Package 'OscillatorGenerator'

July 21, 2025

Title Generation of Customizable, Discretized Time Series of Oscillating Species

Description
The supplied code allows for the generation of discrete time series of oscillating species. General shapes can be selected by means of individual functions, which are widely customizable by means of function arguments. All code was developed in the Biological Information Processing Group at the BioQuant Center at Heidelberg University, Germany.

Version 0.1.0

Depends R (>= 3.4.0)

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

NeedsCompilation no

Author Arne Schoch [aut, cre]

Maintainer Arne Schoch <arne_schoch@gmx.net>

Repository CRAN

Date/Publication 2018-05-07 13:47:53 UTC

Contents

ExpBurst *Generation of a Burst Signal with Exponential Rise and Decline*

Description

This function takes in numeric arguments for a customizable, burst shape with exponential rise and decline. Each oscillation cycle is separated into four phases: the growth phase, in which the oscillator rises from the baseline to the peak concentration, a first drop phase, in which the oscillator declines from the peak to the secondary peak concentration, a second drop phase, in which the oscillator declines from the secondary peak to the baseline concentration and an inactive phase, in which the oscillator stays at baseline concentration. A discretized time course is returned.

Usage

```
ExpBurst(baseline, peak, period, duty_cycle, sec_duty_cycle, sec_peak, trend,
  peak_pos, duration, resolution)
```

Arguments

`baseline`	minimal oscillation value
`peak`	maximal oscillation value
`period`	oscillation period of the oscillating species (reciprocal of the frequency)
`duty_cycle`	ratio of the active phase (oscillator above baseline) to the total oscillation period
`sec_duty_cycle`	ratio of the primary active phase (time interval from cycle start till reaching of sec_peak) to the total active phase
`sec_peak`	intermediary value reached after the end of the primary active phase
`trend`	percental decrease or increase in the peak and secondary peak values for the successive oscillation cycles; if set to 1, values remain unchanged
`peak_pos`	position of the peak value in the primary active phase (example: `peak_pos = 0.5`, peak position is in the middle of the primary active phase)
`duration`	duration of the generated time course
`resolution`	temporal resolution of the generated time course

Details

Standards:

- `peak` and `sec_peak` must be larger than baseline
- `duration` must be larger than `resolution`
- `duration` must be a multiple of `resolution`
- `period` must be a multiple of `resolution`
- `duration`, `resolution`, `peak`, `sec_peak` and `period` must be larger than 0
- `baseline` must be larger or equal to 0

- duty_cycle must be larger than 0 and smaller or equal to 1
- sec_duty_cycle must be larger than 0 and smaller or equal to 1
- trend must be larger than 0
- peak_pos must be larger or equal to 0 and smaller than 1

Value

Returns a matrix with two columns: first column time vector, second column oscillator abundance vector.

