

Package: RhyME (via r-universe)

September 21, 2026

Title Beta Regression Cosinor Models for Circadian Methylation Analysis

Version 0.99.0

Description Fits cosinor-based beta regression models for detecting circadian rhythmicity in DNA methylation array data using Template Model Builder (TMB) for efficient likelihood optimization. Supports fixed-effects (cross-sectional) and mixed-effects (repeated measures) designs with configurable period and adjustment covariates. Implements empirical Bayes shrinkage of the beta distribution precision parameter across CpG sites, analogous to limma's moderated variance, to improve genome-wide inference. Provides amplitude, phase, and MESOR estimation on both logit-link and response scales.

License GPL (>= 3) + file LICENSE

URL <https://github.com/njuelsdke/RhyME>

BugReports <https://github.com/njuelsdke/RhyME/issues>

Encoding UTF-8

Roxygen list(markdown = TRUE)

biocViews Epigenetics, DNAMethylation, TimeCourse, Regression, DifferentialMethylation, Software

Depends R (>= 4.3.0)

Imports SummarizedExperiment, S4Vectors, BiocParallel, TMB, stats, ggplot2

LinkingTo TMB, RcppEigen

Suggests knitr, rmarkdown, BiocStyle, BiocGenerics, RUnit, limma, cowplot

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

Config/pak/sysreqs zlib1g-dev

Repository <https://biocstaging.r-universe.dev>

Date/Publication 2026-08-21 22:32:49 UTC
RemoteUrl <https://github.com/BiocStaging/RhyME>
RemoteRef HEAD
RemoteSha cbe5289e2a0654fc81e8caeabc24950e14c84b351

Contents

fitCosinor	2
plotCosinorFit	4
topCpGs	5
Index	6

fitCosinor	<i>Fit RhyME across CpG sites</i>
------------	-----------------------------------

Description

Fits a RhyME to each CpG site in a methylation array dataset. Supports both fixed-effects (cross-sectional) and mixed-effects (repeated measures with random intercepts) designs, and optionally applies empirical Bayes shrinkage to the precision parameter (ϕ).

Usage

```
fitCosinor(  
  se,  
  time_col = "time",  
  subject_col = NULL,  
  assay_name = NULL,  
  period = 24,  
  prior = c("auto", "none"),  
  test = c("Wald", "LRT"),  
  BPPARAM = BiocParallel::MulticoreParam(),  
  verbose = TRUE,  
  covariates = NULL  
)
```

Arguments

se	A SummarizedExperiment::SummarizedExperiment object. The specified assay should contain methylation beta values in [0, 1]. Values exactly at the boundaries are moved slightly inward for beta regression.
time_col	Character, column name in colData(se) containing collection times in hours (default "time").

subject_col	Character or NULL. Column name in colData(se) for subject identifiers. If non-NULL, random intercepts per subject are included (mixed-effects mode). If NULL, fixed-effects mode is used.
covariates	Character vector or NULL. Optional colData(se) columns included as adjustment covariates in both the rhythmic and null models. Covariate coefficients are treated as nuisance parameters and are not tested or returned.
assay_name	Character or NULL, name of the assay to use. The default NULL uses the first assay in se.
period	Numeric, the oscillation period in hours (default 24 for circadian rhythms). Can be set to other values for ultradian or infradian rhythms.
prior	Character, either "auto" (default) to estimate a robust empirical-Bayes prior for log(phi), or "none" for ordinary maximum likelihood.
test	Rhythmicity test: a covariance-aware joint Wald test of the sine and cosine coefficients ("Wald", default), or a likelihood-ratio test ("LRT").
BPPARAM	A BiocParallel::BiocParallelParam object for execution. Defaults to BiocParallel::MulticoreParam() for parallel fitting on Unix-like systems. Pass BiocParallel::SerialParam() for serial fitting or BiocParallel::SnowParam() for socket-based parallelism.
verbose	Logical, print messages and show supported progress bars.

Details

When covariates are supplied, their centered design columns are included in both the full and null models. Consequently, the reported p-value still tests only the joint sine/cosine rhythmicity effect, while response-scale summaries describe the mean covariate profile.

Value

The input SummarizedExperiment with per-CpG results appended to rowData(se) and design info, EB prior, test settings, and fit summary stored in metadata(se)\$RhyME.

Examples

```
ex <- readRDS(system.file("extdata", "example_data.rds",
                          package = "RhyME"))
se <- ex$se[seq_len(5), ]

fit <- fitCosinor(se, time_col = "time", subject_col = "subject",
                 prior = "none", verbose = FALSE)
head(SummarizedExperiment::rowData(fit))
```

plotCosinorFit	<i>Plot fitted cosinor curves for selected CpGs</i>
----------------	---

Description

Overlays the fitted cosinor curve on raw beta values for a set of CpGs.

Usage

```
plotCosinorFit(
  se,
  cpgs = NULL,
  assay_name = NULL,
  time_col = NULL,
  ncol = 3,
  n_grid = 200
)
```

Arguments

se	A SummarizedExperiment::SummarizedExperiment object returned by fitCosinor() .
cpgs	Character vector of CpG identifiers to plot. If NULL, the top 6 CpGs by p-value are plotted.
assay_name	Character or NULL. By default, use the assay recorded by fitCosinor() .
time_col	Character or NULL. By default, use the time column recorded by fitCosinor() .
ncol	Integer, number of columns in facet layout (default 3).
n_grid	Integer, resolution of fitted curve grid (default 200).

Value

A ggplot2 object.

Examples

```
ex <- readRDS(system.file("extdata", "example_data.rds",
                          package = "RhyME"))
se <- ex$se[seq_len(5), ]
fit <- fitCosinor(se, time_col = "time", subject_col = "subject",
                 prior = "none", verbose = FALSE)
plotCosinorFit(fit, cpgs = rownames(fit)[1], ncol = 1)
```

topCpGs	<i>Return top CpGs by significance</i>
---------	--

Description

Filters to converged CpGs, selects key columns, and returns the top n sorted by sort_by.

Usage

```
topCpGs(se, n = 10, sort_by = "pvalue")
```

Arguments

se	A SummarizedExperiment::SummarizedExperiment object returned by fitCosinor() .
n	Integer, number of top CpGs to return (default 10).
sort_by	Character, column to sort by (default "pvalue").

Value

A [S4Vectors::DataFrame](#) with the top CpGs.

Examples

```
ex <- readRDS(system.file("extdata", "example_data.rds",  
                           package = "RhyME"))  
se <- ex$se[seq_len(5), ]  
fit <- fitCosinor(se, time_col = "time", subject_col = "subject",  
                  prior = "none", verbose = FALSE)  
topCpGs(fit, n = 3)
```

Index

BiocParallel::BiocParallelParam, [3](#)
BiocParallel::MulticoreParam(), [3](#)
BiocParallel::SerialParam(), [3](#)
BiocParallel::SnowParam(), [3](#)

fitCosinor, [2](#)
fitCosinor(), [4](#), [5](#)

plotCosinorFit, [4](#)

S4Vectors::DataFrame, [5](#)
SummarizedExperiment::SummarizedExperiment,
 [2](#), [4](#), [5](#)

topCpGs, [5](#)