Package ‘BrainMetabolism’

August 23, 2015

Type Package

Title calculate brain metabolism rates from extracellular conentrations

Version 1.0

Date 2015-03-03

Author baptiste balanca

Maintainer baptiste balanca <baptiste.balanca@gmail.com>

Depends R (>= 3.1.0), ggplot2, stats

Imports ggplot2

Description calculate CMRO2, CMRgluc, mitochondrialPO2 ... provide functions to process data from biosensors

License MIT

NeedsCompilation no

R topics documented:

BrainMetabolism-package *calculate brain metabolism rate from extracellular conentrations*

Description

calculate CMRO2, CMRgluc, mitochondrialPO2 ...
provide functions to process data from biosensors

Details

Package:	BrainMetabolism
Type:	Package
Version:	1.0
Date:	2015-03-03
License:	MIT

Author(s)

Baptiste Balanca
Lyon Neuroscience Research Center
Team TIGER

Maintainer: baptiste balanca <baptiste.balanca@gmail.com>

References

Piilgaard Lauritzen JCBFM (2009) 29, 1517
Gjedde et al JCBFM (2005) 25(9), 1183
Gjedde et al JCBFM (2000) 20(4), 747
Du et al JCBFM 2012 32(9)
Gruetter et al J Neurochem 1998 70(1)
balanca et al (2016) in preparation

Examples

```
MyTPO2<-23 #mmHg
MyLDF<-95 #percent
MyL<-L.calc(CMR=219, TPO2=MyTPO2, P50=36, h=2.7, Ca=8, cbf=53 )
MyCMRO2<-CMRO2.calc(LDF=MyLDF, MyTPO2, P50=36, h=2.7, Ca=8, L=MyL,cbfbase=53)

MyGlucBrain<-1 #mM
MyGlucPlasma<-6 #mM
MyCRMgluc<-CMRGluc.calc(MyGlucbrain, Vd=0.77, Kt=13.4, Tmax=1.35, Gplasma=MyGlucPlasma)
```

auc *area under the curve*

Description

take a numeric vector and return the area under the curve (AUC)

Usage

```
auc(data, pos = TRUE)
```

Arguments

`data` : a numeric vector

`pos` : a logical value. if TRUE (default) AUC is calculated on positive values, if FALSE on negative values.

Value

auc : area under the curve

max/min : maxiaml value (or minimal if pos=FALSE)

calibration *Sensor calibration*

Description

Fit an nth degree polynom

$$(y = C_n X^n + C_{n-1} X^{n-1} + \ldots + C_1 X + C_0)$$

to vectors x=volt and y=concentration

Usage

```
calibration(volt, mol, order = 1)
```

Arguments

`volt` : a numeric vector of the biosensor voltage

`mol` : a numeric vector of the molecule conentration in the medium (x and y must have the same length)

`order` : a numeric value for the polynomial degree. default is one.

Value

Coef : coeficients in ascending order (i.e. C0, C1, C2, . . . , Cn)

R2 : goodness of fit

CMRGluc.calc *CMRGluc calculation*

Description

take extracellular glucose concentration and return brain metabolic rate of glucose, using a reversible Michaelis-Menten model equation:

Usage

```
CMRGluc.calc(Gbrain, Vd = 0.77, Kt = 1.4, Tmax = 1.27, Gplasma = 7.3)
```

Arguments

`Gbrain` : a vector/numeric value of extracellular glucose concentration in mmol/L, time decay

`Vd` : glucose brain space diffusion (default is 0.77 ml/g)

`Kt` : glucose apparent maximal transport rate (default is 1.4 mmol/L)

`Tmax` : the apparent maximal transport rate (default is 1.27 micormol/g/min)

`Gplasma` : plasma glucose concentration (default=6mmol/L)

Details

$$G_{brain} = V_d \frac{\left(\frac{T_{max}}{CMR_{gluc}} - 1\right) \times G_{plasma} - K_t}{\frac{T_{max}}{CMR_{gluc}} + 1}$$

Value

CMRGlucose : numeric value of Cerebral metabolic rate of glucose in micromol/g/min

