

Package ‘renz’

October 14, 2022

Type Package

Title R-Enzymology

Version 0.1.1

Author Juan Carlos Aledo

Maintainer Juan Carlos Aledo <caledo@uma.es>

Description

Contains utilities for the analysis of Michaelian kinetic data. Beside the classical linearization methods (Lineweaver-Burk, Eadie-Hofstee, Hanes-Woolf and Eisenthal-Cornish-Bowden), features include the ability to carry out weighted regression analysis that, in most cases, substantially improves the estimation of kinetic parameters (Aledo (2021) <[doi:10.1002/bmb.21522](https://doi.org/10.1002/bmb.21522)>). To avoid data transformation and the potential biases introduced by them, the package also offers functions to directly fitting data to the Michaelis-Menten equation, either using ([S], v) or (time, [S]) data. Utilities to simulate substrate progress-curves (making use of the Lambert W function) are also provided. The package is accompanied of vignettes that aim to orientate the user in the choice of the most suitable method to estimate the kinetic parameter of an Michaelian enzyme.

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

Depends R (>= 4.0.0)

Imports graphics, stats, VGAM

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2021-12-02 13:50:02 UTC

R topics documented:

dir.MM	2
ecb	3
eh	4
fE.progress	4
hw	5
int.MM	6
lb	7
ONPG	8
sE.progress	8

Index	10
--------------	-----------

dir.MM	<i>Non-linear Least-squares Fitting of the MM equation</i>
--------	--

Description

Non-linear least-squares fitting of the Michaelis-Menten equation.

Usage

```
dir.MM(data, unit_S = 'mM', unit_v = 'au', plot = TRUE)
```

Arguments

data	a dataframe with two columns. The first column contains the values of the independent variable (substrate concentration), and the second column contains the initial rates.
unit_S	concentration unit.
unit_v	time unit.
plot	logical. If true, the data and fitted curve are plotted.

Details

This function invokes `nls()` to carry out the fitting.

Value

A list of two elements. The first one is a vector containing the enzyme kinetic parameters. The second one is a dataframe with the original data plus the fitted value of v .

Examples

```
dir.MM(ONPG[, c(1,2)])
```

Description

Obtains K_m and V_m using the Eisenthal & Cornish-Bowden method.

Usage

```
ecb(data, unit_S = 'mM', unit_v = 'au', plot = TRUE)
```

Arguments

<code>data</code>	a dataframe where the first column is the independent variable, [S], and the remaining columns (as many as experiment replicates) correspond to the dependent variable, v.
<code>unit_S</code>	concentration unit.
<code>unit_v</code>	time unit.
<code>plot</code>	logical. If TRUE data are plotted.

Details

For each experimental replicate the observations (S, v) are plotted as lines in the K_m - V_m parameter space, instead of points in observation space. Afterwards, the lines tend to intersect at a common point, whose coordinates provide the kinetic parameters. Nevertheless, since the observations are subject to error, there is no unique intersection point for all the lines. In this case, the method computes all the pair-wise intersections. Then, the median value from each series is taken to be the best estimate of K_m and V_m . This procedure is repeated as many times as replicates and finally the mean and sd is returned.

Value

Returns a list with the estimated values of K_m and V_m .

References

Biochem.J.(1974) 139:715-720 (10.1042/bj1390715)

See Also

lb(), hw(), eh()

Examples

```
ecb(ONPG[, c(1,2)])
```

eh *Eadie-Hofstee Transformation*

Description

Obtain Km and Vm using the Eadie-Hofstee transformation.

Usage

```
eh(data, unit_S = 'mM', unit_v = 'au', plot = TRUE)
```

Arguments

data	a dataframe where the first column is the independent variable, [S], and the remaining columns (as many as experiment replicates) correspond to the dependent variable, v.
unit_S	concentration unit.
unit_v	time unit.
plot	logical. If TRUE the data and fitted line are plotted.

Value

A dataframe with the values of the transformed variables is returned. The fitted Km and Vm are given as attributes of this dataframe.

See Also

lb(), hw(), ecb()

Examples

```
eh(ONPG[, c(1,2)])
```

fE.progress *Fitted Progress Curve for Enzyme-Catalyzed Reaction*

Description

Fits the progress curve of an enzyme-catalyzed reaction.

Usage

```
fE.progress(data, unit_S = 'mM', unit_t = 'min')
```

Arguments

data	a dataframe where the first column is the time and the second column is the substrate concentration.
unit_S	concentration unit.
unit_t	time unit.

Value

Returns a list with two elements. The first one contains the fitted kinetic parameters, the second one is a dataframe giving the fitted substrate concentration time course.

References

Biochem Mol Biol Educ.39:117-25 (10.1002/bmb.20479).

See Also

sEprogress(), int.MM()

Examples

```
data <- sE.progress(So = 10, time = 5, Km = 4, Vm = 50, plot = FALSE)
fE.progress(data[, c(1,3)])
```

hw

Hanes-Woolf Transformation

Description

Obtains Km and Vm using the Hanes-Woolf transformation.

