Package 'plantecophys'

March 13, 2019

Type Package

Title Modelling and Analysis of Leaf Gas Exchange Data

Version 1.4-4

Description Coupled leaf gas exchange model, A-Ci curve simulation and fitting, Ball-Berry stomatal conductance models, leaf energy balance using Penman-Monteith, Cowan-Farquhar optimization, humidity unit conversions. See Duursma (2015) <doi:10.1371/journal.pone.0143346>.

URL `https://www.bitbucket.org/remkoduursma/plantecophys`

BugReports `https://www.bitbucket.org/remkoduursma/plantecophys/issues`

Depends R (>= 3.3.0)

Suggests nlstools, testthat, knitr, rmarkdown, DT

License GPL

LazyData yes

RoxygenNote 6.1.1

VignetteBuilder knitr

NeedsCompilation no

Author Remko Duursma [aut, cre]

Maintainer Remko Duursma <`remkoduursma@gmail.com`>

Repository CRAN

Date/Publication 2019-03-13 09:20:03 UTC

R topics documented:

`plantecophys-package` *Modelling and Analysis of Leaf Gas Exchange Data*

Description

Coupled leaf gas exchange model, A-Ci curve simulation and fitting, Ball-Berry stomatal conductance models, leaf energy balance using Penman-Monteith, Cowan-Farquhar optimization, humidity unit conversions. See Duursma (2015) <doi:10.1371/journal.pone.0143346>.

Details

The DESCRIPTION file:

Package:	plantecophys
Type:	Package
Title:	Modelling and Analysis of Leaf Gas Exchange Data
Version:	1.4-4
Authors@R:	person("Remko", "Duursma", role = c("aut", "cre"),email = "remkoduursma@gmail.com")
Description:	Coupled leaf gas exchange model, A-Ci curve simulation and fitting, Ball-Berry stomatal conductance mo
URL:	https://www.bitbucket.org/remkoduursma/plantecophys
BugReports:	https://www.bitbucket.org/remkoduursma/plantecophys/issues
Depends:	R (>= 3.3.0)
Suggests:	nlstools, testthat, knitr, rmarkdown, DT
License:	GPL
LazyData:	yes
RoxygenNote:	6.1.1
VignetteBuilder:	knitr
Author:	Remko Duursma [aut, cre]
Maintainer:	Remko Duursma <remkoduursma@gmail.com>

Index of help topics:

```
AciC4                   C4 Photosynthesis
FARAO                   FARquhar And Opti
Photosyn                Coupled leaf gas exchange model
PhotosynEB              Coupled leaf gas exchange model with energy
```

	balance
PhotosynTuzet	Coupled leaf gas exchange model with Tuzet stomatal conductance
RHtoVPD	Conversions between relative humidity, vapour pressure deficit and dewpoint
acidata1	An example A-Ci curve
findCiTransition	Calculate transition points for fitted A-Ci curves
fitBB	Fit Ball-Berry type models of stomatal conductance
fitBBs	Fit Ball-Berry type models of stomatal conductance to many groups at once
fitaci	Fit the Farquhar-Berry-von Caemmerer model of leaf photosynthesis
fitacis	Fit multiple A-Ci curves at once
manyacidat	An example dataset with multiple A-Ci curves
plantecophys-package	Modelling and Analysis of Leaf Gas Exchange Data

The following functions are the main tools in **plantecophys**:

1. `fitaci` (and `fitaci`) fits A-Ci curves to data.
2. `Photosyn` can be used to simulate A-Ci curves (or `Aci`), and simulate from a coupled leaf gas exchange model.
3. `fitBB` fits Ball-Berry-type stomatal conductance models to data.
4. `FARAO` is an implementation of a numeric solution to Cowan-Farquhar optimization of stomatal conductance.
5. `RHtoVPD` converts relative humidity to vapour pressure deficit (and more similar functions on that help page).

The package also includes the following example datasets:

1. `acidata1` A dataset with a single A-Ci curve.
2. `manyacidat` A dataset with many A-Ci curves.

Author(s)

Remko Duursma

Maintainer: Remko Duursma

References

Duursma, R.A., 2015. Plantecophys - An R Package for Analysing and Modelling Leaf Gas Exchange Data. PLoS ONE 10, e0143346. doi:10.1371/journal.pone.0143346

`AciC4` *C4 Photosynthesis*

Description

An implementation of the A-Ci curve for C4 plants, based on von Caemmerer et al. (2000)

Usage

```
AciC4(Ci, PPFD = 1500, Tleaf = 25, VPMAX25 = 120, JMAX25 = 400,
  Vcmax = 60, Vpr = 80, alpha = 0, gbs = 0.003, O2 = 210,
  x = 0.4, THETA = 0.7, Q10 = 2.3, RD0 = 1, RTEMP = 25,
  TBELOW = 0, DAYRESP = 1, Q10F = 2, FRM = 0.5, ...)
```

Arguments

`Ci`	Intercellular CO2 concentration (ppm)
`PPFD`	Photosynthetic photon flux density (mu mol m-2 s-1)
`Tleaf`	Leaf temperature (C)
`VPMAX25`	The maximum rate of PEP carboxylation (mu mol m-2 s-1)
`JMAX25`	Maximum electron transport rate (at 25C)
`Vcmax`	Maximum rate of carboxylation (mu mol m-2 s-1) (at 25C)
`Vpr`	PEP regeneration (mu mol m-2 s-1)
`alpha`	Fraction of PSII activity in the bundle sheath (-)
`gbs`	Bundle sheath conductance (mol m-2 s-1)
`O2`	Mesophyll O2 concentration
`x`	Partitioning factor for electron transport
`THETA`	Shape parameter of the non-rectangular hyperbola
`Q10`	T-dependence parameter for Michaelis-Menten coefficients.
`RD0`	Respiration at base temperature (RTEMP) (mu mol m-2 s-1)
`RTEMP`	Base leaf temperature for respiration (C)
`TBELOW`	Below this T, respiration is zero.
`DAYRESP`	Fraction respiration in the light vs. that measured in the dark
`Q10F`	T-dependence parameter of respiration
`FRM`	Fraction of day respiration that is mesophyll respiration (Rm)
`...`	Further arguments (currently ignored).

Details

Note that the temperature response parameters have been hardwired in this function, and are based on von Caemmerer (2000).

Note that it is not (yet) possible to fit this curve to observations of photosynthesis (see `fitaci` to fit the C3 model of photosynthesis).

Author(s)

Rhys Whitley

References

Caemmerer, S.V., 2000. Biochemical Models of Leaf Photosynthesis. Csiro Publishing.

Examples

```
# Simulate a C4 A-Ci curve.
aci <- AciC4(Ci=seq(5,600, length=101))
with(aci, plot(Ci, ALEAF, type='l', ylim=c(0,max(ALEAF))))
```

`acidata1` *An example A-Ci curve*

Description

CO2 response of leaf photosynthesis, as measured with a Licor6400.

Format

CO2S CO2 concentration in cuvette (ppm)

Ci Intercellular CO2 concentration (ppm)

Tleaf Leaf temperature (deg C)

Photo Net photosynthesis rate (mu mol m-2 s-1)

`FARAO` *FARquhar And Opti*

Description

The numerical solution of the optimal stomatal conductance model, coupled with the Farquhar model of photosynthesis. The model of Medlyn et al. (2011) is an approximation to this full numeric solution.

