Package ‘LeafGasExchange’

November 2, 2020

Title Modeling photosynthesis at the leaf level

Version 0.0.0.9000

Description Model gas exchanges at the leaf level using a coupled stomatal conductance model (USO) and photosynthesis model (Farquhar) using analytical solutions of the different equations. It is also possible to include leaf energy balance and mesophyll conductance. This package also gathers functions to import data from LICORS 6400 and 6800, fit and display the main types of curves obtained with a gas exchange device: AQ, Aci, Mcurves and simulate data.

License What license it uses

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Depends stats4, bbmle, readxl, stringr, tidyverse, readr, tealeaves,reshape2,ggplot2, BioCro

Imports stats4,
bbmle,
readxl,
stringr,
tidyverse,
readr,
tealeaves,
reshape2,
ggplot2,
BioCro

Remotes ebimodeling/biocro

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Julien Lamour [aut, cre]

Maintainer Julien Lamour <jlamour@bnl.gov>

R topics documented:

`f.A` *Coupled conductance photosynthesis model*

Description

Photosynthesis model at the leaf level using the farquhar equations. The parameters can be defined by the function f.make param and corresponds to the parameters inplemented in different Terrestrial Biosphere Modesl such as ORCHIDEE, JULES, CLM4.5 or FATES

Usage

```
f.A(PFD, cs, Tleaf, Tair, RH, param = f.make.param())
```

Value

List of different variables: - A: Raw assimilation of the leaf in micromol.m-2.s-1 - Ac: Rubisco limitation assimilation of the leaf in micromol.m-2.s-1 - Aj: Electron transport rate assimilation of the leaf in micromol.m-2.s-1 - Ap: TPU rate of the leaf in micromol.m-2.s-1 - Ag: Gross assimilation in micromol.m-2.s-1 - gs: Conductance of the leaf for water vapour in mol m-2 s-1 - ci: Intracellular CO2 concentration in micromol.mol-1 - cc: Mesophyll CO2 concentration in micromol.mol-1 (for the models using mesophyll conductance) - ds: Leaf surface to air vapour pressure deficit in Pa - Trans: Water transpiration in mL m-2 s-1

Examples

```
f.A(PFD=2000,cs=400,Tleaf=273.16+29,Tair=273.16+28,RH=70,param=f.make.param())
```

f.Aci — *Photosynthesis model*

Description

Calculate the assimilation according to Farquhar equations. Contrary to f.A, this function uses intracellular CO2 and not ambiant air CO2

Usage

```
f.Aci(PFD, ci, Tleaf, param = f.make.param())
```

Arguments

param	List of parameters, see f.make.param for details

Value

Assimilation in micromol.m-2.s-1

Examples

```
ci=seq(40,1500,10)
plot(x=ci,y=f.Aci(PFD=2000,ci=ci,Tleaf=300,param=f.make.param())$A)
```

f.arrhenius — *Temperature dependence of Gamma star, Ko, Kc and Rd*

Description

Temperature dependence of Gamma star, Ko, Kc and Rd

Usage

```
f.arrhenius(PRef, Ha, Tleaf, TRef = 298.16, R = 8.314)
```

Arguments

PRef	Value of the parameter at the reference temperature
Ha	Enthalpie of activation in J.mol-1
Tleaf	Temperature of the leaf in Kelvin
TRef	Reference temperature
R	Ideal gas constant

Value

Value of the parameter at the temperature of the leaf

References

VON CAEMMERER, S. (2013), Steady state models of photosynthesis. Plant Cell Environ, 36: 1617-1630. doi:10.1111/pce.12098 Bernacchi, C.J., Singsaas, E.L., Pimentel, C., Portis Jr, A.R. and Long, S.P. (2001), Improved temperature response functions for models of Rubisco-limited photosynthesis. Plant, Cell & Environment, 24: 253-259. doi:10.1111/j.1365-3040.2001.00668.x

Examples

```
plot(x=seq(25,35,0.1),y=f.arrhenius(PRef=1,Ha=46390,Tleaf=seq(273.15+25,273.15+35,0.1),R=8.314),xlab='Tempe
```

`f.arrhenius.inv` *Temperature dependence of Gamma star, Ko, Kc and Rd*

Description

Temperature dependence of Gamma star, Ko, Kc and Rd

Usage

```
f.arrhenius.inv(P, Ha, Tleaf, TRef = 298.16, R = 8.314)
```

Arguments

`P`	Value of the parameter at Tleaf
`Ha`	Enthalpie of activation in J.mol-1
`Tleaf`	Temperature of the leaf in Kelvin
`TRef`	Reference temperature
`R`	Ideal gas constant

