Package ‘Rquefts’

April 30, 2021

Type Package

Title Quantitative Evaluation of the Native Fertility of Tropical Soils

Version 1.2-1

Date 2021-04-30

LinkingTo Rcpp

SystemRequirements C++11

Imports meteor, methods, Rcpp (>= 0.12.4)

Suggests terra, limSolve

Description An implementation of the QUEFTS (Quantitative Evaluation of the Native Fertility of Tropical Soils) model. The model (1) estimates native nutrient (N, P, K) supply of soils from a few soil chemical properties; and (2) computes crop yield given that supply, crop parameters, fertilizer application, and crop attainable yield. See Janssen et al. (1990) <doi:10.1016/0016-7061(90)90021-Z> for the technical details and Sattari et al. (2014) <doi:10.1016/j.fcr.2013.12.005> for a recent evaluation and improvements.

License GPL (>= 3)

BugReports https://github.com/cropmodels/Rquefts/issues

URL https://CRAN.R-project.org/package=Rquefts

NeedsCompilation yes

Author Robert J. Hijmans [cre, aut],
Pieter Pypers [ctb],
Joost Wolff [ctb]

Maintainer Robert J. Hijmans <r.hijmans@gmail.com>

Repository CRAN

Date/Publication 2021-04-30 19:50:10 UTC

R topics documented:

`Rquefts-package` *Quantitative Evaluation of the Native Fertility of Tropical Soils*

Description

This package provides implments the QUEFTS model.

QUEFTS (Quantitative Evaluation of the Native Fertility of Tropical Soils) model (1) estimates native nutrient (N, P, K) supply of soils from a few soil chemical properties; and (2) computes crop yield given that supply, fertilizer application and crop parameters. See Janssen et al. (1990) <doi:10.1016/0016-7061(90)90021-Z> for the technical details and Sattari et al. (2014) <doi:10.1016/j.fcr.2013.12.005> for a recent evaluation and improvements.

The package is particularly useful if you want to make spatial predictions with QUEFTS.

There are also a few functions that can help with computing the amount of nutriets supplied with fertilizer (blends) and compute the optimal use of fertilizer given a goal in nutrients, available products, and their prices.

`fertApp` *Optimal fertilizer application*

Description

Compute the optimal fertilizer application rates given a target nutrient application and the available products (fertilizer blends) and their prices.

Usage

```
fertApp(nutrients, fertilizers, price, exact=TRUE, retCost=FALSE)
```

Arguments

`nutrients`	data.frame with columns "N", "P", "K" in kg (per unit area)
`fertilizers`	data.frame with fertilizer products (see examples)
`price`	numeric. Vector with fertilizer product prices. Should have length of `nrow(fertilizers)`
`exact`	logical. If `FALSE` the cheapest solution is returned that includes at least as much of each nutrient as desired, but possibly more, if that is cheaper than the exact solution; or when there is no exact solution
`retCost`	logical. If `FALSE` the optimal solution is returned (the amounts of fertilizers). If `TRUE`, the price of the optimal solution is returned

Examples

```
# fertilizer product list
fert <- fertilizers()
# shortening some of the names for display
fert[,2] = substr(fert[,2], 1, 20)
# contents are expressed as a percentage.
ferts <- fert[c(8,15:17), 2:5]
ferts

x <- fertApp(data.frame(N=100, P=50, K=50), ferts, c(1, 1.5, 1.25, 1))
# show that it is correct
nutrientRates(ferts, x[,2])

fertApp(data.frame(N=seq(0,200,50), P=50, K=50), ferts, c(1, 1.5, 1.25, 0.75))
fertApp(data.frame(N=seq(0,200,50), P=50), ferts[,-3], c(1, 1.5, 1.25, 0.75))
fertApp(data.frame(N=seq(0,200,50), P=50), ferts[,-3], c(1, 1.5, 1.25, 5.75))
```

`Fertilizers` *Helper functions to go from fertilizers to nutrients*

Description

Computes the amount of nutrients given a rate of fertilizer.

