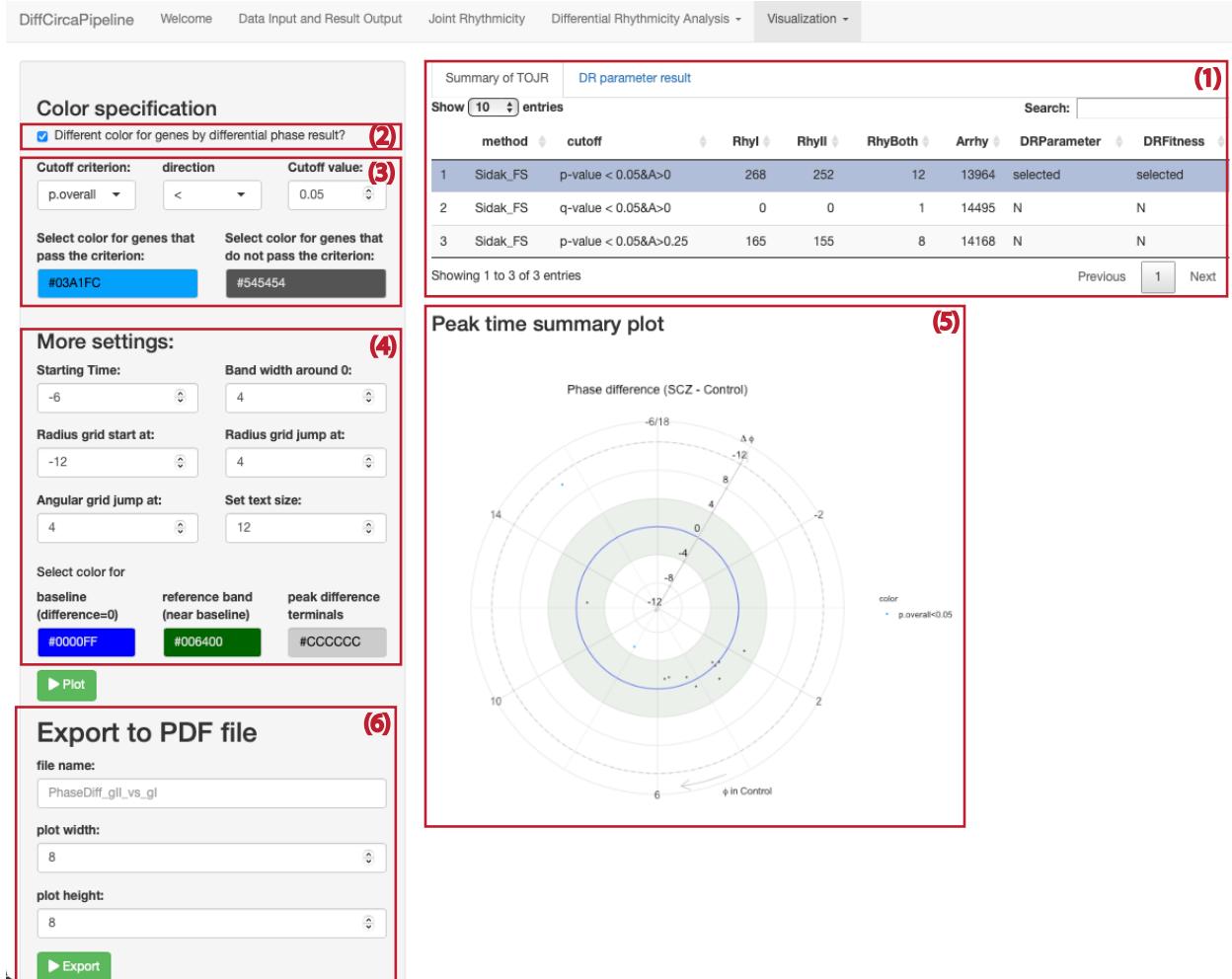


Figure 12: UI for making circos phase difference plots in module Visualization.

- “Band width” defines the width of the reference band, which is the shaded area around the baseline (baseline marks where phase difference = 0);
- “Radius gird start” defines the phase difference for the innermost genes on the circos plot, which has to be a number from  $-\frac{1}{2}$  Period to  $\frac{1}{2}$  Period;
- “Radius grid jump” controls how often the radial grid dumps;
- “Angular grid jump” controls how often the angular grid dumps;
- “Text size” controls size of the text annotations;
- The bottom row controls the color for the baseline, the reference band, and the radial axis limits.

## 7 Single-group analysis

Besides two-group differential rhythmicity analysis, the Shiny also allows the single-group analysis. The functional modules for the single-group analysis are:

- Data Input and Result Output: the use is similar to that of two-group analysis except for that only data from one group is input;

- Joint Rhythmicity: instead of joint rhythmicity analysis, this module fits cosinor model to the single group data. A summary table is also available for number of rhythmic genes by different cutoffs.
- Visualization: scatter plots, heatmap and the radar plots is available.

## References

- Fithian, W., Taylor, J., Tibshirani, R., and Tibshirani, R. (2015). Selective sequential model selection. *arXiv preprint arXiv:1512.02565*.
- Huo, Z., Tang, S., Park, Y., and Tseng, G. (2020). P-value evaluation, variability index and biomarker categorization for adaptively weighted fisher's meta-analysis method in omics applications. *Bioinformatics*, 36(2):524–532.