Details

Retrieve the value of the parameter at Tref knowing its value at Tleaf

---

`f.AT` *Coupled conductance photosynthesis model and energy balance model*

Description

Coupled conductance photosynthesis model and energy balance model

Usage

```
f.AT(PFD, cs, Tair, RH, wind, precision = 0.1, max_it = 10, param)
```

Arguments

`wind`	Wind speed at the surface of the leaf in m.s-1
`precision`	Precision of the leaf temperature prediction. The resolution of the energy balance coupled with the photosynthesis and stomatal conductance is numerical. The smaller the precision, the longer will be the resolution.
`max_it`	Maximum number of iterations to find the solution
`param`	List of parameters given by f.make.param()

Details

This function allows to calculate the photosynthesis from environmental variables PFD, RH, wind, cs and Tair. The energy balance model is calculated using the package Tealeaves (see reference). The energy balance calculation involves the stomatal conductance and the cuticular conductance. Here the cuticular conductance is considered to be equal to g0 as done in some TBMs even if it is probably a wrong representation.This choice was made to prevent unrealistic energy budgets when the conductance is too low (<= 0) for low light levels.

Value

- A: Raw assimilation of the leaf in micromol.m-2.s-1 - Ag: Gross assimilation in micromol.m-2.s-1 - gs: Conductance of the leaf for water vapour in mol m-2 s-1 - ci: Intracellular CO2 concentration in micromol.mol-1 - cc: Mesophyll CO2 concentration in micromol.mol-1 (for the models using mesophyll conductance) - ds: Leaf surface to air vapour pressure deficit in Pa - Trans: Water transpiration in mL m-2 s-1 - Tleaf: Leaf Temperature in K

References

tealeaves: an R package for modelling leaf temperature using energy budgets. Christopher. D. Muir. bioRxiv 529487; doi: https://doi.org/10.1101/529487

Examples

```
f.AT(PFD=1500,cs=400,Tair=299,wind=2,RH=70,param=f.make.param())
```

`f.canopy.interception` *Wrapper of biocro lightME and sunML function to describe the light levels inside the canopy*

Description

Wrapper of biocro lightME and sunML function to describe the light levels inside the canopy

Usage

```
f.canopy.interception(
  meteo_hourly,
  lat,
  t.d,
  DOY,
  n_layers,
```

```
  Height,
  LAI,
  chi.l = 0.9
)
```

Arguments

meteo_hourly	Hourly weather data frame with at least the column at (air temperature in degree C) tl (leaf temperature in degree C) rh (humidity in pc) and sr the total PAR in micro mol m-2 s-1
lat	Latitude of the canopy to model (see lightME from biocro)
t.d	time of the day (see lightME from biocro)
DOY	Day of Year (see lightME from biocro)
n_layers	Number of layers inside the canopy (max = 50, see sunML from biocro)
Height	Total height of the canopy (see sunML from biocro)
LAI	Total LAI of the cnaopy (see sunML from biocro)
chi.l	Orientation of the leaves (see sunML from biocro)

Examples

```
##Simulation of weather data
meteo_hourly=data.frame(time=0:23,rh=80,at=25,sr=sin(seq(0,pi,pi/23))*2000,tl=25)
meteo_hourly[!meteo_hourly$time%in%7:17,'sr']=0
##Representation of the light interception inside the canopy
canopy=f.canopy.interception(meteo_hourly=meteo_hourly,lat = 9.2801048,t.d = 0:23,DOY = 60,n_layers = 50,Heig
```

---

f.ci.treshold *Intracellular CO2 threshold between electron transport and carboxylation limitations*

Description

Intracellular CO2 threshold between electron transport and carboxylation limitations

Usage

```
f.ci.treshold(PFD, Tleaf, param)
```

Value

Intracellular CO2 such as Wc==Wj

Examples

```
f.ci.treshold(PFD=2000,Tleaf=300,param=f.make.param(VcmaxRef=60,JmaxRef=85))
f.ci.treshold(PFD=2000,Tleaf=300,param=f.make.param(VcmaxRef=70,JmaxRef=85))
```

f.ds — *Leaf water vapour pressure deficit calculation*

Description

This function calculates the leaf water pressure deficit (VPDl or Ds) using the temperature of the leaf, the temperature of the air and its relative humidity

