

# Package ‘TimerQuant’

May 23, 2024

**Type** Package

**Title** Timer Quantification

**Version** 1.35.0

**Date** 2015-05-22

**Author** Joseph Barry

**Maintainer** Joseph Barry <joseph.barry@embl.de>

**Depends** shiny

**Suggests** BiocStyle, reshape2, knitr, shinyBS

**Imports** ggplot2, grid, gridExtra, deSolve, dplyr, locfit

**VignetteBuilder** knitr

**Description** Supplementary Data package for tandem timer methods paper by Barry et al. (2015) including TimerQuant shiny applications.

**biocViews** ExperimentData, Danio\_rerio\_Data, HighThroughputImagingData, Tissue

**License** Artistic-2.0

**LazyLoad** yes

**NeedsCompilation** no

**git\_url** <https://git.bioconductor.org/packages/TimerQuant>

**git\_branch** devel

**git\_last\_commit** bfc85cf

**git\_last\_commit\_date** 2024-04-30

**Repository** Bioconductor 3.20

**Date/Publication** 2024-05-23

## Contents

analyticSolutions . . . . .	2
fitCV . . . . .	3
FRETdata . . . . .	4

genRatioHeatmap . . . . .	4
genTimeSteadyStateHeatmap . . . . .	5
getBreaks10 . . . . .	6
getSpacedSeq . . . . .	6
maturationData . . . . .	7
plotPrimordiumProfile . . . . .	7
profileGradients . . . . .	8
ratioSteadyState . . . . .	8
ratioTimeDependent . . . . .	9
runShinyApps . . . . .	10
signal . . . . .	11
simulatedSignal . . . . .	11

<b>Index</b>	<b>13</b>
--------------	-----------

---

analyticSolutions	<i>Analytic model solutions for fluorescence intensity</i>
-------------------	--

---

### Description

Time-dependent and steady-state analytic solution to one-step model for number of mature fluorophores. Where  $f$  is given as a parameter the returned value is transformed from a molecular population into a fluorescence intensity. For the function names, 0 refers to the dark population of non-mature fluorophores, and 1 to the mature, fluorescent population. 'ss' indicates steady-state solutions.  $t_{ss}$  is the time required to reach steady-state.

### Usage

```
tss(m, k)
x0ss(p, m, k)
x1(p, m, k, t, f=1)
x1ss(p, m, k, f=1)
x1fretFP1(p, m1, m2, k, t, E=0, f=1)
x1fretFP1ss(p, m1, m2, k, E=0, f=1)
```

### Arguments

$p$	Protein production rate (molecules produced per unit time).
$m$	Maturation rate of fluorophore, which can be for either FP1 or FP2 (convert to maturation time with $\log(2)/m$ ).
$m1$	Maturation rate of FP1.
$m2$	Maturation rate of FP2.
$k$	Protein degradation rate (convert to half-life with $\log(2)/k$ ).
$t$	Time (must be non-negative).
$E$	FRET coefficient representing energy transfer from FP1 to FP2.
$f$	Proportionality factor relating intensity to the number of molecules. When equal to one then the readout is number of molecules directly.

**Value**

A numeric specifying the model solution for the given parameters.

**Author(s)**

Joseph D. Barry

**Examples**

```
t0 <- seq(0.001, 1000, by=0.1)
plot(t0, x1(p=10, m=log(2)/5, k=log(2)/100, t=t0), type="l", col="darkgreen",
      lwd=2, xlab="Time (min)", ylab="Number of mature fluorophores", cex.lab=1.4)
```

---

fitCV

*fitCV*

---

**Description**

Fits a smoothing line to coefficient of variation profiles.

**Usage**

```
fitCV(x, scaleLog10)
```

**Arguments**

**x** A dataframe with columns Time (FP maturation time) and CV (coefficient of variation of timer signal).

**scaleLog10** A logical indicating whether the points are spaced on the log10 scale or not.

**Value**

A dataframe containing fitted values and the minimum CV of the profile.