References

Morgenthaler et al. Neurochem Int 2006, 48

Ori et al. Anesthesiology 1986, 65(2)

Duarte et al. Front in Neuroenrgetics 2009, 1

Gruetter et al J Neurochem 1998, 70(1)

`CMRO2.calc` *CMRO2 calculation*

Description

take tissue oxygen pressure (tPO2) and cerebral blood flow (relative value, e.g. laser doopler lowmetry BPU)

Usage

```
CMRO2.calc(LDF, TPO2, P50 = 36, h = 2.7, Ca = 8, L = 4.03,
  cbfbase = 53)
```

Arguments

`LDF` : a vector/numeric value of LDF (percentage of baseline, i.e: basal value is 100 %)

`TPO2` : a vector/numeric value of brain oxygnen pressure (mmHg)

`P50` : half-saturation tension of hemoglobine (default is 36 mmHg)

`h` : hill's coeficient

`Ca` : oxygen arterial concentration (default is 8 micromol/ml)

`L` : effective diffusion coefficient of oxygen in brain tissue, default is 4.03 micomol/100g/mmHg, but one should use the L.calc function to calculate it from their data.

`cbfbase` : basal expected value of CBF (default is 53 ml/100g/min wich was used to calculate L)
LBF and TPO2 must be the same length

Details

$$PbtO_2 = P_{50}\cdot \sqrt[h]{\frac{2\cdot C_a\cdot CBF}{CMRO_2} - 1} - \frac{CMRO_2}{2\cdot L}$$

Value

CMRO2 : vector/numeric value of Cerebral Metabolic Rate of Oxygen in micromol/100g/min

References

Gjedde et al JCBFM (2005) 25(9), 1183

Piilgaard et al JCBFM (2009) 29, 1517

`correction.Temp` *Temperature correction*

Description

Biosensor enzymatic reaction, that underpin amperometric measures, has a sigmoid relation to temperature :

$$m(x,P) = \frac{P_1 + (P_2 - P_1)}{(1 + \exp((P_3 - x)/P_4))}$$

parameters are different for each enzyme and has been measured in vitro.
This function correct biosensor signal depending on temperature conditions during calibration and experimentation

Usage

```
correction.Temp(x, enz, temp.calib = 25, temp.exp = 37)
```

Arguments

`x`	: a numeric vector of the biosensor voltage
`enz`	: enzyme on the biosensor i.e. "glucose", "lactate", "glutamate, "daao"
`temp.calib`	: temperature of the medium where sensor has been calibrated the default is 25^oC (i.e. room temperature).
`temp.exp`	: temperature of the medium during experiment the default is 37^oC (i.e. animal central temperature)

Value

volt.temp.cor : a vector of corected x values for temperature

References

balanca et al 2015

`correction.TPO2` *Oxygen Tension correction*

Description

Biosensor enzymatic reaction, that underpin amperometric measures, has an asymptotic relation to oxygen tention in the medium :

$$m(PO_2, P) = P_1 + (P_2 - P_1) \times \exp(-\exp(P_3)PO_2)$$

parameters are different for each enzyme and has been measured in vitro
This function correct biosensor signal depending on PO2 conditions during calibration and experimentation

Usage

```
correction.TPO2(x, enz, TPO2 = 28)
```

Arguments

`x` : a numeric vector of the biosensor voltage

`enz` : enzyme on the biosensor i.e. "glucose", "lactate", "glutamate, "daao"

`TPO2` : oxygene tension in the medium during the experiment. the default is 30mmHg measured in anesthetized rat brain.