Usage

```
FARAO(lambda = 0.002, Ca = 400, VPD = 1, photo = c("BOTH", "VCMAX",
  "JMAX"), energybalance = FALSE, C4 = FALSE, Tair = 25, Wind = 2,
  Wleaf = 0.02, StomatalRatio = 1, LeafAbs = 0.86, ...)

FARAO2(lambda = 0.002, Ca = 400, energybalance = FALSE, ...)
```

Arguments

`lambda`	The marginal cost of water (mol mol-1)
`Ca`	The CO2 concentration.
`VPD`	Vapor pressure deficit (kPa)
`photo`	Which photosynthesis rate should stomata respond to? Defaults to 'BOTH', i.e. the minimum of Vcmax and Jmax limited rates.
`energybalance`	If TRUE (Default = FALSE), calculates leaf temperature from energy balance (and its effects on photosynthesis as well as leaf transpiration), using `PhotosynEB`.
`C4`	If TRUE, uses the C4 photosynthesis routine (`AciC4`)
`Tair`	Air temperature (deg C)
`Wind`	Wind speed (m s-1) (only used if energybalance=TRUE)
`Wleaf`	Leaf width (m) (only used if energybalance=TRUE)
`StomatalRatio`	The stomatal ratio (see `PhotosynEB`) (only used if energybalance=TRUE)
`LeafAbs`	Leaf absorptance (see `PhotosynEB`) (only used if energybalance=TRUE)
`...`	All other parameters are passed to `Aci`

Details

This model finds the Ci that maximizes A - lambda*E (Cowan & Farquhar 1977, see also Medlyn et al. 2011). The new function FARAO2 is a much simpler (and probably more stable) implementation, based on Buckley et al. 2014 (P,C&E). Both functions are provided, as FARAO has a few more options than FARAO2, at the moment.

Author(s)

Remko Duursma

References

Buckley, T.N., Martorell, S., Diaz-Espejo, A., Tomas, M., Medrano, H., 2014. Is stomatal conductance optimized over both time and space in plant crowns? A field test in grapevine (Vitis vinifera). Plant Cell Environ doi:10.1111/pce.12343

Cowan, I. and G.D. Farquhar. 1977. Stomatal function in relation to leaf metabolism and environment. Symposia of the Society for Experimental Biology. 31:471-505.

Medlyn, B.E., R.A. Duursma, D. Eamus, D.S. Ellsworth, I.C. Prentice, C.V.M. Barton, K.Y. Crous, P. De Angelis, M. Freeman and L. Wingate. 2011. Reconciling the optimal and empirical approaches to modelling stomatal conductance. Global Change Biology. 17:2134-2144.

`findCiTransition` *Calculate transition points for fitted A-Ci curves*

Description

Calculates the Ci at the transition points between Ac & Aj (point 1), and Aj and Ap (point 2). The latter is not NA only when TPU was estimated (and estimable), see `fitaci`, argument `fitTPU`.

Usage

```
findCiTransition(object, ...)
```

Arguments

`object`	Either an object returned by `fitaci`, or a copy of the `Photosyn` function.
`...`	Further arguments passed to the `Photosyn` function.

Details

This function is also used by `fitaci`, the results are stored in elements `Ci_transition` and `Ci_transition2`.

`fitaci` *Fit the Farquhar-Berry-von Caemmerer model of leaf photosynthesis*

Description

Fits the Farquhar-Berry-von Caemmerer model of photosynthesis to measurements of photosynthesis and intercellular CO_2 concentration (Ci). Estimates Jmax, Vcmax, Rd and their standard errors. A simple plotting method is also included, as well as the function `fitacis` which quickly fits multiple A-Ci curves (see its help page). Temperature dependencies of the parameters are taken into account following Medlyn et al. (2002), see `Photosyn` for more details.

Usage

```
fitaci(data, varnames = list(ALEAF = "Photo", Tleaf = "Tleaf", Ci = "Ci",
  PPFD = "PARi", Rd = "Rd"), Tcorrect = TRUE, Patm = 100,
  citransition = NULL, quiet = FALSE, startValgrid = TRUE,
  fitmethod = c("default", "bilinear", "onepoint"),
  algorithm = "default", fitTPU = FALSE, alphag = 0, useRd = FALSE,
  PPFD = NULL, Tleaf = NULL, alpha = 0.24, theta = 0.85,
  gmeso = NULL, EaV = 82620.87, EdVC = 0, delsC = 645.1013,
  EaJ = 39676.89, EdVJ = 2e+05, delsJ = 641.3615, GammaStar = NULL,
  Km = NULL, id = NULL, ...)

## S3 method for class 'acifit'
```

```
plot(x, what = c(”data”, ”model”, ”none”),
  xlim = NULL, ylim = NULL, whichA = c(”Ac”, ”Aj”, ”Amin”, ”Ap”),
  add = FALSE, pch = 19, addzeroline = TRUE, addlegend = !add,
  legendbty = ”o”, transitionpoint = TRUE, linecols = c(”black”,
  ”blue”, ”red”), lwd = c(1, 2), lty = 1, ...)
```

Arguments

`data`	Dataframe with Ci, Photo, Tleaf, PPFD (the last two are optional). For `fitacis`, also requires a grouping variable.
`varnames`	List of names of variables in the dataset (see Details).
`Tcorrect`	If TRUE, Vcmax and Jmax are corrected to 25C. Otherwise, Vcmax and Jmax are estimated at measurement temperature. **Warning** : since package version 1.4, the default parameters have been adjusted (see Details).
`Patm`	Atmospheric pressure (kPa)
`citransition`	If provided, fits the Vcmax and Jmax limited regions separately (see Details).
`quiet`	If TRUE, no messages are written to the screen.
`startValgrid`	If TRUE (the default), uses a fine grid of starting values to increase the chance of finding a solution.
`fitmethod`	Method to fit the A-Ci curve. Either ’default’ (Duursma 2015), ’bilinear’ (See Details), or ’onepoint’ (De Kauwe et al. 2016).
`algorithm`	Passed to `nls`, sets the algorithm for finding parameter values.
`fitTPU`	Logical (default FALSE). Attempt to fit TPU limitation (fitmethod set to ’bilinear’ automatically if used). See Details.
`alphag`	When estimating TPU limitation (with `fitTPU`), an additional parameter (see Details).
`useRd`	If Rd provided in data, and useRd=TRUE (default is FALSE), uses measured Rd in fit. Otherwise it is estimated from the fit to the A-Ci curve.
`PPFD`	Photosynthetic photon flux density (’PAR’) (mu mol m-2 s-1)
`Tleaf`	Leaf temperature (degrees C)
`alpha`	Quantum yield of electron transport (mol mol-1)
`theta`	Shape of light response curve.
`gmeso`	Mesophyll conductance (mol m-2 s-1 bar-1). If not NULL (the default), Vcmax and Jmax are chloroplastic rates.
`EaV, EdVC, delsC`	Vcmax temperature response parameters
`EaJ, EdVJ, delsJ`	Jmax temperature response parameters
`Km, GammaStar`	Optionally, provide Michaelis-Menten coefficient for Farquhar model, and Gammastar. If not provided, they are calculated with a built-in function of leaf temperature.
`id`	Names of variables (quoted, can be a vector) in the original dataset to be stored in the result. Most useful when using `fitacis`, see there for examples of its use.

