Package 'BIOMASS'

June 25, 2020

Type Package

Title Estimating Aboveground Biomass and Its Uncertainty in Tropical Forests

Version 2.1.3

Date 2020-06-22

Description
Contains functions to estimate aboveground biomass/carbon and its uncertainty in tropical forests.
These functions allow to (1) retrieve and to correct taxonomy, (2) estimate wood density and its uncertainty,
(3) construct height-diameter models, (4) manage tree and plot coordinates,
(5) estimate the aboveground biomass/carbon at the stand level with associated uncertainty.
To cite BIOMASS, please use citation(``BIOMASS'').
See more in the article of Réjou-Méchain et al. (2017) <doi:10.1111/2041-210X.12753>.

License GPL-2

Encoding UTF-8

LazyData true

Depends R(>= 2.10.0)

URL https://github.com/AMAP-dev/BIOMASS

BugReports https://github.com/AMAP-dev/BIOMASS/issues

Imports minpack.lm, raster, jsonlite, methods, proj4, graphics, stats, utils, data.table (>= 1.9.8), rappdirs, sp

VignetteBuilder knitr

Suggests knitr, rmarkdown, testthat, covr, httr, rvest, prettydoc, usethis, MCMCglmm, taxize, lmfor, sf

RoxygenNote 6.1.1

NeedsCompilation no

Author Maxime Réjou-Méchain [aut, cre, dtc],
Arthur Pere [aut],
Guillaume Cornu [aut] (<https://orcid.org/0000-0002-7523-5176>),
Ariane Tanguy [aut],

Camille Piponiot [aut],
Jerome Chave [dtc],
Bruno Hérault [aut],
Ted Feldpausch [dtc],
Philippe Verley [ctb]

Maintainer Maxime Réjou-Méchain <maxime.rejou@gmail.com>

Repository CRAN

Date/Publication 2020-06-25 09:00:03 UTC

R topics documented:

`AGBmonteCarlo` *Propagating above-ground biomass (AGB) or carbon (AGC) errors to the stand level*

Description

Propagation of the errors throughought the steps needed to compute AGB or AGC.

Usage

```
AGBmonteCarlo(D, WD = NULL, errWD = NULL, H = NULL, errH = NULL,
  HDmodel = NULL, coord = NULL, Dpropag = NULL, n = 1000,
  Carbon = FALSE, Dlim = NULL, plot = NULL)
```

Arguments

D	Vector of tree diameters (in cm)
WD	Vector of wood density estimates (in g/cm3)
errWD	Vector of error associated to the wood density estimates (should be of the same size as WD)
H	(option 1) Vector of tree heights (in m). If set, errH must be set too.
errH	(if H) Residual standard error (RSE) of a model or vector of errors (sd values) associated to tree height values (in the latter case the vector should be of the same length as H).
HDmodel	(option 2) Model used to estimate tree height from tree diameter (output from modelHD(), see example).
coord	(option 3) Coordinates of the site(s), either a vector giving a single site (e.g. c(longitude, latitude)) or a matrix/dataframe with two columns (e.g. cbind(longitude, latitude)). The coordinates are used to predict height-diameter allometry with bioclimatic variables.
Dpropag	This variable can take three kind of values, indicating how to propagate the errors on diameter measurements: a single numerical value or a vector of the same size as D, both representing the standard deviation associated with the diameter measurements or "chave2004" (an important error on 5 percent of the measures, a smaller error on 95 percent of the trees).
n	Number of iterations. Cannot be smaller than 50 or larger than 1000. By default n = 1000
Carbon	(logical) Whether or not the propagation should be done up to the carbon value (FALSE by default).
Dlim	(optional) Minimum diameter (in cm) for which above-ground biomass should be calculated (all diameter below Dlim will have a 0 value in the output).
plot	(optional) Plot ID, must be either one value, or a vector of the same length as D. This argument is used to build stand-specific HD models.

Details

See Rejou-Mechain et al. (2017) for all details on the error propagation procedure.

Value

Returns a list with (if Carbon is FALSE):

- meanAGB: Mean stand AGB value following the error propagation
- medAGB: Median stand AGB value following the error propagation

- sdAGB: Standard deviation of the stand AGB value following the error propagation
- credibilityAGB: Credibility interval at 95% of the stand AGB value following the error propagation
- AGB_simu: Matrix with the AGB of the trees (rows) times the n iterations (columns)

Author(s)

Maxime REJOU-MECHAIN, Bruno HERAULT, Camille PIPONIOT, Ariane TANGUY, Arthur PERE

References

Chave, J. et al. (2004). *Error propagation and scaling for tropical forest biomass estimates*. Philosophical Transactions of the Royal Society B: Biological Sciences, 359(1443), 409-420.

Rejou-Mechain et al. (2017). *BIOMASS: An R Package for estimating above-ground biomass and its uncertainty in tropical forests*. Methods in Ecology and Evolution, 8 (9), 1163-1167.

