

# Package ‘stressaddition’

August 3, 2020

**Type** Package

**Title** Modelling Tri-Phasic Concentration-Response Relationships

**Version** 3.0.1

**Date** 2020-07-29

**Description** The stress addition approach is an alternative to the traditional concentration addition or effect addition models. It allows the modelling of tri-phasic concentration-response relationships either as single toxicant experiments, in combination with an environmental stressor or as mixtures of two toxicants. See Liess et al. (2019) <doi:10.1038/s41598-019-51645-4>.

**License** GPL-3

**Copyright** file inst/COPYRIGHTS

**URL** <https://git.ufz.de/oekotox/stressaddition>

**Encoding** UTF-8

**LazyData** true

**Depends** R (>= 3.5)

**Imports** drc (>= 3.0), plotrix

**Suggests** testthat (>= 2.1.0)

**RoxygenNote** 7.1.1

**NeedsCompilation** no

**Author** Sebastian Henz [aut, cre] (<<https://orcid.org/0000-0001-8299-8852>>),  
Matthias Liess [aut] (<<https://orcid.org/0000-0002-3321-8909>>),  
Naeem Shahid [ctb] (<<https://orcid.org/0000-0001-6581-1654>>),  
Helmholtz-Zentrum fuer Umweltforschung GmbH - UFZ [cph, fnd]

**Maintainer** Sebastian Henz <[sebastian.henz@ufz.de](mailto:sebastian.henz@ufz.de)>

**Repository** CRAN

**Date/Publication** 2020-08-03 10:40:05 UTC

## R topics documented:

stressaddition-package	2
ecxsys	3
lc	5
log10_ticks	6
multiple_stress	7
multi_tox	8
plot_ecxsys	10
predict_ecxsys	11
Stressconversion	12
<b>Index</b>	<b>14</b>

---

stressaddition-package

*stressaddition: Modelling Tri-Phasic Concentration-Response Relationships*

---

### Description

The stress addition approach is an alternative to the traditional concentration addition or effect addition models. It allows the modelling of tri-phasic concentration-response relationships either as single toxicant experiments, in combination with an environmental stressor or as mixtures of two toxicants. See Liess et al. (2019) <doi:10.1038/s41598-019-51645-4>.

### Details

This R package contains the definitions of the [ECx-SyS](#) model and the [Multi-TOX](#) model. See the publications linked below for more information including equations.

Author contributions: M. Liess conceived the ECx-SyS and Multi-TOX models. S. Henz developed this R package. N. Shahid contributed to the optimization of the Multi-TOX model and provided the [multiple\\_stress](#) data set.

### Author(s)

**Maintainer:** Sebastian Henz <sebastian.henz@ufz.de> ([ORCID](#))

Authors:

- Matthias Liess ([ORCID](#))

Other contributors:

- Naeem Shahid ([ORCID](#)) [contributor]
- Helmholtz-Zentrum fuer Umweltforschung GmbH - UFZ [copyright holder, funder]

## References

Liess, M., Henz, S. & Knillmann, S. Predicting low-concentration effects of pesticides. *Sci Rep* 9, 15248 (2019).

Liess, M., Henz, S., Shahid, N., 2020. Modelling the synergistic effects of toxicant mixtures. Manuscript submitted for publication.

## See Also

Useful links:

- <https://git.ufz.de/oekotox/stressaddition>

---

ecxsys

*ECx-SyS*

---

## Description

ECx-SyS is a model for tri-phasic concentration-response relationships where hormetic and sub-hormetic effects are observed at low concentrations. It expands the Stress Addition Model (SAM) by introducing system stress (SyS) which is negatively correlated with toxicant stress. A constant environmental stress can be included. See the publication for details.

## Usage

```
ecxsys(  
  concentration,  
  hormesis_concentration,  
  survival_tox_observed,  
  survival_tox_env_observed = NULL,  
  survival_max = 100,  
  curves_concentration_max = NULL,  
  p = 3.2,  
  q = 3.2  
)
```

## Arguments

**concentration** A vector of concentrations. Must be sorted in ascending order and the first element must be 0 to indicate the control.