`...`	Further arguments (ignored at the moment).
`x`	For plot.acifit, an object returned by `fitaci`
`what`	The default is to plot both the data and the model fit, or specify 'data' or 'model' to plot one of them, or 'none' for neither (only the plot region is set up)
`xlim`	Limits for the X axis, if left blank estimated from data
`ylim`	Limits for the Y axis, if left blank estimated from data
`whichA`	By default all photosynthetic rates are plotted (Aj=Jmax-limited (blue), Ac=Vcmax-limited (red), Hyperbolic minimum (black)), TPU-limited rate (Ap, if estimated in the fit). Or, specify one or two of them.
`add`	If TRUE, adds to the current plot
`pch`	The plotting symbol for the data
`addzeroline`	If TRUE, the default, adds a dashed line at y=0
`addlegend`	If TRUE, adds a legend (by default does not add a legend if add=TRUE)
`legendbty`	Box type for the legend, passed to argument bty in `legend`.
`transitionpoint`	For plot.acifit, whether to plot a symbol at the transition point.
`linecols`	Vector of three colours for the lines (limiting rate, Ac, Aj), if one value provided it is used for all three.
`lwd`	Line widths, can be a vector of length 2 (first element for Ac and Aj, second one for the limiting rate).
`lty`	Line type (only for Amin - the limiting rate).

Details

Fitting method: The default method to fit A-Ci curves (set by `fitmethod="default"`) uses non-linear regression to fit the A-Ci curve. No assumptions are made on which part of the curve is Vcmax or Jmax limited. Normally, all three parameters are estimated: Jmax, Vcmax and Rd, unless Rd is provided as measured (when `useRd=TRUE`, and Rd is contained in the data). This is the method as described by Duursma (2015, Plos One).

The 'bilinear' method to fit A-Ci curves (set by `fitmethod="bilinear"`) linearizes the Vcmax and Jmax-limited regions, and applies linear regression twice to estimate first Vcmax and Rd, and then Jmax (using Rd estimated from the Vcmax-limited region). The transition point is found as the one which gives the best overall fit to the data (i.e. all possible transitions are tried out, similar to Gu et al. 2010, PCE). The advantage of this method is that it *always* returns parameter estimates, so it should be used in cases where the default method fails. Be aware, though, that the default method fails mostly when the curve contains bad data (so check your data before believing the fitted parameters).

When `citransition` is set, it splits the data into a Vcmax-limited (where Ci < citransition), and Jmax-limited region (Ci > citransition). Both parameters are then estimated separately for each region (Rd is estimated only for the Vcmax-limited region). **Note** that the actual transition point as shown in the standard plot of the fitted A-Ci curve may be quite different from that provided, since the fitting method simply decides which part of the dataset to use for which limitation, it does not constrain the actual estimated transition point directly. See the example below. If `fitmethod="default"`, it applies non-linear regression to both parts of the data, and when fitmethod="bilinear", it uses linear regression on the linearized photosynthesis rate. Results will differ between the two methods (slightly).

The 'onepoint' fitting method is a very simple estimation of Vcmax and Jmax for every point in the dataset, simply by inverting the photosynthesis equation. See De Kauwe et al. (2016) for details. The output will give the original data with Vcmax and Jmax added (note you can set `Tcorrect` as usual!). For increased reliability, this method only works if dark respiration (Rd) is included in the data (`useRd` is set automatically when setting `fitmethod='one-point'`). This method is not recommended for full A-Ci curves, but rather for spot gas exchange measurements, when a simple estimate of Vcmax or Jmax is needed, for example when separating stomatal and non-stomatal drought effects on photosynthesis (Zhou et al. 2013, AgForMet). The user will have to decide whether the Vcmax or Jmax rates are used in further analyses. This fitting method can not be used in `fitacis`, because Vcmax and Jmax are already estimated for each point in the dataset.

TPU limitation: Optionally, the `fitaci` function estimates the triose-phosphate utilization (TPU) rate. The TPU can act as another limitation on photosynthesis, and can be recognized by a 'flattening out' of the A-Ci curve at high Ci. When `fitTPU=TRUE`, the fitting method used will always be 'bilinear'. The TPU is estimated by trying out whether the fit improves when the last n points of the curve are TPU-limited (where n=1,2,...). When TPU is estimated, it is possible (though rare) that no points are Jmax-limited (in which case estimated Jmax will be NA). A minimum of two points is always reserved for the estimate of Vcmax and Rd. An additional parameter (`alphag`) can be set that affects the behaviour at high Ci (see Ellsworth et al. 2015 for details, and also `Photosyn`). See examples.

Temperature correction: When `Tcorrect=TRUE` (the default), Jmax and Vcmax are re-scaled to 25C, using the temperature response parameters provided (but Rd is always at measurement temperature). When `Tcorrect=FALSE`, estimates of all parameters are at measurement temperature. If TPU is fit, it is never corrected for temperature. Important parameters to the fit are GammaStar and Km, both of which are calculated from leaf temperature using standard formulations. Alternatively, they can be provided as known inputs. **Warning** : since package version 1.4, the default parameters have been adjusted. The new parameter values (EaV, EdVJ, delSJ, etc.) were based on a comprehensive literature review. See `vignette("new_T_responses")` or the article on remkoduursma.github.io/plantecophys.

Mesophyll conductance: It is possible to provide an estimate of the mesophyll conductance as input (`gmeso`), in which case the fitted Vcmax and Jmax are to be interpreted as chloroplastic rates. When using gmeso, it is recommended to use the 'default' fitting method (which will use the Ethier&Livingston equations inside `Photosyn`). It is also implemented with the 'bilinear' method but it requires more testing (and seems to give some strange results). When gmeso is set to a relatively low value, the resulting fit may be quite strange.

Other parameters: The A-Ci curve parameters depend on the values of a number of other parameters. For Jmax, PPFD is needed in order to express it as the asymptote. If PPFD is not provided in the dataset, it is assumed to equal 1800 mu mol m-2 s-1 (in which case a warning is printed). It is possible to either provide PPFD as a variable in the dataset (with the default name 'PARi', which can be changed), or as an argument to the `fitaci` directly.

Plotting and summarizing: The default **plot** of the fit is constructed with `plot.acifit`, see Examples below. When plotting the fit, the A-Ci curve is simulated using the `Aci` function, with leaf temperature (Tleaf) and PPFD set to the mean value for the dataset. The **coefficients** estimated in the fit (Vcmax, Jmax, and usually Rd) are extracted with `coef`. The summary of the fit is the same as the 'print' method, that is `myfit` will give the same output as `summary(myfit)` (where `myfit` is an object returned by `fitaci`).

Because fitaci returns the fitted `nls` object, more details on statistics of the fit can be extracted with standard tools. The Examples below shows the use of the **nlstools** to extract many details of the fit at once. The fit also includes the **root mean squared error** (RMSE), which can be extracted as `myfit$RMSE`. This is a useful metric to compare the different fitting methods.

Predicting and the CO2 compensation point: The fitted object contains two functions that reproduce the fitted curve exactly. Suppose your object is called 'myfit', then `myfit$Photosyn(200)` will give the fitted rate of photosynthesis at a Ci of 200. The inverse, calculating the Ci where some rate of photosynthesis is achieved, can be done with `myfit$Ci(10)` (find the Ci where net photosynthesis is ten). The (fitted!) CO2 compensation point can then be calculated with : `myfit$Ci(0)`

.