Examples

```
# test effect of changes in period
m1 = ExpBurst(baseline = 200, peak = 1000, period = 10, duty_cycle = 0.6,
sec_duty_cycle = 0.5, sec_peak = 850, trend = 1, peak_pos = 0.3, duration = 500, resolution = 0.1)
m2 = ExpBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
sec_duty_cycle = 0.5, sec_peak = 850, trend = 1, peak_pos = 0.3, duration = 500, resolution = 0.1)
m3 = ExpBurst(baseline = 200, peak = 1000, period = 200, duty_cycle = 0.6,
sec_duty_cycle = 0.5, sec_peak = 850, trend = 1, peak_pos = 0.3, duration = 500, resolution = 0.1)

par(mfrow = c(3,1))
plot(m1, type = "l", xlab = "time", ylab = "abundance")
plot(m2, type = "l", xlab = "time", ylab = "abundance")
plot(m3, type = "l", xlab = "time", ylab = "abundance")

# test effect of changes in duty_cycle
m1 = ExpBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.3,
sec_duty_cycle = 0.5, sec_peak = 850, trend = 1, peak_pos = 0.3, duration = 500, resolution = 0.1)
m2 = ExpBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
sec_duty_cycle = 0.5, sec_peak = 850, trend = 1, peak_pos = 0.3, duration = 500, resolution = 0.1)
m3 = ExpBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.9,
sec_duty_cycle = 0.5, sec_peak = 850, trend = 1, peak_pos = 0.3, duration = 500, resolution = 0.1)
par(mfrow = c(3,1))
plot(m1, type = "l", xlab = "time", ylab = "abundance")
plot(m2, type = "l", xlab = "time", ylab = "abundance")
plot(m3, type = "l", xlab = "time", ylab = "abundance")

# test effect of changes in sec_duty_cycle
m1 = ExpBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
sec_duty_cycle = 0.3, sec_peak = 850, trend = 1, peak_pos = 0.3, duration = 500, resolution = 0.1)
m2 = ExpBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
sec_duty_cycle = 0.6, sec_peak = 850, trend = 1, peak_pos = 0.3, duration = 500, resolution = 0.1)
m3 = ExpBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
sec_duty_cycle = 0.9, sec_peak = 850, trend = 1, peak_pos = 0.3, duration = 500, resolution = 0.1)

par(mfrow = c(3,1))
plot(m1, type = "l", xlab = "time", ylab = "abundance")
plot(m2, type = "l", xlab = "time", ylab = "abundance")
plot(m3, type = "l", xlab = "time", ylab = "abundance")

# test effect of changes in trend
```

```
m1 = ExpBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
sec_duty_cycle = 0.5, sec_peak = 850, trend = 0.7, peak_pos = 0.3, duration = 500, resolution = 0.1)
m2 = ExpBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
sec_duty_cycle = 0.5, sec_peak = 850, trend = 1, peak_pos = 0.3, duration = 500, resolution = 0.1)
m3 = ExpBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
sec_duty_cycle = 0.5, sec_peak = 850, trend = 1.3, peak_pos = 0.3, duration = 500, resolution = 0.1)

par(mfrow = c(3,1))
plot(m1, type = "l", xlab = "time", ylab = "abundance")
plot(m2, type = "l", xlab = "time", ylab = "abundance")
plot(m3, type = "l", xlab = "time", ylab = "abundance")

# test effect of changes in peak_pos
m1 = ExpBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
sec_duty_cycle = 0.5, sec_peak = 850, trend = 1, peak_pos = 0.3, duration = 500, resolution = 0.1)
m2 = ExpBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
sec_duty_cycle = 0.5, sec_peak = 850, trend = 1, peak_pos = 0.6, duration = 500, resolution = 0.1)
m3 = ExpBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
sec_duty_cycle = 0.5, sec_peak = 850, trend = 1, peak_pos = 0.9, duration = 500, resolution = 0.1)

par(mfrow = c(3,1))
plot(m1, type = "l", xlab = "time", ylab = "abundance")
plot(m2, type = "l", xlab = "time", ylab = "abundance")
plot(m3, type = "l", xlab = "time", ylab = "abundance")
```

`ExpSpike` *Generation of a Spike Signal with Exponential Rise and Decline*

Description

This function takes in numeric arguments for a customizable, spike shape, in which rise and decline are modelled by means of an exponential function. A discretized time course is returned.

Usage

```
ExpSpike(baseline, peak, period, duty_cycle, peak_pos, trend, duration,
  resolution)
```

Arguments

`baseline`	minimal oscillation value
`peak`	maximal oscillation value
`period`	oscillation period of the oscillating species (reciprocal of the frequency)
`duty_cycle`	ratio of the active phase (oscillator above baseline) to the total oscillation period
`peak_pos`	position of the peak value in the active phase of an oscillation cycle (example: `peak_pos = 0.5`, peak position is in the middle of the active phase)
`trend`	percental decrease or increase in the peak value for the successive oscillation cycles; if set to 1, peak value remains unchanged

duration	duration of the generated time course
resolution	temporal resolution of the generated time course

Details

Standards:

- peak must be larger than baseline
- duration must be larger than resolution
- duration must be a multiple of resolution
- period must be a multiple of resolution
- duration, resolution, peak and period must be larger than 0
- baseline must be larger or equal to 0
- duty_cycle must be larger than 0 and smaller or equal to 1
- trend must be larger than 0
- peak_pos must be larger or equal to 0 and smaller than 1

Value

Returns a matrix with two columns: a time vector and an oscillator abundance vector.

