

# Package ‘limorhyde’

October 13, 2022

**Title** Differential Analysis of Rhythmic Transcriptome Data

**Version** 1.0.1

**Description** A flexible approach, inspired by cosinor regression, for differential analysis of rhythmic transcriptome data. See Singer and Hughey (2018) <[doi:10.1177/0748730418813785](https://doi.org/10.1177/0748730418813785)>.

**Depends** R (>= 3.4)

**License** GPL-2

**URL** <https://limorhyde.hugheylab.org>,  
<https://github.com/hugheylab/limorhyde>

**Encoding** UTF-8

**RoxygenNote** 7.1.2

**Imports** pbs (>= 1.1)

**Suggests** annotate (>= 1.58.0), data.table (>= 1.12.2), foreach (>= 1.4.4), ggplot2 (>= 2.2.1), knitr (>= 1.20), limma (>= 3.36.1), matrixStats (>= 0.56.0), org.Mm.eg.db (>= 3.6.0), qs (>= 0.25.2), rmarkdown (>= 1.9), testthat (>= 3.0.4)

**VignetteBuilder** knitr

**BugReports** <https://github.com/hugheylab/limorhyde/issues>

**NeedsCompilation** no

**Author** Jake Hughey [aut, cre],  
Jordan Singer [ctb]

**Maintainer** Jake Hughey <[jakejhughey@gmail.com](mailto:jakejhughey@gmail.com)>

**Repository** CRAN

**Date/Publication** 2022-02-18 08:20:05 UTC

## R topics documented:

getCosinorBasis . . . . .	2
getSplineBasis . . . . .	2
limorhyde . . . . .	3

**Index****5**


---

getCosinorBasis	<i>Basis matrix for cosinor</i>
-----------------	---------------------------------

---

**Description**

Generate basis matrix for cosinor regression.

**Usage**

```
getCosinorBasis(x, period, intercept)
```

**Arguments**

x	Values of the predictor variable.
period	Period for the predictor variable.
intercept	If TRUE, a column of ones will be included in the basis.

**Value**

A matrix with a row for each value of x and a column for each component of the decomposition.

**Examples**

```
b = getCosinorBasis(seq(0, 20, 4), period = 24, intercept = FALSE)
```

---

getSplineBasis	<i>Basis matrix for periodic splines</i>
----------------	--

---

**Description**

Generate basis matrix for a periodic B-spline using `pbs::pbs()`.

**Usage**

```
getSplineBasis(x, period, nKnots, intercept)
```

**Arguments**

x	Values of the predictor variable.
period	Period for the predictor variable.
nKnots	Number of internal knots.
intercept	If TRUE, a column of ones will be included in the basis.

**Value**

A matrix with a row for each value of  $x$  and a column for each component of the decomposition.

**Examples**

```
b = getSplineBasis(seq(0, 20, 4), period = 24, nKnots = 3, intercept = FALSE)
```

---

limorhyde	<i>Convert a periodic time variable into components usable in linear models</i>
-----------	---

---

**Description**

Decompose a periodic time variable into multiple components based on either the first harmonic of a Fourier series or on a periodic smoothing spline.

**Usage**

```
limorhyde(
  time,
  colnamePrefix = NULL,
  period = 24,
  sinusoid = TRUE,
  nKnots = 3,
  intercept = FALSE
)
```

**Arguments**

<code>time</code>	Numeric vector of times, e.g., at which samples were acquired.
<code>colnamePrefix</code>	Character string with which to prefix the column names of the basis.
<code>period</code>	Number corresponding to the period to use for the decomposition (in the same units as <code>time</code> ).
<code>sinusoid</code>	If <code>TRUE</code> , the decomposition is based on <code>cosinor</code> , i.e., cosine and sine. If <code>FALSE</code> , the decomposition is based on a periodic smoothing spline from the <code>pbs</code> package.
<code>nKnots</code>	Number of internal knots for the periodic spline. Only used if <code>sinusoid</code> is <code>FALSE</code> .
<code>intercept</code>	If <code>TRUE</code> , a column of ones will be included in the basis.

**Value**

A matrix with a row for each sample and a column for each component of the time decomposition.

**Examples**

```
# create an example data frame
nSamples = 12
d = data.frame(
  sample = paste0('sample_', 1:nSamples),
  genotype = factor(rep(c('WT', 'KO'), each = nSamples / 2),
                    levels = c('WT', 'KO')),
  zt = rep(seq(0, 24 - 24 / nSamples * 2, 24 / nSamples * 2), times = 2),
  stringsAsFactors = FALSE)

# call limorhyde
limo = limorhyde(d$zt, 'zt_')
d = cbind(d, limo)

# create a design matrix that could be used with methods such as limma
design = model.matrix(~ genotype * (zt_cos + zt_sin), data = d)
```

# Index

`getCosinorBasis`, 2

`getSplineBasis`, 2

`limorhyde`, 3

`pbs::pbs()`, 2