Usage

```
f.ds(Tleaf, Tair, RH)
```

Arguments

`Tleaf`	Temperature of the leaf in Kelvin
`Tair`	Temperature of the air in Kelvin
`RH`	Humidity of the air (0 to 100)

Value

Ds in Pascal

Examples

```
f.ds(Tleaf=273.16 + 30, Tair=273.16+28, RH=70)
```

f.fitting — *Fitting function for photosynthesis datadata (light curve or Aci curve)*

Description

Function to fit model to data. The parameters to fit have to be described in the list Start. All the other parameters of the f.Aci functions have to be in param. If the parameters from Start are repeated in param, the later one will be ignored. This function uses two methods to fit the data. First by minimizing the residual sum-of-squares of the residuals and then by maximizing the likelihood function. The first method is more robust but the second one allows to calculate the confident interval of the parameters.

Usage

```
f.fitting(
  measures,
  id.name = NULL,
  Start = list(JmaxRef = 90, VcmaxRef = 70, RdRef = 1),
  param = f.make.param(),
  modify.init = TRUE,
  do.plot = TRUE,
  type = "Aci"
)
```

Arguments

measures	Data frame of measures obtained from gas exchange analyser with at least the columns Photo, Ci, PARi and Tleaf (in K)
id.name	Name of the colums in measures with the identifier for the curve.
Start	List of parameters to fit with their initial values.
param	See f.make.param() for details.
modify.init	TRUE or FALSE, allows to modify the Start values before fitting the data
do.plot	TRUE or FALSE, plot data and fitted curves ?

Examples

```
##Simulation of a CO2 curve
data=data.frame(Tleaf=rep(300,20),
Ci=seq(40,1500,75),PARi=rep(2000,20),Photo=f.Aci(PFD=2000,Tleaf=300,ci=seq(40,1500,75),
param=f.make.param(TBM='FATES'))$A+rnorm(n = 20,mean = 0,sd = 0.5))

f.fitting(measures=data,id.name=NULL,Start=list(JmaxRef=90,VcmaxRef=70,RdRef=1),param=f.make.param(TBM='FA
```

---

f.GPP *Canopy scale GPP calculation*

Description

Generic function to calculate the GPP within a forest (Here GPP = sum of Anet at the canopy level, so it takes into account the leaf mitochondrial respiration)

Usage

```
f.GPP(
  TBM,
  meteo_hourly,
  Vcmax_Profile,
  Jmax_Profile,
  Rd_Profile,
  Tp_Profile,
  g0_Profile,
  g1_Profile,
  gsmin,
  canopy,
  Patm = 100,
  ...
)
```

Arguments

TBM	Specific TBM to use (ORCHIDEE, CLM4.5, FATES or JULES)
meteo_hourly	Hourly weather data frame with at least the column at (air temperature in degree C) tl (leaf temperature in degree C) rh (humidity in pc) and sr the total PAR in micro mol m-2 s-1

Vcmax_Profile	Vector of the values of Vcmax at the reference temperature at each layer of the canopy
Jmax_Profile	Vector of the values of Jmax at the reference temperature at each layer of the canopy
Rd_Profile	Vector of the values of Rd at the reference temperature at each layer of the canopy
Tp_Profile	Vector of the values of Tp at the reference temperature at each layer of the canopy
g0_Profile	Vector of the values of g0 at the reference temperature at each layer of the canopy
g1_Profile	Vector of the values of g1 at the reference temperature at each layer of the canopy
gsmin	Minimum stomatal conductance for water to consider. This value will be used as the minimum conductance value to avoid 0 and negative values obtained from the coupled assimilation and conductance models
canopy	Description of the canopy interception (see canopy_interception function)
Patm	Atmospheric pressure (used to calculate the transpiration)
...	Other parameters of the photosynthetic model, without gradients, for example curvature factor, quantum yield.. see the help of f.make.param()

