Package ‘rpmodel’

December 4, 2019

Type Package

Title P-Model

Version 1.0.4

Maintainer Benjamin Stocker <benjamin.stocker@gmail.com>

Description Implements the P-model (Stocker et al., 2019 <doi:10.5194/gmd-2019-200>), predicting acclimated parameters of the enzyme kinetics of C3 photosynthesis, assimilation, and dark respiration rates as a function of the environment (temperature, CO2, vapour pressure deficit, light, atmospheric pressure).

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.0.0

Imports rlang

Suggests dplyr, purrr, tidyr, knitr

VignetteBuilder knitr

URL https://github.com/stineb/rpmodel

BugReports https://github.com/stineb/rpmodel/issues

NeedsCompilation no

Author Benjamin Stocker [aut, cre] (<https://orcid.org/0000-0003-2697-9096>)

Repository CRAN

Date/Publication 2019-12-04 10:30:02 UTC

R topics documented:

`calc_ftemp_arrh` *Calculates the Arrhenius-type temperature response*

Description

Given a kinetic rate at a reference temperature (argument `tkref`) this function calculates its temperature-scaling factor following Arrhenius kinetics.

Usage

```
calc_ftemp_arrh(tk, dha, tkref = 298.15)
```

Arguments

`tk`	Temperature (Kelvin)
`dha`	Activation energy (J mol-1)
`tkref`	Reference temperature (Kelvin)

Details

To correct for effects by temperature following Arrhenius kinetics, and given a reference temperature T_0, f calculates the temperature scaling. Arrhenius kinetics are described by an equation of form $x(T) = exp(c - \Delta H_a/(TR))$. The temperature-correction function $f(T, \Delta H_a)$ is thus given by $f = x(T)/x(T_0)$ which is:

$$f = exp(\Delta H_a(T - T_0)/(T_0 R T_K))$$

ΔH_a is given by argument `dha`. T is given by argument `tk` and has to be provided in Kelvin. R is the universal gas constant and is 8.3145 J mol-1 K-1. Note that this is equivalent to

$$f = exp((\Delta H_a/R)(1/T_0 - 1/T))$$

Value

A numeric value for f

Examples

```
print("Relative rate change from 25 to 10 degrees Celsius (percent change):")
print( (1.0-calc_ftemp_arrh( 283.15, 100000, tkref = 298.15))*100 )
```

`calc_ftemp_inst_rd` *Calculates the temperature response of dark respiration*

Description

Given the dark respiration at the reference temperature 25 degress Celsius, this function calculates its temperature-scaling factor following Heskel et al. 2016.

Usage

```
calc_ftemp_inst_rd(tc)
```

Arguments

`tc`	Temperature (degrees Celsius)

Details

To correct for effects by temperature Heskel et al. 2016, and given the reference temperature $T0 =$ 25 deg C, this calculates the temperature scaling factor to calculate dark respiration at temperature T (argument `tc`) as:

$$fr = exp(0.1012(T0 - T) - 0.0005(T0^2 - T^2))$$

where T is given in degrees Celsius.

Value

A numeric value for fr

References

Heskel, M., O'Sullivan, O., Reich, P., Tjoelker, M., Weerasinghe, L., Penillard, A.,Egerton, J., Creek, D., Bloomfield, K., Xiang, J., Sinca, F., Stangl, Z., Martinez-De La Torre, A., Griffin, K., Huntingford, C., Hurry, V., Meir, P., Turnbull, M.,and Atkin, O.: Convergence in the temperature response of leaf respiration across biomes and plant functional types, Proceedings of the National Academy of Sciences, 113, 3832–3837, doi:10.1073/pnas.1520282113,2016.