**hormesis\_concentration**

The concentration where the hormesis occurs. This is usually the concentration of the highest survival after the control.

**survival\_tox\_observed**

A vector of survival values observed at the given concentrations and in absence of environmental stress. Values must be between 0 and survival\_max.

<code>survival_tox_env_observed</code>	Survival values observed in the presence of environmental stress. Must be between 0 and <code>survival_max</code> .
<code>survival_max</code>	The maximum value the survival could possibly reach. For survival data in percent this should be 100 (the default).
<code>curves_concentration_max</code>	The maximum concentration of the predicted curves. This might be useful if for example your highest observed concentration is 30 but you would like to know the predicted values on a scale between 0 and 100.
<code>p, q</code>	The shape parameters of the beta distribution. Default is $p = q = 3.2$ .

## Details

It is advised to complete the curve down to zero for optimal prediction. Therefore `survival_tox_observed` in the highest concentration should be at or close to zero. If the model does not fit properly try adding a survival of 0 at ten times the maximum observed concentration.

The vectors `concentration`, `survival_tox_observed` and `survival_tox_env_observed` (if provided) must be of equal length and sorted by increasing concentration.

## Value

A list (of class `ecxsys`) containing many different objects of which the most important are listed below. The survival and stress vectors correspond to the provided concentrations.

**`survival_tox`** Modeled survival resulting from toxicant stress.

**`survival_tox_sys`** Modeled survival resulting from toxicant and system stress.

**`survival_tox_env`** Modeled survival resulting from toxicant and environmental stress.

**`survival_tox_env_sys`** Modeled survival resulting from toxicant, environmental and system stress.

**`survival_tox_LL5`** The survival predicted by the five-parameter log-logistic model derived from the observations under toxicant stress but without environmental stress.

**`survival_tox_env_LL5`** The survival predicted by the five-parameter log-logistic model derived from the observations under toxicant stress with environmental stress.

**`curves`** A data frame containing survival and stress values as returned by `predict_ecxsys`. The concentrations are regularly spaced on a logarithmic scale in the given concentration range. The control is approximated by the lowest non-control concentration times  $1e-7$ . The additional column `concentration_for_plots` is used by the plotting functions of this package to approximate the control and generate a nice concentration axis.

## References

Liess, M., Henz, S. & Knillmann, S. Predicting low-concentration effects of pesticides. *Sci Rep* 9, 15248 (2019).

## Examples

```

model <- ecxsys(
  concentration = c(0, 0.05, 0.5, 5, 30),
  hormesis_concentration = 0.5,
  survival_tox_observed = c(90, 81, 92, 28, 0),
  survival_tox_env_observed = c(29, 27, 33, 5, 0)
)

# Use survival_max if for example the survival is given as the average number
# of surviving animals and the initial number of animals is 21:
model <- ecxsys(
  concentration = c(0, 0.03, 0.3, 3, 30),
  hormesis_concentration = 0.3,
  survival_tox_observed = c(17, 15.2, 18.8, 7, 0),
  survival_tox_env_observed = c(4.8, 4.6, 6.4, 0, 0),
  survival_max = 21
)

```

---

lc *Lethal Concentrations*

---

## Description

Estimate the concentration to reach a certain mortality relative to the control.

## Usage

```
lc(model, response_name, response_level, reference, warn = TRUE)
```

## Arguments

model	This can be one of three types of objects: Either the output of <code>ecxsys</code> or the output of <code>predict_ecxsys</code> or a data frame with a "concentration" column and a response_name column. In the case of the data frame the first row is assumed to be the control. See the examples.
response_name	The name of the survival or stress for which you want to calculate the LC.
response_level	The desired response level as a percentage between 0 and 100. For example with the value 10 the function will return the LC10, i.e. the concentration where the response falls below 90 % of the control response. In other words: where a mortality of 10 % relative to the control is reached.
reference	The reference value of the response, usually the control at concentration 0. This argument is optional. If it is missing, the first value of the response is used as control. This value determines what number response_level actually corresponds to. For example if response_level is 10 and reference is 45, then the function returns the concentration where the curve is reduced by 10% relative to 45 = 40.5.
warn	Logical. Should the function emit a warning if the calculation of the lethal concentration is not possible?