Examples

```
## Simulation of photosynthetic gradients
LAI=seq(0,6.2,6.2/49)
Vcmax=f.VcmaxRef.LAI(kn=0.11,LAI=LAI,Vcmax0=70)
Jmax=1.7*Vcmax; Tp=1/5*Vcmax; Rd=0.03*Vcmax
##Simulation of weather data
meteo_hourly=data.frame(time=0:23,rh=80,at=25,sr=sin(seq(0,pi,pi/23))*2000,tl=25)
meteo_hourly[!meteo_hourly$time%in%7:17,'sr']=0
##Representation of the light interception inside the canopy
canopy=f.canopy.interception(meteo_hourly=meteo_hourly,lat = 9.2801048,t.d = 0:23,DOY = 60,n_layers = 50,Heig
GPP_sc1=f.GPP(TBM = "FATES",meteo_hourly = meteo_hourly,Vcmax_Profile = Vcmax,
Jmax_Profile =Jmax ,Rd_Profile =Rd ,Tp_Profile = Tp,
g0_Profile = rep(0.02,length(Vcmax)),g1_Profile = rep(4,length(Vcmax)),canopy=canopy,gsmin = 0.01)
```

---

f.gs *Conductance model for stomatal conductance to water vapour*

Description

Semi-empirical model of the leaf conductance to water vapour

Usage

```
f.gs(A, cs, ds = NULL, RH = NULL, g0, g1, power = 0.5, model = "USO")
```

Arguments

`A`	Net assimilation in micromol.m-2.s-1, i-e, the assimilation in presence of respiration
`cs`	CO2 at the surface of the leaf in ppm
`ds`	Leaf surface to air vapour pressure deficit in Pa
`RH`	Humidity at the surface of the leaf (0 - 100). ds or RH as to be specified
`g0`	Constant of the USO model, representing the conductance when A is 0, in mol.m-2.s-1
`g1`	Slope parameter, between 1.14 and 3.58 KPa^0.5 (Wu et al., 2019)
`power`	Power of the VPDl in USO model. By default is is 0.5 as in Medlin publication
`model`	Stomatal model ("USO", "USO_simpl" or "BWB")

Value

This function returns the optimal stomatal conductance to water vapour in mol.m-2.s-1

References

Medlyn, B.E., Duursma, R.A., Eamus, D., Ellsworth, D.S., Colin Prentice, I., Barton, C.V.M., Crous, K.Y., de Angelis, P., Freeman, M. and Wingate, L. (2012), Reconciling the optimal and empirical approaches to modelling stomatal conductance. Glob Change Biol, 18: 3476-3476. doi:10.1111/j.1365-2486.2012.02790.x Wu, J, Serbin, SP, Ely, KS, et al. The response of stomatal conductance to seasonal drought in tropical forests. Glob Change Biol. 2020; 26: 823– 839. https://doi.org/10.1111/gcb.14820

Examples

```
gs=f.gs(A=30,cs=400,ds=1500,g0=0.01,g1=2,power=0.5)
```

`f.gsmax` *Maximum theoretical stomatal conductance*

Description

Maximum theoretical stomatal conductance

Usage

```
f.gsmax(
  Sarea = 0.78,
  Sdensity = 400,
  Sdepth = 5,
  Diffusivity = 0.282/1000,
  mvair = 24.5/1000
)
```

Arguments

`Sarea`	Maximum area of the aperture of stomata when open (microm2)
`Sdensity`	Number of stomata per mm2 of leaf
`Sdepth`	Stomatal pore depth (micro m)
`Diffusivity`	Diffusivity of water vapor in air (m2 s-1)
`mvair`	Molar volume of air (m3 mol-1)

Details

This function calculates the maximum theoretical conductance value according to morphological data and the physics of diffusion through pores. It follows the equation from Franks and Berling 2009.

Value

Maximum stomatal conductance to water vapour in mol m-2 s-1

References

Franks PJ, Beerling DJ. Maximum leaf conductance driven by CO2 effects on stomatal size and density over geologic time. Proc Natl Acad Sci U S A. 2009;106(25):10343-10347. doi:10.1073/pnas.0904209106

Examples

```
## The density of stomata is around 400 stomata.mm-2 in the tropical species.
## The length of the stomata is around 20 micro m. Following Franks and Beerling 2009 we can estimate the
## Sarea of the stomata: pi*(20/4*10^-6)^2 and the Sdepth: 20*10^-6/4
f.gsmax(Sarea=0.78,Sdensity=400,Sdepth=5)
```

`f.gsmin` *Calculation of the minimal conductance given by a particular coupled conductance and photosynthesis model*

Description

The minimal conductance of a model depends on the parameters of the model (ie g0 and g1) but also on the minimum A value, which corresponds to the dark respiration. Knowing the minimal conductance is important because the conductance can become negative and lead to unrealistic values in photosynthesis models