Usage

```
fertilizers()
nutrientRates(supply, treatment)
```

Arguments

`supply`	data.frame with columns "N", "P", "K" expressed as percentage of the product (row)
`treatment`	amounts applied

Examples

```
# fertilizer product list
fert <- fertilizers()
# shortening some of the names for display
fert[,2] = substr(fert[,2], 1, 20)
# contents are expressed as a percentage.
fert

myferts <- fert[c(8,15), ]
nutrientRates(myferts, c(100,50))
```

`nutSupply` *Soil nutrients supply for QUEFTS model*

Description

`nutSupply1` computes the base (unfertilized) soil supply of N, P and K according to Janssen et al. (1990), Table 2. For use with the QUEFTS model.

`nutSupply2` is a modified version following Sattari et al. (2014). It has an additional variable "temperature", and P-total is required. Sattari et al suggest that, for soils that have not been fertilized with P, you can estimate P-total as `95 * P-Olsen`. Using AfSIS data I found `55 * P-Olsen`.

Usage

```
nutSupply1(pH, SOC, Kex, Polsen, Ptotal=NA)
nutSupply2(temp, pH, SOC, Kex, Polsen, Ptotal)
```

Arguments

`temp`	average growing season temperature (C)
`pH`	soil pH (H2O)
`SOC`	soil organic carbon (g/kg)
`Kex`	exchangeable K in the soil (mmol/kg)
`Polsen`	soil P measured with the P-Olsen method (mg/kg)
`Ptotal`	total soil P (mg/kg)

Value

Matrix with three columns: Nsup, Psup and Ksup. These are the potential supply of N, P and K of the unfertilized soil (kg/ha).

References

Janssen B.H., F.C.T. Guiking, D. van der Eijk, E.M.A. Smaling, J. Wolf and H. van Reuler, 1990. A system for the quantitative evaluation of the fertility of tropical soils (QUEFTS). Geoderma 46: 299-318

Sattari, S.Z., M.K. van Ittersum, A.F. Bouwman, A.L. Smit, and B.H. Janssen, 2014. Crop yield response to soil fertility and N, P, K inputs in different environments: Testing and improving the QUEFTS model. Field Crops Research 157: 35-46

Examples

```
s1 <- nutSupply1(6, c(23, 11, 35), 15, c(1.6, 2.6, 2.4))
s1
s2 <- nutSupply2(20, 6, c(23, 11, 35), 15, c(1.6, 2.6, 2.4), 225)
s2
```

---

predict *Spatial QUEFTS model predictions*

Description

Make spatial predictions with a QUEFTS model. First create a model, then use the model with a SpatRaster of soil properties to make spatial predictons.

Usage

```
## S4 method for signature 'Rcpp_QueftsModel'
predict(object, supply, yatt, var="yield", filename="", overwrite=FALSE, ...)
```

Arguments

`object`	QUEFTSModel
`supply`	SpatRaster with nutrient supply data (Ns, Ps, Ks)
`yatt`	SpatRaster with attainable yield
`var`	character. Output variable name. Either "yield" or "gap"
`filename`	character. Output filename. Optional
`overwrite`	logical. If `TRUE`, `filename` is overwritten
`...`	list. Options for writing files as in `writeRaster`

Value

SpatRaster

Examples

```
library(Rquefts)
library(terra)

ff <- list.files(system.file("sp", package="Rquefts"), full.names=TRUE)
r <- rast(ff)

soil <- r[[c("Tavg", "pH", "SOC", "Kex", "Pex", "Ptot")]]
supply <- lapp(soil, nutSupply2)
plot(supply)

yatt <- rast(system.file("sp/Ya.tif", package="Rquefts"))

maize <- quefts_crop("Maize")
fertilizer <- list(N=0, P=0, K=0)
q <- quefts(crop=maize, fert=fertilizer)

p <- predict(q, supply, yatt)
plot(p)

g <- predict(q, supply, yatt, "gap")
plot(g)
```

---

quefts *QUEFTS model*

Description

Create a QUEFTS model, set parameters, and run it to compute nutrient requirements and nutrient limited yield.

A number of default crop parameter sets are provided, as well as one example soil. You need to provide attainable crop production (in this context that is the maximum production in the absence of nutrient limitation), or target dry-matter biomass for leaves, stems and the storage organ (e.g. grain, root or tuber). Some crops are grown for the stems/leaves, in which case there is no relevant storage organ (e.g. sugarcane, jute). production yield estimates can be obtained with a crop growth model.