Examples

```
# test effect of changes in period
m1 = ExpSpike(baseline = 200, peak = 1000, period = 50, duty_cycle = 0.6,
peak_pos = 0.3, trend = 1, duration = 500, resolution = 0.1)
m2 = ExpSpike(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
peak_pos = 0.3, trend = 1, duration = 500, resolution = 0.1)
m3 = ExpSpike(baseline = 200, peak = 1000, period = 200, duty_cycle = 0.6,
peak_pos = 0.3, trend = 1, duration = 500, resolution = 0.1)

par(mfrow = c(3,1))
plot(m1, type = "l", xlab = "time", ylab = "abundance")
plot(m2, type = "l", xlab = "time", ylab = "abundance")
plot(m3, type = "l", xlab = "time", ylab = "abundance")

# test effect of changes in duty_cycle
m1 = ExpSpike(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.3,
peak_pos = 0.3, trend = 1, duration = 500, resolution = 0.1)
m2 = ExpSpike(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
peak_pos = 0.3, trend = 1, duration = 500, resolution = 0.1)
m3 = ExpSpike(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.9,
peak_pos = 0.3, trend = 1, duration = 500, resolution = 0.1)

par(mfrow = c(3,1))
plot(m1, type = "l", xlab = "time", ylab = "abundance")
plot(m2, type = "l", xlab = "time", ylab = "abundance")
plot(m3, type = "l", xlab = "time", ylab = "abundance")
```

```
# test effect of changes in peak_pos
m1 = ExpSpike(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
peak_pos = 0.3, trend = 1, duration = 500, resolution = 0.1)
m2 = ExpSpike(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
peak_pos = 0.6, trend = 1, duration = 500, resolution = 0.1)
m3 = ExpSpike(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
peak_pos = 0.9, trend = 1, duration = 500, resolution = 0.1)

par(mfrow = c(3,1))
plot(m1, type = "l", xlab = "time", ylab = "abundance")
plot(m2, type = "l", xlab = "time", ylab = "abundance")
plot(m3, type = "l", xlab = "time", ylab = "abundance")

# test effect of changes in trend
m1 = ExpSpike(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
peak_pos = 0.3, trend = 0.7, duration = 500, resolution = 0.1)
m2 = ExpSpike(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
peak_pos = 0.3, trend = 1, duration = 500, resolution = 0.1)
m3 = ExpSpike(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
peak_pos = 0.3, trend = 1.3, duration = 500, resolution = 0.1)

par(mfrow = c(3,1))
plot(m1, type = "l", xlab = "time", ylab = "abundance")
plot(m2, type = "l", xlab = "time", ylab = "abundance")
plot(m3, type = "l", xlab = "time", ylab = "abundance")
```

`LinBurst` *Generation of a Burst Signal with Linear Rise and Decline*

Description

This function takes in numeric arguments for a customizable, burst shape with linear rise and decline. Each oscillation cycle is separated into four phases: the growth phase, in which the oscillator rises from the baseline to the peak concentration, a first drop phase, in which the oscillator declines from the peak to the secondary peak concentration, a second drop phase, in which the oscillator declines from the secondary peak to the baseline concentration and an inactive phase, in which the oscillator stays at baseline concentration. A discretized time course is returned.