Usage

```
hw(data, unit_S = 'mM', unit_v = 'au', plot = TRUE)
```

Arguments

data	a dataframe where the first column is the independent variable, [S], and the remaining columns (as many as experiment replicates) correspond to the dependent variable, v.
unit_S	concentration unit.
unit_v	time unit.
plot	logical. If TRUE the data and fitted line are plotted.

Value

A dataframe with the values of the transformed variables is returned. The fitted Km and Vm are given as attributes of this dataframe.

See Also

lb(), eh(), ecb()

Examples

```
hw(ONPG[, c(1,2)])
```

int.MM

Linearization of The Integrated Michaelis-Menten Equation

Description

Estimates the kinetic parameters using an linearized form of the integrated Michaelis-Menten equation.

Usage

```
int.MM(data, unit_S = 'mM', unit_t = 'min')
```

Arguments

data	a dataframe with two columns. The first column contains the values of the independent variable time, t, and the second column contains the substrate concentrations.
unit_S	concentration unit.
unit_t	time unit.

Details

The r-squared value of the model can be checked using attributes().

Value

A list of two elements. The first element is named vector containing the Km and Vm. The second element is a dataframe where the first two columns are the original data and the last two columns are the transformed variables. Also a linear plot of the transformed variables together with the parameters values are provided.

Examples

```
int.MM(data = sE.progress(So = 10, time = 5, Km = 4, Vm = 50)[, c(1,3)])
```

lb *Lineweaver-Burk Transformation*

Description

Obtains Km and Vm using double reciprocal transformation

Usage

```
lb(data, unit_S = 'mM', unit_v = 'au', weighting = FALSE, plot = TRUE)
```

Arguments

<code>data</code>	a dataframe where the first column is the independent variable, [S], and the remaining columns (as many as experiment replicates) correspond to the dependent variable, v.
<code>unit_S</code>	concentration unit.
<code>unit_v</code>	time unit.
<code>weighting</code>	logical. When TRUE the weight v^4 is employed.
<code>plot</code>	logical. If TRUE the data and fitted line are plotted.

Value

A double reciprocal plot and the Km and Vm computed using averaged $1/v$ (when more than one replicate is provided). In addition, this function returns a list of five elements. The first and second ones are vectors with the Km and Vm, respectively, computed individually for each replicate. The third one provides the R-squared values of the fits. The fourth element of the list gives the fitted Km and Vm. The last element of the list is a dataframe with the values of the transformed variables.

References

J. Am. Chem. Soc.1934, 56, 3,658-666 (doi.org/10.1021/ja01318a036)

See Also

`hw()`, `eh()`, `ecb()`

Examples

```
lb(ONPG[, c(1,2)], weighting = TRUE)
```

ONPG

Kinetic data for the hydrolysis of ONPG catalyzed by Beta-galactosidase (EC. 3.2.1.23)

Description

In the University of Málaga, Enzymology is a second-year subject that all Biochemistry students must take. In the context of this subject, students carry out different experiments in the laboratory, using Beta-galactosidase (EC. 3.2.1.23) as an enzyme model, to illustrate the effect of different variables on the rate of the enzyme-catalyzed reaction (hydrolysis of o-nitrophenyl-Beta-d-galactopyranoside, ONPG). One of these experiments consists in assessing the effect of the substrate (ONPG) concentration on the initial rate. The current dataframe shows the results obtained by eight different student groups, as were presented in their reports.

Usage

ONPG

Format

A dataframe with 10 rows (one per substrate concentration) and 9 columns. The first column give the ONPG concentrations assayed (in mM). The remaining columns provide the determined initial rates. Please, note that rates are given using different units, which can be checked typing in the console: `attributes(ONPG)`.

sE.progress

Progress Curve for Enzyme-Catalyzed Reaction

Description

Simulates the evolution of the substrate concentration along time.

Usage

```
sE.progress(So, time, Km, Vm, unit_S = 'mM', unit_t = 'min',
            I = 0, Kic = Inf, Kiu = Inf, replicates = 3,
            error = 'a', sd = 0.005, plot = TRUE)
```

Arguments

So	initial substrate concentration.
time	reaction timespan.
Km	Michaelis constant.
Vm	maximal velocity.

<code>unit_S</code>	concentration unit.
<code>unit_t</code>	time unit.
<code>I</code>	inhibitor concentration.
<code>Kic</code>	competitive inhibition constant.
<code>Kiu</code>	uncompetitive inhibition constant.
<code>replicates</code>	number of replicates for the dependent variable
<code>error</code>	it should be one among c('absolute', 'relative').
<code>sd</code>	standard deviation of the error.
<code>plot</code>	logical. If TRUE, the progress curve is plotted.

Details

When `sd` is different to 0, then an absolute error normally distributed is added to the variable `St`.

Value

Returns a dataframe where the two first columns are time and `St` (without error). The two last columns are the mean and `sd` of the variable `St`.

See Also

`fE.progress()`

Examples

```
sE.progress(So = 10, time = 5, Km = 4, Vm = 50, plot = FALSE)
```

Index

* **datasets**

ONPG, 8

dir.MM, 2

ecb, 3

eh, 4

fE.progress, 4

hw, 5

int.MM, 6

lb, 7

ONPG, 8

sE.progress, 8