For a cereal crop you can assume that 50

Usage

```
quefts(soil, crop, fert, biom)
quefts_soil()
quefts_fert()
quefts_crop(name="")
quefts_biom()
crop(x) <- value
soil(x) <- value
```

```
fert(x) <- value
biom(x) <- value
run(x, ...)
```

Arguments

`soil`	list with named soil parameters. See Details. An example is returned by `quefts_soil()`
`crop`	list with named crop parameters. See Details. An example is returned by `quefts_crop()`
`fert`	list with named fertilizer parameters (N, P and K). An example is returned by `quefts_fert()`
`biom`	list with named biomass and growing season length parameters. An example is returned by `quefts_biom()`
`name`	character. crop name
`x`	QueftsModel object
`value`	list with soil, crop, fertilizer, or biomass parameters as above
`...`	additional arguments. None implemented

Details

Input Parameters	**Explanation**
Soil	.
`N_base_supply`, `P_base_supply`, `K_base_supply`	Potential supply (kg/ha) of N, P and K of the (unfertilized) s
`N_recovery`, `P_recovery`, `K_recovery`	Fertilizer recovery, that is, the fraction of applied fertilizer th
`UptakeAdjust`	Two-column matrix to compute the fraction uptake from soi
Crop	.
`_minVeg`, `_maxVeg`, `_minStore`, `_maxStore`	minimum and maximum concentration of "_" (N, P, or K) in
`Yzero`	the maximum biomass of vegetative organs at zero yield of s
`Nfix`	the fraction of a crop's nitrogen uptake supplied by biologica
Management	.
`N`, `P`, `K`	N, P, and K fertilizer applied.
Crop yield	.
`leaf_att`, `stem_att`, `store_att`	Attainable (in the absence of nutrient limitation), or target cr
`SeasonLength`	Length of the growing season (days)
.	
.	
Output Variables	**Explanation**
`N_actual_supply`, `P_actual_supply`, `K_actual_supply`	nutrient uptake from soil (not fertilizer) (kg/ha)
`leaf_lim`, `stem_lim`, `store_lim`	nutrient limited biomass of leaves, stems, and storage organ
`N_gap`, `P_gap`, `K_gap`	fertilizer required to reach the specified biomass (kg/ha)

Value

vector with output variables as described in the Details

References

Janssen B.H., F.C.T. Guiking, D. van der Eijk, E.M.A. Smaling, J. Wolf and H. van Reuler, 1990. A system for the quantitative evaluation of the fertility of tropical soils (QUEFTS). Geoderma 46: 299-318

Sattari, S.Z., M.K. van Ittersum, A.F. Bouwman, A.L. Smit, and B.H. Janssen, 2014. Crop yield response to soil fertility and N, P, K inputs in different environments: Testing and improving the QUEFTS model. Field Crops Research 157: 35-46

Examples

```
# create a QUEFTS model
# 1. get parameters
soiltype <- quefts_soil()
barley <- quefts_crop("Barley")
fertilizer <- list(N=0, P=0, K=0)
att_yield <- list(leaf_att=2200, stem_att=2700, store_att=4800, SeasonLength=110)

# 2. create a model
q <- quefts(soiltype, barley, fertilizer, att_yield)

# 3. run the model
run(q)

# change some parameters
q$SeasonLength <- 162
q$leaf_att <- 2651
q$stem_att <- 5053
q$store_att <- 8208

q$N <- 100
q$P <- 50
q$K <- 50

run(q)


## note that Rquefts uses C++ reference classes.
## This means that if you copy a quefts model, you do not create a
## new instance of the model, but you point to the same one!
q <- quefts()
q["N"]
k <- q
k["N"] <- 150
k["N"]
# the value of q has also changed!
q["N"]
```

```
## different ways of subsetting / replacement
q <- quefts()
q$N
q$N <- 30
q["N"]
q["N"] <- 90
q["model", "N"]
q["model", "N"] <- 60
q$N

q$soil$N_recovery
q["soil$N_recovery"]
q["soil$N_recovery"] <- .6
q["soil", "N_recovery"]
q["soil", "N_recovery"] <- .4
q$soil$N_recovery
```

Index