Usage

```
f.gsmin(
  RdRef = 0.825,
  RdHa = 46390,
  RdHd = 150650,
  RdS = 490,
  Tleaf = 300,
  cs = 400,
  ds = 1000,
```

```
  g0 = 0.02,
  g1 = 4.1,
  power = 0.5,
  model = "USO"
)
```

Arguments

RdRef	Respiration value at the reference temperature
RdHa	Energie of activation for Rd in J.mol-1
g0	Constant of the USO model, representing the conductance when A is 0, in mol.m-2.s-1
g1	Slope parameter, between 1.14 and 3.58 KPa^0.5 (Wu et al., 2019)
power	Power of VPDl in USO model. By default power=0.5 as in Medlyn article

Value

Minimal conductance

Examples

```
gs_min=f.gsmin(RdRef= 0.825,RdHa= 46390,RdHd=150650,RdS=490,Tleaf=300,cs=400,ds=1000,g0=0.02,g1=4.1,power=0
```

`f.import_licor6400` *Import Licor 6400 file*

Description

This functions allows to import the text file produced by LICOR as a data.frame

Usage

```
f.import_licor6400(
  file,
  column_display = c("Photo", "Cond", "PARi", "Ci", "Leaf_Barcode", "Species",
    "Tree Canopy", "Age", "file")
)
```

Arguments

file	File to import by the function
column_display	The first lines of the file which are part of this list are displayed by this function after being imported.

Value

dataframe

References

Adapted from http://www.ericrscott.com/2018/01/17/li-cor-wrangling/

f.import_licor6800 *Import Licor 6800 excel file*

Description

This functions allows to import the excel file produced by LICOR as a data.frame. The files have to be open in Excel and saved before using his function so the result of the formula are calculated. The formula are sotred into the Excel file but not computed until the file is open.

Usage

```
f.import_licor6800(
  file,
  column_display = c("A", "gsw", "Qin", "Ci", "Species", "Canopy", "Pheno_Age",
    "Barcode", "file")
)
```

Arguments

`file`	File to import by the function
`column_display`	The first lines of the file which are part of this list are displayed by this function after being imported.

Value

dataframe

References

Adapted from http://www.ericrscott.com/2018/01/17/li-cor-wrangling/

f.logistic *Logistic function*

Description

This function takes it values in -Inf;+Inf and returns values in 0;1. It is the inverse function of f.logit, ie f.logistic(logit(x))=x

Usage

```
f.logistic(x)
```

Details

f.logistic(x)=1/(1+exp(-x)) if x<0, = exp(x)/(1+exp(x)) if x<=0

Examples

```
plot(x=seq(-10,10,0.1),y=f.logistic(x=seq(-10,10,0.1)))
```

f.logit *Function logit*

Description

This function takes it values in 0;1- and returns values in Inf;+Inf. It is the inverse function of f.logistic

Usage

```
f.logit(x)
```

Examples

```
plot(x=seq(0,1,0.01),y=f.logit(x=seq(0,1,0.01)))
```

f.make.param *Photosynthesis and stomata model parameters*

Description

Function to create a list of parameters to be used in most of the functions of this package. Depending on the function, all the parameters are not used. For example go and g1 are not used in f.Aci. The parameters from different TBM are implemented and can be chosen by selecting a TBM

Usage

```
f.make.param(
  TBM = "FATES",
  R = NA,
  O2 = NA,
  TRef = NA,
  Patm = NA,
  JmaxRef = NA,
  JmaxHa = NA,
  JmaxHd = NA,
  JmaxS = NA,
  VcmaxRef = NA,
  VcmaxHa = NA,
  VcmaxHd = NA,
  VcmaxS = NA,
  VcmaxQ10 = NA,
  Tlow = NA,
  Tup = NA,
  TpRef = NA,
  TpHa = NA,
  TpHd = NA,
  TpS = NA,
  thetacj = NA,
```

```
    thetaip = NA,
    RdRef = NA,
    RdHa = NA,
    RdHd = NA,
    RdS = NA,
    KcRef = NA,
    KcHa = NA,
    KcQ10 = NA,
    KoRef = NA,
    KoHa = NA,
    KoQ10 = NA,
    GstarRef = NA,
    GstarHa = NA,
    TauRef = NA,
    TauQ10 = NA,
    abso = NA,
    aQY = NA,
    Theta = NA,
    model.gs = NA,
    g0 = NA,
    g1 = NA,
    power = NA,
    gmRef = NA,
    gmS = NA,
    gmHa = NA,
    gmHd = NA
)
```