Examples

```
## Relative change in Rd going (instantaneously, i.e. not
## acclimatedly) from 10 to 25 degrees (percent change):
print( (calc_ftemp_inst_rd(25)/calc_ftemp_inst_rd(10)-1)*100 )
```

`calc_ftemp_inst_vcmax` *Calculates the instantaneous temperature response of Vcmax*

Description

Given Vcmax at a reference temperature (argument `tcref`) this function calculates its temperature-scaling factor following modified Arrhenius kinetics based on Kattge & Knorr (2007). Calculates f for the conversion

$$V = fVref$$

Usage

```
calc_ftemp_inst_vcmax(tcleaf, tcgrowth = tcleaf, tcref = 25)
```

Arguments

`tcleaf`	Leaf temperature, or in general the temperature relevant for photosynthesis (degrees Celsius)
`tcgrowth`	(Optional) Growth temperature, in the P-model, taken to be equal to `tcleaf` (in degrees Celsius). Defaults to `tcgrowth = tcleaf`.
`tcref`	Reference temperature (in degrees Celsius)

Details

The function is given by Kattge & Knorr (2007) as

$$fv = f(T, \Delta Hv)A/B$$

where $f(T, \Delta Hv)$ is a regular Arrhenius-type temperature response function (see calc_ftemp_arrh) with $Hv = 71513$ J mol-1,

$$A = 1 + exp((T0\Delta S - Hd)/(T0R))$$

and

$$B = 1 + exp((T\Delta S - Hd)/(TKR))$$

Here, T is in Kelvin, $T0 = 293.15$ K, $Hd = 200000$ J mol-1 is the deactivation energy and R is the universal gas constant and is 8.3145 J mol-1 K-1, and

$$\Delta S = aS - bST$$

with $aS = 668.39$ J mol-1 K-1, and $bS = 1.07$ J mol-1 K-2, and T given in degrees Celsius (!)

Value

A numeric value for fv

References

Kattge, J. and Knorr, W.: Temperature acclimation in a biochemical model of photosynthesis: a reanalysis of data from 36 species, Plant, Cell and Environment, 30,1176–1190, 2007.

Examples

```
## Relative change in Vcmax going (instantaneously, i.e.
## not acclimatedly) from 10 to 25 degrees (percent change):
print((calc_ftemp_inst_vcmax(25)/calc_ftemp_inst_vcmax(10)-1)*100 )
```

---

calc_ftemp_kphio — *Calculates the temperature dependence of the quantum yield efficiency*

Description

Calculates the temperature dependence of the quantum yield efficiency following the temperature dependence of the maximum quantum yield of photosystem II in light-adapted tobacco leaves, determined by Bernacchi et al. (2003)

Usage

```
calc_ftemp_kphio(tc)
```

Arguments

`tc`	Temperature, relevant for photosynthesis (degrees Celsius)

Details

The temperature factor is calculated as

$$\phi(T) = 0.352 + 0.022T - 0.00034T^2$$

The factor $\phi(T)$ is to be multiplied with leaf absorptance and the fraction of absorbed light that reaches photosystem II. In the P-model these additional factors are lumped into a single apparent quantum yield efficiency parameter (argument `kphio` to function rpmodel).

Value

A numeric value for $\phi(T)$

References

Bernacchi, C. J., Pimentel, C., and Long, S. P.: In vivo temperature response func-tions of parameters required to model RuBP-limited photosynthesis, Plant Cell Environ., 26, 1419–1430, 2003

Examples

```
## Relative change in the quantum yield efficiency
## between 5 and 25 degrees celsius (percent change):
print(paste((calc_ftemp_kphio(25.0)/calc_ftemp_kphio(5.0)-1)*100 ))
```

---

calc_gammastar   *Calculates the CO2 compensation point*

Description

Calculates the photorespiratory CO2 compensation point in absence of dark respiration, $\Gamma*$ (Farquhar, 1980).

Usage

```
calc_gammastar(tc, patm)
```

Arguments

`tc`	Temperature, relevant for photosynthesis (degrees Celsius)
`patm`	Atmospheric pressure (Pa)

Details

The temperature and pressure-dependent photorespiratory compensation point in absence of dark respiration $\Gamma * (T, p)$ is calculated from its value at standard temperature ($T0 = 25$deg C) and atmospheric pressure ($p0 = 101325$ Pa), referred to as $\Gamma * 0$, quantified by Bernacchi et al. (2001) to 4.332 Pa (their value in molar concentration units is multiplied here with 101325 Pa to yield 4.332 Pa). $\Gamma * 0$ is modified by temperature following an Arrhenius-type temperature response function $f(T, \Delta Ha)$ (implemented by calc_ftemp_arrh) with activation energy $\Delta Ha = 37830$ J mol-1 and is corrected for atmospheric pressure $p(z)$ (see calc_patm) at elevation z.

$$\Gamma* = \Gamma * 0 f(T, \Delta Ha) p(z)/p_0$$

$p(z)$ is given by argument `patm`.