Examples

```
# Load a database
data(NouraguesHD)
data(KarnatakaForest)

# Modelling height-diameter relationship
HDmodel <- modelHD(D = NouraguesHD$D, H = NouraguesHD$H, method = "log2")

# Retrieving wood density values
KarnatakaWD <- getWoodDensity(KarnatakaForest$genus, KarnatakaForest$species,
  stand = KarnatakaForest$plotId
)

# Propagating errors with a standard error in wood density in one plot
filt <- KarnatakaForest$plotId == "BSP20"
set.seed(10)
resultMC <- AGBmonteCarlo(
  D = KarnatakaForest$D[filt], WD = KarnatakaWD$meanWD[filt],
  errWD = KarnatakaWD$sdWD[filt], HDmodel = HDmodel
)
str(resultMC)

# If only the coordinates are available
lat <- KarnatakaForest$lat[filt]
long <- KarnatakaForest$long[filt]
coord <- cbind(long, lat)
## Not run:
resultMC <- AGBmonteCarlo(
  D = KarnatakaForest$D[filt], WD = KarnatakaWD$meanWD[filt],
  errWD = KarnatakaWD$sdWD[filt], coord = coord
)
str(resultMC)
```

```
## End(Not run)

# Propagating errors with a standard error in wood density in all plots at once
KarnatakaForest$meanWD <- KarnatakaWD$meanWD
KarnatakaForest$sdWD <- KarnatakaWD$sdWD
## Not run:
resultMC <- by(
  KarnatakaForest, KarnatakaForest$plotId,
  function(x) AGBmonteCarlo(
      D = x$D, WD = x$meanWD, errWD = x$sdWD,
      HDmodel = HDmodel, Dpropag = "chave2004"
    )
)
meanAGBperplot <- unlist(sapply(resultMC, "[", 1))
credperplot <- sapply(resultMC, "[", 4)

## End(Not run)
```

`attributeTree` *Attribute trees to subplots*

Description

Fonction to attribute the trees on each subplot, the trees that are at the exterior of the subplot will be marked as NA

Usage

```
attributeTree(xy, plot, coordAbs)
```

Arguments

`xy`	The coordinates of the trees for each plot
`plot`	The label of the plot (same length as the number of rows of `xy`)
`coordAbs`	Output of the function `cutPlot()`

Value

A vector with the code of the subplot for each trees, the code will be `plot_X_Y`. `X` and `Y` are the coordinate where the tree is inside the plot in regards to the corresponding subplot.

Author(s)

Arthur PERE

Examples

```
# Trees relative coordinates
xy <- data.frame(x = runif(200, min = 0, max = 200), y = runif(200, min = 0, max = 200))


# cut the plot in multiple part
coord <- data.frame(X = rep(c(0, 200, 0, 200), 2), Y = rep(c(0, 0, 200, 200), 2))
coord[1:4, ] <- coord[1:4, ] + 5000
coord[5:8, ] <- coord[5:8, ] + 6000
corner <- rep(c(1, 2, 4, 3), 2)
plot <- rep(c("plot1", "plot2"), each = 4)

cut <- cutPlot(coord, plot, corner, gridsize = 100, dimX = 200, dimY = 200)


# Assign a plot to 200 trees
plot <- rep(c("plot1", "plot2"), 100)

# attribute trees to subplots
attributeTree(xy, plot, cut)
```

---

attributeTreeCoord *Attribute trees to GPS coordinates*

Description

Attribute trees to GPS coordinates

Usage

```
attributeTreeCoord(xy, plot, dim, coordAbs)
```

Arguments

xy	The relative coordinates of the trees within each plot
plot	The label of the plot (same length as the number of rows of xy or length of 1)
dim	The dimension of the plot (either one value if the plot is a square or a vector if a rectangle)
coordAbs	The result of the function cutPlot() or numberCorner()

Value

A data frame with two columns: - Xproj: The X coordinates in the absolute coordinate system - Yproj: The Y coordinates in the absolute coordinate system

Examples

```
# Trees relative coordinates
xy <- data.frame(x = runif(200, min = 0, max = 200), y = runif(200, min = 0, max = 200))


# cut the plot in multiple part
coord <- data.frame(X = rep(c(0, 200, 0, 200), 2), Y = rep(c(0, 0, 200, 200), 2))
coord[1:4, ] <- coord[1:4, ] + 5000
coord[5:8, ] <- coord[5:8, ] + 6000
corner <- rep(c(1, 2, 4, 3), 2)
Forestplot <- rep(c("plot1", "plot2"), each = 4)

Outcut <- cutPlot(coord, Forestplot, corner, gridsize = 100, dimX = 200, dimY = 200)


# Assign a plot to 200 trees
Forestplot <- rep(c("plot1", "plot2"), 100)

# attribute trees to subplots
attributeTreeCoord(xy, Forestplot, dim =100,coordAbs = Outcut)
```

---

cacheManager *Manage the cache*

Description

The function creates a folder (only once) and then controls files that need to be downloaded and placed in this folder.

Usage

```
cacheManager(nameFile)
```

Arguments

nameFile	the name of the file or folder

Value

the path to the file we need

Localisation

The localisation of the folder is :

- On Linux : ~/.local/share/R/BIOMASS
- On Mac OS X : ~/Library/Application Support/R/BIOMASS

- On Windows 7 up to 10 : `C:\Users\<username>\AppData\Local\R\BIOMASS\R\BIOMASS`
- On Windows XP : `C:\Documents and Settings\<username>\Data\R\BIOMASS\R\BIOMASS`

See this function for more information : `rappdirs::user_data_dir()`

Author(s)

Arthur PERE

See Also

`rappdirs::user_data_dir()`

`computeAGB` *Computing tree above-ground biomass (AGB)*

Description

This function uses Chave et al. 2014's pantropical models to estimate the above-ground biomass of tropical trees.