Arguments

`TBM`	Type of model (FATES, ORCHIDEE, CLM4.5 or JULES). Default is FATES
`R`	Ideal gas constant
`O2`	O2 concentration in ppm
`TRef`	Reference temperature for Kc, Ko, Rd,GammaStar Vcmax, Jmax
`Patm`	Atmospheric pressure in kPa
`JmaxRef`	Maximum electron transport rate in micromol.m-2.s-1
`JmaxHa`	Energy of activation for Jmax in J.mol-1
`JmaxHd`	Energy of desactivation for Jmax in J.mol-1
`JmaxS`	Entropy term for Jmax in J.mol-1.K-1
`VcmaxRef`	Maximum rate of Rubisco for carboxylation micromol.m-2.s-1
`VcmaxHa`	Energy of activation for Vcmax in J.mol-1
`VcmaxHd`	Energy of desactivation for Vcmax in J.mol-1
`VcmaxS`	Entropy term for Vcmax in J.mol-1.K-1
`RdRef`	Respiration value at the reference temperature
`RdHa`	Energie of activation for Rd in J.mol-1
`KcRef`	Michaelis-Menten constant of Rubisco for CO2 at the reference temperature in micromol.mol-1
`KcHa`	Energy of activation for Kc in J.mol-1

KoRef	ichaelis-Menten constant of Rubisco for CO2 at the reference temperature in milimol.mol-1
KoHa	Energy of activation for Ko in J.mol-1
GstarRef	CO2 compensation point in absence of respiration in micromol.mol-1
GstarHa	Enthalpie of activation for Gstar in J.mol-1
abso	Absorptance of the leaf in the photosynthetic active radiation wavelenghts
aQY	Apparent quantum yield
Theta	Theta is the empirical curvacture factor for the response of J to PFD. It takes its values between 0 and 1.
model.gs	Type of conductance model (USO, USO_simpl,BWB)
g0	Constant of the USO model, representing the conductance when A is 0, in mol.m-2.s-1
g1	Slope parameter, between 1.14 and 3.58 KPa^0.5 (Wu et al., 2019)
power	Power of VPDl in USO model. By default power=0.5 as in Medlyn article
gmRef	Mesophyll conductance at Tref (25 deg C) mol m-2 s-1
gmS	Entropy term for gm J K-1 mol-1
gmHa	Energy of activation for gm in J.mol-1
gmHd	Energy of deactivation for gm in J.mol-1

Details

The call of this function is made using f.make.param(). If a parameter is modified for example writting f.make.param(VcmaxRef=10), this function will return all the default parameters from FATES TBM with VcmaxRef = 10 instead of its default value

Value

List of parameters that can be used in f.A

References

Bernacchi, C.J., Singsaas, E.L., Pimentel, C., Portis Jr, A.R. and Long, S.P. (2001), Improved temperature response functions for models of Rubisco-limited photosynthesis. Plant, Cell & Environment, 24: 253-259. doi:10.1111/j.1365-3040.2001.00668.x CLM4.5: http://www.cesm.ucar.edu/models/cesm2/land/CL
ORCHIDEE: https://forge.ipsl.jussieu.fr/orchidee/wiki/Documentation/OrchideeParameters AND https://agupubs.online
JULES: https://www.geosci-model-dev.net/4/701/2011/gmd-4-701-2011.pdf FATES: https://fates-docs.readthedocs.io/en/latest/fates_tech_note.html# Medlyn, B.E., Duursma, R.A., Eamus, D., Ellsworth, D.S., Colin Prentice, I., Barton, C.V.M., Crous, K.Y., de Angelis, P., Freeman, M. and Wingate, L. (2012), Reconciling the optimal and empirical approaches to modelling stomatal conductance. Glob Change Biol, 18: 3476-3476. doi:10.1111/j.1365-2486.2012.02790.x

Examples

```
param1=f.make.param(TBM='FATES',JmaxRef=100,VcmaxRef=60,RdRef=1,TpRef=10)
 param2=f.make.param(TBM='CLM4.5',JmaxRef=100,VcmaxRef=60,RdRef=1,TpRef=10)
 f.A(PFD=1500,cs=400,Tleaf=300,Tair=299,RH=70,param=param1)
 f.A(PFD=1500,cs=400,Tleaf=300,Tair=299,RH=70,param=param2)
```

`f.modified.arrhenius` *Temperature dependence of Jmax and Vcmax*

Description

The temperature dependence of the photosynthetic parameters Vcmax, the maximum catalytic rate of the enzyme Rubisco, and Jmax, the maximum electron transport rate is modelled by a modified Arrehenius equation. It is modified to account for decreases in each parameter at high temperatures.