Atmospheric pressure correction: Note that atmospheric pressure (Patm) is taken into account, assuming the original data are in molar units (Ci in mu mol mol-1, or ppm). During the fit, Ci is converted to mu bar, and Km and Gammastar are recalculated accounting for the effects of Patm on the partial pressure of oxygen. When plotting the fit, though, molar units are shown on the X-axis. Thus, you should get (nearly) the same fitted curve when Patm was set to a value lower than 100kPa, but the fitted Vcmax and Jmax will be higher. This is because at low Patm, photosynthetic capacity has to be higher to achieve the same measured photosynthesis rate.

Value

A list of class 'acifit', with the following components:

df A dataframe with the original data, including the measured photosynthetic rate (Ameas), the fitted photosynthetic rate (Amodel), Jmax and Vcmax-limited gross rates (Aj, Ac), TPU-limited rate (Ap), dark respiration (Rd), leaf temperature (Tleaf), chloroplastic CO2 (Cc), PPFD, atmospheric pressure (Patm), and 'original Ci, i.e. the Ci used as input (which is different from the Ci used in fitting if Patm was not set to 100kPa)

pars Contains the parameter estimates and their approximate standard errors

nlsfit The object returned by `nls`, and contains more detail on the quality of the fit

Tcorrect whether the temperature correction was applied (logical)

Photosyn A copy of the `Photosyn` function with the arguments adjusted for the current fit. That is, Vcmax, Jmax and Rd are set to those estimated in the fit, and Tleaf and PPFD are set to the mean value in the dataset. All other parameters that were set in fitaci are also used (e.g. temperature dependency parameters, TPU, etc.).

Ci As Photosyn, except the opposite: calculate the Ci where some rate of net photosynthesis is achieved.

Ci_transition The Ci at which photosynthesis transitions from Vcmax to Jmax limited photosynthesis.

Ci_transition2 The Ci at which photosynthesis transitions from Jmax to TPU limitation. Set to NA is either TPU was not estimated, or it could not be estimated from the data.

Rd_measured Logical - was Rd provided as measured input?

GammaStar The value for GammaStar, either calculated or provided to the fit.

Km he value for Km, either calculated or provided to the fit.

kminput Was Km provided as input? (If FALSE, it was calculated from Tleaf)

gstarinput Was GammaStar provided as input? (If FALSE, it was calculated from Tleaf)

fitmethod The fitmethod uses, either default or bilinear

citransition The input citransition (NA if it was not provided as input)

gmeso The mesophyll conductance used in the fit (NA if it was not set)

fitTPU Was TPU fit?

alphag The value of alphag used in estimating TPU.

RMSE The Root-mean squared error, calculated as `sqrt(sum((Ameas-Amodel)^2))`.

runorder The data returned in the ’df’ slot are ordered by Ci, but in rare cases the original order of the data contains information; ’runorder’ is the order in which the data were provided.

References

Duursma, R.A., 2015. Plantecophys - An R Package for Analysing and Modelling Leaf Gas Exchange Data. PLoS ONE 10, e0143346. doi:10.1371/journal.pone.0143346

De Kauwe, M. G. et al. 2016. A test of the ’one-point method’ for estimating maximum carboxylation capacity from field-measured, light-saturated photosynthesis. New Phytol 210, 1130-1144.

Examples

```
## Not run:
# Fit an A-Ci curve on a dataframe that contains Ci, Photo and optionally Tleaf and PPFD.
# Here, we use the built-in example dataset 'acidata1'.
f <- fitaci(acidata1)

# Note that the default behaviour is to correct Vcmax and Jmax for temperature,
# so the estimated values are at 25C. To turn this off:
f2 <- fitaci(acidata1, Tcorrect=FALSE)

# To use different T response parameters (see ?Photosyn),
f3 <- fitaci(acidata1, Tcorrect=TRUE, EaV=25000)

# Make a standard plot
plot(f)

# Look at a summary of the fit
summary(f)

# Extract coefficients only
coef(f)

# The object 'f' also contains the original data with predictions.
# Here, Amodel are the modelled (fitted) values, Ameas are the measured values.
with(f$df, plot(Amodel, Ameas))
```

```
abline(0,1)

# The fitted values can also be extracted with the fitted() function:
fitted(f)

# The non-linear regression (nls) fit is stored as well,
summary(f$nlsfit)

# Many more details can be extracted with the nlstools package
library(nlstools)
overview(f$nlsfit)

# The curve generator is stored as f$Photosyn:
# Calculate photosynthesis at some value for Ci, using estimated
# parameters and mean Tleaf, PPFD for the dataset.
f$Photosyn(Ci=820)

# Photosynthetic rate at the transition point:
f$Photosyn(Ci=f$Ci_transition)$ALEAF

# Set the transition point; this will fit Vcmax and Jmax separately. Note that the *actual*
# transition is quite different from that provided, this is perfectly fine :
# in this case Jmax is estimated from the latter 3 points only (Ci>800), but the actual
# transition point is at ca. 400ppm.
g <- fitaci(acidata1, citransition=800)
plot(g)
g$Ci_transition

# Use measured Rd instead of estimating it from the A-Ci curve.
# The Rd measurement must be added to the dataset used in fitting,
# and you must set useRd=TRUE.
acidata1$Rd <- 2
f2 <- fitaci(acidata1, useRd=TRUE)
f2

# Fit TPU limitation
ftpu <- fitaci(acidata1, fitTPU=TRUE, PPFD=1800, Tcorrect=TRUE)
plot(ftpu)

## End(Not run)
```

`fitacis` *Fit multiple A-Ci curves at once*

Description

A convenient function to fit many curves at once, by calling `fitaci` for every group in the dataset. The data provided must include a variable that uniquely identifies each A-Ci curve.

Usage

```
fitacis(data, group, fitmethod = c("default", "bilinear"),
  progressbar = TRUE, quiet = FALSE, id = NULL, ...)

## S3 method for class 'acifits'
plot(x, how = c("manyplots", "oneplot"),
  highlight = NULL, ylim = NULL, xlim = NULL, add = FALSE,
  what = c("model", "data", "none"), colour_by_id = FALSE,
  id_legend = TRUE, linecol = "grey", linecol_highlight = "black",
  lty = 1, ...)
```

Arguments

`data`	Dataframe with Ci, Photo, Tleaf, PPFD (the last two are optional). For `fitacis`, also requires a grouping variable.
`group`	The name of the grouping variable in the dataframe (an A-Ci curve will be fit for each group separately).
`fitmethod`	Method to fit the A-Ci curve. Either ’default’ (Duursma 2015), or ’bilinear’. See Details.
`progressbar`	Display a progress bar (default is TRUE).
`quiet`	If TRUE, no messages are written to the screen.
`id`	Names of variables (quoted, can be a vector) in the original dataset to return as part of the coef() statement. Useful for keeping track of species names, treatment levels, etc. See Details and Examples.
`...`	Further arguments passed to `fitaci` (in the case of `fitacis`), or `plot.acifit` (in the case of `plot.acifits`).
`x`	For `plot.acifits`, an object returned from `fitacis`
`how`	If ’manyplots’, produces a single plot for each A-Ci curve. If ’oneplot’ overlays all of them.
`highlight`	If a name of a curve is given (check names(object), where object is returned by acifits), all curves are plotted in grey, with the highlighted one on top.
`xlim, ylim`	The X and Y axis limits.
`add`	If TRUE, adds the plots to a current plot.
`what`	What to plot, either ’model’ (the fitted curve), ’data’ or ’none’. See examples.
`colour_by_id`	If TRUE, uses the ’id’ argument to colour the curves in the standard plot (only works when `how = 'oneplot'`, see Examples)
`id_legend`	If `colour_by_id` is set, place a legend (topleft) or not.
`linecol`	Colour(s) to use for the non-highlighted curves (can be a vector).
`linecol_highlight`	Colour to use for the ’highlighted’ curve.
`lty`	Line type(s), can be a vector (one for each level of the factor, will be recycled).