**Details**

If the response level occurs multiple times because of hormesis, which may happen for low values of `response_level`, then the occurrence with the smallest concentration is returned.

This function only makes sense for curves which generally go down with increasing concentration, i.e. all `survival_*` curves and also `sys_tox` and `sys_tox_env`. Others are untested and may give unexpected results, if any.

**Value**

A list containing the lethal concentration and the corresponding survival. The survival will be NA if its calculation is impossible using the supplied data.

**Examples**

```
# Calculate the LC10, the concentration where the survival falls
# below 90% of the survival in the control.

model <- ecxsys(
  concentration = c(0, 0.05, 0.5, 5, 30),
  hormesis_concentration = 0.5,
  survival_tox_observed = c(90, 81, 92, 28, 0)
)

# using the ecxsys() output or the curves therein directly:
lc(model, "survival_tox_sys", 10)
lc(model$curves, "survival_tox_sys", 10)

# using the output of predict_ecxsys() with custom concentrations:
conc <- 10^seq(-9, 1, length.out = 1000)
curves <- predict_ecxsys(model, conc)
lc(curves, "survival_tox_sys", 10)

# using a custom data frame:
df_custom <- data.frame(
  concentration = curves$concentration,
  foo = curves$survival_tox_sys
)
lc(df_custom, "foo", 10)

# Calculate the LC50 relative to an survival of 100
# instead of relative to the control:
lc(model, "survival_tox_sys", 50, reference = 100)
```

---

log10\_ticks

*Logarithmic axis tick marks*


---

**Description**

Calculate the positions and labels of major and minor tick marks for a base 10 logarithmic axis.

**Usage**

```
log10_ticks(x, label_zero = TRUE)
```

**Arguments**

**x** A vector of axis values. Can be arbitrarily long but only the minimum and maximum are necessary.

**label\_zero** Whether or not to replace the smallest major label with "0". This defaults to TRUE and is useful for some types of plots used to display concentration-response data where the leftmost data point represents the control.

**Value**

A list with the positions and labels of the major and minor tick marks. The labels are formatted without trailing zeros using `formatC(labels, format = "fg")`.

**Examples**

```
x <- c(0.01, 0.2, 3, 10, 50)
plot(x, c(5, 4, 2.5, 1, 0), xaxt = "n", log = "x")
ticks <- log10_ticks(x)
axis(1, at = ticks$major, labels = ticks$major_labels)
axis(1, at = ticks$minor, labels = FALSE, tcl = -0.25)
```

---

multiple\_stress

*Survival of Daphnia magna exposed to multiple stressors*


---

**Description**

A data set of mixture toxicity experiments. Individuals of *Daphnia magna* were exposed to combinations of food stress and multiple concentrations of the pesticides esfenvalerate and prochloraz. The survival was recorded at 21 days after contamination.

**Usage**

```
multiple_stress
```

**Format**

A data frame with 58 rows and 4 variables:

**food** The amount of food in percent.

**esfenvalerate** The concentration of esfenvalerate in  $\mu\text{g/L}$ .

**prochloraz** The concentration of prochloraz in  $\mu\text{g/L}$ .

**survival** The mean survival in percent.

## Source

Shahid, N., Liess, M., Knillmann, S., 2019. Environmental Stress Increases Synergistic Effects of Pesticide Mixtures on *Daphnia magna*. *Environ. Sci. Technol.* 53, 12586–12593.

Liess, M., Henz, S., Shahid, N., 2020. Modelling the synergistic effects of toxicant mixtures. Manuscript submitted for publication.

---

multi\_tox

*Predict the survival of binary toxicant mixtures*

---

## Description

The Multi-TOX model predicts the effects of binary toxicant mixtures based on three-phasic concentration-response relationships. See the publication for details.