Usage

```
f.modified.arrhenius(PRef, Ha, Hd, s, Tleaf, TRef = 298.16, R = 8.314)
```

Arguments

`PRef`	Value of the parameter, here Vcmax or Jmax, at the reference temperature in micromol.m-2.s-1
`Ha`	Energy of activation in J.mol-1
`Hd`	Energy of desactivation in J.mol-1
`s`	Entropy term in J.mol-1.K-1
`Tleaf`	Temperature of the leaf in Kelvin
`TRef`	Reference temperature
`R`	Ideal gas constant

Value

Value of the parameter Jmax or Vcmax at a given temperature

References

Leuning, R. (2002), Temperature dependence of two parameters in a photosynthesis model. Plant, Cell & Environment, 25: 1205-1210. doi:10.1046/j.1365-3040.2002.00898.x

Examples

```
plot(x=seq(25,35,0.1),y=f.modified.arrhenius(PRef=50,Ha=73637,Hd=149252,s=486,Tleaf=seq(273.15+25,273.15+35
```

`f.modified.arrhenius.inv` *Temperature dependence of Jmax and Vcmax*

Description

Retrieve the reference temperature value of a parameter knowing its value at Tleaf

Usage

```
f.modified.arrhenius.inv(P, Ha, Hd, s, Tleaf, TRef = 298.16, R = 8.314)
```

Arguments

P	Value of the parameter, here Vcmax or Jmax, at the leaf temperature in micromol.m-2.s-1
Ha	Energy of activation in J.mol-1
Hd	Energy of desactivation in J.mol-1
s	Entropy term in J.mol-1.K-1
Tleaf	Temperature of the leaf in Kelvin
TRef	Reference temperature
R	Ideal gas constant

---

f.plot *Plot data and model*

Description

Plot a generic graphic with observed data and predictions. Be careful to sort the data.frame beforehand.

Usage

```
f.plot(measures = NULL, list_legend, param, name = "", type = "Aci")
```

Arguments

measures	Data frame obtained from CO2 or light curve with at least columns Photo, Ci, PARi and Tleaf Data frame obtained from CO2 or light curve with at least columns Photo, Ci, PARi and Tleaf
list_legend	Named list where the name and values will appear in the legend
name	Name of the curve to be displayed
type	Type of the curve to plot (light curve: Aq or CO2 curve Aci)

Value

Plot a figure

Examples

```
param=f.make.param()
Photo=f.Aci(PFD=2000,Tleaf=300,ci=seq(40,1500,50),param=param)$A+rnorm(n = 30,mean = 0,sd = 0.5)
data=data.frame(Tleaf=rep(300,30),Ci=seq(40,1500,50),PARi=rep(2000,30),Photo=Photo)
f.plot(measures=data,param=param,list_legend=param['VcmaxRef'],name='Example 01',type='Aci')
```

`f.Q10` *Temperature dependence of photosynthetic parameters*

Description

Temperature dependence of photosynthetic parameters

Usage

```
f.Q10(Pref, Q10, Tleaf, TRef)
```

Arguments

`TRef`	Reference temperature for Kc, Ko, Rd,GammaStar Vcmax, Jmax

Details

This equation is used in JULES TBM model

Value

Value of the photosynthetic parameter at the specified leaf temperature

References

Clark, D. B., Mercado, L. M., Sitch, S., Jones, C. D., Gedney, N., Best, M. J., . Cox, P. M. (2011). The Joint UK Land Environment Simulator (JULES), model description - Part 2: Carbon fluxes and vegetation dynamics. Geoscientific Model Development, 4(3), 701-722. doi:10.5194/gmd-4-701-2011