Usage

```
LinBurst(baseline, peak, period, duty_cycle, sec_duty_cycle, sec_peak, trend,
  peak_pos, duration, resolution)
```

Arguments

`baseline`	minimal oscillation value
`peak`	maximal oscillation value
`period`	oscillation period of the oscillating species (reciprocal of the frequency)

duty_cycle	ratio of the active phase (oscillator above baseline) to the total oscillation period
sec_duty_cycle	ratio of the primary active phase (time interval from cycle start till reaching of the secondary peak) to the total active phase
sec_peak	intermediary value reached after the end of the primary active phase
trend	percental decrease or increase in the peak and secondary peak values for the successive oscillation cycles; if set to 1, values remain unchanged
peak_pos	position of the peak value in the primary active phase (example: peak_pos = 0.5, peak position is in the middle of the primary active phase)
duration	duration of the generated time course
resolution	temporal resolution of the generated time course

Details

Standards:

- peak and sec_peak must be larger than baseline
- duration must be larger than resolution
- duration must be a multiple of resolution
- period must be a multiple of resolution
- duration, resolution, peak, sec_peak and period must be larger than 0
- baseline must be larger or equal to 0
- duty_cycle must be larger than 0 and smaller or equal to 1
- sec_duty_cycle must be larger than 0 and smaller or equal to 1
- trend must be larger than 0
- peak_pos must be larger or equal to 0 and smaller than 1

Value

Returns a matrix with two columns: first column time vector, second column oscillator abundance vector.

Examples

```
# test effect of changes in period
m1 = LinBurst(baseline = 200, peak = 1000, period = 50, duty_cycle = 0.6,
sec_duty_cycle = 0.5, sec_peak = 850, trend = 1, peak_pos = 0.3, duration = 500, resolution = 0.1)
m2 = LinBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
sec_duty_cycle = 0.5, sec_peak = 850, trend = 1, peak_pos = 0.3, duration = 500, resolution = 0.1)
m3 = LinBurst(baseline = 200, peak = 1000, period = 200, duty_cycle = 0.6,
sec_duty_cycle = 0.5, sec_peak = 850, trend = 1, peak_pos = 0.3, duration = 500, resolution = 0.1)

par(mfrow = c(3,1))
plot(m1, type = "l", xlab = "time", ylab = "abundance")
plot(m2, type = "l", xlab = "time", ylab = "abundance")
plot(m3, type = "l", xlab = "time", ylab = "abundance")
```

```
# test effect of changes in duty_cycle
m1 = LinBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.3,
sec_duty_cycle = 0.5, sec_peak = 850, trend = 1, peak_pos = 0.3, duration = 500, resolution = 0.1)
m2 = LinBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
sec_duty_cycle = 0.5, sec_peak = 850, trend = 1, peak_pos = 0.3, duration = 500, resolution = 0.1)
m3 = LinBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.9,
sec_duty_cycle = 0.5, sec_peak = 850, trend = 1, peak_pos = 0.3, duration = 500, resolution = 0.1)

par(mfrow = c(3,1))
plot(m1, type = "l", xlab = "time", ylab = "abundance")
plot(m2, type = "l", xlab = "time", ylab = "abundance")
plot(m3, type = "l", xlab = "time", ylab = "abundance")

# test effect of changes in sec_duty_cycle
m1 = LinBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
sec_duty_cycle = 0.3, sec_peak = 850, trend = 1, peak_pos = 0.3, duration = 500, resolution = 0.1)
m2 = LinBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
sec_duty_cycle = 0.6, sec_peak = 850, trend = 1, peak_pos = 0.3, duration = 500, resolution = 0.1)
m3 = LinBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
sec_duty_cycle = 0.9, sec_peak = 850, trend = 1, peak_pos = 0.3, duration = 500, resolution = 0.1)

par(mfrow = c(3,1))
plot(m1, type = "l", xlab = "time", ylab = "abundance")
plot(m2, type = "l", xlab = "time", ylab = "abundance")
plot(m3, type = "l", xlab = "time", ylab = "abundance")

# test effect of changes in trend
m1 = LinBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
sec_duty_cycle = 0.5, sec_peak = 850, trend = 0.7, peak_pos = 0.3, duration = 500, resolution = 0.1)
m2 = LinBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
sec_duty_cycle = 0.5, sec_peak = 850, trend = 1, peak_pos = 0.3, duration = 500, resolution = 0.1)
m3 = LinBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
sec_duty_cycle = 0.5, sec_peak = 850, trend = 1.3, peak_pos = 0.3, duration = 500, resolution = 0.1)

par(mfrow = c(3,1))
plot(m1, type = "l", xlab = "time", ylab = "abundance")
plot(m2, type = "l", xlab = "time", ylab = "abundance")
plot(m3, type = "l", xlab = "time", ylab = "abundance")

# test effect of changes in peak_pos
m1 = LinBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
sec_duty_cycle = 0.5, sec_peak = 850, trend = 1, peak_pos = 0.3, duration = 500, resolution = 0.1)
m2 = LinBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
sec_duty_cycle = 0.5, sec_peak = 850, trend = 1, peak_pos = 0.6, duration = 500, resolution = 0.1)
m3 = LinBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
sec_duty_cycle = 0.5, sec_peak = 850, trend = 1, peak_pos = 0.9, duration = 500, resolution = 0.1)

par(mfrow = c(3,1))
plot(m1, type = "l", xlab = "time", ylab = "abundance")
plot(m2, type = "l", xlab = "time", ylab = "abundance")
plot(m3, type = "l", xlab = "time", ylab = "abundance")
```