Details

Troubleshooting - When using the default fitting method (see `fitaci`), it is common that some curves cannot be fit. Usually this indicates that the curve is poor quality and should not be used to estimate photosynthetic capacity, but there are exceptions. The `fitacis` function now refits the non-fitting curves with the 'bilinear' method (see `fitaci`), which will always return parameter estimates (for better or worse).

Summarizing and plotting - Like `fitaci`, the batch utility `fitacis` also has a standard plotting method. By default, it will make a single plot for every curve that you fit (thus generating many plots). Alternatively, use the setting `how="oneplot"` (see Examples below) for a single plot. The fitted **coefficients** are extracted with `coef`, which gives a dataframe where each row represents a fitted curve (the grouping label is also included).

Adding identifying variables - after fitting multiple curves, the most logical next step is to analyze the coefficient by some categorical variable (species, treatment, location). You can use the `id` argument to store variables from the original dataset in the output. It is important that the 'id' variables take only one value per fitted curve, if this is not the case only the first value of the curve will be stored (this will be rarely useful). See examples.

References

Duursma, R.A., 2015. Plantecophys - An R Package for Analysing and Modelling Leaf Gas Exchange Data. PLoS ONE 10, e0143346. doi:10.1371/journal.pone.0143346

Examples

```
## Not run:
# Fit many curves (using an example dataset)
# The bilinear method is much faster, but compare using 'default'!
fits <- fitacis(manyacidat, "Curve", fitmethod="bilinear")
with(coef(fits), plot(Vcmax, Jmax))

# The resulting object is a list, with each component an object as returned by fitaci
# So, we can extract one curve:
fits[[1]]
plot(fits[[1]])

# Plot all curves in separate figures with plot(fits)
# Or, in one plot:
plot(fits, how="oneplot")

# Note that parameters can be passed to plot.acifit. For example,
plot(fits, how="oneplot", what="data", col="blue")
plot(fits, how="oneplot", add=TRUE, what="model", lwd=c(1,1))

# Other elements can be summarized with sapply. For example, look at the RMSE:
rmses <- sapply(fits, "[[", "RMSE")
plot(rmses, type='h', ylab="RMSE", xlab="Curve nr")

# And plot the worst-fitting curve:
plot(fits[[which.max(rmses)]])
```

```
# It is very straightforward to summarize the coefficients by a factor variable
# that was contained in the original data. In manyacidat, there is a factor variable
# 'treatment'.
# We first have to refit the curves, using the 'id' argument:
fits <- fitacis(manyacidat, "Curve", fitmethod="bilinear", id="treatment")

# And now use this to plot Vcmax by treatment.
boxplot(Vcmax ~ treatment, data=coef(fits), ylim=c(0,130))

# As of package version 1.4-2, you can also use the id variable for colouring curves,
# when plotting all fitted curves in one plot.
# Set colours to be used. Also note that the 'id' variable has to be a factor,
# colours will be set in order of the levels of the factor.
# Set palette of colours:
palette(rainbow(8))

# Use colours, add legend.
plot(fits, how="oneplot", colour_by_id = TRUE, id_legend=TRUE)


## End(Not run)
```

`fitBB` *Fit Ball-Berry type models of stomatal conductance*

Description

Fits one of three versions of the Ball-Berry type stomatal conductance models to observations of stomatal conductance (gs), photosynthesis (A), atmospheric CO2 concentration (Ca) and vapour pressure deficit (VPD).

Usage

```
fitBB(data, varnames = list(ALEAF = "Photo", GS = "Cond", VPD = "VpdL",
  Ca = "CO2S", RH = "RH"), gsmodel = c("BBOpti", "BBLeuning",
  "BallBerry", "BBOptiFull"), fitg0 = FALSE)
```

Arguments

`data`	Input dataframe, containing all variables needed to fit the model.
`varnames`	List of names of variables in the input dataframe. Relative humidity (RH) is only needed when the original Ball-Berry model is to be fit.
`gsmodel`	One of BBOpti (Medlyn et al. 2011), BBLeuning (Leuning 1995), BallBerry (Ball et al. 1987), or BBOptiFull (Medlyn et al. 2011 but with an extra parameter gk, see Duursma et al. 2013)
`fitg0`	If TRUE, also fits the intercept term (g0, the 'residual conductance'). Default is FALSE.

Details

Note that unlike in some publications (e.g. Leuning et al. 1995), the models fit here do not include the CO2 compensation point. This correction may be necessary but can be added by the user (by replacing Ca with the corrected term).

Note that all models use atmospheric CO2 concentration (Ca) instead of, as sometimes argued, intercellular CO2 concentration (Ci). Using the latter makes these models far more difficult to use in practice, and we have found no benefit of using Ci instead of Ca (and Ca arises from an optimization argument, see Medlyn et al. 2011). The idea that we should use Ci because 'stomata sense Ci, not Ca' is probably not valid (or at least, not sufficient), and note that Ci plays a central role in the steady-state solution to stomatal conductance anyway (see `Photosyn`).

To fit the Ball-Berry models for each group in a dataframe, for example species, see the `fitBBs` function.

Value

A list with several components, most notably `fit`, the object returned by `nls`. If the user needs more information on the goodness of fit etc, please further analyze this object. For example, use the **broom** package for quick summaries. Or use `confint` to calculate confidence intervals on the fitted parameters.

References

Ball, J.T., Woodrow, I.E., Berry, J.A., 1987. A model predicting stomatal conductance and its contribution to the control of photosynthesis under different environmental conditions., in: Biggins, J. (Ed.), Progress in Photosynthesis Research. Martinus-Nijhoff Publishers, Dordrecht, the Netherlands, pp. 221-224.

Leuning, R. 1995. A critical-appraisal of a combined stomatal-photosynthesis model for C-3 plants. Plant Cell and Environment. 18:339-355.

Medlyn, B.E., R.A. Duursma, D. Eamus, D.S. Ellsworth, I.C. Prentice, C.V.M. Barton, K.Y. Crous, P. De Angelis, M. Freeman and L. Wingate. 2011. Reconciling the optimal and empirical approaches to modelling stomatal conductance. Global Change Biology. 17:2134-2144.