Value

A numeric value for $\Gamma*$ (in Pa)

References

Farquhar, G. D., von Caemmerer, S., and Berry, J. A.: A biochemical model of photosynthetic CO2 assimilation in leaves of C 3 species, Planta, 149, 78–90, 1980.

Bernacchi, C. J., Singsaas, E. L., Pimentel, C., Portis, A. R. J., and Long, S. P.:Improved temperature response functions for models of Rubisco-limited photosyn-thesis, Plant, Cell and Environment, 24, 253–259, 2001

Examples

```
print("CO2 compensation point at 20 degrees Celsius and standard atmosphere (in Pa):")
print(calc_gammastar(20, 101325))
```

`calc_kmm` *Calculates the Michaelis Menten coefficient for Rubisco-limited photosynthesis*

Description

Calculates the Michaelis Menten coefficient of Rubisco-limited assimilation as a function of temperature and atmospheric pressure.

Usage

```
calc_kmm(tc, patm)
```

Arguments

`tc`	Temperature, relevant for photosynthesis (deg C)
`patm`	Atmospheric pressure (Pa)

Details

The Michaelis-Menten coefficient K of Rubisco-limited photosynthesis is determined by the Michalis-Menten constants for O2 and CO2 (Farquhar, 1980) according to:

$$K = Kc(1 + pO2/Ko)$$

where Kc is the Michaelis-Menten constant for CO2 (Pa), Ko is the Michaelis-Menten constant for O2 (Pa), and $pO2$ is the partial pressure of oxygen (Pa), calculated as $0.209476p$, where p is given by argument `patm`. Kc and Ko follow a temperature dependence, given by the Arrhenius Equation f (implemented by calc_ftemp_arrh):

$$Kc = Kc25f(T, \Delta Hkc)$$

$$Ko = Ko25f(T, \Delta Hko)$$

Values ΔHkc (79430 J mol-1), ΔHko (36380 J mol-1), $Kc25$ (39.97 Pa), and $Ko25$ (27480 Pa) are taken from Bernacchi et al. (2001) and have been converted from values given therein to units of Pa by multiplication with the standard atmosphere (101325 Pa). T is given by the argument `tc`.

Value

A numeric value for K (in Pa)

References

Farquhar, G. D., von Caemmerer, S., and Berry, J. A.: A biochemical model of photosynthetic CO2 assimilation in leaves of C 3 species, Planta, 149, 78–90, 1980.

Bernacchi, C. J., Singsaas, E. L., Pimentel, C., Portis, A. R. J., and Long, S. P.:Improved temperature response functions for models of Rubisco-limited photosyn-thesis, Plant, Cell and Environment, 24, 253–259, 2001

Examples

```
print("Michaelis-Menten coefficient at 20 degrees Celsius and standard atmosphere (in Pa):")
print(calc_kmm(20, 101325))
```

`calc_patm` *Calculates atmospheric pressure*

Description

Calculates atmospheric pressure as a function of elevation, by default assuming standard atmosphere (101325 Pa at sea level)

Usage

```
calc_patm(elv, patm0 = 101325)
```

Arguments

`elv`	Elevation above sea-level (m.a.s.l.)
`patm0`	(Optional) Atmospheric pressure at sea level (Pa), defaults to 101325 Pa.