LinSpike *Generation of a Spike Signal with Linear Rise and Decline*

Description

This function takes in numeric arguments for a customizable, spike shape, in which rise and decline are modelled by means of a linear function. A discretized time course is returned.

Usage

```
LinSpike(baseline, peak, period, duty_cycle, peak_pos, trend, duration,
  resolution)
```

Arguments

`baseline`	minimal oscillation value
`peak`	maximal oscillation value
`period`	oscillation period of the oscillating species (reciprocal of the frequency)
`duty_cycle`	ratio of the active phase (oscillator above baseline) to the total oscillation period
`peak_pos`	position of the peak value in the active phase of an oscillation cycle (example: `peak_pos = 0.5`, peak position is in the middle of the active phase)
`trend`	percental decrease or increase in the peak value for the successive oscillation cycles; if set to 1, peak value remains unchanged
`duration`	duration of the generated time course
`resolution`	temporal resolution of the generated time course

Details

Standards:

- `peak` must be larger than `baseline`
- `duration` must be larger than `resolution`
- `duration` must be a multiple of `resolution`
- `period` must be a multiple of `resolution`
- `duration`, `resolution`, `peak` and `period` must be larger than 0
- `baseline` must be larger or equal to 0
- `duty_cycle` must be larger than 0 and smaller or equal to 1
- `trend` must be larger than 0
- `peak_pos` must be larger or equal to 0 and smaller than 1

Value

Returns a matrix with two columns: a time vector and an oscillator abundance vector.

Examples

```
# test effect of changes in period
m1 = LinSpike(baseline = 200, peak = 1000, period = 50, duty_cycle = 0.6,
peak_pos = 0.3, trend = 1, duration = 500, resolution = 0.1)
m2 = LinSpike(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
peak_pos = 0.3, trend = 1, duration = 500, resolution = 0.1)
m3 = LinSpike(baseline = 200, peak = 1000, period = 200, duty_cycle = 0.6,
peak_pos = 0.3, trend = 1, duration = 500, resolution = 0.1)

par(mfrow = c(3,1))
plot(m1, type = "l", xlab = "time", ylab = "abundance")
plot(m2, type = "l", xlab = "time", ylab = "abundance")
plot(m3, type = "l", xlab = "time", ylab = "abundance")

# test effect of changes in duty_cycle
m1 = LinSpike(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.3,
peak_pos = 0.3, trend = 1, duration = 500, resolution = 0.1)
m2 = LinSpike(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
peak_pos = 0.3, trend = 1, duration = 500, resolution = 0.1)
m3 = LinSpike(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.9,
peak_pos = 0.3, trend = 1, duration = 500, resolution = 0.1)

par(mfrow = c(3,1))
plot(m1, type = "l", xlab = "time", ylab = "abundance")
plot(m2, type = "l", xlab = "time", ylab = "abundance")
plot(m3, type = "l", xlab = "time", ylab = "abundance")

# test effect of changes in peak_pos
m1 = LinSpike(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
peak_pos = 0.3, trend = 1, duration = 500, resolution = 0.1)
m2 = LinSpike(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
peak_pos = 0.6, trend = 1, duration = 500, resolution = 0.1)
m3 = LinSpike(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
peak_pos = 0.9, trend = 1, duration = 500, resolution = 0.1)

par(mfrow = c(3,1))
plot(m1, type = "l", xlab = "time", ylab = "abundance")
plot(m2, type = "l", xlab = "time", ylab = "abundance")
plot(m3, type = "l", xlab = "time", ylab = "abundance")

# test effect of changes in trend
m1 = LinSpike(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
peak_pos = 0.3, trend = 0.7, duration = 500, resolution = 0.1)
m2 = LinSpike(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
peak_pos = 0.3, trend = 1, duration = 500, resolution = 0.1)
m3 = LinSpike(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
peak_pos = 0.3, trend = 1.3, duration = 500, resolution = 0.1)

par(mfrow = c(3,1))
plot(m1, type = "l", xlab = "time", ylab = "abundance")
plot(m2, type = "l", xlab = "time", ylab = "abundance")
plot(m3, type = "l", xlab = "time", ylab = "abundance")
```