Duursma, R.A., Payton, P., Bange, M.P., Broughton, K.J., Smith, R.A., Medlyn, B.E., Tissue, D.T., 2013. Near-optimal response of instantaneous transpiration efficiency to vapour pressure deficit, temperature and [CO2] in cotton (Gossypium hirsutum L.). Agricultural and Forest Meteorology 168, 168-176. doi:10.1016/j.agrformet.2012.09.005

Examples

```
## Not run:
# If 'mydfr' is a dataframe with 'Photo', 'Cond', 'VpdL' and 'CO2S', you can do:
myfit <- fitBB(mydfr, gsmodel = "BBOpti")

# Coefficients and a message:
myfit

# Coefficients only
coef(myfit)
```

```
# If you have a species variable, and would like to fit the model for each species,
# use fitBBs (see its help page ?fitBBs)
myfits <- fitBBs(mydfr, "species")

## End(Not run)
```

`fitBBs` *Fit Ball-Berry type models of stomatal conductance to many groups at once*

Description

A batch utility for the `fitBB` function, to fit the model for each group in a dataframe.

Usage

```
fitBBs(data, group, ...)
```

Arguments

`data`	Input dataframe, containing all variables needed to fit the model.
`group`	Name of the grouping variable in the dataframe (quoted), the model will be fit for each group defined by this variable.
`...`	Further parameters passed to `fitBB`, see there for a full description.

Examples

```
## Not run:
# If you have a factor variable in your dataset called 'species', and you
# want to fit the Ball-Berry model for each of the species:
myfits <- fitBBs(mydata, "species", model="BallBerry")

# A dataframe with coefficients is returned by coef()
coef(myfits)


## End(Not run)
```

`manyacidat` *An example dataset with multiple A-Ci curves*

Description

CO2 response of leaf photosynthesis, as measured with a Licor6400, for multiple leaves.

Format

Curve An identifier for the A-Ci curve (28 curves in total, 13-14 points per curve)

Ci Intercellular CO2 concentration (ppm)

Photo Net photosynthesis rate (mu mol m-2 s-1)

Tleaf Leaf temperature (deg C)

PPFD Photosynthetic photon flux density (mu mol m-2 s-1)

`Photosyn` *Coupled leaf gas exchange model*

Description

A coupled photosynthesis - stomatal conductance model, based on the Farquhar model of photosynthesis, and a Ball-Berry type model of stomatal conductance. Includes options for temperature sensitivity of photosynthetic parameters, day respiration (optionally calculated from leaf temperature), and mesophyll conductance.

Usage

```
Photosyn(VPD = 1.5, Ca = 400, PPFD = 1500, Tleaf = 25,
  Patm = 100, RH = NULL, gsmodel = c("BBOpti", "BBLeuning",
  "BallBerry", "BBdefine"), g1 = 4, g0 = 0, gk = 0.5, vpdmin = 0.5,
  D0 = 5, GS = NULL, BBmult = NULL, alpha = 0.24, theta = 0.85,
  Jmax = 100, Vcmax = 50, gmeso = NULL, TPU = 1000, alphag = 0,
  Rd0 = 0.92, Q10 = 1.92, Rd = NULL, TrefR = 25, Rdayfrac = 1,
  EaV = 58550, EdVC = 2e+05, delsC = 629.26, EaJ = 29680,
  EdVJ = 2e+05, delsJ = 631.88, GammaStar = NULL, Km = NULL,
  Ci = NULL, Tcorrect = TRUE, returnParsOnly = FALSE,
  whichA = c("Ah", "Amin", "Ac", "Aj"))

Aci(Ci, ...)
```

Arguments

`VPD`	Vapour pressure deficit (kPa) (not needed when RH provided)
`Ca`	Atmospheric CO2 concentration (ppm)
`PPFD`	Photosynthetic photon flux density ('PAR') (mu mol m-2 s-1)
`Tleaf`	Leaf temperature (degrees C)
`Patm`	Atmospheric pressure (kPa) (but see warning below!)
`RH`	Relative humidity (in %) (not needed when VPD provided)
`gsmodel`	One of BBOpti (Medlyn et al. 2011), BBLeuning (Leuning 1995), BallBerry (Ball et al. 1987), or BBdefine (for full control; see Details).
`g0, g1`	Parameters of Ball-Berry type stomatal conductance models.
`gk`	Optional, exponent of VPD in gs model (Duursma et al. 2013)
`vpdmin`	Below vpdmin, VPD=vpdmin, to avoid very high gs.
`D0`	Parameter for the BBLeuning stomatal conductance model.
`GS`	Optionally, stomatal conductance (to H2O). If provided, `Photosyn` calculates Ci and photosynthesis. See Details.
`BBmult`	Optional, only used when `gsmodel = "BBdefine"`, see Details.
`alpha`	Quantum yield of electron transport (mol mol-1)
`theta`	Shape of light response curve.
`Jmax`	Maximum rate of electron transport at 25 degrees C (mu mol m-2 s-1)
`Vcmax`	Maximum carboxylation rate at 25 degrees C (mu mol m-2 s-1)
`gmeso`	Mesophyll conductance (mol m-2 s-1). If not NULL (the default), Vcmax and Jmax are chloroplastic rates.
`TPU`	Triose-phosphate utilization rate (mu mol m-2 s-1); optional.
`alphag`	Fraction of glycolate not returned to the chloroplast; parameter in TPU-limited photosynthesis (optional, only to be used when TPU is provided) (0 - 1)
`Rd0`	Day respiration rate at reference temperature (`TrefR`). Must be a positive value.
`Q10`	Temperature sensitivity of Rd.
`Rd`	Day respiration rate (mu mol m-2 s-1), optional (if not provided, calculated from Tleaf, Rd0, Q10 and TrefR). Must be a positive value (an error occurs when a negative value is supplied).
`TrefR`	Reference temperature for Rd (Celcius).
`Rdayfrac`	Ratio of Rd in the light vs. in the dark.
`EaV, EdVC, delsC`	Vcmax temperature response parameters
`EaJ, EdVJ, delsJ`	Jmax temperature response parameters
`Km, GammaStar`	Optionally, provide Michaelis-Menten coefficient for Farquhar model, and Gammastar. If not provided, they are calculated with a built-in function of leaf temperature.

`Ci`	Optional, intercellular CO2 concentration (ppm). If not provided, calculated via gs model.
`Tcorrect`	If TRUE, corrects input Vcmax and Jmax for actual Tleaf (if FALSE, assumes the provided Vcmax and Jmax are at the Tleaf provided). **Warning** : since package version 1.4, the default parameters have been adjusted (see Details).
`returnParsOnly`	If TRUE, returns calculated Vcmax,Jmax,Km and GammaStar based on leaf temperature.
`whichA`	Which assimilation rate does gs respond to?
`...`	Further arguments passed to `Photosyn`

Details

The coupled photosynthesis - stomatal conductance model finds the intersection between the supply of CO2 by diffusion, and the demand for CO2 by photosynthesis. See Farquhar and Sharkey (1982) for basic description of this type of model, Duursma (2015) for more details on the implementation in the `plantecophys` package, and Duursma et al. (2014) for an example application (that uses this implementation).

Photosynthesis model and temperature response - The model of Farquhar et al. (1980) is used to estimate the dependence of leaf net photosynthesis rate (ALEAF) on intercellular CO2 concentration (Ci), accounting for all three limitations (electron transport, carboxylation, and TPU limitation). The equations for the temperature response of photosynthetic parameters, including Vcmax, Jmax, Gammastar, and Km follow Medlyn et al. (2002). However, **note that the default temperature response parameter values are not taken from Medlyn, and likely will have to be adjusted for your situation**. **Warning** : since package version 1.4, the default parameters have been adjusted. The new parameter values (EaV, EdVJ, delSJ, etc.) were based on a comprehensive literature review. See vignette("new_T_responses") or the article on remkoduursma.github.io/plantecophys.