Details

The elevation-dependence of atmospheric pressure is computed by assuming a linear decrease in temperature with elevation and a mean adiabatic lapse rate (Berberan-Santos et al., 1997):

$$p(z) = p0(1 - Lz/TK0)^(gM/(RL))$$

where z is the elevation above mean sea level (m, argument `elv`), g is the gravity constant (9.80665 m s-2), $p0$ is the atmospheric pressure at 0 m a.s.l. (argument `patm0`, defaults to 101325 Pa), L is the mean adiabatic lapse rate (0.0065 K m-2), M is the molecular weight for dry air (0.028963 kg mol-1), R is the universal gas constant (8.3145 J mol-1 K-1), and $TK0$ is the standard temperature (298.15 K, corresponds to 25 deg C).

Value

A numeric value for p

References

Allen, R. G., Pereira, L. S., Raes, D., Smith, M.: FAO Irrigation and Drainage Paper No. 56, Food and Agriculture Organization of the United Nations, 1998

Examples

```
print("Standard atmospheric pressure, in Pa, corrected for 1000 m.a.s.l.:")
print(calc_patm(1000))
```

---

calc_soilmstress *Calculates an empirical soil moisture stress factor*

Description

Calculates an empirical soil moisture stress factor as a function of relative soil moisture (fraction of field capacity).

Usage

```
calc_soilmstress(soilm, meanalpha = 1, apar_soilm = 0, bpar_soilm = 0.685)
```

Arguments

`soilm`	Relative soil moisture as a fraction of field capacity (unitless). Defaults to 1.0 (no soil moisture stress).
`meanalpha`	Local annual mean ratio of actual over potential evapotranspiration, measure for average aridity. Defaults to 1.0.
`apar_soilm`	(Optional, used only if `do_calc_soilmstress==TRUE`) Parameter determining the sensitivity of the empirical soil moisture stress function. Defaults to 0.0, the empirically fitted value as presented in Stocker et al. (2019) Geosci. Model Dev. for model setup 'FULL' (corresponding to a setup with `method_jmaxlim="wang17",do_ftemp_kphio=T`
`bpar_soilm`	(Optional, used only if `do_calc_soilmstress==TRUE`) Parameter determining the sensitivity of the empirical soil moisture stress function. Defaults to 0.685, the empirically fitted value as presented in Stocker et al. (2019) Geosci. Model Dev. for model setup 'FULL' (corresponding to a setup with `method_jmaxlim="wang17",do_ftemp_kph`

Details

The soil moisture stress factor is calculated using a quadratic function that is 1 above `soilm` = 0.6 and has a sensitivity, given by the y-axis cutoff, (zero soil moisture), determined by average aridity (argument `meanalpha`) as:

$$\beta = q(\theta - \theta*)^2 + 1$$

for $\theta < \theta*$ and $\beta = 1.0$ otherwise. $\theta*$ is fixed at 0.6. q is the sensitivity parameter and is calculated as a linear function of average aridity, quantified by the local annual mean ratio of actual over potential evapotranspiration, termed α:

$$q = (\beta 0 - 1)/(\theta * - \theta 0)^2$$

$\theta 0$ is 0.0, and

$$\beta 0 = a + b\alpha$$

a is given by argument `apar`, b is given by argument `bpar`.

Value

A numeric value for β

References

Stocker, B. et al. Geoscientific Model Development Discussions (in prep.)

Examples

```
## Relative reduction (%) in GPP due to soil moisture stress at
## relative soil water content ('soilm') of 0.2:
print((calc_soilmstress(0.2)-1)*100 )
```

`rpmodel` *Invokes a P-model function call*

Description

R implementation of the P-model and its corrolary predictions (Prentice et al., 2014; Han et al., 2017).