`Sinusoid` *Generation of a Sinusoidal Signal*

Description

This function takes in numeric arguments for a customizable, sinusoidal shape. A discretized time course is returned.

Usage

```
Sinusoid(baseline, peak, period, duty_cycle, trend, duration, resolution)
```

Arguments

`baseline`	minimal oscillation value
`peak`	maximal oscillation value
`period`	oscillation period of the oscillator (reciprocal of the frequency)
`duty_cycle`	ratio of the active phase (oscillator above baseline) to the total oscillation period
`trend`	percental decrease or increase in the peak value for the successive oscillation cycles; if set to 1, peak value remains unchanged
`duration`	duration of the generated time course
`resolution`	temporal resolution of the generated time course

Details

Standards:

- `peak` must be larger than `baseline` value
- `duration` must be larger than `resolution`
- `duration` must be a multiple of `resolution`
- `period` must be a multiple of `resolution`
- `duration`, `resolution`, `peak` and `period` must be larger than 0
- `baseline` must be larger or equal to 0
- `duty_cycle` must be larger than 0 and smaller or equal to 1
- `trend` must be larger than 0

Value

Returns a matrix with two columns: a time vector and an oscillator abundance vector.

Examples

```
# test effect of changes in period
m1 = Sinusoid(baseline = 200, peak = 1000, period = 50, duty_cycle = 0.6,
trend = 1, duration = 500, resolution = 0.1)
m2 = Sinusoid(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
trend = 1, duration = 500, resolution = 0.1)
m3 = Sinusoid(baseline = 200, peak = 1000, period = 200, duty_cycle = 0.6,
trend = 1, duration = 500, resolution = 0.1)

par(mfrow = c(3,1))
plot(m1, type = "l", xlab = "time", ylab = "abundance")
plot(m2, type = "l", xlab = "time", ylab = "abundance")
plot(m3, type = "l", xlab = "time", ylab = "abundance")

# test effect of changes in duty_cycle
m1 = Sinusoid(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.3,
trend = 1, duration = 500, resolution = 0.1)
m2 = Sinusoid(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
trend = 1, duration = 500, resolution = 0.1)
m3 = Sinusoid(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.9,
trend = 1, duration = 500, resolution = 0.1)

par(mfrow = c(3,1))
plot(m1, type = "l", xlab = "time", ylab = "abundance")
plot(m2, type = "l", xlab = "time", ylab = "abundance")
plot(m3, type = "l", xlab = "time", ylab = "abundance")

# test effect of changes in trend
m1 = Sinusoid(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
trend = 0.7, duration = 500, resolution = 0.1)
m2 = Sinusoid(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
trend = 1, duration = 500, resolution = 0.1)
m3 = Sinusoid(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
trend = 1.3, duration = 500, resolution = 0.1)

par(mfrow = c(3,1))
plot(m1, type = "l", xlab = "time", ylab = "abundance")
plot(m2, type = "l", xlab = "time", ylab = "abundance")
plot(m3, type = "l", xlab = "time", ylab = "abundance")
```