#' By default, the `Photosyn` function returns the hyperbolic minimum of Vcmax and Jmax-limited photosynthetic rates, as well as the hyperbolic minimum of Jmax-limited and TPU-limited rates. This approach avoids the discontinuity at the transition between the two rates (thus allowing use of `Photosyn` and `fitaci` in optimization or fitting routines). The individual rates (Ac, Aj and Ap) are also returned as output should they be needed. Note that those rates are output as gross photosynthetic rates (leaf respiration has to be subtracted to give net leaf photosynthesis).

Coupled leaf gas exchange When Ci is not provided, Ci is calculated from the intersection between the 'supply' and 'demand', where 'demand' is given by the Farquhar model of photosynthesis (A=f(Ci)), and supply by the stomatal conductance. The latter is, by default, estimated using the stomatal conductance model of Medlyn et al. (2011), but two other models are provided as well (Ball-Berry and Leuning, see `gsmodel` argument). Otherwise, stomatal conductance may be directly provided via the `GS` argument.

Stomatal conductance models - At the moment, three stomatal conductance models are implemented. The 'BBOpti' model is a slightly more general form of the model of Medlyn et al. 2011 (see Duursma et al. 2013). It is given by (in notation of the parameters and output variables of `Photosyn`),

$$GS = g0 + 1.6 * (1 + g1/D^(1 - gk)) * ALEAF/CA$$

where `gk = 0.5` if stomata behave optimally (cf. Medlyn et al. 2011).

The 'BBLeuning' model is that of Leuning (1995). It is given by,

$$GS = g0 + g1 * ALEAF/(Ca * (1 + VPD/D0))$$

Note that this model also uses the `g1` parameter, but it needs to be set to a much higher value to be comparable in magnitude to the BBOpti model.

The 'BallBerry' model is that of Ball et al. (1987). It is given by,

$$GS = g0 + g1 * RH * ALEAF/Ca$$

Where RH is relative humidity. Again, the `g1` value is not comparable to that used in the previous two models.

Finally, `Photosyn` provides a very flexible Ball-Berry model, where the multiplier has to be specified by the user, the model is:

$$GS = g0 + BBmult * ALEAF$$

This interface can be used to quickly simulate what happens if stomata do not respond to humidity at all (in which case BBmult=g1/Ca, or ca. 5/400), or to use the Tuzet model of stomatal conductance inside another model that provides the leaf water potential function.

For the full numerical solution to the Cowan-Farquhar optimization, use the `FARAO` function (which was used in Medlyn et al. 2011 for comparison to the approximation there presented). See Duursma (2015) for more details.

Mesophyll conductance -

If the mesophyll conductance `gmeso` is provided as an input, it is assumed that Vcmax and Jmax are the chloroplastic rates, and leaf photosynthesis is calculated following the equations from Ethier and Livingston (2004). When very low mesophyll conductance rates are input, the model may return poor solutions (and sometimes they may not exist).

Simulating A-Ci curves

If Ci is provided as an input, this function calculates an A-Ci curve. For example, you may do `Photosyn(Ci=300)`, for which the function `Aci` is included as a shortcut (`Aci(300)`).

Atmospheric pressure -

A correction for atmospheric pressure (Patm) is implemented in `fitaci`, but **not in Photosyn**. In `fitaci`, the necessary corrections are applied so that estimated Vcmax and Jmax are expressed at standard pressure (Patm=100kPa). In Photosyn, however, the corrections are much more complicated and tend to be very small, because effects of Patm on partial pressures are largely offset by increases in diffusivity (Terashima et al. 1995, Gale 1973).

Note that Patm is an argument to the Photosyn function, but it only affects calculations of Km and GammaStar (as used by fitaci), and transpiration rate. Setting only Patm **does not correct for atmospheric pressure effects on photosynthesis rates**.

The simulation of limitation of the photosynthetic rate to triose-phosphate utilization follows details in Ellsworth et al. (2015), their Eq. 7. Note that the parameter `alphag` is set to zero by default.

Value

Returns a dataframe.

References

Duursma, R.A., Payton, P., Bange, M.P., Broughton, K.J., Smith, R.A., Medlyn, B.E., Tissue, D. T., 2013, Near-optimal response of instantaneous transpiration efficiency to vapour pressure deficit, temperature and [CO2] in cotton (Gossypium hirsutum L.). Agricultural and Forest Meteorology 168 : 168 - 176.

Duursma, R.A., Barton, C.V.M., Lin, Y.-S., Medlyn, B.E., Eamus, D., Tissue, D.T., Ellsworth, D.S., McMurtrie, R.E., 2014. The peaked response of transpiration rate to vapour pressure deficit in field conditions can be explained by the temperature optimum of photosynthesis. Agricultural and Forest Meteorology 189 - 190, 2-10. doi:10.1016/j.agrformet.2013.12.007

Duursma, R.A., 2015. Plantecophys - An R Package for Analysing and Modelling Leaf Gas Exchange Data. PLoS ONE 10, e0143346. doi:10.1371/journal.pone.0143346

Ellsworth, D.S., Crous, K.Y., Lambers, H., Cooke, J., 2015. Phosphorus recycling in photorespiration maintains high photosynthetic capacity in woody species. Plant Cell Environ 38, 1142-1156. doi:10.1111/pce.12468

Ethier, G. and N. Livingston. 2004. On the need to incorporate sensitivity to CO2 transfer conductance into the Farquhar von Caemmerer Berry leaf photosynthesis model. Plant, Cell & Environment. 27:137-153.

Farquhar, G.D., S. Caemmerer and J.A. Berry. 1980. A biochemical model of photosynthetic CO2 assimilation in leaves of C3 species. Planta. 149:78-90.

Farquhar, G. D., & Sharkey, T. D. (1982). Stomatal conductance and photosynthesis. Annual review of plant physiology, 33(1), 317-345.

Gale, J., 1972. Availability of Carbon Dioxide for Photosynthesis at High Altitudes: Theoretical Considerations. Ecology 53, 494-497. doi:10.2307/1934239

Leuning, R. 1995. A critical-appraisal of a combined stomatal-photosynthesis model for C-3 plants. Plant Cell and Environment. 18:339-355.

Medlyn, B.E., E. Dreyer, D. Ellsworth, M. Forstreuter, P.C. Harley, M.U.F. Kirschbaum, X. Le Roux, P. Montpied, J. Strassemeyer, A. Walcroft, K. Wang and D. Loustau. 2002. Temperature response of parameters of a biochemically based model of photosynthesis. II. A review of experimental data. Plant Cell and Environment. 25:1167-1179.

Medlyn, B.E., R.A. Duursma, D. Eamus, D.S. Ellsworth, I.C. Prentice, C.V.M. Barton, K.Y. Crous, P. De Angelis, M. Freeman and L. Wingate. 2011. Reconciling the optimal and empirical approaches to modelling stomatal conductance. Global Change Biology. 17:2134-2144.