Usage

```
rpmodel(
  tc,
  vpd,
  co2,
  fapar,
  ppfd,
  patm = NA,
  elv = NA,
 kphio = ifelse(do_ftemp_kphio, ifelse(do_soilmstress, 0.087182, 0.081785), 0.049977),
  beta = 146,
  soilm = 1,
  meanalpha = 1,
  apar_soilm = 0,
```

```
  bpar_soilm = 0.733,
  c4 = FALSE,
  method_optci = "prentice14",
  method_jmaxlim = "wang17",
  do_ftemp_kphio = TRUE,
  do_soilmstress = FALSE,
  returnvar = NULL,
  verbose = FALSE
)
```

Arguments

`tc`	Temperature, relevant for photosynthesis (deg C)
`vpd`	Vapour pressure deficit (Pa)
`co2`	Atmospheric CO2 concentration (ppm)
`fapar`	(Optional) Fraction of absorbed photosynthetically active radiation (unitless, defaults to `NA`)
`ppfd`	(Optional) Photosynthetic photon flux density (mol m-2 d-1, defaults to `NA`). Note that the units of `ppfd` (per area and per time) determine the units of outputs `lue`, `gpp`, `vcmax`, and `rd`. For example, if `ppfd` is provided in units of mol m-2 month-1, then respective output variables are returned as per unit months.
`patm`	Atmospheric pressure (Pa). When provided, overrides `elv`, otherwise `patm` is calculated using standard atmosphere (101325 Pa), corrected for elevation (argument `elv`), using the function calc_patm.
`elv`	Elevation above sea-level (m.a.s.l.). Is used only for calculating atmospheric pressure (using standard atmosphere (101325 Pa), corrected for elevation (argument `elv`), using the function calc_patm), if argument `patm` is not provided. If argument `patm` is provided, `elv` is overridden.
`kphio`	Apparent quantum yield efficiency (unitless). Defaults to 0.0817 for `method_jmaxlim="wang17",do_fte` 0.0870 for `method_jmaxlim="wang17",do_ftemp_kphio=TRUE,do_soilmstress=TRUE`, and 0.0492 for `method_jmaxlim="wang17",do_ftemp_kphio=FALSE,do_soilmstress=FALSE`, corresponding to the empirically fitted value as presented in Stocker et al. (2019) Geosci. Model Dev. for model setup 'BRC', 'FULL', and 'ORG' respectively.
`beta`	Unit cost ratio. Defaults to 146.0 (see Stocker et al., 2019).
`soilm`	(Optional, used only if `do_soilmstress==TRUE`) Relative soil moisture as a fraction of field capacity (unitless). Defaults to 1.0 (no soil moisture stress). This information is used to calculate an empirical soil moisture stress factor (calc_soilmstress) whereby the sensitivity is determined by average aridity, defined by the local annual mean ratio of actual over potential evapotranspiration, supplied by argument `meanalpha`.
`meanalpha`	(Optional, used only if `do_soilmstress==TRUE`) Local annual mean ratio of actual over potential evapotranspiration, measure for average aridity. Defaults to 1.0.
`apar_soilm`	(Optional, used only if `do_soilmstress==TRUE`) Parameter determining the sensitivity of the empirical soil moisture stress function. Defaults to 0.0, the empirically fitted value as presented in Stocker et al. (2019) Geosci. Model Dev. for model setup 'FULL' (corresponding to a setup with `method_jmaxlim="wang17",do_ftemp_kphio=TRUE`

`bpar_soilm` (Optional, used only if `do_soilmstress==TRUE`) Parameter determining the sensitivity of the empirical soil moisture stress function. Defaults to 0.685, the empirically fitted value as presented in Stocker et al. (2019) Geosci. Model Dev. for model setup 'FULL' (corresponding to a setup with `method_jmaxlim="wang17",do_ftemp_kphio=T`

`c4` (Optional) A logical value specifying whether the C3 or C4 photosynthetic pathway is followed. Defaults to `FALSE`. If `TRUE`, the leaf-internal CO2 concentration is assumed to be very large and m (returned variable `mj`) tends to 1, and m' tends to 0.669 (with `c = 0.41`).

`method_optci` (Optional) A character string specifying which method is to be used for calculating optimal ci:ca. Defaults to `"prentice14"`.

`method_jmaxlim` (Optional) A character string specifying which method is to be used for factoring in Jmax limitation. Defaults to `"wang17"`, based on Wang Han et al. 2017 Nature Plants and (Smith 1937). Available is also `"smith19"`, following the method by Smith et al., 2019 Ecology Letters, and `"none"` for ignoring effects of Jmax limitation.