**Author(s)**

Joseph D. Barry

**Examples**

```
if (interactive()) vignette(topic="genPaperFigures", package="TimerQuant")
```

---

 FRETdata

*FRET Data*


---

### Description

A three-dimensional array of dimensions FRET value x tFT x assay type containing FRET readouts.

### Usage

FRETdata

---

genRatioHeatmap

*generate ratio heatmap*


---

### Description

Visualize timer ratios as a function of either FP1 or FP2 maturation time, and protein half-life.

### Usage

```
genRatioHeatmap(tRangeFP, Tfixed, TA, TB, channel, E, f=1, n, ramp)
```

### Arguments

tRangeFP	Vector containing two numerics specifying the range of maturation times to display for the chosen fluorescence channel.
Tfixed	The maturation time of the FP that will remain fixed.
TA	tFT half-life in location A.
TB	tFT half-life in location B.
channel	Integer specifying fluorescence channel to be varied (1 or 2).
E	FRET value representing transfer from FP1 to FP2.
f	$f=f_2/f_1$ , the ratio of prefactors relating the number of molecules to the fluorescence intensity.
n	Integer specifying the number of data points. Choose a higher n for a higher pixel density.
ramp	Colour ramp, see <code>colorRampPalette</code> for more details.

### Value

Returns a ggplot2 heatmap.

### Author(s)

Joseph D. Barry

### Examples

```
if (interactive()) vignette(topic="genPaperFigures", package="TimerQuant")
```

---

`genTimeSteadyStateHeatmap`  
*Time to reach steady-state*

---

### Description

Visualize the time to reach steady-state as a function of FP2 maturation time and protein half-life.

### Usage

```
genTimeSteadyStateHeatmap(tRangeFP2, tRangeHlife, n, ramp)
```

### Arguments

<code>tRangeFP2</code>	Vector containing two numerics specifying the range of FP2 maturation times.
<code>tRangeHlife</code>	Vector containing two numerics specifying the range of protein half-lives.
<code>n</code>	Integer specifying the number of data points. Choose a higher <code>n</code> for a higher pixel density.
<code>ramp</code>	Colour ramp, see <code>colorRampPalette</code> for more details.

### Value

Returns a `ggplot2` heatmap.

### Author(s)

Joseph D. Barry

### Examples

```
if (interactive()) vignette(topic="genPaperFigures", package="TimerQuant")
```

---

getBreaks10	<i>Get log10 breaks</i>
-------------	-------------------------

---

**Description**

Return breaks for each half-decade on the log10 scale, e.g. 1, 5, 10, 50, ...

**Usage**

```
getBreaks10(x)
```

**Arguments**

x                    A vector of numbers. Breaks will be calculated across the range of x.

**Value**

A sequence of breaks useful for ticks or labels on the log10 scale.

**Author(s)**

Joseph D. Barry

**Examples**

```
getBreaks10(c(1, 100))
```

---

getSpacedSeq	<i>Get Spaced Sequence</i>
--------------	----------------------------

---

**Description**

Return points nicely spaced for on the log10 scale.

**Usage**

```
getSpacedSeq(x, n)
```

**Arguments**

x                    A vector of two numbers containing the minimum and maximum of the desired sequence.

n                    The desired length of the sequence to be returned.

**Value**

A sequence of numbers with appropriate spacing for the log10 scale.

**Author(s)**

Joseph D. Barry

**Examples**

```
getSpacedSeq(c(1, 1000), n=10)
```

---

maturationData	<i>Maturation Data</i>
----------------	------------------------

---

**Description**

A five-dimensional array of dimensions time (in minutes) x data columns x sample x view x tFT (identified by the RFP since sfGFP is present for all) containing fluorescence intensity readouts for the fluorophore maturation curves.

**Usage**

```
maturationData
```

---

plotPrimordiumProfile	<i>Plot Primordium Profile</i>
-----------------------	--------------------------------

---

**Description**

Visualizes primordium signal as a function of position with median and median absolute deviation across samples.