SquareBurst *Generation of a Square-wave Burst Signal*

Description

This function takes in numeric arguments for a customizable, square-wave burst shape. Each oscillation cycle is separated into three phases: a primary active phase, in which the oscillator resides at peak concentration, a secondary active phase, in which the oscillator stays at secondary peak concentration and an inactive phase, in which the oscillator is fixed to baseline concentration. A discretized time course is returned.

Usage

```
SquareBurst(baseline, peak, period, duty_cycle, sec_duty_cycle, sec_peak, trend,
  duration, resolution)
```

Arguments

baseline	minimal oscillation value
peak	maximal oscillation value
period	oscillation period of the oscillating species (reciprocal of the frequency)
duty_cycle	ratio of the active phase (oscillator above baseline) to the total oscillation period
sec_duty_cycle	ratio of the primary active phase (time interval from cycle start till reaching of the secondary peak level) to the total active phase
sec_peak	intermediary value reached after the end of the primary active phase
trend	percental decrease or increase in the peak and secondary peak values for the successive oscillation cycles; if set to 1, values remain unchanged
duration	duration of the generated time course
resolution	temporal resolution of the generated time course

Details

Standards:

- peak and sec_peak must be larger than baseline
- duration must be larger than resolution
- duration must be a multiple of the resolution
- period must be a multiple of resolution
- duration, resolution, peak, sec_peak and period must be larger than 0
- baseline must be larger or equal to 0
- duty_cycle must be larger than 0 and smaller or equal to 1
- sec_duty_cycle must be larger than 0 and smaller or equal to 1
- trend must be larger than 0

Value

Returns a matrix with two columns: a time vector and an oscillator abundance vector.

Examples

```
# test effect of changes in period
m1 = SquareBurst(baseline = 200, peak = 1000, period = 50, duty_cycle = 0.6,
sec_duty_cycle = 0.5, sec_peak = 700, trend = 1, duration = 500, resolution = 0.1)
m2 = SquareBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
sec_duty_cycle = 0.5, sec_peak = 700, trend = 1, duration = 500, resolution = 0.1)
m3 = SquareBurst(baseline = 200, peak = 1000, period = 200, duty_cycle = 0.6,
sec_duty_cycle = 0.5, sec_peak = 700, trend = 1, duration = 500, resolution = 0.1)
```

```
par(mfrow = c(3,1))
plot(m1, type = "l", xlab = "time", ylab = "abundance")
plot(m2, type = "l", xlab = "time", ylab = "abundance")
plot(m3, type = "l", xlab = "time", ylab = "abundance")

# test effect of changes in duty_cycle
m1 = SquareBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.3,
sec_duty_cycle = 0.5, sec_peak = 700, trend = 1, duration = 500, resolution = 0.1)
m2 = SquareBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
sec_duty_cycle = 0.5, sec_peak = 700, trend = 1, duration = 500, resolution = 0.1)
m3 = SquareBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.9,
sec_duty_cycle = 0.5, sec_peak = 700, trend = 1, duration = 500, resolution = 0.1)

par(mfrow = c(3,1))
plot(m1, type = "l", xlab = "time", ylab = "abundance")
plot(m2, type = "l", xlab = "time", ylab = "abundance")
plot(m3, type = "l", xlab = "time", ylab = "abundance")

# test effect of changes in sec_duty_cycle
m1 = SquareBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
sec_duty_cycle = 0.3, sec_peak = 700, trend = 1, duration = 500, resolution = 0.1)
m2 = SquareBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
sec_duty_cycle = 0.6, sec_peak = 700, trend = 1, duration = 500, resolution = 0.1)
m3 = SquareBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
sec_duty_cycle = 0.9, sec_peak = 700, trend = 1, duration = 500, resolution = 0.1)

par(mfrow = c(3,1))
plot(m1, type = "l", xlab = "time", ylab = "abundance")
plot(m2, type = "l", xlab = "time", ylab = "abundance")
plot(m3, type = "l", xlab = "time", ylab = "abundance")

# test effect of changes in trend
m1 = SquareBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
sec_duty_cycle = 0.6, sec_peak = 700, trend = 0.7, duration = 500, resolution = 0.1)
m2 = SquareBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
sec_duty_cycle = 0.6, sec_peak = 700, trend = 1, duration = 500, resolution = 0.1)
m3 = SquareBurst(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
sec_duty_cycle = 0.6, sec_peak = 700, trend = 1.3, duration = 500, resolution = 0.1)

par(mfrow = c(3,1))
plot(m1, type = "l", xlab = "time", ylab = "abundance")
plot(m2, type = "l", xlab = "time", ylab = "abundance")
plot(m3, type = "l", xlab = "time", ylab = "abundance")
```