Value

volt.O2.cor : a vector of corected x values for TPO2

References

balanca et al 2015

`IO2.calc` *Oxidative index calculation*

Description

take CMRO2 and LDF to give an oxidative index

Usage

```
IO2.calc(CMRO2, LDF, cbfbasal = 53)
```

Arguments

`CMRO2` : vector/numeric value of Cerebral metabolic rate of oxygen in micromol/100g/min

`LDF` : a vector/numeric value of LDF (percentage from baseline, baseline is 100 %)

`cbfbasal` : basale expected value of CBF from the litterature (default is 53 ml/100g/min wich was used to calculate L)

Details

$$IO2 = CMRO2/(cbfbasal * LDF)$$

Value

IO2 : oxidative index, reflect the degree of flow metabolism coupling

References

Gjedde et al JCBFM (2000) 20(4), 747

`L.calc` *Calculate the effective diffusion coefficient of oxygen in brain tissue, L.*

Description

take CMRO2 and cerebral blood flow (CBF) to calculate the effective diffusion coefficient of oxygen in brain tissue (L)

Usage

```
L.calc(CMR = 219, TPO2, P50 = 36, h = 2.7, Ca = 8, cbf = 53)
```

Arguments

`CMR`	: Cerebral Metabolic Rate of Oxygen, default is 219 micro mol/100
`TPO2`	: numeric value of brain oxygnen pressure (mmHg)
`P50`	: half-saturation tension of hemoglobine (default is 36 mmHg)
`h`	: hill's coeficient
`Ca`	: oxygen arterial concentration (default is 8 micromol/ml)
`cbf`	: expected value of CBF (default is 53 ml/100g/min wich was used to calculate L

Details

$$PbtO_2 = P_{50}\cdot \sqrt[h]{\frac{2\cdot C_a\cdot CBF}{CMRO_2} - 1} - \frac{CMRO_2}{2\cdot L}$$

Value

L : numeric value of the effective diffusion coefficient of oxygen in brain tissue micomol/100g/mmHg

References

Gjedde et al JCBFM (2005) 25(9), 1183

Piilgaard et al JCBFM (2009) 29, 1517

noise.na *Remove artifacts from biosensor signal, based on data's standar deviation*

Description

Remove artifacts from biosensor signal, based on data's standar deviation

Usage

```
noise.na(data, z = 20, width = 30)
```

Arguments

`data`	: a numeric vector
`z`	: a numeric value. Number of SD over which values should be exculded
`width`	: size of the window used to roll SD over data (see roll.funct)

Value

a vector with NA remplacing exculded values

OGI.calc *Oxygene Glucose index (OGI)*

Description

take CMRO2 and CMRGlucose to give an OGI

Usage

```
OGI.calc(CMRO2, CMRGluc)
```

Arguments

`CMRO2`	: vector (numeric value) of Cerebral metabolic rate of oxygen in micromol/100g/min
`CMRGluc`	: a vector (numeric value) of CMRGlucose in micromol/100g/min

Details

$$OGI = CMRO2/CMRGlucose$$

Value

OGI

`polyfit` *polynomial fit*

Description

Fit a nth degree polynom

$$(y = C_n X^n + C_{n-1} X^{n-1} + \ldots + C_1 X + C_0)$$

to vectors x and y

Usage

```
polyfit(x, y, order = 1)
```

Arguments

`x`	: a numeric vector
`y`	: a numeric vector (x and y must have the same length)
`order`	: a numeric value for the polynomial degree. default is one.

Value

model formula

Coef: coeficients in ascending order (i.e. C0, C1, C2, . . . , Cn)

R2: Rsquare

`polyval` *polynomial evaluation*

Description

evaluation a nth degree polynom

$$(y = C_n X^n + C_{n-1} X^{n-1} + \ldots + C_1 X + C_0)$$

at given values of x

Usage

```
polyval(x, coef)
```

Arguments

`x`	: a numeric vector
`coef`	: a n dimention vector corresponding to the polynom coefficient (ascending order, i.e. C0, C1, C2, ... , Cn)

Value

y

`roll.funct` *apply a function FUN on a rooling windows of a vector*

Description

apply a function FUN on a rooling windows of a vector

Usage

```
roll.funct(data, width, FUN, size = T, ...)
```

Arguments

`data`	: a numeric vector
`width`	: the size of the rolling window
`FUN`	: the function to apply
`size`	: a logical value indicating if the returned vector have the same length as original data. default is TRUE.
`...`	: additional argument to pass to the FUN

Value

a vector with FUN result

Index