## Usage

```
multi_tox(  
  model_a,  
  model_b,  
  concentration_a,  
  concentration_b,  
  sa_contribution = 0.5,  
  survival_max = 100  
)
```

## Arguments

model\_a, model\_b

The ecxsys models of the toxicants.

concentration\_a, concentration\_b

The concentrations of the toxicants in the mixture. Both vectors must either be the same length or the longer length must be a multiple of the shorter length. That's because the shorter concentration vector gets recycled to the length of the longer one.

sa\_contribution

The proportion of stress addition contributing to the calculation of the toxicant stress in the mixture. Must be between 0 and 1 where 1 stands for 100 % stress addition.

survival\_max

Controls the scaling of the result. This represents the maximum value the survival could possibly reach. For survival data in percent this should be 100 (the default).

## Details

The predictions are symmetric, i.e. it does not matter which of the toxicant models is a or b as long as the concentration arguments are supplied in the same order.

This method is only suitable for experiments without or with low environmental stress. Any environmental stress supplied as arguments to `ecxsys` in `model_a` or `model_b` is ignored.



**Value**

A data frame with columns of the supplied concentrations and the corresponding mixture survival and stresses.

**References**

Liess, M., Henz, S., Shahid, N., 2020. Modelling the synergistic effects of toxicant mixtures. Manuscript submitted for publication.

**Examples**

```
# Using a data set which is included in this package. See ?multiple_stress
ms <- multiple_stress
esfen <- ms[ms$food == 1 & ms$prochloraz == 0, ]
proch <- ms[ms$food == 1 & ms$esfenvalerate == 0, ]

model_esfen <- ecxsys(
  concentration = esfen$esfenvalerate,
  survival_tox_observed = esfen$survival,
  hormesis_concentration = 0.1
)
model_proch <- ecxsys(
  concentration = proch$prochloraz,
  survival_tox_observed = proch$survival,
  hormesis_concentration = 100
)

# Predict the survival at 8 different esfenvalerate concentrations
# but keep the prochloraz concentration constant at 32:
mt <- multi_tox(
  model_esfen,
  model_proch,
  c(0, 0.0001, 0.001, 0.01, 0.1, 0.316, 1, 3.16),
  32,
  sa_contribution = 0.8
)
mt[1:3] # The concentrations and survival of the 8 mixtures.

# Predict the survival at 4 different combinations
# of esfenvalerate and prochloraz:
mt <- multi_tox(
  model_esfen,
  model_proch,
  c(0.1, 0.2, 0.3, 0.4),
  c(0, 1, 32, 100),
  sa_contribution = 0.8
)
mt[1:3] # The concentrations and survival of the 4 mixtures.
```

plot\_ecxsys

*Plot the results of the ECx-SyS model*

---

**Description**

Plot the observed and modeled survivals and stresses.

**Usage**

```
plot_stress(  
  model,  
  which = NA,  
  show_legend = FALSE,  
  xlab = "concentration",  
  ylab = "stress",  
  main = NULL  
)
```

```
plot_survival(  
  model,  
  which = NA,  
  show_legend = FALSE,  
  xlab = "concentration",  
  ylab = "survival",  
  main = NULL  
)
```

**Arguments**

model	The object returned from <a href="#">ecxsys</a> .
which	A vector of names to plot. Allowed are the column names of the <code>model\$curves</code> data frame. There is also "survival_tox_observed" and "survival_tox_env_observed" for the observed survival and "sys_tox_observed" and "sys_tox_env_observed" for the observed Sys. The default NA only plots the most important curves. Use <code>which = "all"</code> to display all curves. An empty vector or NULL creates just the axes.
show_legend	Should the plot include a legend? Defaults to FALSE because it may cover some parts of the plot depending on the plot size and the number of elements shown.
xlab, ylab, main	Axis labels and title.

**Value**

None

## Examples

```
model <- ecxsys(
  concentration = c(0, 0.05, 0.5, 5, 30),
  hormesis_concentration = 0.5,
  survival_tox_observed = c(90, 81, 92, 28, 0),
  survival_tox_env_observed = c(29, 27, 33, 5, 0)
)
plot_survival(model, show_legend = TRUE)
plot_stress(model, show_legend = TRUE)

# Plot all curves:
plot_survival(model, which = "all")
plot_stress(model, which = "all")

# Plot only some selected curves:
plot_survival(model, which = c("survival_tox_sys", "survival_tox_env_sys"))
plot_stress(model, which = c("sys_tox", "sys_tox_env"))

# Plot only the observed values:
plot_survival(model, which = c("survival_tox_observed", "survival_tox_env_observed"))
plot_stress(model, which = c("sys_tox_observed", "sys_tox_env_observed"))
```

---

predict\_ecxsys

*Predict survival and stress*

---

## Description

Calculate the survivals and stresses of an ECx-SyS model at arbitrary concentrations.