**Usage**

```
plotPrimordiumProfile(x, add, ylab, lwd, cex.lab, cex.axis, xlim, ylim, main,
  col, lty, alpha)
```

**Arguments**

x	A matrix of data where rows are samples and columns are sequential positions.
add	A logical indicating whether or not to add to the existing plot.
ylab	The y-axis label.
lwd	Integer specifying width of lines.
cex.lab	Integer specifying size of labels.
cex.axis	Integer specifying size of axis labels.
xlim	An optional vector of length 2 specifying the limits for the x-axis.
ylim	An optional vector of length 2 specifying the limits for the y-axis.

main	Plot title.
col	Line colour.
lty	Style of line
alpha	A numeric between zero and one specifying the level of transparency for the shaded region.

**Value**

Produces a plot of signal vs position summarizing across multiple primordium samples.

**Author(s)**

Joseph D. Barry

**Examples**

```
if (interactive()) vignette(topic="genPaperFigures", package="TimerQuant")
```

---

profileGradients      *Profile Gradients*

---

**Description**

A three-dimensional array of dimensions tFT x sample x position containing ratio readouts for migrating posterior lateral line primordia.

**Usage**

```
profileGradients
```

---

ratioSteadyState      *analytic function ratioSteadyState*

---

**Description**

Steady-state analytic solution to one-step model for the ratio of mature to non-mature fluorophores.

**Usage**

```
ratioSteadyState(T1, T2, halfLife, E=0, f=1)
```



**Arguments**

T1	Maturation time of fluorescent protein 1 (FP1, fast maturing).
T2	Maturation time of fluorescent protein 2 (FP2, slow maturing).
halfLife	Protein half-life.
E	FRET value representing transfer from FP1 to FP2.
f	$f=f_2/f_1$ , the ratio of prefactors relating the number of molecules to fluorescence intensity for each fluorescence channel.

**Value**

A numeric specifying the model steady-state solution for the given parameters.

**Author(s)**

Joseph D. Barry

**Examples**

```
halfLifeSeq <- seq(1, 2000, by=0.1)
plot(halfLifeSeq, ratioSteadyState(T1=5, T2=100, halfLife=halfLifeSeq),
     type="l", lwd=2, ylim=c(0, 1), xlab="tFT half-life (min)",
     ylab="Steady-state ratio", cex.lab=1.4, log="x", col="red")
```

---

ratioTimeDependent     *analytic function ratioTimeDependent*

---

**Description**

Steady-state analytic solution to one-step model for the ratio of mature to non-mature fluorophores.

**Usage**

```
ratioTimeDependent(T1, T2, halfLife, t, E=0, f=1)
```

**Arguments**

T1	Maturation time of fluorescent protein 1 (FP1, fast maturing).
T2	Maturation time of fluorescent protein 2 (FP2, slow maturing).
halfLife	Protein half-life.
t	Time, which must be non-negative.
E	FRET value representing energy transfer from FP1 to FP2.
f	$f=f_2/f_1$ , the ratio of prefactors relating the number of molecules to fluorescence intensity.

**Value**

A numeric specifying the model time-dependent solution for the given parameters.

**Author(s)**

Joseph D. Barry

**Examples**

```
tSeq <- seq(0.1, 300, by=0.1)
plot(tSeq, ratioTimeDependent(T1=5, T2=100, halfLife=30, t=tSeq, E=0, f=1), type="l", lwd=2,
     xlab="time (min)", ylab="ratio", cex.lab=1.4, col="black", ylim=c(0, 0.3))
points(tSeq, ratioTimeDependent(T1=5, T2=100, halfLife=30, t=tSeq, E=0.4, f=1), type="l", lwd=2,
      col="red")
abline(h=ratioSteadyState(T1=5, T2=100, halfLife=30, E=0, f=1), lty=2, col="black")
abline(h=ratioSteadyState(T1=5, T2=100, halfLife=30, E=0.4, f=1), lty=2, col="red")
```

---

runShinyApps

*Run R-shiny applications*

---

**Description**

Wrapper functions that run shiny apps located in extdata subdirectory of R package.