`do_ftemp_kphio` (Optional) A logical specifying whether temperature-dependence of quantum yield efficiency after Bernacchi et al., 2003 is to be accounted for. Defaults to `TRUE`.

`do_soilmstress` (Optional) A logical specifying whether an empirical soil moisture stress factor is to be applied to down-scale light use efficiency (and only light use efficiency). Defaults to `FALSE`.

`returnvar` (Optional) A character string of vector of character strings specifying which variables are to be returned (see return below).

`verbose` Logical, defines whether verbose messages are printed. Defaults to `FALSE`.

Value

A named list of numeric values (including temperature and pressure dependent parameters of the photosynthesis model, P-model predictions, including all its corollary). This includes :

- `ca`: Ambient CO2 expressed as partial pressure (Pa)
- `gammastar`: Photorespiratory compensation point $\Gamma*$, (Pa), see calc_gammastar.
- `kmm`: Michaelis-Menten coefficient K for photosynthesis (Pa), see calc_kmm.
- `ns_star`: Change in the viscosity of water, relative to its value at 25 deg C (unitless).

  $$\eta* = \eta(T)/\eta(25degC)$$

  This is used to scale the unit cost of transpiration. Calculated following Huber et al. (2009).
- `chi`: Optimal ratio of leaf internal to ambient CO2 (unitless). Derived following Prentice et al. (2014) as:

  $$\chi = \Gamma * /ca + (1 - \Gamma * /ca)\xi/(\xi + \sqrt{D})$$

  with

  $$\xi = \sqrt{(}\beta(K + \Gamma*)/(1.6\eta*))$$

  β is given by argument `beta`, K is `kmm` (see calc_kmm), $\Gamma*$ is `gammastar` (see calc_gammastar). $\eta*$ is `ns_star`. D is the vapour pressure deficit (argument `vpd`), ca is the ambient CO2 partial pressure in Pa (`ca`).

- `ci`: Leaf-internal CO2 partial pressure (Pa), calculated as (χca).
- `lue`: Light use efficiency (g C / mol photons), calculated as

$$LUE = \phi(T)\phi 0 m' Mc$$

where $\phi(T)$ is the temperature-dependent quantum yield efficiency modifier (calc_ftemp_kphio) if `do_ftemp_kphio==TRUE`, and 1 otherwise. $\phi 0$ is given by argument `kphio`. $m' = m$ if `method_jmaxlim=="none"`, otherwise

$$m' = m\sqrt(1 - (c/m)^(2/3))$$

with $c = 0.41$ (Wang et al., 2017) if `method_jmaxlim=="wang17"`. Mc is the molecular mass of C (12.0107 g mol-1). m is given returned variable `mj`. If `do_soilmstress==TRUE`, LUE is multiplied with a soil moisture stress factor, calculated with calc_soilmstress.
- `mj`: Factor in the light-limited assimilation rate function, given by

$$m = (ci - \Gamma*)/(ci + 2\Gamma*)$$

where $\Gamma*$ is given by `gammastar`.
- `mc`: Factor in the Rubisco-limited assimilation rate function, given by

$$mc = (ci - \Gamma*)/(ci + K)$$

where K is given by `kmm`.
- `gpp`: Gross primary production (g C m-2), calculated as

$$GPP = IabsLUE$$

where $Iabs$ is given by `fapar*ppfd` (arguments), and is `NA` if `fapar==NA` or `ppfd==NA`. Note that gpp scales with absorbed light. Thus, its units depend on the units in which `ppfd` is given.
- `iwue`: Intrinsic water use efficiency (iWUE, Pa), calculated as

$$iWUE = ca(1 - \chi)/(1.6)$$

- `gs`: Stomatal conductance (gs, in mol C m-2 Pa-1), calculated as

$$gs = A/(ca(1 - \chi))$$

where A is `gpp`$/Mc$.
- `vcmax`: Maximum carboxylation capacity $Vcmax$ (mol C m-2) at growth temperature (argument `tc`), calculated as

$$Vcmax = \phi(T)\phi 0 Iabsn$$

where n is given by $n = m/mc$, or

$$n = (ci + K)/(ci + 2\Gamma*)$$

- `vcmax25`: Maximum carboxylation capacity $Vcmax$ (mol C m-2) normalised to 25 deg C following a modified Arrhenius equation, calculated as $Vcmax25 = Vcmax/fv$, where fv is the instantaneous temperature response by Vcmax and is implemented by function calc_ftemp_inst_vcmax.

