# Package: tipa (via r-universe)

September 12, 2024

**Type** Package

**Title** Tau-Independent Phase Analysis for Circadian Time-Course Data

**Version** 1.0.8

**Description** Accurately estimates phase shifts by accounting for period changes and for the point in the circadian cycle at which the stimulus occurs. See Tackenberg et al. (2018) <doi:10.1177/0748730418768116>.

**URL** https://tipa.hugheylab.org, https://github.com/hugheylab/tipa

**License** GPL-2

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.2.3

**Roxygen** list(markdown = TRUE)

**Depends** R (>= 3.4)

**Imports** optimx (>= 2023-8.13)

**Suggests** knitr, rmarkdown

**Repository** https://hugheylab.r-universe.dev

**RemoteUrl** https://github.com/hugheylab/tipa

**RemoteRef** HEAD

**RemoteSha** 0790af919c919f52f70c42c51f9e0e31dfb941ec

## Contents

`tipaCosinor` *Use cosinor regression to estimate the phase shift induced by a stimulus during a circadian time-course.*

## Description

Calculate the phase shift based on fitting sine curves to waveform data before and after the stimulus, accounting for possible period changes and for the point in the circadian cycle at which the stimulus occurred. This function will work best for measurements whose rhythms are approximately sinusoidal, or at least smoothly increasing and decreasing. If your data are not sinusoidal, you can first define the phase reference points and then use `tipaPhaseRef()`.

## Usage

```
tipaCosinor(
  time,
  y,
  stimOnset,
  stimDuration = 0,
  periodGuess = 24,
  trend = TRUE,
  shortcut = TRUE
)
```

## Arguments

| | |
|---|---|
| `time` | Vector of time values for the full time-course. |
| `y` | Vector of measurements (e.g., bioluminescence) for the full time-course. |
| `stimOnset` | Time at which the stimulus started. |
| `stimDuration` | Duration of the stimulus and any transients. Data between `stimOnset` and `stimOnset` + `stimDuration` will be ignored. |
| `periodGuess` | Approximate period of the oscillations (in the same units used in `time`), used as initial value in fitting the sine curves. |
| `trend` | Model a long-term trend in the cosinor fit for each epoch. Uses a natural cubic spline with 4 degrees of freedom. It is strongly recommended to keep as `TRUE`. If set to `FALSE`, the function may give an error or give completely invalid results. |
| `shortcut` | Calculate phase shift using the standard TIPA procedure or using a shortcut based on the phases of the sine curve fits. The two methods give exactly the same result. |

## Value

A list.

| | |
|---|---|
| `phaseShift` | Estimated phase shift in circadian hours. Negative values correspond to a delay, positive values an advance. |

epochInfo — Dataframe containing information about the sine curve fits for each epoch: period (in the same units used in `time`), phase (in radians), and root mean square error (in the same units as `y`). If the RMS errors pre-stimulus and post-stimulus are substantially different, then the stimulus may have induced a change in the waveform and thus phase shift estimates may be invalid.

## See Also

`tipaPhaseRef()`

## Examples

```
# Time-course data from multiple (simulated) experiments
getTimecourseFile = function() {
  system.file('extdata', 'timecourses.csv', package = 'tipa')}
df = read.csv(getTimecourseFile(), stringsAsFactors = FALSE)

resultList = lapply(sort(unique(df$expId)), function(ii) {
  time = df$time[df$expId == ii]
  y = df$intensity[df$expId == ii]
  tipaCosinor(time, y, stimOnset = 0)})

phaseShifts = sapply(resultList, function(r) r$phaseShift)
```

---

# tipaPhaseRef — *Use phase reference points to estimate the phase shift induced by a stimulus during a circadian time-course.*

---

## Description

Calculate the phase shift based on the times of a phase reference point (e.g., onset of activity), accounting for possible period changes and for the point in the circadian cycle at which the stimulus occurred. If the rhythms of the measurement are approximately sinusoidal, it is recommended to instead use `tipaCosinor()`.

## Usage

```
tipaPhaseRef(phaseRefTimes, stimOnset, stimDuration = 0, period = NULL)
```

## Arguments

phaseRefTimes — Vector of times of the chosen phase reference point.

stimOnset — Time at which the stimulus started.

stimDuration — Duration of the stimulus and any transients. Data between `stimOnset` and `stimOnset + stimDuration` will be ignored.

`period` Optional list with elements "pre" and "post" corresponding to the period of the oscillations prior to and subsequent to the stimulus. If not supplied, the periods for pre- and post-stimulus are calculated as the mean time between occurrences of the phase reference point within the respective epoch. Using this argument is not recommended.

## Value

A list.

`phaseShift` Estimated phase shift in circadian hours. Negative values indicate a delay, positive values an advance.

`epochInfo` `data.frame` containing estimated period for each epoch.

## See Also

`tipaCosinor()`

## Examples

```
# Peak times of bioluminescence (in hours)
phaseRefTimes = c(-75.5, -51.5, -27.4, -3.8,
                  20.5, 42.4, 65.5, 88.0)
result = tipaPhaseRef(phaseRefTimes, stimOnset = 0)

# Data from multiple (simulated) experiments
getExtrFile = function() {
  system.file('extdata', 'phaseRefTimes.csv', package = 'tipa')}
getStimFile = function() {
  system.file('extdata', 'stimOnsets.csv', package = 'tipa')}

extrDf = read.csv(getExtrFile(), stringsAsFactors = FALSE)
stimDf = read.csv(getStimFile(), stringsAsFactors = FALSE)

resultList = lapply(stimDf$expId, function(ii) {
  phaseRefTimes = extrDf$phaseRefTime[extrDf$expId == ii]
  stimOnset = stimDf$stimOnset[stimDf$expId == ii]
  tipaPhaseRef(phaseRefTimes, stimOnset)})

phaseShifts = sapply(resultList, function(r) r$phaseShift)
```

# Index