Usage

```
computeAGB(D, WD, H = NULL, coord = NULL, Dlim = NULL)
```

Arguments

`D`	Tree diameter (in cm), either a vector or a single value.
`WD`	Wood density (in g/cm3), either a vector or a single value. If not available, see `getWoodDensity()`.
`H`	(optional) Tree height (H in m), either a vector or a single value. If not available, see `retrieveH()` and `modelHD()`. Compulsory if the coordinates `coord` are not given.
`coord`	(optional) Coordinates of the site(s), either a vector giving a single site (e.g. c(longitude, latitude)) or a matrix/dataframe with two columns (e.g. cbind(longitude, latitude)). The coordinates are used to account for variation in height-diameter relationship thanks to an environmental proxy (parameter E in Chave et al. 2014). Compulsory if tree heights `H` are not given.
`Dlim`	(optional) Minimum diameter (in cm) for which aboveground biomass should be calculated (all diameter below `Dlim` will have a 0 value in the output).

Details

This function uses two different ways of computing the above-ground biomass of a tree:

1. If tree height data are available, the AGB is computed thanks to the following equation (Eq. 4 in Chave et al., 2014):

$$AGB = 0.0673 * (WD * H * D^2)^0.976$$

2. If no tree height data is available, the AGB is computed thanks to the site coordinates with the following equation, slightly modified from Eq. 7 in Chave et al., 2014 (see Réjou-Méchain et al. 2017):

$$AGB = exp(-2.024 - 0.896 * E + 0.920 * log(WD) + 2.795 * log(D) - 0.0461 * (log(D)^2))$$

where `E` is a measure of environmental stress estimated from the site coordinates (`coord`).

Value

The function returns the ABG in Mg (or ton).

Author(s)

Maxime REJOU-MECHAIN, Ariane TANGUY, Arthur PERE

References

Chave et al. (2014) *Improved allometric models to estimate the aboveground biomass of tropical trees*, Global Change Biology, 20 (10), 3177-3190

See Also

`computeE()`

Examples

```
# Create variables
D <- 10:99
WD <- runif(length(D), min = 0.1, max = 1)
H <- D^(2 / 3)

# If you have height data
AGB <- computeAGB(D, WD, H)

# If you do not have height data and a single site
lat <- 4.08
long <- -52.68
coord <- cbind(long, lat)
## Not run:
AGB <- computeAGB(D, WD, coord = coord)

## End(Not run)
```

```
# If you do not have height data and several sites (here three)
lat <- c(rep(4.08, 30), rep(3.98, 30), rep(4.12, 30))
long <- c(rep(-52.68, 30), rep(-53.12, 30), rep(-53.29, 30))
coord <- cbind(long, lat)
## Not run:
AGB <- computeAGB(D, WD, coord = coord)

## End(Not run)
```

`computeFeldRegion` *Retrieving Feldpausch regions*

Description

Extract the Feldpausch et al. (2012)'s regions using local coordinates.

Usage

```
computeFeldRegion(coord, level = "region")
```

Arguments

`coord`	Coordinates of the site(s), a matrix/dataframe with two columns (e.g. cbind(longitude, latitude)) (see examples).
`level`	a string or a vector of string, the length must match the number of rows of the parameter coord. This parameter gives the scale at which Feldpausch regions should be assigned. There are tree levels: • `region`: Models assign at sub-continent levels, value by default • `continent`: Models assign at the Africa, South America, Asia and Australia levels • `world`: Pantropical model

Value

The function returns a vector with the Feldpausch et al. (2012)'s regions that can be incorporated in the `retrieveH` function.

Author(s)

Arthur PERE

References

Feldpausch, T.R., et al. (2012). *Tree height integrated into pantropical forest biomass estimates.* Biogeosciences, 9, 3381–3403.

Examples

```
#' # One study site
lat <- 4.08
long <- -52.68
coord <- cbind(long, lat)
## Not run:
FeldRegion <- computeFeldRegion(coord)

## End(Not run)

# Several study sites (here three sites)
long <- c(-52.68, -51.12, -53.11)
lat <- c(4.08, 3.98, 4.12)
coord <- cbind(long, lat)
## Not run:
FeldRegion <- computeFeldRegion(coord)

## End(Not run)
```

`correctCoordGPS` *Correct the GPS coordinates*

Description

This function builds the most probable GPS coordinates of the plot corners from multiple GPS measurements.

Usage

```
correctCoordGPS(longlat = NULL, projCoord = NULL, coordRel, rangeX,
  rangeY, maxDist = 15, drawPlot = FALSE, rmOutliers = TRUE)
```

Arguments

`longlat`	(optionnal) data frame with the coordinates in longitude latitude (e.g., cbind(longitude, latitude)).
`projCoord`	(optionnal) data frame with the projected coordinates (e.g., cbind(X, Y))
`coordRel`	data frame with the plot's relative coordinates corresponding to the longlat or projCoord GPS measurements
`rangeX`	a vector of length 2 giving the size of the plot along the X coordinates
`rangeY`	a vector of length 2 giving the size of the plot along the Y coordinates
`maxDist`	a numeric giving the maximum distance above which GPS measurements should be considered as outliers (by default 15 m)
`drawPlot`	a logical: if true, a graphical representation will be displayed
`rmOutliers`	a logical: if true, outliers will be removed from coordinates calculation (default)

Details

GPS coordinates should be either given in longitude latitude (longlat) or in projected coordinates (projCoord)

Value

If there are no outliers or rmOutliers = TRUE, a list with:

- `cornerCoords`: a data.frame with the coordinates of the corners
- `correctedCoord`: a data.frame with the adjusted coordinates given as input
- `polygon`: a spatial polygon
- `outliers`: index of coordinates lines considered as outliers, if any
- `codeUTM`: the UTM code of the coordinates if the parameter `longlat` is set

Author(s)

Arthur PERE, Maxime REJOU-MECHAIN

Arthur PERE, Maxime REJOU-MECHAIN

Examples

```
projCoord <- data.frame(
  X = c(
    runif(5, min = 9, max = 11), runif(5, min = 8, max = 12),
    runif(5, min = 80, max = 120), runif(5, min = 90, max = 110)
  ),
  Y = c(
    runif(5, min = 9, max = 11), runif(5, min = 80, max = 120),
    runif(5, min = 8, max = 12), runif(5, min = 90, max = 110)
  )
)
projCoord <- projCoord + 1000
coordRel <- data.frame(
  X = c(rep(0, 10), rep(100, 10)),
  Y = c(rep(c(rep(0, 5), rep(100, 5)), 2))
)

aa <- correctCoordGPS(
  projCoord = projCoord, coordRel = coordRel,
  rangeX = c(0, 100), rangeY = c(0, 100)
)
bb <- correctCoordGPS(
  projCoord = projCoord, coordRel = coordRel,
  rangeX = c(0, 100), rangeY = c(0, 100), rmOutliers = TRUE
)
## Not run:
correctCoordGPS(
  projCoord = projCoord, coordRel = coordRel,
  rangeX = c(0, 100), rangeY = c(0, 100), drawPlot = TRUE
)
```

```
## End(Not run)
```

`correctTaxo` *Checking typos in names*

Description

This function corrects typos for a given taxonomic name using the Taxonomic Name Resolution Service (TNRS) via the Taxosaurus interface. This function has been adapted from the `tnrs` function from the taxize package (`taxize::tnrs()`).

Usage

```
correctTaxo(genus, species = NULL, score = 0.5, useCache = TRUE,
  verbose = TRUE)
```

Arguments

`genus`	Vector of genera to be checked. Alternatively, the whole species name (genus + species) or (genus + species + author) may be given (see example).
`species`	(optional) Vector of species to be checked (same size as the genus vector).
`score`	Score of the matching (see http://tnrs.iplantcollaborative.org/instructions.html#match) below which corrections are discarded.
`useCache`	logical. Whether or not use a cache to reduce online search of taxa names (NULL means use cache but clear it first)
`verbose`	logical. If TRUE various messages are displayed during process

Details

This function create a file named correctTaxo.log (see Localisation), this file have the memory of all the previous requests, as to avoid the replication of time-consuming servor requests.

By default, names are queried in batches of 50, with a 0.5s delay between each query. These values can be modified using options: `options(BIOMASS.batch_size=50)` for batch size, `options(BIOMASS.wait_delay=0.5)` for delay.

Value

The function returns a dataframe with the corrected (or not) genera and species.

Localisation

The localisation of the folder is :

- On Linux : `~/.local/share/R/BIOMASS`
- On Mac OS X : `~/Library/Application Support/R/BIOMASS`
- On Windows 7 up to 10 : `C:\Users\<username>\AppData\Local\R\BIOMASS\R\BIOMASS`
- On Windows XP : `C:\Documents and Settings\<username>\Data\R\BIOMASS\R\BIOMASS`

See this function for more information : `rappdirs::user_data_dir()`

Author(s)

Ariane TANGUY, Arthur PERE, Maxime REJOU-MECHAIN, Guillaume CORNU

References

Boyle, B. et al. (2013). *The taxonomic name resolution service: An online tool for automated standardization of plant names*. BMC bioinformatics, 14, 1.

Chamberlain, S. A. and Szocs, E. (2013). *taxize: taxonomic search and retrieval in R*. F1000Research, 2.

Examples

```
## Not run:
correctTaxo(genus = "Astrocarium", species = "standleanum")
correctTaxo(genus = "Astrocarium standleanum")

## End(Not run)
```

---

cutPlot *Divides a plot in subplots*

Description

This function divides a plot in subplots (with dimX and dimY) and gives the coordinates of the grid in return. This function uses a procrustes analysis to fit the rectangle you gave to the plot you have.

Usage

```
cutPlot(projCoord, plot, corner, gridsize = 100, dimX = 200,
  dimY = 200)
```

Arguments

projCoord	A data frame with the projected coordinates with X and Y on the first and second colonne respectively
plot	Vector with the code of the plot
corner	Vector with the corner numbered from 1 to 4 for each plot, the numbered must be conter clockwise (see the result of the numberCorner())
gridsize	The size of the grid
dimX	A vector of the real size for the X axis for the plot (can be given one value it will be replicate for each plot)
dimY	A vector of the real size for the Y axis for the plot (can be given one value it will be replicate for each plot)

Value

This function return a data frame with :

- plot: The code of the plot you use
- subplot: The code of the subplot automaticaly generated
- XRel: The relative coordinate for the axis X (following the corner 1->2) for the plot
- YRel: The relative coordinate for the axis Y (following the corner 1->4) for the plot
- XAbs: The absolute coordinate (projected) for the axis X (following the corner 1->2)
- YAbs: The absolute coordinate (projected) for the axis Y (following the corner 1->4)

Author(s)