## Usage

```
predict_ecxsys(model, concentration)
```

## Arguments

**model** An ECx-SyS model as returned by [ecxsys](#).  
**concentration** A numeric vector of concentrations.

## Value

A data frame (of class "ecxsys\_predicted") with the following columns:

**concentration** The supplied concentrations.

**survival\_tox\_LL5** The survival predicted by the five-parameter log-logistic model derived from the observations under toxicant stress but without environmental stress.

**survival\_tox** Modeled survival resulting from toxicant stress.

**survival\_tox\_sys** Modeled survival resulting from toxicant and system stress.

**stress\_tox** The toxicant stress.

**sys\_tox** System stress under toxicant stress conditions without environmental stress.

**stress\_tox\_sys** The sum of `stress_tox` and `sys_tox`.

**survival\_tox\_env\_LL5** The survival predicted by the five-parameter log-logistic model derived from the observations under toxicant stress with environmental stress.

**survival\_tox\_env** Modeled survival resulting from toxicant and environmental stress.

**survival\_tox\_env\_sys** Modeled survival resulting from toxicant, environmental and system stress.

**stress\_env** Environmental stress.

**stress\_tox\_env** The sum of toxicant and environmental stress.

**sys\_tox\_env** System stress under toxicant and environmental stress conditions.

**stress\_tox\_env\_sys** The sum of `stress_tox_env` and `sys_tox_env`.

### Examples

```
model <- ecxsys(
  concentration = c(0, 0.05, 0.5, 5, 30),
  hormesis_concentration = 0.5,
  survival_tox_observed = c(90, 81, 92, 28, 0),
  survival_tox_env_observed = c(29, 27, 33, 5, 0)
)
p <- predict_ecxsys(model, c(0.001, 0.01, 0.1, 1, 10))
```

---

Stressconversion

*Convert Between Stress and Survival*

---

### Description

Functions to convert survival to general stress or vice versa using the beta distribution.

### Usage

```
survival_to_stress(survival, p = 3.2, q = 3.2)
```

```
stress_to_survival(stress, p = 3.2, q = 3.2)
```

### Arguments

<code>survival</code>	One or more survival values to convert to general stress. Should be a value between 0 and 1. Smaller or bigger values are treated as 0 or 1 respectively.
<code>p, q</code>	The shape parameters of the <a href="#">beta</a> distribution. Default is 3.2.
<code>stress</code>	One or more stress values to convert to survival.

**Details**

These are simple wrappers around the beta distribution function [pbeta](#) and the beta quantile function [qbeta](#).

**Value**

`stress_to_survival` returns a vector the same length as `stress` giving the survival caused by each amount of stress.

`survival_to_stress` returns a vector the same length as `survival` containing the stress values associated with each survival.

**Examples**

```
stress <- 0.3
survival <- stress_to_survival(stress)
survival_to_stress(survival)
```

# Index

## \* datasets

multiple\_stress, [7](#)

beta, [12](#)

ECx-SyS, [2](#)

ecxsys, [3](#), [5](#), [8](#), [10](#), [11](#)

lc, [5](#)

log10\_ticks, [6](#)

Multi-TOX, [2](#)

multi\_tox, [8](#)

multiple\_stress, [2](#), [7](#)

pbeta, [13](#)

plot\_ecxsys, [10](#)

plot\_stress (plot\_ecxsys), [10](#)

plot\_survival (plot\_ecxsys), [10](#)

predict\_ecxsys, [4](#), [5](#), [11](#)

qbeta, [13](#)

stress\_to\_survival (Stressconversion),  
[12](#)

stressaddition  
(stressaddition-package), [2](#)

stressaddition-package, [2](#)

Stressconversion, [12](#)

survival\_to\_stress (Stressconversion),  
[12](#)