Terashima, I., Masuzawa, T., Ohba, H., Yokoi, Y., 1995. Is photosynthesis suppressed at higher elevations due to low CO2 pressure? Ecology 76, 2663-2668. doi:10.2307/2265838

See Also

`FARAO`, `fitaci`, `AciC4`

Examples

```
# Run the coupled leaf gas exchange model, set only a couple of parameters
Photosyn(VPD=2, g1=4, Ca=500)

# It is easy to set multiple values for inputs (and these can be mixed with single inputs);
r <- Photosyn(VPD=seq(0.5, 4, length=25), Vcmax=50, Jmax=100)
with(r, plot(VPD, ALEAF, type='l'))

# Set the mesophyll conductance
run1 <- Photosyn(PPFD=seq(50,1000,length=25), gmeso=0.15, Vcmax=40, Jmax=85)
with(run1, plot(PPFD, GS, type='l'))

# Run A-Ci curve only (provide Ci instead of calculating it).
arun1 <- Aci(Ci=seq(50, 1200, length=101), Vcmax=40, Jmax=85)
arun2 <- Aci(Ci=seq(50, 1200, length=101), Vcmax=30, Jmax=70)
with(arun1, plot(Ci, ALEAF, type='l'))
with(arun2, points(Ci, ALEAF, type='l', lty=5))

# Find the intersection between supply of CO2 and demand for CO2 (cf. Farquhar and Sharkey 1982).

# Set some parameters
gs <- 0.2  # stomatal conductance to H2O
Ca <- 400  # ambient CO2
gctogw <- 1.57  # conversion
gc <- gs / gctogw  # stomatal conductance to CO2

# Demand curve (Farquhar model)
p <- Aci(seq(60,500,length=101), Ca=400)

# Provide stomatal conductance as input, gives intersection point.
g <- Photosyn(GS=gs, Ca=Ca)

# Intersection point visualized
par(yaxs="i")
with(p, plot(Ci, ALEAF, type='l', ylim=c(0,max(ALEAF))))
with(g, points(Ci, ALEAF, pch=19, col="red"))
abline(gc * Ca, -gc, lty=5)

legend("topleft", c(expression("Demand:"~~A==f(C[i])),
                    expression("Supply:"~~A==g[c]*(C[a]-C[i])),
                    "Operating point"),
       lty=c(1,5,-1),pch=c(-1,-1,19),
       col=c("black","black","red"),
       bty='n', cex=0.9)
```

PhotosynEB *Coupled leaf gas exchange model with energy balance*

Description

As `Photosyn`, but calculates the leaf temperature based on the leaf's energy balance. Including sensible and long-wave heat loss, latent heat loss from evaporation, and solar radiation input.

#'**Warning:**Do not provide GS as an input to `PhotosynEB` directly; the results will not be as expected (filed as issue #27)

Usage

```
PhotosynEB(Tair = 25, VPD = 1.5, Wind = 2, Wleaf = 0.02,
  StomatalRatio = 1, LeafAbs = 0.86, RH = NULL, ...)

FindTleaf(gs, Tair, ...)
```

Arguments

`Tair`	Air temperature (C)
`VPD`	The vapour pressure deficit of the air (i.e. not the leaf-to-air VPD) (kPa).
`Wind`	Wind speed (m s-1)
`Wleaf`	Leaf width (m)
`StomatalRatio`	The stomatal ratio (cf. Licor6400 terminology), if it is 1, leaves have stomata only on one side (hypostomatous), 2 for leaves with stomata on both sides (amphistomatous).
`LeafAbs`	Leaf absorptance of solar radiation (0-1).
`RH`	The relative humidity of the air (i.e. not calculated with leaf temperature) (in percent).
`...`	Further parameters passed to `Photosyn`. Note that Tleaf is not allowed as an input, since that is calculated by `PhotosynEB` from energy balance.
`gs`	For `FindTleaf`, the stomatal conductance (mol m-2 s-1).

Details

Uses the Penman-Monteith equation to calculate the leaf transpiration rate, and finds Tleaf by solving the leaf energy balance iteratively. In the solution, it is accounted for that stomatal conductance (via the dependence of photosynthesis on Tleaf) and net radiation depend on Tleaf.

Also included is the function `FindTleaf`, which calculates the leaf temperature if the stomatal conductance is known. The **limitation** to this function is that input stomatal conductance (gs) is not vectorized, i.e. you can only provide one value at a time.

`PhotosynTuzet` *Coupled leaf gas exchange model with Tuzet stomatal conductance*

Description

An implementation of the coupled photosynthesis - stomatal conductance model, using the Tuzet et al. (2003) model of stomatal conductance. Accepts all arguments of `Photosyn` (except `gsmodel`, of course).

Usage

```
PhotosynTuzet(g1 = 8, Ca = 400, psis = 0, kl = 2, sf = 3,
  psif = -2, ...)
```

Arguments

`g1`	The slope parameter. Note that the default value should be much higher than that used in the Medlyn et al. (2011) model to give comparable predictions.
`Ca`	Atmospheric CO2 concentration.
`psis`	Soil water potential (MPa). Note that soil-to-root hydraulic conductance is not implemented.
`kl`	Leaf-specific hydraulic conductance (mmol m-2 s-1 MPa-1)
`sf`	Shape parameter (-) of sigmoidal function of leaf water potential (see Tuzet et al. 2003)
`psif`	Leaf water potential at which stomatal conductance is 50% of maximum (MPa).
`...`	All other arguments in `Photosyn`

`RHtoVPD` *Conversions between relative humidity, vapour pressure deficit and dewpoint*

Description

A collection of functions to convert between relative humidity (RH) (%), vapour pressure deficit (VPD) (kPa), dew point temperature, and leaf- or air temperature-based VPD or RH. To convert from relative humidity to VPD, use the `RHtoVPD` function, use `VPDtoRH` for the other way around. The water vapor saturation pressure is calculated with `esat`. Use `DewtoVPD` to convert from dewpoint temperature to VPD. The functions `VPDleafToAir` and `VPDairToLeaf` convert VPD from a leaf temperature to an air-temperature basis and vice versa. The functions `RHleafToAir` a `RHairToLeaf` do the same for relative humidity.

Usage

```
RHtoVPD(RH, TdegC, Pa = 101)

VPDtoRH(VPD, TdegC, Pa = 101)

esat(TdegC, Pa = 101)

VPDtoDew(VPD, TdegC, Pa = 101)

DewtoVPD(Tdew, TdegC, Pa = 101)

VPDleafToAir(VPD, Tleaf, Tair, Pa = 101)

VPDairToLeaf(VPD, Tair, Tleaf, Pa = 101)

RHleafToAir(RH, Tleaf, Tair, Pa = 101)

RHairToLeaf(RH, Tair, Tleaf, Pa = 101)
```

Arguments

`RH`	Relative humidity (%)
`TdegC`	Temperature (degrees C) (either leaf or air)
`Pa`	Atmospheric pressure (kPa)
`VPD`	Vapour pressure deficit (kPa)
`Tdew`	Dewpoint temperature (degrees C)
`Tleaf`	Leaf temperature (degrees C)
`Tair`	Air temperature (degrees C)

Details

The function describing saturated vapor pressure with temperature is taken from Jones (1992). All other calculations follow directly from the standard definitions, for which Jones (1992) may also be consulted.

Author(s)

Remko Duursma

References

Jones, H.G. 1992. Plants and microclimate: a quantitative approach to environmental plant physiology. 2nd Edition., 2nd Edn. Cambridge University Press, Cambridge. 428 p.

Index