`f.Q10.modified` *Temperature dependence of photosynthetic parameters*

Description

Temperature dependence of photosynthetic parameters

Usage

```
f.Q10.modified(Pref, Q10, Tleaf, TRef, Tlow, Tup)
```

Arguments

`TRef`	Reference temperature for Kc, Ko, Rd,GammaStar Vcmax, Jmax

Details

This equation is used in JULES TBM model

Value

Value of the photosynthetic parameter at the specified leaf temperature

References

Clark, D. B., Mercado, L. M., Sitch, S., Jones, C. D., Gedney, N., Best, M. J., . Cox, P. M. (2011). The Joint UK Land Environment Simulator (JULES), model description - Part 2: Carbon fluxes and vegetation dynamics. Geoscientific Model Development, 4(3), 701-722. doi:10.5194/gmd-4-701-2011

`f.smooth` *Smoothing functions between photosynthesis limitations (for example between rubisco carboxylation and light limitation)*

Description

Smoothing functions between photosynthesis limitations (for example between rubisco carboxylation and light limitation)

Usage

```
f.smooth(A1, A2, theta)
```

Arguments

`theta`	Smoothing factor

Value

Smoothed value

Examples

```
A1= seq(0,20,1)
A2= seq(9,11,2/20)
Asmooth=f.smooth(A1=A1,A2=A2,theta=0.99)
plot(A1,type='l')
lines(A2)
lines(Asmooth,col='blue')
```

`f.VcmaxRef.LAI` *Gradients of photosynthetic parameters*

Description

Several versions of gradients can be found in the litterature, see for example Lloyd et al. 2010 (Fig. 10 and equation A2), but also the equation A14 from Krinner et al. 2005 and the equation 33 from Clark et al. 2011 The simpler model describing the gradients is Vcmax(LAI)=Vcmax0 x exp(-kn x LAI) with Vcmax0 Vcmax at the top of the canopy kn can be also calculated as a function of Vcmax0: kn=exp(alpha x Vcmax0+beta) If kn is NULL, then the function will use the default alpha and beta to calculate kn. If, on the contrary, kn is given, this specific one will be used to calculate the gradients. Krinner et al use a slightly different version of this equation with the parameter lambda: Vcmax(LAI)=Vcmax0 x (1-lambda x (1-exp(-kn*LAI))). The previous equation is a particular case of this one for lambda = 1

Usage

```
f.VcmaxRef.LAI(
  alpha = 0.00963,
  beta = -2.43,
  Vcmax0 = 50,
  LAI = 0:8,
  kn = NULL,
  lambda = 1
)
```

Arguments

`alpha`	Slope of the relationship between Vcmax0 and log(kn), see Lloyd et al. 2010
`beta`	Intercept of the relationship between Vcmax0 and log(kn), see Lloyd et al. 2010
`Vcmax0`	Vcmax at 25 degree C at the top of the canopy
`LAI`	Vector of Leaf Area Index (or depth within the canopy see Clark et al. 2011)
`kn`	Exponential decrease
`lambda`	Asymptot of the decrease (see Krinner et al. 2005)

Value

Vector of Vcmax at the different LAI specified in the call of the function

References

Krinner, G., Viovy, N., de Noblet-Ducoudr?, N., Og?e, J., Polcher, J., Friedlingstein, P., . Prentice, I. C. (2005). A dynamic global vegetation model for studies of the coupled atmosphere-biosphere system. Global Biogeochemical Cycles, 19(1). doi:10.1029/2003gb002199 Clark, D. B., Mercado, L. M., Sitch, S., Jones, C. D., Gedney, N., Best, M. J., . Cox, P. M. (2011). The Joint UK Land Environment Simulator (JULES), model description - Part 2: Carbon fluxes and vegetation dynamics. Geoscientific Model Development, 4(3), 701-722. doi:10.5194/gmd-4-701-2011 Lloyd, J., Pati?o, S., Paiva, R. Q., Nardoto, G. B., Quesada, C. A., Santos, A. J. B., . Mercado, L. M. (2010). Optimisation of photosynthetic carbon gain and within-canopy gradients of associated foliar traits for Amazon forest trees. Biogeosciences, 7(6), 1833-1859. doi:10.5194/bg-7-1833-2010

Examples

```
LAI=seq(0,6.2,6.2/49)
Vcmax=f.VcmaxRef.LAI(kn=0.11,LAI=LAI,Vcmax0=70)
Vcmax2=f.VcmaxRef.LAI(kn=0.11,LAI=LAI,Vcmax0=70,lambda=0.7)
plot(Vcmax)
lines(Vcmax2)
```

Index