Arthur PERE

Examples

```
coord <- data.frame(X = c(0, 200, 0, 200), Y = c(0, 0, 200, 200)) + 5000
corner <- c(1, 2, 4, 3)
plot <- rep("plot1", 4)

cut <- cutPlot(coord, plot, corner, gridsize = 100, dimX = 200, dimY = 200)

# plot the result
plot(coord, main = "example", xlim = c(4900, 5300), ylim = c(4900, 5300), asp = 1)
text(coord, labels = corner, pos = 1)
points(cut$XAbs, cut$YAbs, pch = "+")
legend("bottomright", legend = c("orignal", "cut"), pch = c("o", "+"))
```

getTaxonomy *Retrieving the taxonomy*

Description

From a genus, the function `getTaxonomy` finds the APG III family, and optionally the order, from the genusFamily database and the apgFamilies dataset

Usage

```
getTaxonomy(genus, findOrder = FALSE)
```

Arguments

`genus`	Vector of genus names
`findOrder`	(Boolean) If `TRUE`, the output will contain the taxonomical orders of the families.

Value

Data frame with the order (if `findOrder` is `TRUE`), family and genus.

Author(s)

Ariane TANGUY, Arthur PERE, Maxime REJOU-MECHAIN

Examples

```
# Find the Family of the Aphelandra genus
getTaxonomy("Aphelandra")
# ... and the order
getTaxonomy("Aphelandra", findOrder = TRUE)
```

getWoodDensity *Estimating wood density*

Description

The function estimates the wood density (WD) of the trees from their taxonomy or from their congeners using the global wood density database (Chave et al. 2009, Zanne et al. 2009) or any additional dataset. The WD can either be attributed to an individual at a species, genus, family or stand level.

Usage

```
getWoodDensity(genus, species, stand = NULL, family = NULL,
  region = "World", addWoodDensityData = NULL, verbose = TRUE)
```

Arguments

genus	Vector of genus names
species	Vector of species names
stand	(optional) Vector with the corresponding stands of your data. If set, the missing wood densities at the genus level will be attributed at stand level. If not, the value attributed will be the mean of the whole tree dataset.
family	(optional) Vector of families. If set, the missing wood densities at the genus level will be attributed at family level if available.
region	Region (or vector of region) of interest of your sample. By default, Region is set to 'World', but you can restrict the WD estimates to a single region :

- AfricaExtraTrop: Africa (extra tropical)
- AfricaTrop: Africa (tropical)
- Australia: Australia
- AustraliaTrop: Australia (tropical)
- CentralAmericaTrop: Central America (tropical)
- China: China
- Europe: Europe
- India: India
- Madagascar: Madagascar
- Mexico: Mexico
- NorthAmerica: North America
- Oceania: Oceania
- SouthEastAsia: South-East Asia
- SouthEastAsiaTrop: South-East Asia (tropical)
- SouthAmericaExtraTrop: South America (extra tropical)
- SouthAmericaTrop: South America (tropical)
- World: World

addWoodDensityData	A dataframe containing additional wood density data to be combined with the global wood density database. The dataframe should be organized in a dataframe with three (or four) columns: "genus","species","wd", the fourth column "family" is optional.
verbose	A logical, give some statistic whith the database

Details

The function assigns to each taxon a species- or genus- level average if at least one wood density value at the genus level is available for that taxon in the reference database. If not, the mean wood density of the family (if set) or of the stand (if set) is given.

The function also provides an estimate of the error associated with the wood density estimate (i.e. a standard deviation): a mean standard deviation value is given to the tree at the appropriate taxonomic level using the sd_10 dataset.

Value

Returns a dataframe containing the following information:

- `family`: (if set) Family
- `genus`: Genus
- `species`: Species
- `meanWD`: Mean wood density
- `sdWD`: Standard deviation of the wood density that can be used in error propagation (see sd_10 and `AGBmonteCarlo()`)
- `levelWD`: Level at which wood density has been calculated. Can be species, genus, family, dataset (mean of the entire dataset) or, if stand is set, the name of the stand (mean of the current stand)
- `nInd`: Number of individuals taken into account to compute the mean wood density

Author(s)

Maxime REJOU-MECHAIN, Arthur PERE, Ariane TANGUY

References

Chave, J., et al. *Towards a worldwide wood economics spectrum*. Ecology letters 12.4 (2009): 351-366. Zanne, A. E., et al. *Global wood density database*. Dryad. Identifier: http://hdl. handle. net/10255/dryad 235 (2009).

See Also

wdData, sd_10

Examples

```
# Load a data set
data(KarnatakaForest)

# Compute the Wood Density up to the genus level and give the mean wood density of the dataset
WD <- getWoodDensity(
  genus = KarnatakaForest$genus,
  species = KarnatakaForest$species
)

# Compute the Wood Density up to the genus level and then give the mean wood density per stand
WD <- getWoodDensity(
  genus = KarnatakaForest$genus,
  species = KarnatakaForest$species,
  stand = KarnatakaForest$plotId
)

# Compute the Wood Density up to the family level and then give the mean wood density per stand
WD <- getWoodDensity(
  family = KarnatakaForest$family,
```

```
  genus = KarnatakaForest$genus,
  species = KarnatakaForest$species,
  stand = KarnatakaForest$plotId
)
str(WD)
```

HDmethods *HDmethods*

Description

Methods used for modeling height-diameter relationship

Usage

```
loglogFunction(data, method)

michaelisFunction(data, weight = NULL)

weibullFunction(data, weight = NULL)
```

Arguments

data	Dataset with the informations of height (H) and diameter (D)
method	In the case of the loglogFunction, the model is to be chosen between log1, log2 or log3.
weight	(optional) Vector indicating observation weights in the model.