- `rd`: Dark respiration Rd (mol C m-2), calculated as

$$Rd = b0 Vcmax (fr/fv)$$

where $b0$ is a constant and set to 0.015 (Atkin et al., 2015), fv is the instantaneous temperature response by Vcmax and is implemented by function calc_ftemp_inst_vcmax, and fr is the instantaneous temperature response of dark respiration following Heskel et al. (2016) and is implemented by function calc_ftemp_inst_rd.

Additional variables are contained in the returned list if argument `method_jmaxlim=="smith19"`

- `omega`: Term corresponding to ω, defined by Eq. 16 in Smith et al. (2019), and Eq. E19 in Stocker et al. (2019).
- `omega_star`: Term corresponding to ω^*, defined by Eq. 18 in Smith et al. (2019), and Eq. E21 in Stocker et al. (2019).

References

Bernacchi, C. J., Pimentel, C., and Long, S. P.: In vivo temperature response func-tions of parameters required to model RuBP-limited photosynthesis, Plant Cell Environ., 26, 1419–1430, 2003

Heskel, M., O'Sullivan, O., Reich, P., Tjoelker, M., Weerasinghe, L., Penillard, A.,Egerton, J., Creek, D., Bloomfield, K., Xiang, J., Sinca, F., Stangl, Z., Martinez-De La Torre, A., Griffin, K., Huntingford, C., Hurry, V., Meir, P., Turnbull, M.,and Atkin, O.: Convergence in the temperature response of leaf respiration across biomes and plant functional types, Proceedings of the National Academy of Sciences, 113, 3832–3837, doi:10.1073/pnas.1520282113,2016.

Huber, M. L., Perkins, R. A., Laesecke, A., Friend, D. G., Sengers, J. V., Assael,M. J., Metaxa, I. N., Vogel, E., Mares, R., and Miyagawa, K.: New international formulation for the viscosity of H2O, Journal of Physical and Chemical ReferenceData, 38, 101–125, 2009

Prentice, I. C., Dong, N., Gleason, S. M., Maire, V., and Wright, I. J.: Balancing the costs of carbon gain and water transport: testing a new theoretical frameworkfor plant functional ecology, Ecology Letters, 17, 82–91, 10.1111/ele.12211,http://dx.doi.org/10.1111/ele.12211, 2014.

Wang, H., Prentice, I. C., Keenan, T. F., Davis, T. W., Wright, I. J., Cornwell, W. K.,Evans, B. J., and Peng, C.: Towards a universal model for carbon dioxide uptake by plants, Nat Plants, 3, 734–741, 2017. Atkin, O. K., et al.: Global variability in leaf respiration in relation to climate, plant func-tional types and leaf traits, New Phytologist, 206, 614–636, doi:10.1111/nph.13253, https://nph.onlinelibrary.wiley.com/doi/abs/10.1111/nph.13253.

Smith, N. G., Keenan, T. F., Colin Prentice, I. , Wang, H. , Wright, I. J., Niinemets, U. , Crous, K. Y., Domingues, T. F., Guerrieri, R. , Yoko Ishida, F. , Kattge, J. , Kruger, E. L., Maire, V. , Rogers, A. , Serbin, S. P., Tarvainen, L. , Togashi, H. F., Townsend, P. A., Wang, M. , Weerasinghe, L. K. and Zhou, S. (2019), Global photosynthetic capacity is optimized to the environment. Ecol Lett, 22: 506-517. doi:10.1111/ele.13210

Stocker, B. et al. Geoscientific Model Development Discussions (in prep.)

Examples

```
rpmodel( tc = 20, vpd = 1000, co2 = 400, fapar = 1, ppfd = 300, elv = 0)
```

Index