**Usage**

```
runChooseFP2App()
runTimerModellingApp()
```

**Author(s)**

Joseph D. Barry

**Examples**

```
runChooseFP2App()

runTimerModellingApp()
```

---

signal	<i>Signal</i>
--------	---------------

---

**Description**

Computes timer signal (without additive noise) for a set of model parameters.

**Usage**

```
signal(T1, T2, TA, TB, E=0)
```

**Arguments**

T1	Maturation time of fluorescent protein 1 (fast maturing).
T2	Maturation time of fluorescent protein 2 (slow maturing).
TA	Minimum protein half-life.
TB	Maximum protein half-life.
E	FRET value representing transfer from FP1 to FP2.

**Value**

A numeric specifying the timer signal.

**Author(s)**

Joseph D. Barry

**Examples**

```
signal(T1=5, T2=60, TA=30, TB=180, E=0)
signal(T1=5, T2=60, TA=30, TB=180, E=0.5)
```

---

simulatedSignal	<i>Simulated Timer Signal</i>
-----------------	-------------------------------

---

**Description**

Additive error model for timer signal.

**Usage**

```
simulatedRatio(T1, T2, hLife, sigmaAdd, p, E)
simulatedSignal(T1, T2, TA, TB, sigmaAdd, p, E)
simulatedSignalN(T1, T2, TA, TB, sigmaAdd, N, p, E)
```

**Arguments**

T1	Maturation time of fluorescent protein 1 (FP1, fast maturing).
T2	Maturation time of fluorescent protein 2 (FP2, slow maturing).
hLife	Protein half-life.
TA	Minimum protein half-life.
TB	Maximum protein half-life.
sigmaAdd	Standard deviation of normal distribution from which noise terms are drawn.
p	Protein production rate (molecules produced per unit time).
E	FRET value representing energy transfer from FP1 to FP2.
N	Number of simulation realizations.

**Value**

Returns simulated values for ratios or timer signal.

**Author(s)**

Joseph D. Barry

**Examples**

```
if (interactive()) vignette(topic="genPaperFigures", package="TimerQuant")
```

# Index

- \* **FRETdata**
    - FRETdata, 4
  - \* **datasets**
    - maturationData, 7
    - profileGradients, 8
  - \* **fitCV**
    - fitCV, 3
  - \* **genRatioHeatmap**
    - genRatioHeatmap, 4
  - \* **genTimeSteadyStateHeatmap**
    - genTimeSteadyStateHeatmap, 5
  - \* **getBreaks10**
    - getBreaks10, 6
  - \* **getSpacedSeq**
    - getSpacedSeq, 6
  - \* **plotPrimordiumProfile**
    - plotPrimordiumProfile, 7
  - \* **ratioSteadyState**
    - ratioSteadyState, 8
  - \* **ratioTimeDependent**
    - ratioTimeDependent, 9
  - \* **runShinyApps**
    - runShinyApps, 10
  - \* **signal**
    - signal, 11
  - \* **simulatedSignal**
    - simulatedSignal, 11
  - \* **x1**
    - analyticSolutions, 2
- analyticSolutions, 2
- fitCV, 3
- FRETdata, 4
- genRatioHeatmap, 4
- genTimeSteadyStateHeatmap, 5
- getBreaks10, 6
- getSpacedSeq, 6
- maturationData, 7
- plotPrimordiumProfile, 7
- profileGradients, 8
- ratioSteadyState, 8
- ratioTimeDependent, 9
- runChooseFP2App (runShinyApps), 10
- runShinyApps, 10
- runTimerModellingApp (runShinyApps), 10
- signal, 11
- simulatedRatio (simulatedSignal), 11
- simulatedSignal, 11
- simulatedSignalN (simulatedSignal), 11
- tss (analyticSolutions), 2
- x0ss (analyticSolutions), 2
- x1 (analyticSolutions), 2
- x1fretFP1 (analyticSolutions), 2
- x1fretFP1ss (analyticSolutions), 2
- x1ss (analyticSolutions), 2