Details

These functions model the relationship between tree height (H) and diameter (D). **loglogFunction** Compute two types of log model (log and log2) to predict H from D. The model can be:

- log 1: $log(H) = a + b * log(D)$ (equivalent to a power model)
- log 2: $log(H) = a + b * log(D) + c * log(D)^2$

michaelisFunction Construct a Michaelis Menten model of the form:

$$H = (A * D)/(B + D)$$

(A and B are the model parameters to be estimated)

weibullFunction Construct a three parameter Weibull model of the form:

$$H = a * (1 - exp(-(D/b)^c))$$

(a, b, c are the model parameters to be estimated)

Value

All the functions give an output similar to the one given by `stats::lm()`, obtained for `michaelisFunction` and `weibullFunction` from minpack.lm::nlsLM).

Author(s)

Maxime REJOU-MECHAIN, Ariane TANGUY

References

Michaelis, L., & Menten, M. L. (1913). *Die kinetik der invertinwirkung*. Biochem. z, 49(333-369), 352. Weibull, W. (1951). *Wide applicability*. Journal of applied mechanics, 103. Baskerville, G. L. (1972). *Use of logarithmic regression in the estimation of plant biomass*. Canadian Journal of Forest Research, 2(1), 49-53.

See Also

`modelHD()`, `lmfor::HDmodels()`

KarnatakaForest *Karnataka forest dataset*

Description

Dataset from 96 forest plots (1 ha) established in the central Western Ghats of India by Ramesh et al. (2010).

Usage

```
data("KarnatakaForest")
```

Format

A data frame with 65889 observations on the following 8 variables :

- `plotId`: Names of the plots
- `treeId`: Tree Id, contains a letter (A, B, C...) when an individual has multiple stems
- `family`: Family
- `genus`: Genus
- `species`: Species
- `D`: Diameter (cm)
- `lat`: Latitude
- `long`: Longitude

References

Ramesh, B. R. et al. (2010). *Forest stand structure and composition in 96 sites along environmental gradients in the central Western Ghats of India* Ecological Archives E091-216. Ecology, 91(10), 3118-3118.

Examples

```
data(KarnatakaForest)
str(KarnatakaForest)
```

`latlong2UTM` *Translate the long lat coordinate in UTM coordinate*

Description

Translate the long lat coordinate in UTM coordinate

Usage

```
latlong2UTM(coord)
```

Arguments

`coord`	Coordinates of the site(s), a matrix/dataframe with two columns (e.g. cbind(longitude, latitude)) (see examples).

Value

a data frame whith :

- `long`: The longitude of the entry
- `lat`: The latitude of the entry
- `codeUTM`: The code `proj` for UTM
- `X`: The X UTM coordinate
- `Y`: The Y UTM coordinate

Examples

```
long <- c(-52.68, -51.12, -53.11)
lat <- c(4.08, 3.98, 4.12)
coord <- cbind(long, lat)
## Not run:
UTMcoord <- latlong2UTM(coord)

## End(Not run)
```

`modelHD` *Fitting height-diameter models*

Description

This function fits and compares (optional) height-diameter models.

Usage

```
modelHD(D, H, method = NULL, useWeight = FALSE, drawGraph = FALSE,
  plot = NULL)
```

Arguments

`D`	Vector with diameter measurements (in cm). NA values are accepted but a minimum of 10 valid entries (i.e. having a corresponding height in H) is required.
`H`	Vector with total height measurements (in m). NA values are accepted but a minimum of 10 valid entries (i.e. having a corresponding diameter in D) is required.
`method`	Method used to fit the relationship. To be chosen between: • log1, log2 – log 1: $(log(H) = a + b * log(D))$ (equivalent to a power model) – log 2: $(log(H) = a + b * log(D) + c * log(D)^2)$ • weibull: $H = a * (1 - exp(-(D/b)^c))$ • michaelis: $H = (A * D)/(B + D)$ If `NULL`, all the methods will be compared.
`useWeight`	If weight is `TRUE`, model weights will be $(D^2) * H$ (i.e. weights are proportional to tree volume, so that larger trees have a stronger influence during the construction of the model).
`drawGraph`	If `TRUE`, a graphic will illustrate the relationship between H and D. Only if argument `plot` is null.
`plot`	(optional) Plot ID, must be either one value, or a vector of the same length as D. This argument is used to build stand-specific HD models.

Details

All the back transformations for log-log models are done using the Baskerville correction ($0.5 * RSE^2$, where RSE is the Residual Standard Error).