SquareSpike *Generation of a Square-wave Signal*

Description

This function takes in numeric arguments for a customizable, square-wave spike shape. A discretized time course is returned.

Usage

```
SquareSpike(baseline, peak, period, duty_cycle, trend, duration, resolution)
```

Arguments

baseline	minimal oscillation value
peak	maximal oscillation value
period	oscillation period of the oscillating species (reciprocal of the frequency)
duty_cycle	ratio of the active phase (oscillator above baseline) to the total oscillation period
trend	percental decrease or increase in the peak value for the successive oscillation cycles; if set to 1, peak value remains unchanged
duration	duration of the generated time course
resolution	temporal resolution of the generated time course

Details

Standards:

- peak must be larger than baseline
- duration must be larger than resolution
- duration must be a multiple of resolution
- period must be a multiple of resolution
- duration, resolution, peak and period must be larger than 0
- baseline must be larger or equal to 0
- duty_cycle must be larger than 0 and smaller or equal to 1
- trend must be larger than 0

Value

Returns a matrix with two columns: a time vector and an oscillator abundance vector.

Examples

```
# test effect of changes in period
m1 = SquareSpike(baseline = 200, peak = 1000, period = 50, duty_cycle = 0.6,
trend = 1, duration = 500, resolution = 0.1)
m2 = SquareSpike(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
trend = 1, duration = 500, resolution = 0.1)
m3 = SquareSpike(baseline = 200, peak = 1000, period = 200, duty_cycle = 0.6,
trend = 1, duration = 500, resolution = 0.1)
```

```
par(mfrow = c(3,1))
plot(m1, type = "l", xlab = "time", ylab = "abundance")
plot(m2, type = "l", xlab = "time", ylab = "abundance")
plot(m3, type = "l", xlab = "time", ylab = "abundance")

# test effect of changes in duty_cycle
m1 = SquareSpike(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.3,
trend = 1, duration = 500, resolution = 0.1)
m2 = SquareSpike(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
trend = 1, duration = 500, resolution = 0.1)
m3 = SquareSpike(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.9,
trend = 1, duration = 500, resolution = 0.1)

par(mfrow = c(3,1))
plot(m1, type = "l", xlab = "time", ylab = "abundance")
plot(m2, type = "l", xlab = "time", ylab = "abundance")
plot(m3, type = "l", xlab = "time", ylab = "abundance")

# test effect of changes in trend
m1 = SquareSpike(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
trend = 0.7, duration = 500, resolution = 0.1)
m2 = SquareSpike(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
trend = 1, duration = 500, resolution = 0.1)
m3 = SquareSpike(baseline = 200, peak = 1000, period = 100, duty_cycle = 0.6,
trend = 1.3, duration = 500, resolution = 0.1)

par(mfrow = c(3,1))
plot(m1, type = "l", xlab = "time", ylab = "abundance")
plot(m2, type = "l", xlab = "time", ylab = "abundance")
plot(m3, type = "l", xlab = "time", ylab = "abundance")
```

Index