Value

If you have just one plot or NULL, there will be just the one of those result. However, if there is multiple plot, there will be the list with the names of the plot and inside each item their is those results. Returns a list with if the parameter model is not null:

- `input`: list of the data used to construct the model (list(H, D))
- `model`: outputs of the model (same outputs as given by `stats::lm()`, `stats::nls()`)
- `RSE`: Residual Standard Error of the model
- `RSElog`: Residual Standard Error of the log model (`NULL` if other model)
- `residuals`: Residuals of the model
- `coefficients`: Coefficients of the model
- `R.squared`: R^2 of the model
- `formula`: Formula of the model
- `method`: Name of the method used to construct the model
- `predicted`: Predicted height values

If the parameter model is null, the function return a graph with all the methods for comparison, the function also return a data.frame with:

- `method`: The method that had been used to construct the graph
- `color`: The color of the curve in the graph
- `RSE`: Residual Standard Error of the model
- `RSElog`: Residual Standard Error of the log model (`NULL` if other model)
- `Average_bias`: The average bias for the model

Author(s)

Maxime REJOU-MECHAIN, Arthur PERE, Ariane TANGUY

See Also

`retrieveH()`

Examples

```
# Load a data set
data(NouraguesHD)

# To model the height from a dataset
## Not run:
HDmodel <- modelHD(D = NouraguesHD$D, H = NouraguesHD$H, drawGraph = TRUE)

## End(Not run)

# If the method needed is known
HDmodel <- modelHD(D = NouraguesHD$D, H = NouraguesHD$H, method = "weibull", drawGraph = TRUE)
HDmodel <- modelHD(D = NouraguesHD$D, H = NouraguesHD$H, method = "log1", drawGraph = TRUE)

# Using weights
HDmodel <- modelHD(
  D = NouraguesHD$D, H = NouraguesHD$H, method = "weibull", useWeight = TRUE,
  drawGraph = TRUE
)
```

NouraguesHD *Height-Diameter data*

Description

Dataset from two 1-ha plots from the Nouragues forest (French Guiana)

Usage

```
data("NouraguesHD")
```

Format

A data frame with 1051 observations on the following variables :

- `plotId`: Names of the plots
- `genus`: Genus
- `species`: Species
- `D`: Diameter (cm)
- `H`: Height (m)
- `lat`: Latitude
- `long`: Longitude

References

Réjou-Méchain, M. et al. (2015). *Using repeated small-footprint LiDAR acquisitions to infer spatial and temporal variations of a high-biomass Neotropical forest* Remote Sensing of Environment, 169, 93-101.

Examples

```
data(NouraguesHD)
str(NouraguesHD)
```

numberCorner *Get the UTM coordinates with the corner of the plot*

Description

Get the UTM coordinates from the latitude and longitude of the corners of a plot. The function also assign a number to the corners in a clockwise or counterclockwise way, with the number 1 for the XY origin. Corner numbering is done as followed:

- axis X: the corner 1 to the corner 2
- axis Y: the corner 1 to the corner 4

Usage

```
numberCorner(longlat = NULL, projCoord = NULL, plot, origin, clockWise)
```

Arguments

longlat	(optionnal) data frame with the coordinates in longitude latitude (eg. cbind(longitude, latitude)).
projCoord	(optionnal) data frame with the projected coordinates in X Y
plot	A vector of codes (names) of the plots
origin	A logical vector with TRUE corresponding of the origin of the axis of each plot.
clockWise	A logical, whether the numbering should be done in a clockwise (TRUE) or counterclockwise (FALSE) way.

Value

A data frame with:

- plot: The code of the plot
- X: The coordinates X in UTM
- Y: The coordinates Y in UTM
- corner: The corner numbers

Author(s)

Arthur PERE, Maxime REJOU-MECHAIN

Examples

```
coord <- data.frame(X = c(0, 200, 0, 200), Y = c(0, 0, 200, 200)) + 5000
plot <- rep("plot1", 4)
origin <- c(FALSE, FALSE, TRUE, FALSE)

# if you turn clock wise
corner <- numberCorner(projCoord = coord, plot = plot, origin = origin, clockWise = TRUE)

# Plot the plot
plot(coord, asp = 1)
text(coord, labels = corner$corner, pos = 1)


# Using a counterclockwise way
corner <- numberCorner(projCoord = coord, plot = plot, origin = origin, clockWise = FALSE)

# Plot the plot
plot(coord, asp = 1)
text(coord, labels = corner$corner, pos = 1)
```

predictHeight *Predicting tree height*

Description

The function predicts height from diameter based on a fitted model.

Usage

```
predictHeight(D, model, err = FALSE, plot = NULL)
```

Arguments

D	Vector of diameter (in cm).
model	A height-diameter model output by the function modelHD()
err	If TRUE, An error is taken randomly from a normal distribution with a mean of zero and a standard deviation equalled to the residual standard error of the model (RSE). Only used for the Monte Carlo approach (see AGBmonteCarlo()), otherwise it should be let as FALSE, the default case.
plot	(optional) Plot ID, must be either one value, or a vector of the same length as D. This argument is used to build stand-specific HD models.

Details

In the case where the error is FALSE and the model is a log-log model, we use the Baskerville correction, a bias correction factor used to get unbiased backtransformation values.

Value

Returns a vector of total tree height (in m).

Author(s)

Maxime REJOU-MECHAIN, Ariane TANGUY, Arthur PERE

See Also

minpack.lm::nlsLM()

procrust *Procrust analysis*

Description

Do a procrust analysis. X is the target matrix, Y is the matrix we want to fit to the target. This function returns a translation vector and a rotation matrix After the procrust problem you **must** do the rotation before the translation. **Warning : The order of the value on both matrix is important**

Usage

```
procrust(X, Y)
```

Arguments

X	the target matrix
Y	the matrix we want to fit to the target

Value

A list with the translation vector and the matrix of rotation

Author(s)

Arthur PERE

retrieveH *Retrieving tree height from models*

Description

From the diameter and either i) a model, ii) the coordinates of the plot or iii) the region, this function gives an estimation of the total tree height.

Usage

```
retrieveH(D, model = NULL, coord = NULL, region = NULL,
  plot = NULL)
```

Arguments

D	Vector of diameters.
model	A model output by the function modelHD().
coord	Coordinates of the site(s), either a vector (e.g. c(longitude, latitude)) or a matrix/dataframe with two columns (e.g. cbind(longitude, latitude)).
region	Area of your dataset to estimate tree height thanks to Weibull-H region-, continent-specific and pantropical models proposed by Feldpausch et al. (2012). To be chosen between: • Africa: Africa • CAfrica: Central Africa • EAfrica: Eastern Africa • WAfrica: Western Africa • SAmerica: Southern America • BrazilianShield: Brazilian Shield • ECAmazonia: East-Central Amazonia • GuianaShield: Guiana Shield • WAmazonia: Western Amazonia • SEAsia: South-Eastern Asia • NAustralia: Northern Australia • Pantropical: Pantropical
plot	(optional) Plot ID, must be either one value, or a vector of the same length as D. This argument is used to build stand-specific HD models.

Value

Returns a list with:

- H: H predicted by the model
- RSE Residual Standard Error of the model, or a vector of those for each plot

Author(s)

Ariane TANGUY, Maxime REJOU-MECHAIN, Arthur PERE

References

Feldpausch et al. *Tree height integrated into pantropical forest biomass estimates*. Biogeosciences (2012): 3381-3403. Chave et al. *Improved allometric models to estimate the aboveground biomass of tropical trees*. Global change biology 20.10 (2014): 3177-3190.

See Also

modelHD()

Examples

```
# Load a database
data(NouraguesHD)
model <- modelHD(D = NouraguesHD$D, H = NouraguesHD$H, method = "log2")

# If any height model is available
H <- retrieveH(D = NouraguesHD$D, model = model)

# If the only data available are the coordinates of your spot
lat <- 4.08
long <- -52.68
coord <- cbind(long, lat)
## Not run:
H <- retrieveH(D = NouraguesHD$D, coord = coord)

## End(Not run)

# If the only data available is the region of your spot
H <- retrieveH(D = NouraguesHD$D, region = "GuianaShield")
```

`summaryByPlot` *Summarize by plot (or subplot) the posterior distribution of AGB values*

Description

This function summarizes the matrix `AGB_val` given by the function `AGBmonteCarlo()` by plot. Or just do the sums for each plot of the AGB if the argument `AGB_val` is the resulting vector from the function `computeAGB()`.

Usage

```
summaryByPlot(AGB_val, plot, drawPlot = FALSE, subplot = NULL)
```

Arguments

`AGB_val`	Matrix resulting from the function `AGBmonteCarlo()` (AGB_val element of the list), or just the output of the function `AGBmonteCarlo()`. Or the output of the function `computeAGB()`
`plot`	Vector with the code of plot
`drawPlot`	a logical to draw the plot (see Details)
`subplot`	Data frame, output of the function `cutPlot()`

Details

If some trees belong to an unknown plot (i.e. NA value in the plot arguments), their AGB values are randomly assigned to a plot at each iteration of the AGB monte Carlo approach. Or discarded when using output from `computeAGB()`.

The `drawPlot` argument is a logical that if it is set `TRUE`, a graph will appear with the plot given on absciss and the value of AGB on ordinate, the red segments are the quantile, if `AGB_val` is the result of the function `AGBmonteCarlo()`. If the `subplot` arguments is set and the `drawPlot` is set `TRUE`, a graph is drawn with the spatialisation of the plots.

Value

a data frame where:

- `plot`: the code of the plot
- `AGB`: AGB value at the plot level
- `Cred_2.5`: the quantile 2.5% for the plot (when output of `AGBmonteCarlo()` is used)
- `Cred_97.5`: the quantile 97.5% for the plot (when output of `AGBmonteCarlo()` is used)

If the `subplot` is set, the output is a list with the previous data frame and a simple features (sf) geometry object.

Examples

```
# Load a database
data(NouraguesHD)
data(KarnatakaForest)

# Modelling height-diameter relationship
HDmodel <- modelHD(D = NouraguesHD$D, H = NouraguesHD$H, method = "log2")

# Retrieving wood density values
KarnatakaWD <- getWoodDensity(KarnatakaForest$genus, KarnatakaForest$species,
  stand = KarnatakaForest$plotId
)

# Propagating errors
filt <- KarnatakaForest$plotId %in% c("BSP20", "BSP14")
resultMC <- AGBmonteCarlo(
  D = KarnatakaForest$D[filt], WD = KarnatakaWD$meanWD[filt],
  errWD = KarnatakaWD$sdWD[filt], HDmodel = HDmodel
)

plot <- KarnatakaForest$plotId[ filt ]

# The summary by plot
summaryByPlot(AGB_val = resultMC$AGB_simu, plot)

# The summary by plot for computeAGB
H <- retrieveH(KarnatakaForest$D[filt], model = HDmodel)$H
```

```
AGB <- computeAGB(KarnatakaForest$D[filt], WD = KarnatakaWD$meanWD[filt], H = H)
summaryByPlot(AGB, plot)
```

---

updateCache *Update the cache for the different function*

Description

This function update the cache for the environmental variables:

- wc2-5
- CWD
- E

Usage

```
updateCache(nameFile = NULL)
```

Arguments

nameFile	The name of the file you want to update. If it's NULL the function will update all the files.

Author(s)

Arthur PERE

Examples

```
## Not run:
updateCache()

## End(Not run)
```

Index